

# **FULL PAPERS**



# Combining techniques to refine item to skills Q-matrices with a partition tree

Michel C. Desmarais  
Polytechnique Montreal  
michel.desmarais@polymtl.ca

Peng Xu  
Polytechnique Montreal  
peng.xu@polymtl.ca

Behzad Beheshti  
Polytechnique Montreal  
behzad.beheshti@polymtl.ca

## ABSTRACT

The problem of mapping items to skills is gaining interest with the emergence of recent techniques that can use data for both defining this mapping, and for refining mappings given by experts. We investigate the problem of refining mapping from an expert by combining the output of different techniques. The combination is based on a partition tree that combines the suggested refinements of three known techniques from the literature. Each technique is given as input a Q-matrix, that maps items to skills, and student test outcome data, and outputs a modified Q-matrix that constitutes suggested improvements. We test the accuracy of the partition tree combination techniques over both synthetic and real data. The results over synthetic data show a high improvement over the best single technique with a 86% error reduction on average for four different Q-matrices. For real data, the error reduction is 55%. In addition to the substantial error reduction, the partition tree refinements provide a much more stable performance than any single technique. These results suggest that the partition tree is a valuable refinement combination approach that can effectively take advantage of the complementarity of the Q-matrix refinement techniques. It brings the goal of using a data driven approach to refine the item to skill mapping closer to real applications, although challenges remain and are discussed.

## 1. INTRODUCTION

Defining which skills are involved in a task is non trivial. Whereas task outcome is observable, skills are not. This layer of opacity leaves a world of possibilities to define which skills are behind task performance, and no obvious evidence to know if the proposed definition is correct or not. Means to provide such feedback would be highly valuable to teachers and designers of learning environments, and we find numerous recent efforts towards this end in the last few years. They are reviewed in section 2.

We developed an approach that takes the output of a combination of techniques to detect likely errors of task to skills

mappings given by experts. We investigate the combination of three data-driven techniques [3, 2, 7] based on a partition tree algorithm that creates binary partitions. See also [6] for a more detailed comparison of the performance of these three techniques.

The performance of the partition tree approach is tested over synthetic and real data. But even in the case of real data, the approach to grow the partition tree trains on synthetic performance data generated from a set of Q-matrices that are similar to the Q-matrix to refine. This procedure is chosen because only synthetic data provides a large enough training set, and because it also provides ground truth labelling of latent variables.

In the rest of this text we use the term *items* to refer to questions or tasks that can be part of a formative or summative assessment, or exercises within an e-learning environment. Skills can be the mastery of concepts, factual knowledge, or any ability involved in item outcome success. However, the models reviewed here assume a static student skills state, as opposed to the Knowledge Tracing model and its derivatives [11], for example, which rely on dynamic data. We return to this limitation in the Discussion.

The different techniques to validate a Q-matrix are first described, followed by the description of the approach, the experiments, and the results.

## 2. Q-MATRICES AND TECHNIQUES TO VALIDATE THEM FROM DATA

A mapping of item to skills is termed a Q-matrix. An example of a 11 items and 3 skills Q-matrix is given beside. It corresponds to the Q-matrix labelled QM 1 in the results section below. From this example, item 4 requires skill 1 only, whereas item 11 requires skills 1 and 2. If all specified skills are required to succeed the item, the Q-matrix is labeled **conjunctive**. If a any of the required skill is sufficient to the item's success, then it is labeled **disjunctive**. The **compensatory** version corresponds to the case

Q-matrix QM-1

Item	Skill		
	1	2	3
1	1	1	0
2	1	0	1
3	1	0	1
4	1	0	0
5	1	1	0
6	1	1	0
7	1	0	1
8	1	0	1
9	1	0	0
10	1	0	0
11	1	1	0

where each required item increases the chances of success in some way. Conjunctive Q-matrices the most common and all matrices of the experiments here are of this type.

The conjunctive/disjunctive distinction is also referred to as AND/OR gates. Skills models such as DINA (Deterministic Input Noisy AND) and DINO (Deterministic Input Noisy Or) make reference to this AND/OR gates terminology.

The DINA model [10] defines the probability of success to an item as a function of whether the skills required are mastered, and of two parameters, the *slip* and *guess* factors. Mastery is a binary value based on the conjunctive framework: if all required skills are mastered then the value is 1, else it is 0. Slip and guess parameters are values that generally vary on a  $[0, 0.2]$  scale. The probability of success to an item  $j$  by a student  $i$  is thereby defined as:

$$P(X_{ij}=1 | \xi_{ij}) = (1 - s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}}$$

where  $\xi_{ij}$  is 1 if student  $i$  masters all required skills of item  $j$ , 0 otherwise.  $s_j$  and  $g_j$  are the *slip* and *guess* factors.

Two techniques for Q-matrix validation surveyed here rely on the DINA model, whereas the third one relies on a matrix factorization technique called ALS (Alternative Least Squares), or more precisely ALSC for the *conjunctive* version of the technique. We briefly review each technique below.

## 2.1 Technique 1: MinRSS

Chiu defines a method that minimizes the residual sum of square (RSS) between the real responses and the ideal responses that follow from a given Q-matrix [2] under the DINA model. The algorithm adjusts the Q-matrix by first estimating the mastery of each student, then choosing the item with the worst RSS over to the data, and replacing it with a q-vector that has the lowest RSS, and iterates until convergence. We refer to this technique as MinRSS .

## 2.2 Technique 2: MaxDiff

The method defined by de la Torre [3] searches for a Q-matrix that maximizes the difference in the probabilities of a correct response to an item between examinees who possess all the skills required for a correct response to that item and examinees who do not. It also relies on the DINA model to determine item outcome probability, and on an EM algorithm to estimate the slip and guess parameters. Probability differences represents an item discrimination index: the greater the difference between the probability of a correct response given the skills required and the probability given missing skills, the greater the item is discriminant. As such, we can consider that the method finds a Q-matrix that maximizes item discrimination over all items. We refer to this technique as MaxDiff .

## 2.3 Technique 3: Conjunctive alternate Least-Square Factorization (ALSC)

The Conjunctive alternate Least-Square Factorization (ALSC) method is defined in [7]. Contrary to the other two methods, it does not rely on the DINA model as it has no slip and guess parameters. ALSC decomposes the results matrix  $\mathbf{R}_{m \times n}$  of  $m$  items by  $n$  students as the inner product two

smaller matrices:

$$\neg \mathbf{R} = \mathbf{Q} \neg \mathbf{S} \quad (1)$$

where  $\neg \mathbf{R}$  is the negation of the results matrix ( $m$  items by  $n$  students),  $\mathbf{Q}$  is the  $m$  items by  $k$  skills Q-matrix, and  $\neg \mathbf{S}$  is negation of the the mastery matrix of  $k$  skills by  $n$  students (normalized for rows columns to sum to 1). By negation, we mean the 0-values are transformed to 1, and non-0-values to 0. Negation is necessary for a conjunctive Q-matrix.

The factorization consists of alternating between estimates of  $\mathbf{S}$  and  $\mathbf{Q}$  until convergence. Starting with the initial expert defined Q-matrix,  $\mathbf{Q}_0$ , a least-squares estimate of  $\mathbf{S}$  is obtained:

$$\neg \hat{\mathbf{S}}_0 = (\mathbf{Q}_0^T \mathbf{Q}_0)^{-1} \mathbf{Q}_0^T \neg \mathbf{R} \quad (2)$$

Then, a new estimate of the Q-matrix,  $\hat{\mathbf{Q}}_1$ , is again obtained by the least-squares estimate:

$$\hat{\mathbf{Q}}_1 = \neg \mathbf{R} \neg \hat{\mathbf{S}}_0^T (\neg \hat{\mathbf{S}}_0 \neg \hat{\mathbf{S}}_0^T)^{-1} \quad (3)$$

And so on until convergence. Alternating between equations (2) and (3) yields progressive refinements of the matrices  $\hat{\mathbf{Q}}_i$  and  $\hat{\mathbf{S}}_i$  that more closely approximate  $\mathbf{R}$  in equation (1). The final  $\hat{\mathbf{Q}}_i$  is rounded to yield a binary matrix.

Note that  $(\neg \mathbf{Q}_i^T \neg \mathbf{Q}_i)$  or  $(\neg \hat{\mathbf{S}}_i \neg \hat{\mathbf{S}}_i^T)$  may not be invertible, for example in the case where the matrix  $\mathbf{Q}_i$  is not column full-rank, or the matrix  $\mathbf{S}_i$  is not row full-rank. This is resolved by adding a very small Gaussian noise before attempting the matrix inverse.

## 2.4 Other techniques

We chose the three techniques described above as the candidates to combine refinements that can potentially provide more accurate suggestions than any of the individual ones, but any other equivalent technique could also be combined in the same fashion instead of the three chosen ones here. Potential candidates could be, for example, a technique based on a Bayesian approach by DeCarlo et al. [5], and recent techniques that rely on time information [13, 12]. Yet another recent approach relies item text [8] to establish the mapping of items to skills.

Although the results obtained through a combination of techniques may vary as a function of the specific techniques chosen, the general principle remains valid for all possible combinations. And there is no reason to believe that the particular combination of the current study is better or worse than other potential combinations.

## 2.5 General validation principle

The general idea behind the validation of Q-matrices is to introduce a perturbation to a matrix and run a refinement technique that takes the perturbed matrix and test data as input, and outputs a set of refinements. In all, 8 cases can occur and they are listed in table 1. The 8 cases are a combination of the original cell value, perturbation, and value proposed ( $2 \times 2 \times 2$ ).

The outcome of a proposed value from the refinement technique is considered correct if it corresponds to the original value before the perturbation, and incorrect otherwise. We

Table 1: Refinement outcomes

Perturbation		Refinement		
Value before	Value after	Value proposed	Outcome	
<b>Perturbed cell</b>				
(1)	0	1	0	correct (TP)
(2)	1	0	1	correct (TP)
(3)	0	1	1	wrong (FN)
(4)	1	0	0	wrong (FN)
<b>Non Perturbed cell</b>				
(5)	0	0	0	correct (TN)
(6)	1	1	1	correct (TN)
(7)	0	0	1	wrong (FP)
(8)	1	1	0	wrong (FP)

also refer to the signal detection terminology with respect to perturbations to introduce further classification of the error types:

- **True Positives (TP)**: perturbed cell that was correctly changed
- **True Negatives (TN)**: non perturbed cell left unchanged
- **False Positives (FP)**: non perturbed cell incorrectly changed
- **False Negatives (FN)**: perturbed cell left unchanged

### 3. COMBINING TECHNIQUES WITH A PARTITION TREE

Each of the technique described above uses a different algorithm to provide a potentially improved Q-matrix. In that respect, their respective outcome may be complementary, and their combined outcome can be more reliable than any single one. This is the first hypothesis and objective of our study. Furthermore, some algorithms are more effective in general, but may not be the best performer in all context. Identifying in which context an algorithm provides the most reliable outcome is another objective of combining these techniques. We will see that the first hypothesis is confirmed in the results of the partition tree labeled (1) and the second is also confirmed by the results of partition tree (3).

#### 3.1 Partitioning tree

To implement the partition tree combination of the three techniques, we chose the `rpart` package for this purpose [19].

The `rpart` package builds classification models that can be represented as binary trees. The tree is constructed in a top-down recursive divide and conquer approach. At each node in the tree, cases are split into two groups based on their attribute value.

#### 3.1.1 Tree building

Attribute selection is done on the basis of Gini index in `rpart`. The Gini index [16] can be calculated as :

$$\text{Gini}(D) = 1 - \sum_{j=1}^n p_j^2$$

where  $n$  is the number of classes and  $p_j$  is the relative frequency of class  $j$  in dataset  $D$ . If attribute  $A$  is chosen to be a split on dataset  $D$  into two subset  $D_1$  and  $D_2$ , then the Gini index for attribute  $A$  is defined as:

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2)$$

Once we get the Gini index to add attributes we can calculate a Delta reduction for each attribute:

$$\Delta \text{Gini}(A) = \text{Gini}(D) - \text{Gini}_A(D)$$

The attribute that creates the largest reduction can be chosen as a splitting point in the decision tree.

#### 3.1.2 Classification with the tree

In our case, attributes are sometimes numeric, such as factors, and sometimes binary, such as cell values in the Q-matrix. And the class is binary since it is also a Q-matrix cell value. At each point of decision from the root node of the tree to a leaf node, a choice is made to go left or right based on the splitting point of each node. The nodes in the partition trees of this experiment are the output of the techniques (suggested values) and the factors considered (they are described in the next section).

Once a leaf node is reached, classification is based on the majority vote of the cases that fall under that leaf node: if the training set contained more case labeled '0', this is the proposed value, else it is a '1'.

### 3.2 Factors considered

The partition tree relies on each technique's output, the Q-matrix refinement proposition, and on a number of factors that may provide information about the most reliable technique refinement in a given context. The factors considered to be relevant are the following:

- **Skills per row**. Items can require one or more skills. The skills per row indicates the number of skills required.
- **Skills per column**. The sum of the skills per columns is an indicator of how often this skill is measured by the different items of the Q-matrix.
- **Stickiness**. If a technique systematically proposes a change to a cell of the Q-matrix, no matter what the perturbation is, this is an indication that this particular change to the original Q-matrix is an artifact of the structure of the Q-matrix and the algorithm. We call this property the *stickiness* of a cell of the matrix and it is measured by the proportion of times the value of the cell is incorrectly changed over all perturbations.

Recall that we train the partition tree over synthetic data for which the ground truth is known. We can therefore reliably identify incorrect changes. This is detailed below.

### 3.3 Training of the partition tree

The partition tree is trained on data that contains the following set of attributes:

- $\text{original}_{(j,k)}$ : value of cell  $(j, k)$  in the original matrix. This is the target class of the partition tree and it corresponds to “Value before” in table 1.
- $\text{MaxDiff}_{(j,k)}$ ,  $\text{MinRSS}_{(j,k)}$ ,  $\text{ALSC}_{(j,k)}$ : the three values proposed as refinements by the respective technique in place of the original value. For every record, at least one of these must be different from the original one, or else it is a perturbed cell record. This corresponds to “Value proposed” in table 1, one for each refinement technique.
- $\text{RS}_{Q_i,j}$ ,  $\text{CS}_{Q_i,k}$ : the number of skills per row and column attributes (see section 3.2). These factors are per Q-matrix,  $Q_i$ , and per row  $j$  and column  $k$ .
- $\text{SF}_{\text{MaxDiff}(Q_i,j,k)}$ ,  $\text{SF}_{\text{MinRSS}(Q_i,j,k)}$ ,  $\text{SF}_{\text{ALSC}(Q_i,j,k)}$ : the stickiness factors of the cell, one for each matrix and technique.

The training data is generated through a perturbation process. Each cell of a Q-matrix is perturbed, in turn and one at a time, to create a new training record containing the above attributes. However, non perturbed cells that are left unchanged by all refinements techniques, cases (5) and (6) in table 1, are left out of the training data because they were assumed to be uninformative.

The size of the data set to train the partition trees over is very large. For the permutations of a single Q-matrix, the number of perturbed and non perturbed cells ranges from approximately 50,000 to 250,000.

*Training of the partition tree for expert Q-matrices with synthetic data.* Whereas for synthetic data, we can generate a large array of Q-matrices and ample training and testing data, real data poses a challenge in that respect. Typically, for a single data set, we have only a few expert Q-matrices, and often a single one is available. For a 3 skills  $\times$  11 items matrix, only 33 single perturbations are possible to train a partition tree. Furthermore, and unlike synthetic data, we do not know what are the valid refinements in the Q-matrix. A “sticky” cell might be a valid refinement, and so can some of the perturbations that are presumed incorrect.

To get around these issues, the training of the partition tree is conducted over synthetic data where the ground truth is known and where we can use a large span of matrices similar to the expert one. Similarity to the Q-matrix to refine is achieved by random permutations the cells of the original Q-matrix. For each Q-matrix, a total of 1000 Q-matrices are generated through this permutation process. Item outcome data for 400 simulated students is also generated. The R package CDM and the `sim.din` function [15] is used for generating synthetic student item outcome data, using 0.2 slip and guess factors.

## 4. REAL DATA AND Q-MATRICES

The primary source of real data for our study, from which the synthetic data is also mimicked, is the well known data set

**Table 2: Four Q-matrices over 11 items of Tatsuoka’s data set on student item outcome**

	Number of			Description
	skills	items	cases	
QM 1	3	11	536	Expert driven. Skill 1 shared by all items. From [9]
QM 2	5	11	536	Expert driven. From [3]
QM 3	3	11	536	Expert driven. Single skill per item. [15]
QM 4	3	11	536	Data driven, SVD-based.

on fraction algebra problems from Tatsuoka [17] (see table 1 in [4] for a description of the problems and of the skills). The data contains complete answers of 536 students to 20 questions items, but only a subset of 11 items are used by the Q-matrices in the current study. It corresponds to the set of common items to the different Q-matrices of the experiment.

The original Q-matrix of this data set contains 8 skills and, as mentioned, 20 items. However, a number of variations of this matrix have been proposed and studied with a smaller number of skills and items [9, 3, 15]. We also chose to focus on this smaller skills set since they offer three very different expert-defined Q-matrices over the same set of items. Moreover, a smaller set of skills allows us to better establish the validity of the approach on a simpler problem, leaving for later the demonstration of whether it scales correctly to larger sets. The Q-matrices are described below.

Four Q-matrices are considered. Three of them have been studied in the literature and one is defined by ourselves. Their main attributes are reported in table 2 and the actual Q-matrices are shown in figure 1 (except for QM 1 which is introduced in section 2).

Item	Skills of										
	QM 2					QM 3			QM 4		
	1	2	3	4	5	1	2	3	1	2	3
1	1	1	1	1	0	0	1	0	1	1	0
2	1	1	1	1	1	0	0	1	1	0	1
3	0	0	1	0	0	0	0	1	0	1	0
4	1	1	1	1	0	1	0	0	1	0	0
5	1	1	1	1	0	0	1	0	1	0	0
6	1	1	0	0	0	0	1	0	0	0	1
7	1	0	1	1	1	0	0	1	1	0	1
8	1	0	1	0	0	0	0	1	0	1	1
9	1	0	1	1	0	1	0	0	1	0	0
10	1	1	1	1	0	1	0	0	1	0	1
11	1	1	1	1	0	0	1	0	1	0	0

**Figure 1: Q-matrices 2, 3, and 4.**

As mentioned, all Q-matrices are derivatives of the Tatsuoka [17] 20 item set. QM-1, QM-2 and QM-3 are available from the CDM package. All data sets have 3 skills, except for data set 2 which has 5 skills. Data set 3 is the only one

with a single skill per item. Matrix QM 4 was created for the purpose of this study, using the three largest singular values and the items to skills  $\mathbf{V}$  matrix of the SVD decomposition of the Tatsuoaka data mentioned above.

Therefore, while these four Q-matrices all share the same 11 items, they vary by the number of skills, item monotonicity or not, whether a skill is common to all items, and whether they are driven from data or driven from expert analysis of item skills involved.

## 5. GENERAL PROCEDURE SUMMARY AND METHODOLOGICAL NOTES

To ease the understanding of the general process of the experiments, and at the expense of introducing some redundancy, figure 2 summarizes the main steps and dependencies. The top greyed box illustrates the process to generate the data for partition trees training, and the synthetic data for performance evaluation. The bottom greyed box illustrates the two test procedures for real and synthetic data. We explain the figure below and fill in some details as well.

*Data generation.* For each of the four Q-matrices ( $QM_i$ ), the data generation process (1) 1000 permutations (2). Duplicates are kept if any. For each permutation, synthetic test outcome data of 400 simulated students is created with the CDM utility `sim.din` (3). Finally, each QM is perturbed, and that Q-matrix is fed to each of the three techniques to generate training data for the partition tree described in section 3.3 (4).

*Test over real data.* The experiment to assess the performance over real data takes three sources of input: the Q-matrices (1), the fraction algebra data set of Tatsuoaka as described in 4 (6), and finally a partition tree (5) trained from data generated (4). It outputs a set of refinements from the different partition trees and for each of the three techniques as well (7). Finally, the refinements are compared with the original Q-matrices in (1).

*Test over synthetic data.* For assessing the performance over synthetic data (9), the process is similar, with the main difference that refinements are based on the synthetic test outcome data generated in (3) instead of real data. And the comparison is not done over the Q-matrices in (1), but instead over the permuted Q-matrices in (2), which constitute the ground truth as they are used to generate the data.

### 5.1 Data set size, cross-validation, and the assumption of correctness of expert Q-matrices

As shown in figure 2, synthetic test outcome data (3) is used for both the training of the partition trees and testing over synthetic data. This large data set (see sect. 3.3) leaves little space for over fitting of the partition trees, and therefore the cross-validations bring very small differences in performance: accuracy/RSS error reduction is the same between a cross-validated and a non cross-validated performance assessment

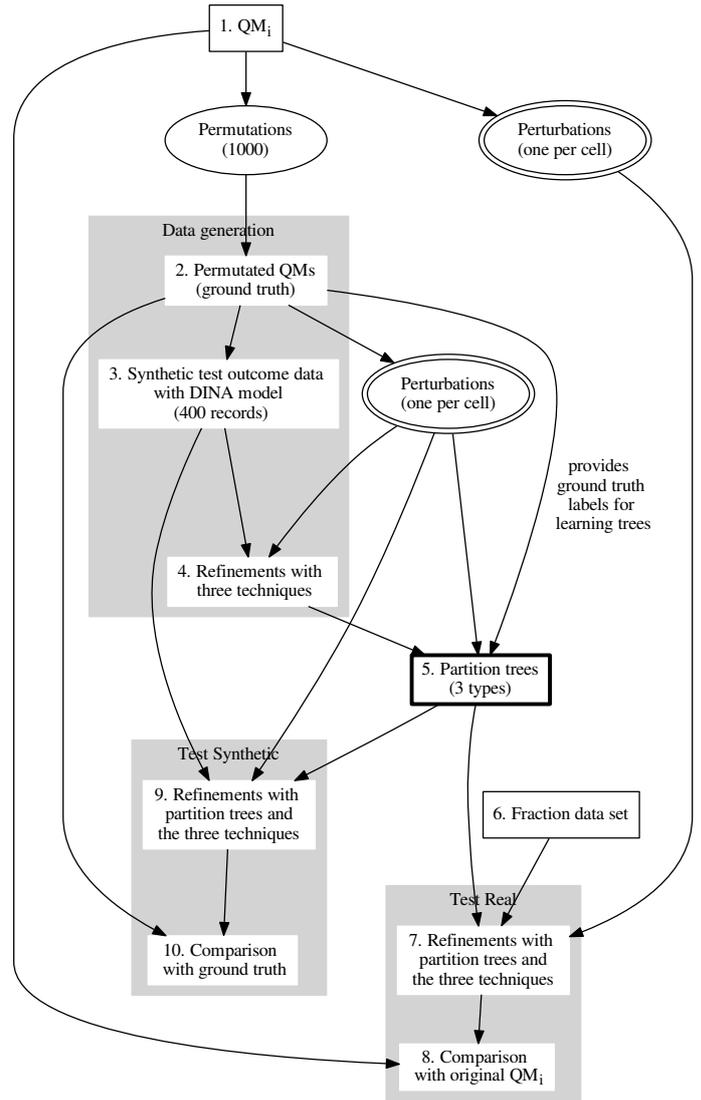


Figure 2: General validation procedure for each Q-matrix ( $QM_i$ ). See section 5 for details.

at the 0.01 level reported in the results below.

However, for real data, the size of the testing data set is much smaller. It varies between 366 (QM-2) and 561 (QM-3), because the test data is based solely on the permutations of the four Q-matrices. But because the test procedure uses partition trees trained from synthetic data, there are no bias issues and cross validation is not required here.

Note also that, for real data, the expert-defined Q-matrix is not necessarily consistent with the (unknown) ground truth. Nevertheless, we consider this Q-matrix as valid and the evaluation of the proposed refinements are made by comparing refinements with expert-defined Q-matrices, as though

Table 3: Results for synthetic data

QM	Technique			Partition tree		
	MinRSS	MaxDiff	ALSC	(1)	(2)	(3)
Accuracy of perturbed cells						
1	0.81	0.47	0.82	0.81	0.88	<b>0.95</b>
2	0.07	0.26	0.36	0.52	0.53	<b>0.83</b>
3	0.96	0.49	0.95	0.99	<b>1.00</b>	<b>1.00</b>
4	0.90	0.49	0.85	0.90	0.92	<b>0.96</b>
$\bar{X}$	0.69	0.43	0.75	0.81	0.83	<b>0.93</b>
Accuracy of non perturbed cells						
1	0.97	0.56	0.44	0.97	0.91	<b>0.99</b>
2	<b>0.99</b>	0.53	0.50	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>
3	0.95	0.26	0.74	0.95	0.94	<b>0.99</b>
4	0.97	0.56	0.44	0.97	0.97	<b>1.00</b>
$\bar{X}$	0.97	0.48	0.53	0.97	0.95	<b>0.99</b>
F-score						
1	0.88	0.51	0.58	0.88	0.90	<b>0.97</b>
2	0.13	0.35	0.42	0.68	0.69	<b>0.90</b>
3	0.96	0.34	0.83	0.97	0.97	<b>1.00</b>
4	0.93	0.52	0.58	0.93	0.94	<b>0.98</b>
$\bar{X}$	0.72	0.43	0.60	0.87	0.87	<b>0.96</b>

they were the ground truth. We should keep in mind that the performance score may be negatively biased if this assumption was false, but for the purpose of comparing the relative techniques performance among themselves, and if we assume that all techniques are equally affected by this bias, then it makes no difference to our relative results.

## 6. PERFORMANCE MEASURES

To measure the performance of the proposed refinements, we use the difference between the original Q-matrix and the proposed refinement of a technique. We use the classification of correct and incorrect refinements introduced in table 1. Cells that are neither perturbed nor incorrectly suggested as refinements by any of the technique are ignored in the analysis (the *true negatives* of table 1, TN). This is the case of the large majority and it also is consistent with the training of the partition tree for which they are also filtered out.

Recovery of a perturbed cell to its original value can be considered as a *recall* measure, whereas the non perturbed cells that are left unchanged can be considered as a *precision* measure. In that respect, we define a performance measure that combines precision and recall of the refinement technique into a single F-score measure:

$$\begin{aligned} \text{F-score} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ &= 2 \times \frac{\text{Acc}_{-P} \times \text{Acc}_P}{\text{Acc}_{-P} + \text{Acc}_P} \end{aligned}$$

where  $\text{Acc}_P$  and  $\text{Acc}_{-P}$  are respectively the accuracy measure of the proposed refinements for the perturbed and non perturbed cells. This measure gives equal weight to both types of accuracies and avoids a bias in favour of the accuracy of the non perturbed cells which can considerably

Table 4: Results for real data

QM	Technique			Partition tree		
	MinRSS	MaxDiff	ALSC	(1)	(2)	(3)
Accuracy of perturbed cells						
1	0.39	0.17	0.52	0.39	0.36	<b>0.67</b>
2	0.35	0.09	0.56	0.60	0.62	<b>0.64</b>
3	0.27	0.09	0.36	0.61	<b>1.00</b>	0.88
4	0.42	0.11	0.58	0.42	0.48	<b>0.61</b>
$\bar{X}$	0.36	0.12	0.51	0.51	0.62	<b>0.70</b>
Accuracy of non perturbed cells						
1	0.45	<b>0.68</b>	0.56	0.45	0.38	0.60
2	0.93	0.93	0.28	0.94	0.94	<b>0.97</b>
3	0.64	0.83	0.42	0.69	0.76	<b>0.78</b>
4	0.55	<b>0.89</b>	0.32	0.55	0.52	0.51
$\bar{X}$	0.52	0.68	0.32	0.62	0.62	<b>0.68</b>
F-score						
1	0.42	0.27	0.54	0.42	0.37	<b>0.63</b>
2	0.50	0.17	0.37	0.73	0.74	<b>0.77</b>
3	0.38	0.16	0.39	0.64	<b>0.86</b>	0.83
4	0.48	0.20	0.42	0.48	0.50	<b>0.56</b>
$\bar{X}$	0.45	0.20	0.43	0.57	0.62	<b>0.70</b>

outweigh in number the single perturbed cell, even after filtering out non-perturbed cells that are left unchanged.

## 7. RESULTS

The results are reported in tables 3 and 4. The format of these tables first described below.

### 7.1 Description

The respective results of the four Q-matrices (column QM) in table 2 are reported. They correspond to a single run (real data can vary a few percentage points by run, but it is practically stable for synthetic data due to the large number of cases). The accuracy of refinement for perturbed and non perturbed cells are reported separately, followed by the F-score which combines both types of accuracy. The averages of the four matrices for each of these three performance measures is also reported as  $\bar{X}$ .

The accuracy and F-score of each individual technique is reported under columns **MinRSS**, **MaxDiff**, and **ALSC**.

The three columns under **Partition tree** correspond to the performance as a function of different factors used for building the tree:

- (1) **MinRSS + MaxDiff + ALSC**. Only the output of the three refinement techniques is considered.
- (2) **MinRSS + MaxDiff + ALSC + SR + SC**. The number of skills per row (SR) and skills per column (SC) of the target cell are taken into account in addition to the output of each technique. If some technique performs better under some combination of SR and SC, this tree will be able to take these factor into account.

(3) **MinRSS + MaxDiff + ALSC + SR + SC + Stickiness.MinRSS + Stickiness.MaxDiff + Stickiness.ALSC.** The tendency of a cell to be a false positive for the MinRSS and ALSC methods are added. The Stickiness factor with MaxDiff is omitted here because it did not yield improvements.

## 7.2 Synthetic data

The results for synthetic data in 3 show large differences between the different matrices and across the individual techniques.

The MinRSS method is clearly superior in terms of general accuracy, except for the 5-skills Q-matrix where it can only identify the perturbed cell 7% of the time, and which brings its average below the ALSC technique. However, because it introduces fewer *false positives* (incorrect refinements) than other techniques, it outperforms the other two methods on the F-Score.

On average, the ALSC technique is good at identifying the perturbed cell with a 75% average, but it also tends to introduce more false positives and consequently obtains a lower global F-score than MinRSS .

Another noticeable result is that the results for QM 3 are very good, in particular for the partition trees which have perfect performance (rounding at the second decimal). This is likely attributed to the fact that it defines a single-skill mapping.

Turning to the main questions addressed in this study, the results of partition tree (1), which uses only the three techniques' output, is equal or better on all scores than any individual one. This confirms the initial hypothesis for synthetic data. Furthermore, the inclusion of factors (partition trees (2) and (3)) also substantially improves all scores, confirming the other hypothesis that some techniques perform better under a combination of factors and that the partition tree is effectively able to take advantage of this information. The stickiness factor is by far the most effective.

## 7.3 Real data

The results over the real data reported in table 4 show the same trends as the synthetic data, but bring less pronounced improvements. They also support both hypothesis.

We do find an exception with the non perturbed cells where the MaxDiff accuracy is above the partition trees (1) and (2) and close to (3). This is mainly due to the fact that more "false positives" are generated by the MinRSS and ALSC techniques for real data than for synthetic data, whereas the MaxDiff technique outputs very few changes in both contexts. That observation is consistent with the results in [6].

The balance between true positives and true negatives illustrates why the F-score should be the reference: a perfect score could be obtained over the accuracy of non perturbed cells if no changes are always suggested, but that would make such refinement technique useless.

Therefore, turning to the F-scores, the tendencies are highly

consistent with the synthetic data. The F-score of the best performer, 0.41 of MinRSS , is improved to 0.55 with the combination of the three techniques, and to 0.66 when all factors are included in the partition tree.

## 8. DISCUSSION

The results of the above experiments show that the combination of Q-matrix refinement techniques using a partition tree can bring substantial improvements over the best performance of the individual techniques. For synthetic data the average best F-score of the MinRSS technique, 0.72, is improved to 0.96, and for real data it is raised from 0.41 to 0.66. These results represent a 86% and 55% error reduction for the F-score of the synthetic and the real data respectively (error reduction =  $1 - (1 - F')/(1 - F)$ , where  $F$  is the initial F-score and  $F'$  is the improved F-score).

In practical terms, if the best technique finds an error in a Q-matrix 5 out of 10 times, an error reduction of 40% represents an increase from 5, to 7 out of 10 times, and the same ratio applies to false errors reduction. And these figures rest on the assumption that we would know which technique is the best, whereas according to table 4's results the best technique varies across Q-matrices.

Another positive note on the results is that the partition tree F-scores are more stable across Q-matrices and are systematically better than any individual technique when all factors are taken into account (partition tree 3). This regularity incurs that, at least in the space of Q-matrices surveyed, one can safely choose partition tree refinements without concerns that, maybe, another technique could deliver better refinements for a specific Q-matrix.

In spite of these encouraging results, limitations and issues remain.

One limit is that the results are from a single 11 items set, and from a single domain. We can reasonably believe that the results vary across contexts and more investigation is required to assess this variability.

Another limitation is the models investigated in the current study use *static* student data: they assume that skill mastery does *not* change for a single student. This assumption is false for most data gathered in learning environments, where students take on exercises as they learn and are being assessed throughout the learning process. This type of data can be labeled as *dynamic* item outcome data because a student will be in different states of skills mastery as learning occurs.

In order to effectively use the existing techniques of Q-matrix refinement, we would need to be able to detect the moment when the state of skill mastery changed. Failure to do so would create noise in the data and impair the effectiveness of these techniques. Fortunately, substantial progress has been done in the recent decade or two towards detecting the moment of learning, such as the large body of work on Bayesian Knowledge Tracing and Tensor factorization (for eg. [1, 18]). We can also cite the work of [14] who refer to a time-varying skills matrix for students and test their approach on synthetic data. But apart from this recent

contribution, little work has been done on using this type of data for refining a Q-matrix, and we can only expect existing techniques to under perform with dynamic student data.

## 9. ACKNOWLEDGEMENTS

This research was funded under the NSERC Discovery grant of the first author.

## 10. REFERENCES

- [1] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting learning moment-by-moment. *International Journal of Artificial Intelligence in Education*, 21(1):5–25, 2011.
- [2] C.-Y. Chiu. Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, 2013.
- [3] J. De La Torre. An empirically based method of Q-Matrix validation for the DINA model: Development and applications. *Journal of educational measurement*, 45(4):343–362, 2008.
- [4] L. T. DeCarlo. On the analysis of fraction subtraction data: The DINA model, classification, latent class sizes, and the Q-Matrix. *Applied Psychological Measurement*, 35:8–26, 2011.
- [5] L. T. DeCarlo. Recognizing uncertainty in the Q-Matrix via a bayesian extension of the DINA model. *Applied Psychological Measurement*, 36(6):447–468, 2012.
- [6] M. C. Desmarais, B. Beheshti, and P. Xu. The refinement of a Q-matrix: Assessing methods to validate tasks to skills mapping. In *7th Educational Data Mining Conference*, pages 208–311, 2014.
- [7] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-Matrices. In *6th International Conference, AIED 2013, Memphis, TN, USA*, pages 441–450, 2013.
- [8] C. Goutte, G. Durand, and S. Léger. Towards automatic description of knowledge components. In *Proceedings of the 8th International Conference on Educational Data Mining*, page (to appear), Madrid, Spain 2015.
- [9] R. A. Henson, J. L. Templin, and J. T. Willse. Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2):191–210, 2009.
- [10] B. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.
- [11] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. Technical report, Carnegie-Mellon University, Human Computer Interaction Institute, 2011.
- [12] J. Nižnan, R. Pelánek, and J. Řihák. Mapping problems to skills combining expert opinion and student data. In *Mathematical and Engineering Methods in Computer Science*, pages 113–124. Springer, 2014.
- [13] J. Nižnan, R. Pelánek, J. Řihák, et al. Using problem solving times and expert opinion to detect skills. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 433–434, London, UK 2014.
- [14] S. Oeda, Y. Ito, and K. Yamanishi. Extracting latent skills from time series of asynchronous and incomplete examinations. In *Proceedings of EDM 2014, The 7th International Conference on Educational Data Mining*, pages 367–368, London, UK 2014.
- [15] A. Robitzsch, T. Kiefer, A. George, A. Uenlue, and M. Robitzsch. Package CDM. 2012.
- [16] L. Rokach. *Data mining with decision trees: theory and applications*. World scientific, 2007.
- [17] K. Tatsuoka, U. of Illinois at Urbana-Champaign. Computer-based Education Research Laboratory, and N. I. of Education (US). *Analysis of errors in fraction addition and subtraction problems*. Computer-based Education Research Laboratory, University of Illinois, 1984.
- [18] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. Matrix and tensor factorization for predicting student performance. In A. Verbraeck, M. Helfert, J. Cordeiro, and B. Shishkov, editors, *CSEdu 2011 - Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011*, pages 69–78. SciTePress, 2011.
- [19] T. Therneau, B. Atkinson, B. Ripley, and M. B. Ripley. Package rpart, 2014.

# On the Performance Characteristics of Latent-Factor and Knowledge Tracing Models

Severin Klingler  
Department of Computer  
Science  
ETH Zurich, Switzerland  
kseverin@inf.ethz.ch

Tanja Käser  
Department of Computer  
Science  
ETH Zurich, Switzerland  
kaesert@inf.ethz.ch

Barbara Solenthaler  
Department of Computer  
Science  
ETH Zurich, Switzerland  
sobarbar@inf.ethz.ch

Markus Gross  
Department of Computer  
Science  
ETH Zurich, Switzerland  
grossm@inf.ethz.ch

## ABSTRACT

Modeling student knowledge is a fundamental task of an intelligent tutoring system. A popular approach for modeling the acquisition of knowledge is Bayesian Knowledge Tracing (BKT). Various extensions to the original BKT model have been proposed, among them two novel models that unify BKT and Item Response Theory (IRT). Latent Factor Knowledge Tracing (LFKT) and Feature Aware Student knowledge Tracing (FAST) exhibit state of the art prediction accuracy. However, only few studies have analyzed the characteristics of these different models. In this paper, we therefore evaluate and compare properties of the models using synthetic data sets. We sample from a combined student model that encompasses all four models. Based on the true parameters of the data generating process, we assess model performance characteristics for over 66'000 parameter configurations and identify best and worst case performance. Using regression we analyze the influence of different sampling parameters on the performance of the models and study their robustness under different model assumption violations.

## Keywords

Knowledge Tracing, Item Response Theory, synthetic data, predictive performance, robustness

## 1. INTRODUCTION

A fundamental part of an intelligent tutoring system (ITS) is the student model. Task selection and evaluation of the student's learning progress are based on this model, and therefore it influences the learning experience and the learning outcome of a student. Thus, accurately modeling and predicting student knowledge is essential.

Approaches for student modeling are usually based on two popular techniques: Item Response Theory (IRT) [36] and Bayesian Knowledge Tracing (BKT) [9]. The concept of IRT assumes that the probability of a correct response to an item is a mathematical function of student and item parameters. The Additive Factors Model (AFM) [7, 8] fits a learning curve to the data by applying a logistic regression. Another technique called Performance Factors Analysis (PFA) [27] is based on the Rasch item response model [12]. BKT models student knowledge as a binary variable that can be inferred by binary observations. Performance of the original BKT model has been improved by using individualization techniques such as modeling the parameters by student and skill [23, 35, 39] or per school class [34]. Clustering approaches [25] have also proven successful in improving the prediction accuracy of BKT. Furthermore, hybrid models combining the approaches of IRT and BKT have been proposed. In [17] a dynamic mixture model has been presented to trace performance and affect simultaneously. The KT-IDEM model extends BKT by introducing item difficulty parameters [22]. Other work focused on individualizing the initial mastery probability of BKT by using IRT [38]. Logistic regression has also been used to integrate subskills into BKT [37]. Recently, two models have been introduced which synthesize IRT and BKT. Latent Factor Knowledge Tracing (LFKT) [18] individualizes the guess and slip probabilities of BKT based on student ability and item difficulty. Feature Aware Student Knowledge Tracing (FAST) [14] generalizes the individualized guess and slip probabilities to arbitrary features.

Lately, the analysis of properties of BKT has gained increasing attention. It has been shown [5] that learning BKT models exhibits fundamental identifiability problems, i.e., different model parameter estimates may lead to identical predictions about student performance. This problem was addressed by using an approach that biases the model search by Dirichlet priors to get statistically reliable improvements in predictive performance. [33] extended this work by performing a fixed point analysis of the solutions of the BKT learning task and by deriving constraints on the range of parameters that lead to unique solutions. Furthermore, it has been shown that the parameter space of BKT models

can be reduced using clustering [30]. Other research focused on analyzing convergence properties [24] of the expectation maximization algorithm (EM) for learning BKT models and exploring parameter estimates produced by EM [15]. It has been shown that convergence in the log likelihood space does not necessarily mean convergence in the parameter space. [11] have studied how good BKT is at predicting the moment of mastery. Different thresholds to assess mastery and their corresponding lag, i.e., the number of tasks that BKT needs to assess mastery (after mastery has already been achieved), have been investigated. Using multiple model fitting procedures, BKT has been compared to PFA [13]. While no differences in predictive accuracy between the models have been reported, it has been shown that for knowledge tracing EM achieves significantly higher predictive accuracy than Brute Force. Findings from other studies, however, suggest the opposite [1, 2]. In [4], upper bounds on the predictive performance have been investigated by employing various cheating models. It has been concluded that BKT and PFA perform close to these limits, suggesting that other factors such as robust learning or optimal waiting intervals should be considered to improve tutorial decision making. The predictive performance of LFKT and FAST has been compared to KT and IRT models in [19]. The evaluation is based on data from different intelligent tutoring systems.

In this work, we are interested in the properties of hybrid approaches combining latent factor and knowledge tracing models. In extension to previous work and especially to [19], we empirically evaluate the performance characteristics of the two recent hybrid models LFKT and FAST on synthetic data and compare them to the underlying approaches of BKT and IRT. We sample from a combined student model that encompasses all four models. By using synthetic data generated from the combined model, we show the robustness of the models under breaking model assumptions. By evaluating the models on 66'000 different parameter configurations we are able to rigorously explore the parameter space to demonstrate the relative performance gain between models for various regions of the parameter space. Our findings show that for the generated data sets FAST significantly outperforms all other methods for predicting the task outcome and that BKT is significantly better than FAST and LFKT at predicting the latent knowledge state. Furthermore we are able to identify the influence of different properties of a data set on model performance using regression and show best and worst case performances of the models.

## 2. INVESTIGATED MODELS

In an intelligent tutoring system a student is typically presented with a set of tasks to learn a specific skill. For each student  $n$  the system chooses at time  $t$  an item  $i$  from a set of items corresponding to a particular skill. The system then observes the answer  $y_{n,t}$  of the student, which is assumed to be binary in this work. In the following, we briefly present four common techniques to model various latent states of the student and the tutoring environment.

**BKT.** Bayesian Knowledge Tracing (BKT) [9] models the knowledge acquisition of a single skill and is a special case of a Hidden Markov Model (HMM) [29]. BKT uses two latent states (*known* and *unknown*) to model if a student  $n$  has mastered a particular skill  $k_{n,t}$  at time  $t$ , and two

observable states (*correct* and *incorrect*) to represent the outcome of a particular task. Therefore, the probabilistic model can be fully described by a set of five probabilities. The initial probability of knowing a skill a-priori  $p(k_{n,0})$  is denoted by  $p_I$ . The transition from one knowledge state  $k_{n,t-1}$  to the next state  $k_{n,t}$  is described by the probability  $p_L$  of transitioning from the *unknown* latent state to the *known* state and the probability  $p_F$  of transitioning from the *known* to the *unknown* state:

$$p(k_{n,t}) = k_{n,t-1}(1 - p_F) + (1 - k_{n,t-1})p_L. \quad (1)$$

In the case of BKT,  $p_F$  is fixed at 0. Finally, the task outcomes  $y_{n,t}$  are modeled as

$$p(y_{n,t}) = k_{n,t}(1 - p_S) + (1 - k_{n,t})p_G, \quad (2)$$

where  $p_S$  denotes the *slip probability*, which is the probability of solving a task incorrectly despite knowing the skill, and  $p_G$  is the *guess probability*, which is the probability of correctly answering a task without having mastered the skill. Learning the parameters for a BKT model is done using maximum likelihood estimation (MLE).

**IRT.** Item Response Theory (IRT) [36] models the response of a student to an item as a function of latent student abilities  $\theta_n$  and latent item difficulties  $d_i$ . The simplest form of an IRT model is the Rasch model, where each student  $n$  and each item  $i$  are treated independently. The outcome  $y_{n,t}$  at time  $t$  is modeled using the logistic function

$$p(y_{n,t}) = \left(1 + e^{-(\theta_n - d_i)}\right)^{-1}. \quad (3)$$

A student with an ability of  $\theta_n = d_i$  has a 50% chance of getting item  $i$  correct. In contrast to BKT, IRT does not model knowledge acquisition. The model parameters for the Rasch model are learned using EM.

**LFKT.** The Latent Factor Knowledge Tracing (LFKT) [18] model combines BKT and IRT using a hierarchical Bayesian model. On the basis of the BKT model, slip and guess probabilities are individualized based on student ability and item difficulty as

$$p_{G_{n,t}} = \left(1 + e^{-(d_i - \theta_n + \gamma_G)}\right)^{-1} \quad (4)$$

$$p_{S_{n,t}} = \left(1 + e^{-(\theta_n - d_i + \gamma_S)}\right)^{-1}, \quad (5)$$

where  $\gamma_G$  and  $\gamma_S$  are offsets for the guess and slip probabilities. The model is fit by calculating Bayesian parameter posteriors using Markov Chain Monte Carlo.

**FAST.** Feature Aware Student Knowledge Tracing (FAST) [14] allows for unification of BKT and IRT as well, but generalizes the individualized slip and guess probabilities to arbitrary features. Given a vector of features  $\mathbf{f}_{n,t}$  for a student  $n$  at time  $t$  the adapted emission probability reads as

$$p(y_{n,t}) = \left(1 + e^{-(\boldsymbol{\omega}^T \mathbf{f}_{n,t})}\right)^{-1}, \quad (6)$$

where  $\boldsymbol{\omega}$  is a vector of learned feature weights. If a set of binary indicator functions for the items and the students are used, FAST is able to represent the item difficulties  $d_i$  and student abilities  $\theta_n$  from the IRT model. The parameters are fit using a variant of EM [6].

### 3. SYNTHETIC DATA GENERATION

Synthetic data is needed to have ground truth about the underlying data generating model, which enables the experimental evaluation of various properties of a model.

The sampling procedure starts by generating  $N$  student abilities  $\theta_n$  from a normal distribution  $N(0, \sigma)$ . Then, it generates  $I$  item difficulties  $d_i$  from a uniform distribution  $U(-\delta, \delta)$ . Based on the initial probability  $p_I$  and the learn probability  $p_L$  a sequence of knowledge states  $k_{n,0}, k_{n,1}, \dots, k_{n,T}$  is sampled based on (1) and we therefore simulate data from only one skill. The time  $t^*$  at which  $k_{n,t^*} = 1$  for the first time is considered as the moment of mastery. The number of sampled knowledge states is then given as  $T = t^* + L$ , where  $L$  denotes the lag of the simulated mastery learning system. For each student we generate a random sequence of items, i.e., item indices  $i$ . Arbitrary features from the training environment, such as answer times, help calls, problem solving strategy, engagement state of the student and gaming attempts, can have an influence on the performance of a student. To simulate those influences in a principled way, a single feature  $f$  is added to the data generating model with a varying feature weight  $\omega$  (and thus varying correlation to the task outcomes  $y_{n,t}$ ).

Based on these quantities, we sample the observations  $y_{n,t}$  from a Bernoulli distribution with probability

$$p(y_{n,t}) = \left(1 + e^{-(\theta_n - d_i - \log \gamma_{n,t} + \omega f_{n,t})}\right)^{-1}, \quad (7)$$

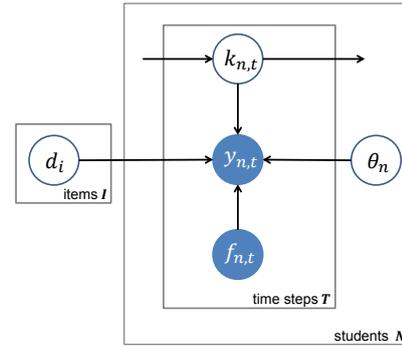
where

$$\gamma_{n,t} = (k_{n,t}(1 - p_S) + (1 - k_{n,t})p_G)^{-1} - 1.$$

Figure 1 gives a graphical overview of the described sampling procedure. Our sampling model has the following nine parameters:  $p_I, p_L, p_S, p_G, \delta, \sigma, \omega, I, N$ . The described sampling procedure allows sampling of data that exactly matches the model assumptions of all four models. To sample BKT data we set  $\delta = \sigma = \omega = 0$  and (7) simplifies to the standard BKT formulation. By setting  $p_S = p_G = 0.5$  and  $\omega = 0$  we can sample from an IRT model. To sample from an LFKT model we set  $\omega = 0$  and for FAST none of the parameters are restricted.

### 4. EXPERIMENTAL SETUP

**Parameter space.** We generated a vast number of parameter configurations in order to analyze the four models. The set of parameter configurations has been carefully designed to match real world conditions. The BKT parameters ( $p_I, p_G, p_S, p_L$ ) are based on the parameter clusters found on real world data [30]. Using a normal distribution with a standard deviation of 0.02, we sampled up to 30 points (depending on the cluster size) around each cluster mean. According to common practice [16] we scaled the student abilities  $\theta_n$  to have a mean of 0 and a variance of 1 and therefore  $\sigma = 1$ . We sampled the parameter  $\delta$  (determining the range of the item difficulties) uniformly from  $[0, 3]$  (according to [16]). Despite simulating only one skill, we varied the item difficulties to account for the fact that skill models tend to be imperfect in practice [7, 32, 20]. In accordance to the item difficulties, the feature weight  $\omega$  was varied uniformly across  $[0, 1.5]$ . Feature values  $f_{n,t}$  were sampled from the uniform distribution  $U(-1, 1)$ .



**Figure 1: Combined student model used for synthetic data generation. The model corresponds to LFKT with the addition of a single feature. The relative dependencies of the observable nodes (blue) and the latent nodes (white) are shown.  $k_{n,s}$  denotes the latent knowledge state,  $d_i$  the item difficulty,  $\theta_n$  the student ability,  $y_{n,t}$  the observation, and  $f_{n,t}$  the feature value.**

For every parameter configuration we generated five folds with  $N = 300$  simulated students. Each fold was randomly split up into two parts of equal number of students. The first part was used as training data and the second part for testing. Therefore, the training data did contain unseen students only. As we simulated data from a mastery learning environment the number of tasks simulated for each student was determined by the moment of mastery. Based on the results presented by [11], we set the lag of the simulated system to  $L = 4$  tasks from the moment of mastery. We simulated  $I = 15$  different items with random item order.

In total, we generated 66'000 parameter configurations for  $p_I, p_G, p_S, p_L, \delta, \omega$ , this amounts total evaluation time (training and test) of 1'280 hours and 1'351 hours for LFKT and FAST respectively. The evaluation time for the BKT was 99 minutes and all configurations were evaluated in 58 minutes for the IRT model.

**Implementation.** To train BKT models we used our custom code that trains BKT using the Nelder-Mead simplex algorithm minimizing the log-likelihood. We thoroughly tested our implementation against the BKT implementation of [39]. The IRT models were fit by joint maximum likelihood estimation [21] implemented in the psychometrics library<sup>1</sup>. FAST using IRT features was shown to be equivalent to LFKT except for the parameter estimation procedure [19]. As this work did not investigate different parameter estimation techniques, both models were trained and evaluated using the publicly available FAST student modeling toolkit<sup>2</sup>.

### 5. RESULTS AND DISCUSSION

Using the generated data, we investigated the performance characteristics of the four models and evaluated their predictive power and robustness under varying parameter configurations. For our results we generated 66'000 parameter

<sup>1</sup>An open source Java library for measurement, available at <https://github.com/meyerjp3/psychometrics>.

<sup>2</sup><http://ml-smores.github.io/fast/>

configurations, and for each of them we generated synthetic data for 1'500 students. Note that there are many ways to characterize performance differences among student models and we only cover a subset of these possibilities.

## 5.1 Error Metrics

The right choice of error metrics when evaluating student models has recently gained increased interest in the EDM community. In [28] some of the common error metric choices are discussed, highlighting possible issues with the accuracy and area under the ROC curve (AUC) measure. Correlations between various performance metrics and the accuracy of predicting the moment of mastering a skill has been investigated in [26], showing that the F-measure (equaling to the harmonic mean of precision and recall) and the recall are two metrics with a high correlation to the accuracy of knowledge estimation. The root mean squared error (RMSE) and log-likelihood, on the other hand, are well suited if one wants to recover the true learning parameters. Similarly, [10] concluded from results of 26 synthetic data sets that RMSE is better at fitting parameters than the log-likelihood.

In line with this previous work we investigated correlations between accuracy, RMSE and F-measure across all four models. For this, all models were trained and evaluated on data using 66'000 different parameter configurations. All metrics are strongly correlated  $|\rho| > 0.75, p \ll 0.001$ . Our inspections of the metric correlations revealed no significant differences in the metric correlations among the different models. Thus, to a large extent the measures capture equal characteristics for the models we considered in this work. In the following, we therefore focus our analysis on the RMSE measure.

## 5.2 Model Comparison

**Overall Performance.** In a first step we investigated the overall performance of the models. For every parameter configuration, we calculated the average RMSE over the five generated folds. Table 1 summarizes the parameters for the best and worst data set for every model when model assumptions are met (see Section 3). Results show that all models that model a knowledge state (all except IRT) perform best if the slip probability is low and the guess probability is high. This leads to a data set that exhibits a high ratio of correct observations. IRT performs best on data that has very distinguished item difficulties ( $\delta$  is high). Notably the best performance of FAST is achieved on a data set without features ( $\omega = 0$ ). We assume that this is due to the decreased complexity of the data set, compared to one that exhibits high  $\omega$ . Consistently, worst case data sets exhibit high symmetric values for guess and slip probabilities. In the case of LFKT and FAST worst case data sets additionally do not distinguish between items (difficulty range  $\delta = 0$ ) and for FAST the feature weights are low.

We then performed the non-parametric Friedman test over all parameter configurations to assess performance differences between the models. We found that there is a statistically significant difference in the performance of the models ( $\chi^2(3) = 13'065, p < 0.0001$ ). Performing a post-hoc analysis using Scheffe's S procedure [31] shows all model differences to be significant at  $p < 0.0001$  with mean ranks of 1.7156, 2.3017, 2.6898 and 3.2929 for FAST, LFKT, BKT,

**Table 1: Parameters of best and worst case data sets for each model. We only considered data sets that meet the model assumptions. Parameters denoted with \* are fixed according to the model assumptions (see Section 3).**

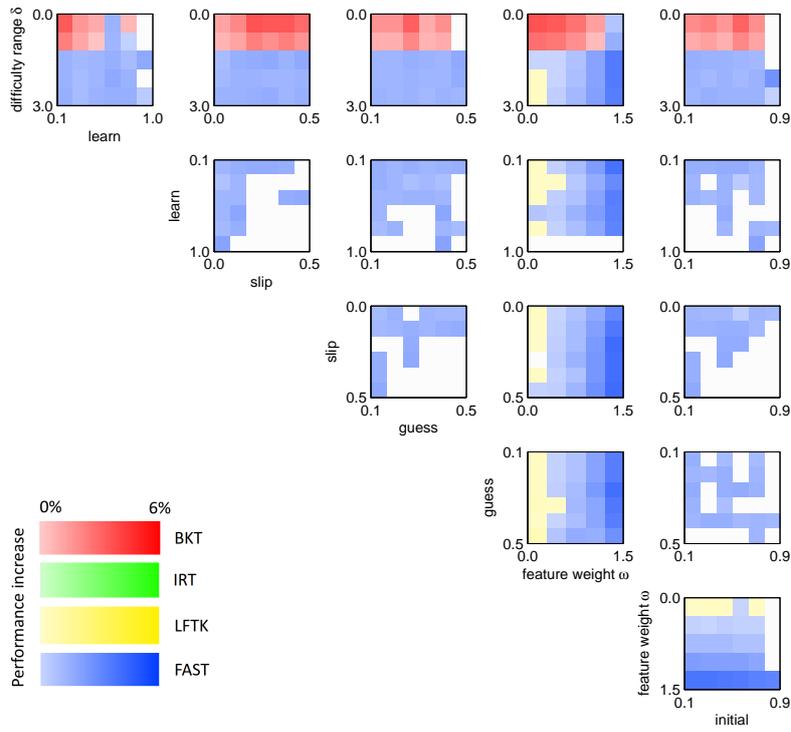
Model	$\delta$	$pI$	$pL$	$pS$	$pG$	$\omega$	RMSE
<b>BKT</b>							
Best	0.00*	0.71	0.41	0.01	0.47	0.00*	0.25
Worst	0.00*	0.10	0.12	0.50	0.49	0.00*	0.48
<b>IRT</b>							
Best	3.00	0.10	0.08	0.50*	0.50*	0.00*	0.42
Worst	0.00	0.10	0.10	0.50*	0.50*	0.00*	0.50
<b>LFKT</b>							
Best	0.75	0.69	0.40	0.01	0.46	0.00*	0.25
Worst	0.00	0.53	0.16	0.28	0.29	0.00*	0.51
<b>FAST</b>							
Best	0.75	0.67	0.40	0.01	0.46	0.00	0.25
Worst	0.00	0.56	0.16	0.28	0.28	0.00	0.51

and IRT, respectively. FAST therefore significantly outperforms the other methods on our synthetic data sets. In [19] IRT performed not significantly worse than LFKT and FAST on four different data sets. The good performance of IRT was attributed to the deterministic item ordering that allows IRT to infer knowledge acquisition confounded with item difficulty. Our results support this hypothesis as in our synthetic data set the items are in random order and IRT exhibits the worst overall performance.

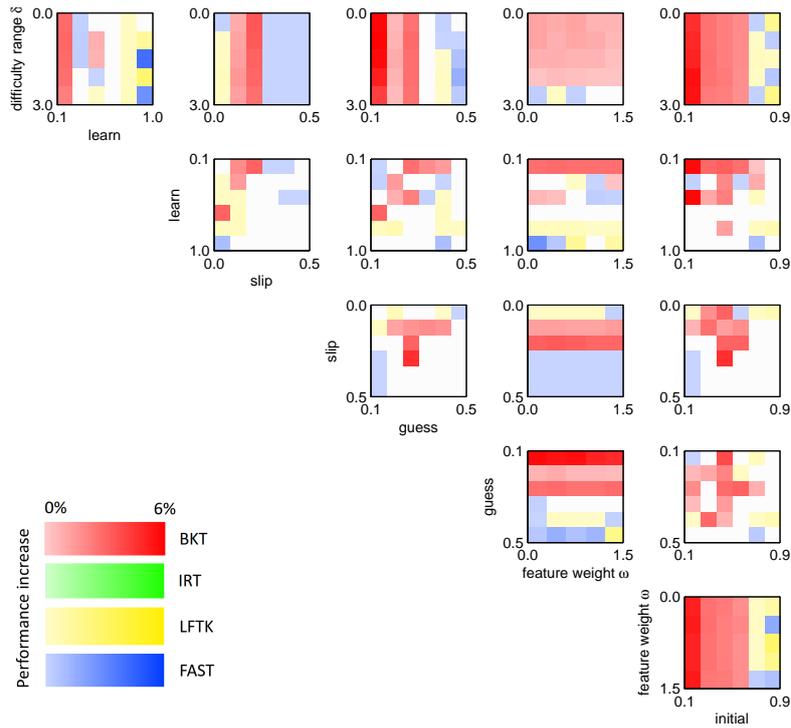
**Parameter Space Investigation.** To gain a better understanding of the performance characteristics of the different models, we analyzed their performances across the parameter space. For every pair of parameters  $p_i$  and  $p_j$ , we divided the parameter configurations into bins with similar values for  $p_i$  and  $p_j$ . We used five bins for each parameter ( $p_i$  and  $p_j$ ) resulting in a total of 25 bins. Performance of each model was assessed by calculating the mean RMSE for each bin. Significance of the observed performance differences was computed using the Friedman test and  $p < 0.05$ .

Figure 2a shows the relative performance of the best model for each parameter pair. The models are color-coded: BKT is shown in red, IRT in green, LFKT in yellow, and FAST in blue. The color gradient indicates the relative improvement of the winning model over the second best model, where darker colors indicate higher values. White-colored areas indicate that there is no significant difference between the models. The plot shows that FAST is robust to parameter variations and outperforms the other models in large parts of the parameter space. In parts with low feature weights, i.e., where the feature  $f$  shows only a low correlation with task outcomes, LFKT outperforms FAST. When the variance  $\delta$  of item difficulties  $d_i$  is low, BKT is the best model. A low variance in  $d_i$  implies a good skill model, with all tasks having approximately the same difficulty.

In contrast to Figure 2a, where we assessed the prediction



(a) Relative improvement in task outcome prediction (RMSE).



(b) Relative improvement in knowledge state prediction (RMSE).

**Figure 2: Best performing models (RMSE) regarding prediction of task outcomes (a) and knowledge state prediction (b). The color for each bin indicates the best performing model, averaged over all other parameters. We investigated BKT (red), IRT (green), LFTK(yellow), and FAST(blue). White-colored bins exhibit no significant difference in model performance. The color brightness indicates the relative improvement of the best performing model over competing models, with dark colors referring to higher values. FAST is robust to parameter variations and outperforms the other models in large parts of the parameter space when predicting task outcomes (a). BKT is the best model if the variance of the item difficulty is low (a). BKT is superior to the other models in large parts of the parameter space when predicting knowledge states (b).**

of task outcomes, we analyzed the quality of the prediction of knowledge states  $k_{n,t}$  using the RMSE in Figure 2b. Ultimately, we want to predict whether a student has mastered a skill or not [26, 3]. The plot uses the same parameter pairs and color codings as Figure 2a. Interestingly, LFKT and FAST are not superior to BKT when it comes to prediction of the latent state. The additional parameters that LFKT and FAST use have a direct influence on the predicted task outcomes and therefore improve performance when predicting task outcomes. They have, however, no direct influence on the latent state  $k_{n,t}$  of the model.

**Robustness.** Next, we tested the robustness of the different models against each other. We generated ideal data (meeting the model assumptions) for all the models and then interpolated the parameter values between these ideal cases. The classes of data sets that meet the model assumptions for the four models are described in Section 3. From every class of data sets, we selected the extreme case with the least amount of noise. In the following, we describe these cases.

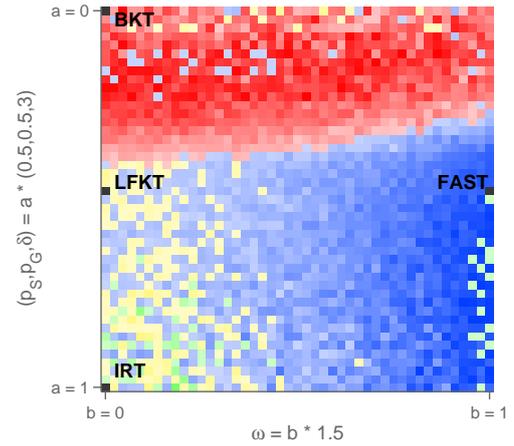
For BKT, data is generated using  $\delta = \omega = 0$ , assuming a perfect skill model (all tasks with same difficulty) and setting the influence of additional (not captured) features to 0. Furthermore, we removed the randomness by setting  $p_G = p_S = 0$ . For IRT, the extreme case data was generated using  $p_G, p_S = 0.5$ ,  $\omega = 0$  and by additionally setting  $\delta = 3$ . As LFKT is a combination of IRT and BKT, we set the parameters to  $p_G, p_S = 0.25$  and  $\delta = 1.5$ . Furthermore, we set  $\omega = 0$ , again assuming no influence of not captured features. For FAST we used the same parameters as for LFKT, but additionally introduced a feature influence by setting  $\omega = 1.5$ . We linearly interpolated the parameter space in-between these extreme cases to assess model robustness when model assumptions are violated. Figure 3 displays the model with best RMSE in this subspace that contains the extreme (ideal) cases, where  $p_L$  and  $p_I$  are averaged over the BKT parameter clusters presented in [30]. From these results, we can see that BKT tends to be robust to increased feature influence as long as  $p_G, p_S \leq 0.15$ . If the feature weight  $\omega > 0.75$ , FAST outperforms all the other classifiers. For large differences in item difficulties and large guess and slip probabilities, LFKT has a slight advantage over IRT.

### 5.3 Parameter Influence

To analyze the influence of the model parameters on the performance of the student models, we used linear regression to predict the RMSE based on the parameters of the sampling model. This allowed us to identify statistically significant correlations between the sampling parameters and the performance of the models despite the high dimensionality of the parameter space.

The sampling parameters have a direct influence on the ratio of correct observations in the data, e.g., a high learning probability with low guess and slip parameters leads to a high ratio of correct observations. Further, if the parameters model fast learners then the average number of tasks tends to be low since we are simulating a mastery learning environment. The three models IRT, LFKT and FAST which explicitly model items are sensitive to this kind of lacking data, as by having fewer observed items per student the estimation of item difficulty becomes more difficult. To

Best performing model under breaking assumptions



**Figure 3: Relative model performance on ideal data sets generated by linearly interpolating between parameters. The colors refer to the models BKT (red), IRT (green), LFKT (yellow) and FAST (blue). The color gradient indicates the relative performance as in Figure 2a. BKT and FAST are more robust to the invalid assumptions of our experiment than IRT and LFKT.**

investigate the effect of both factors, we added the two variables *correct ratio* and *average number of tasks* as predictors to the regression model. In order to make correlation coefficients comparable, all sampling parameters have been normalized to have mean 0 and standard deviation 1.

Figure 4 shows the regression coefficients for all four models, with red and green denoting statistically significant and not significant coefficients, respectively. The variables *correct ratio* and *average number of tasks* have the largest influence on the RMSE. Both effects are significant and positive (reducing the RMSE). A larger range of item difficulties  $\delta$  has a positive influence on the performance of all models except for the BKT model. This is expected as BKT does not account for variations in item difficulty and thus larger variations in item difficulties are treated as noise by BKT, which makes prediction harder. IRT, LFKT and FAST, on the other hand, benefit from larger variations. We assume that this is due to the better identifiability of the effects of the different items. Interestingly, increasing the feature range  $\omega$  has no significant negative effect for the models that do not take features into account (BKT, IRT, LFKT), but has a positive effect for FAST. The initial probability and the learning probability have a small negative and small positive effect on performance, respectively. While these coefficients are partially significant they have very small magnitude. The positive effect of the slip probability  $p_S$  for all models except BKT (the effect is not significant) is rather surprising. However, the effect of a high slip probability in our sampling model is that it weakens the influence of the latent knowledge state on the task outcomes. This could explain the positive influence for models that estimate item difficulty, since the difficulty estimates are less convoluted with effects from the knowledge state. Further work is needed to prove this effect.

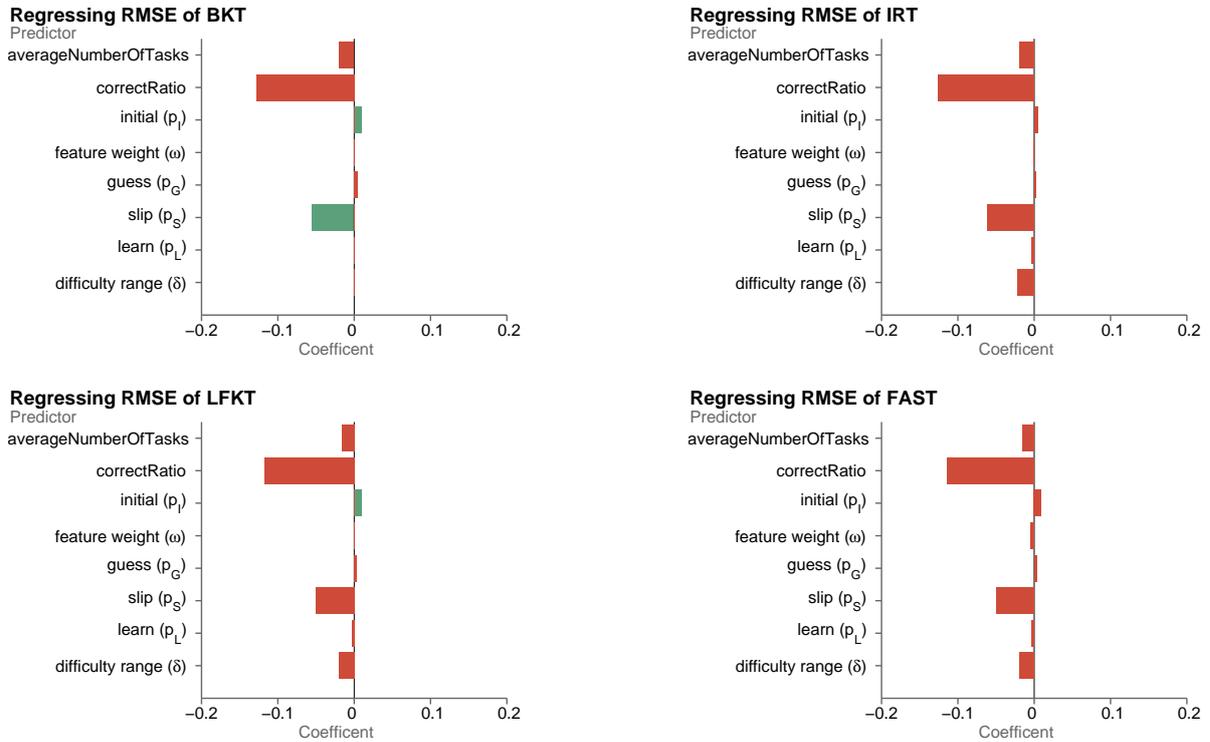


Figure 4: Regression coefficients to predict RMSE based on the sampling parameter values for the models BKT, IRT, LFKT and FAST. Parameters with positive coefficients have a negative effect on the performance and vice versa. Red denotes significant coefficients with  $p < 0.001$ , green coefficients are not significant.

## 6. CONCLUSIONS

In this work, we investigated the performance characteristics of latent factor and knowledge tracing models by exploring their parameter space. To do so, we generated a vast amount of 66'000 synthetic data sets for different parameter configurations containing data for 1'500 students each. Synthetic data allowed us to study the model performances under different parameter settings, and to test the robustness of the models against violations of specific model assumptions.

We showed best and worst case performances for all the models and investigated the relative performance gain in various regions of the parameter space. Our results showed that the two recently developed models LFKT and FAST, which synthesize item response theory and knowledge tracing, perform better than BKT and IRT. FAST even significantly outperformed LFKT if reasonable features can be extracted from the learning environment. Interestingly, IRT exhibited the worst performance, which supports the hypothesis by [19] that random item ordering has a negative influence on the performance of IRT models. However, more analyses are needed to investigate this effect thoroughly. Further, we investigated the models' abilities to predict the latent knowledge state and demonstrated that LFKT and FAST are outperformed by BKT. This raises the question of how to adjust the two recent methods LFKT and FAST if the aim is to predict knowledge states; we leave this exploration for future work. The analysis of the model robustness revealed that BKT is robust to increased feature influence for small guess and slip probabilities. For larger guess and slip, FAST outperformed the other methods.

While all sampling parameters have been carefully chosen to match real world conditions, we expect real world data to exhibit more noise and additional effects not covered by our synthetic data. Thus, the achieved performance can be considered an upper bound on the performance achievable in real world settings. The performance of BKT depends on the quality of the underlying skill model. We have simulated imperfect skill models by introducing item effects, but we did not take other sources for imperfect skill models into account. Furthermore, the simulated data consisted of a fixed set of items. For tutoring systems offering many variations of tasks, reliable estimation of item effects is challenging, which in turn influences the performance of IRT, LFKT and FAST. Moreover, the performance of FAST is driven by feature quality, which may vary between different tutoring systems.

Finally, it remains questionable whether and how the performance of the investigated techniques influences the learning outcome of students in a tutoring system. We show relative improvements in RMSE between models of up to 6%. However, the effect of small-scale improvements in the accuracy of student models on the learning outcome has been discussed controversially [4, 39].

**Acknowledgments.** This work was supported by ETH Research Grant ETH-23 13-2.

## 7. REFERENCES

- [1] R. S. Baker, A. T. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual

- Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In *Proc. ITS*, 2008.
- [2] R. S. Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere. Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In *Proc. UMAP*, 2010.
- [3] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting the moment of learning. In *Proc. ITS*, 2010.
- [4] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In *Proc. EDM*, 2013.
- [5] J. E. Beck and K. M. Chang. Identifiability: A fundamental problem of student modeling. In *Proc. UM*, 2007.
- [6] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with features. In *Proc. NAACL-HLT*, 2010.
- [7] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary? - improving learning efficiency with the cognitive tutor through educational data mining. In *Proc. AIED*, 2007.
- [8] H. Cen, K. R. Koedinger, and B. Junker. Comparing two IRT models for conjunctive skills. In *Proc. ITS*, 2008.
- [9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 1994.
- [10] A. Dhanani, S. Y. Lee, P. Phothilimthana, and Z. Pardos. A comparison of error metrics for learning model parameters in Bayesian knowledge tracing. Technical report, UCB/EECS-2014-131, EECS Department, University of California, Berkeley, 2014.
- [11] S. Fancsali, T. Nixon, and S. Ritter. Optimal and worst-case performance of mastery learning assessment with Bayesian knowledge tracing. In *Proc. EDM*, 2013.
- [12] G. H. Fischer and I. W. Molenaar. *Rasch models: Foundations, recent developments, and applications*. Springer Science & Business Media, 1995.
- [13] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proc. ITS*, 2010.
- [14] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *Proc. EDM*, 2014.
- [15] J. Gu, H. Cai, and J. E. Beck. Investigate performance of expected maximization on the knowledge tracing model. In *Proc. ITS*, 2014.
- [16] D. Harris. Comparison of 1-, 2-, and 3-parameter IRT models. *Educational Measurement: Issues and Practice*, 1989.
- [17] J. Johns and B. Woolf. A dynamic mixture model to detect student motivation and proficiency. In *Proc. Artificial intelligence*, 2006.
- [18] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proc. EDM*, 2014.
- [19] M. M. Khajah, Y. Huang, J. P. González-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. *Personalization Approaches in Learning Environments*, 2014.
- [20] K. Koedinger, J. Stamper, E. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *Proc. AIED*, 2013.
- [21] J. Meyer and E. Hailey. A study of Rasch, partial credit, and rating scale model parameter recovery in WINSTEPS and jMetrik. *Journal of Applied Measurement*, 2011.
- [22] Z. A. Pardos and N. Heffernan. Introducing item difficulty to the knowledge tracing model. In *Proc. UMAP*, 2011.
- [23] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proc. UMAP*, 2010.
- [24] Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of Bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. In *Proc. EDM*, 2010.
- [25] Z. A. Pardos, S. Trivedi, N. T. Heffernan, and G. N. Sárközy. Clustered knowledge tracing. In *Proc. ITS*, 2012.
- [26] Z. A. Pardos and M. Yudelson. Towards moment of learning accuracy. In *AIED Workshops*, 2013.
- [27] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. In *Proc. AIED*, 2009.
- [28] R. Pelánek. A brief overview of metrics for evaluation of student models. In *Approaching Twenty Years of Knowledge Tracing Workshop*, 2014.
- [29] J. Reye. Student modelling based on belief networks. *IJAIED*, 2004.
- [30] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the knowledge tracing space. In *Proc. EDM*, 2009.
- [31] H. Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, 1999.
- [32] J. C. Stamper and K. R. Koedinger. Human-machine student model discovery and improvement using datashop. In *Proc. AIED*, 2011.
- [33] B. van de Sande. Properties of the Bayesian knowledge tracing model. *JEDM*, 2013.
- [34] Y. Wang and J. Beck. Class vs. student in a Bayesian network student model. In *Proc. AIED*, 2013.
- [35] Y. Wang and N. T. Heffernan. The student skill model. In *Proc. ITS*, 2012.
- [36] M. Wilson and P. De Boeck. Descriptive and explanatory item response models. 2004.
- [37] Y. Xu and J. Mostow. Using logistic regression to trace multiple subskills in a dynamic Bayes net. In *Proc. EDM*, 2011.
- [38] Y. Xu and J. Mostow. Using item response theory to refine knowledge tracing. In *Proc. EDM*, 2013.
- [39] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian knowledge tracing models. In *Proc. AIED*, 2013.

# Mixture Modeling of Individual Learning Curves

Matthew Streeter  
Duolingo, Inc.  
Pittsburgh, PA  
matt@duolingo.com

## ABSTRACT

We show that student learning can be accurately modeled using a mixture of learning curves, each of which specifies error probability as a function of time. This approach generalizes Knowledge Tracing [7], which can be viewed as a mixture model in which the learning curves are step functions. We show that this generality yields order-of-magnitude improvements in prediction accuracy on real data. Furthermore, examination of the learning curves provides actionable insights into how different segments of the student population are learning.

To make our mixture model more expressive, we allow the learning curves to be defined by generalized linear models with arbitrary features. This approach generalizes Additive Factor Models [4] and Performance Factors Analysis [16], and outperforms them on a large, real world dataset.

## 1. INTRODUCTION

In the mid-1980s, a now-famous study demonstrated the potential impact of adaptive, personalized education: students tutored one-on-one outperformed those taught in a conventional classroom by two standard deviations [3]. Remarkably, subsequent research has achieved similar gains using interactive, computerized tutors that maintain an accurate model of the student’s knowledge and skills [6]. In the past few years, widespread access to smartphones and the web has allowed such systems to be deployed on an unprecedented scale. Duolingo’s personalized language courses have enrolled over 90 million students, more than the total number of students in all U.S. elementary and secondary schools combined.

A central component of an intelligent tutoring system is the student model, which infers a student’s latent skills and knowledge from observed data. To make accurate inferences from the limited data available for a particular student, one must make assumptions about how students learn. How do students differ in their learning of a particular skill or concept? Is the primary difference in the initial error rate, the rate at which error decreases with time, the shape of the learning curve, or something else? The answers to these questions have implications for the choice of model class (e.g., Hidden Markov Model, logistic regression), as well as the choice of model parameters.

Previous approaches to student modeling typically make strong assumptions about the shape of each student’s learning curve (i.e., the error rate as a function of the number of trials). Additive Factor Models [4] use the student and the number of trials as features in a logistic regression model, which implies a sigmoidal learning curve with the same steepness for each student, but different horizontal offset. Knowledge Tracing [7] is a two-state Hidden Markov Model where, conditioned on the trial  $t$  at which the student first transitions from not knowing the skill to mastering it, the learning curve is a step function.

In empirical studies, it has been observed that aggregate learning curves often follow a power law, a phenomenon so ubiquitous it has been called the *power law of practice* [13]. Later work suggested that, although error rates follow a power law when averaged over an entire population, individual learning curves are more accurately modeled by exponentials [10]. That is, the power law curve observed in aggregate data is actually a mixture of exponentials, with each student’s data coming from one component of the mixture.

These observations led us to seek out a more general approach to student modeling, in which individual learning curves could be teased apart from aggregate data, without making strong assumptions about the shape of the curves. Such an approach has the potential not only to make the student model more accurate, but also to explain and summarize the data in a way that can produce actionable insights into the behavior of different subsets of the student population.

This work makes several contributions to student modeling. First, we present models of student learning that generalize several prominent existing models and that outperform them on real-world datasets from Duolingo. Second, we show how our models can be used to visualize student performance in a way that gives insights into how well an intelligent tutoring system “works”, improving upon the population-level learning curve analysis that is typically used for this purpose [11]. Finally, by demonstrating that relatively simple mixture models can deliver these benefits, we hope to inspire further work on more sophisticated approaches that use mixture models as a building block.

## 1.1 Related Work

The problem of modeling student learning is multifaceted. In full generality it entails modeling a student’s latent abilities, modeling how latent abilities relate to observed performance, and modeling how abilities change over time as a result of learning and forgetting. For an overview of various approaches to student modeling, see [5, 8].

This work focuses on the important subproblem of modeling error probability as a function of trial number for a particular task. Following the influential work of Corbett and Anderson [7], Knowledge Tracing has been used to solve this problem in many intelligent tutoring systems. Recent work has sought to overcome two limitations of the basic Knowledge Tracing model: its assumption that each observed data point requires the use of a single skill, and its assumption that model parameters are the same for all students. To address the first limitation, Additive Factor Models [4] and Performance Factors Analysis [16] use logistic regressions that include parameters for each skill involved in some trial. The second limitation has been addressed by adapting the basic Knowledge Tracing model to individual students, for example by fitting per-student odds multipliers [7], or by learning per-student initial mastery probabilities [14].

Our work seeks to address a third limitation of Knowledge Tracing: its strong assumptions about the shape of the learning curve. Following Knowledge Tracing, we first attempt to model performance on a task that requires only a single skill. In §4, we generalize this approach to obtain a mixture model that includes both Additive Factor Models and Performance Factors Analysis as special cases, and that outperforms both on a large, real-world dataset.

## 2. SINGLE-TASK MIXTURE MODEL

In this section we present a simple mixture model that is appropriate for use on datasets with a single task. This model is a viable alternative to the basic (non-individualized) version of Knowledge Tracing, and is useful for exploratory data analysis. In §4, we generalize this model to handle datasets with multiple tasks.

### 2.1 The Probabilistic Model

A student’s performance on a task after  $T$  trials can be represented as an error vector  $v \in \{0, 1\}^T$ , where  $v_t = 1$  if the student made an error on trial  $t$  and is 0 otherwise. Thus a task, together with a distribution over students, defines a distribution over binary error vectors. In this work, we model this distribution as a mixture of  $K$  distributions, where each component of the mixture is a *learning curve*, or equivalently a product of Bernoulli distributions (one for each trial).

To formally define this model, define the probability of observing outcome  $o \in \{0, 1\}$  when sampling from a Bernoulli distribution with parameter  $p$  as

$$\mathcal{B}(p, o) = \begin{cases} p & o = 1 \\ 1 - p & o = 0 \end{cases}.$$

A learning curve  $q \in [0, 1]^\infty$  specifies, for each trial  $t$ , the

probability  $q_t$  that the student makes an error on trial  $t$ . The probability of the error vector  $v$  according to learning curve  $q$  is  $\prod_t \mathcal{B}(q_t, v_t)$ . A  $K$ -component mixture over learning curves is a set  $q^1, q^2, \dots, q^K$  of learning curves, together with prior probabilities  $p^1, p^2, \dots, p^K$ . The probability of an error vector  $v \in \{0, 1\}^T$  according to the mixture model is

$$\sum_{j=1}^K p^j \prod_{t=1}^T \mathcal{B}(q_t^j, v_t).$$

Inference in a mixture model consists of applying Bayes’ rule to compute a posterior distribution over the  $K$  components of the mixture, given an observed error vector. The model parameters can be fit from data using the EM algorithm, pseudo code for which is given in Algorithm 1.

---

#### Algorithm 1 EM Algorithm for single-task mixture model

---

**Parameters:** number of components  $K$ , error vector  $v^s$  for each student  $s$ , prior parameters  $\alpha \geq 1, \beta \geq 1$ .  
**Initialize**  $p^j \leftarrow \frac{1}{K} \forall j$ , and  $q_t^j \leftarrow \text{Rand}(0, 1) \forall j, t$ .  
**while** not converged **do**  
 $L_{s,j} \leftarrow p^j \prod_{t=1}^T \mathcal{B}(q_t^j, v_t^s) \forall s, j$   
 $z_{s,j} \leftarrow \frac{L_{s,j}}{\sum_{j'} L_{s,j'}} \forall s, j$   
 $q_t^j \leftarrow \frac{\alpha - 1 + \sum_s z_{s,j} v_t^s}{\alpha + \beta - 2 + \sum_s z_{s,j}} \forall j, t$   
 $p^j \leftarrow \frac{\sum_s z_{s,j}}{\sum_s \sum_{j'} z_{s,j'}} \forall j$   
**end while**

---

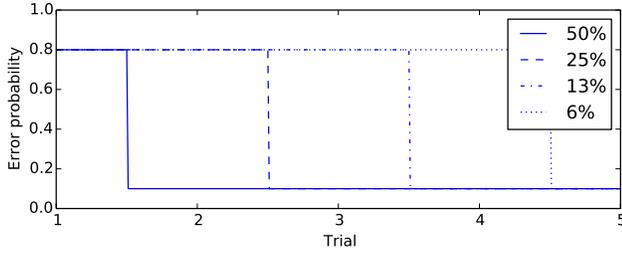
To make Algorithm 1 perform well when data is sparse, it is useful to place a Bayesian prior over the set of possible learning curves. In this work we use a product of Beta distributions for the prior:  $\mathbb{P}[q] = \prod_t \text{Beta}(\alpha, \beta)(q_t)$ . This choice of prior gives a simple closed form for the maximization step of the EM algorithm, which can be thought of computing the maximum-likelihood estimate of  $q_t^j$  after “hallucinating”  $\alpha - 1$  correct responses and  $\beta - 1$  errors (see pseudo code).

### 2.2 Knowledge Tracing as a Mixture Model

Knowledge Tracing is typically presented as a two-state Hidden Markov Model, where the student’s state indicates whether or not they have mastered a particular skill. In this section, we show that if the maximum number of trials is  $T$ , Knowledge Tracing can also be thought of as a mixture model with  $T + 1$  components, each of which is a step function. Thus, Knowledge Tracing can be viewed as a constrained mixture model, in contrast to the unconstrained model discussed in the previous section.

To see this relationship, recall that in a Knowledge Tracing model, the student makes an error with slip probability  $p_s$  if they have mastered the skill, and with probability  $1 - p_g$  otherwise, where  $p_g$  is the probability of a correct guess. The probability of mastery is  $p_0$  initially, and after each trial, a student who has not yet mastered the skill transitions to the mastered state with probability  $p_T$ .

Let  $V$  be an error vector, so  $V_t = 1$  if the student makes an error on trial  $t$  and is 0 otherwise, and let  $M$  be the state vector:  $M_t = 1$  if the student has mastered the skill at the beginning of trial  $t$  and is 0 otherwise. The distribution over



**Figure 1: Mixture model representation of a Knowledge Tracing model with guess probability  $p_g = 0.2$ , slip probability  $p_s = 0.1$ , transition probability  $p_T = 0.5$ , and initial mastery probability  $p_0 = 0$ .**

error vectors defined by Knowledge Tracing is given by

$$\mathbb{P}[V = v] = \sum_m \mathbb{P}[M = m] \mathbb{P}[V = v | M = m].$$

Because the student never leaves the mastered state after reaching it, there are only  $T + 1$  possibilities for the state vector  $M$ . Letting  $m^j$  be the  $j$ th possibility ( $m_t^j = 0$  if  $t < j$ , 1 otherwise), and letting  $p^j = \mathbb{P}[M = m^j]$ , we have

$$\mathbb{P}[V = v] = \sum_{j=1}^{T+1} p^j \cdot \mathbb{P}[V = v | M = m^j].$$

Because the components of  $V$  are conditionally independent given  $M$ ,

$$\mathbb{P}[V = v | M = m^j] = \prod_{t=1}^T \mathcal{B}(q_t^j, v_t)$$

where

$$q_t^j = \begin{cases} 1 - p_g & t < j \\ p_s & t \geq j \end{cases}.$$

Putting these facts together, we see that the probability of a particular error vector under Knowledge Tracing is the same as under a mixture model with  $T+1$  components, where each learning curve  $q^j$  is a step function with the same initial and final height but a different horizontal offset (see Figure 1).

Because the HMM and the mixture model are both generative models that specify the same distribution over binary vectors, the conditional distributions over binary vectors given a sequence of observations are also the same, and Bayesian inference yields exactly the same predictions when performed on either model.

Viewing Knowledge Tracing in this way, it is natural to consider generalizations that remove some of the constraints, for example allowing the step functions to have different initial or final heights (perhaps students who master the skill earlier are less likely to slip later on). In the model presented in §2.1 we simply remove all the constraints, allowing us to fit a mixture model over learning curves of arbitrary shape.

We note that later work on Knowledge Tracing allowed for the possibility of forgetting (transitioning from the mastered

to unmastered state). This version can still be modeled as a mixture model, but with  $2^T$  rather than  $T + 1$  components.

## 2.3 Statistical Consistency

A model is *statistically consistent* if, given enough data, it converges to the ground truth. In this section we show that the “hard” version of EM algorithm 1 is consistent, provided the number of components in the mixture model grows with the amount of available data (the hard EM algorithm is the same as algorithm 1, except that it sets  $z_{s,j} = 1$  for the  $j$  that maximizes  $L_{s,j}$ , and  $z_{s,j} = 0$  otherwise). For simplicity we assume the number of trials  $T$  is the same for all students, but this is not essential. Also, though the data requirements suggested by this analysis are exponential  $T$ , in practice we find that near-optimal predictions are obtained using a much smaller number of components.

**THEOREM 1.** *Consider the “hard” version of EM algorithm 1, and suppose that the number of trials is  $T$  for all students. This algorithm is statistically consistent, provided the number of curves  $K$  in the mixture model grows as a function of the number of data points  $n$ .*

**PROOF.** Recall that an event occurs *with high probability* (whp) if, as  $n \rightarrow \infty$ , the probability of the event approaches 1. The idea of the proof is to show that, whp, each of the  $2^T$  possible error vectors will be placed into its own cluster on the first iteration of the EM algorithm. This will imply that the EM algorithm converges on the first iteration to a mixture model that is close to the true distribution.

Consider a particular error vector  $v^s \in \{0, 1\}^T$ , and let  $j$  be the index of the likelihood-maximizing curve on the first iteration of the algorithm (i.e.,  $z_{s,j} = 1$ ). If  $Q \in [0, 1]^T$  is a random curve, the probability that  $\prod_{t=1}^T \mathcal{B}(Q_t, v_t^s) > \frac{1}{2}$  is positive. Thus, as  $K \rightarrow \infty$ , whp at least one of the  $K$  random curves will satisfy this inequality, and in particular for the likelihood-maximizing curve  $q^j$  we have  $\prod_{t=1}^T \mathcal{B}(q_t^j, v_t^s) > \frac{1}{2}$ , which implies  $\mathcal{B}(q_t^j, v_t^s) > \frac{1}{2}$  for all  $t$ . For any error vector  $v^{s'} \neq v^s$ , there must be some  $t$  such that  $v_t^s \neq v_t^{s'}$ , which implies  $\mathcal{B}(q_t^j, v_t^{s'}) < \frac{1}{2}$ . This means that whp,  $q^j$  cannot be the likelihood-maximizing curve for  $v^{s'}$ , and so each binary vector will have a unique likelihood-maximizing curve.

If each binary vector  $v$  has a unique likelihood-maximizing curve  $q^j$ , then the M step of the algorithm will simply set  $q^j \leftarrow v$ , and will set  $p^j$  to the empirical frequency of  $v$  within the dataset. As  $n \rightarrow \infty$ , this empirical frequency approaches the true probability, which shows that the algorithm is consistent.  $\square$

In the worst case, statistical consistency requires a constant amount of data for every possible error vector, hence the data requirements grow exponentially with  $T$ . However, this is not as bad as it may seem. In intelligent tutoring systems, it is often the case that  $T$  is small enough that even in the worst case we can guarantee near-optimal performance. Furthermore, as we show experimentally in §3.2, near-optimal performance can often be achieved with a much smaller number of components in practice.

## 2.4 Use in an Intelligent Tutoring System

How should the predictions of a mixture model be used to schedule practice within an intelligent tutoring system? When using Knowledge Tracing, a typical approach is to schedule practice for a skill until the inferred probability of having mastered it exceeds some threshold such as 0.95. With a mixture model, we can no longer take this approach since we don't make explicit predictions about whether the student has mastered a skill. Nevertheless, we can define a reasonable practice scheduling rule in terms of predicted future performance.

In particular, note that another way of formulating the scheduling rule typically used in Knowledge Tracing is to say that we stop practice once we are 95% confident that performance has reached an asymptote. With a mixture model, it is unlikely that the marginal value of practice will be exactly 0, so this precise rule is unlikely to work well (it would simply schedule indefinite practice). However, we can compute the expected marginal benefit of practice (in terms of reduction in error rate), and stop scheduling practice once this drops below some threshold.

Note that when practice scheduling is defined in terms of expected marginal benefit, the practice schedule is a function of the predicted distribution over error vectors, so mixture models that make the same predictions will result in the same practice schedule even if the model parameters are different. This is in contrast to Knowledge Tracing, where multiple globally optimal models (in terms of likelihood) can lead to very different practice schedules, because the inferred probability of mastery can be different even for two models that make identical predictions [2].

## 2.5 Identifiability

A statistical model is identifiable if there is a unique set of parameters that maximize likelihood. Our mixture model is not identifiable, since in general there are many ways to express a given distribution over binary vectors as a mixture of learning curves. However, as we argued in the previous section, non-identifiability does not pose a problem for practice scheduling if the schedule is defined in terms of the model's predictions rather than its parameters.

## 3. EXPERIMENTS WITH SINGLE-TASK MODEL

In this section we evaluate the single-task mixture model of §2 on data from Duolingo. These experiments serve two purposes. First, they show that the mixture model can give much more accurate predictions than Knowledge Tracing on real data. Second, inspection of the learning curves produced by the mixture model reveals interesting facts about the student population that are not apparent from conventional learning curve analysis. In §4 we present a more general mixture model that is appropriate for datasets with multiple skills.

### 3.1 The Duolingo Dataset

We collected log data from Duolingo, a free language learning application with over 90 million students. Students who

use Duolingo progress through a sequence of lessons, each of which takes a few minutes to complete and teaches certain words and grammatical concepts. Within each lesson, the student is asked to solve a sequence of self-contained challenges, which can be of various types. For example, a student learning Spanish may be asked to translate a Spanish sentence into English, or to determine which of several possible translations of an English sentence into Spanish is correct.

For these experiments, we focus on *listen challenges*, in which the student listens to a recording of a sentence spoken in the language they are learning, then types what they hear. Listen challenges are attractive because, unlike challenges which involve translating a sentence, there is only one correct answer, which simplifies error attribution. For these experiments we use a simple bag-of-words knowledge component (KC) model. There is one KC for each word in the correct answer, and a KC is marked correct if it appears among the words the student typed. For example, if a student learning English hears the spoken sentence "I have a business card" and types "I have a business car", we would mark the KC *card* as incorrect, while marking the KCs for the other four words correct. This approach is not perfect because it ignores word order as well as the effects of context (students may be able to infer which word is being said from context clues, even if they cannot in general recognize the word when spoken). However, the learning curves generated by this KC model are smooth and monotonically decreasing, suggesting that it performs reasonably well.

Our experiments use data from the Spanish course for English speakers, one of the most popular courses on Duolingo. In this section, we focus on modeling acquisition of a single skill, using data for the KC *una* (the feminine version of the indefinite article "a"). In §4 we consider more general mixture models, and in §5 we evaluate them on datasets with multiple KCs. The full dataset has roughly 700,000 data points (there is one data point for each combination of student, trial, and KC), while the *una* dataset contains around 15,000.

### 3.2 Prediction Accuracy

To evaluate the mixture model's prediction accuracy, we divided the Duolingo dataset into equal-sized training and test sets by assigning each student to one of the two groups at random. We then ran the EM algorithm on the training data to fit mixture models with various numbers of components, as well as a Knowledge Tracing model, and computed the predictions of these models on the test data. We evaluate prediction accuracy using two commonly-used metrics.

1. *Average log-likelihood.* Log-likelihood measures how probable the test data is according to the model. Specifically, if the dataset  $D$  consists of  $n$  independent data points  $D_1, D_2, \dots, D_n$  (each data point is the binary performance of a particular student on a particular trial), and  $p_i = \mathbb{P}[D_i|M]$  is the conditional probability of the  $i$ th data point  $D_i$  given the model  $M$ , then

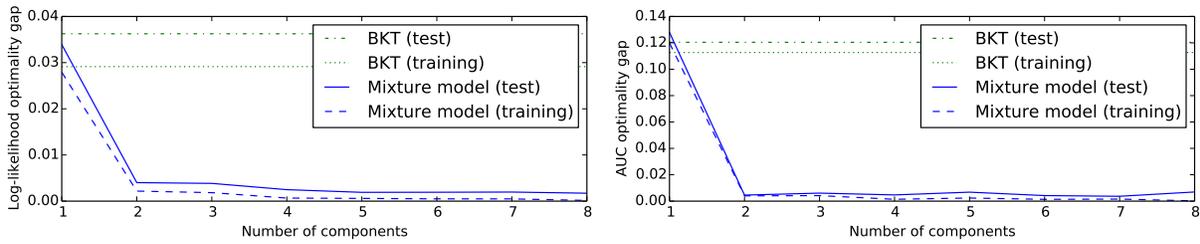


Figure 2: Optimality gaps for log likelihood (left) and AUC (right) as a function of number of components in the mixture model, compared to Knowledge Tracing (horizontal lines). The optimality gap is the absolute difference between the model’s accuracy and the maximum possible accuracy on the dataset.

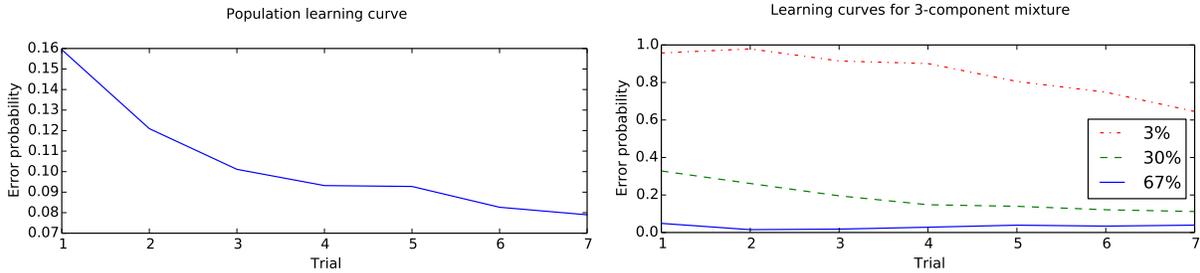


Figure 3: Learning curves for recognizing the Spanish word *una* in a Duolingo listen challenge. The population curve (left) suggests a reasonable rate of learning in aggregate, but the mixture model (right) reveals large differences among different clusters of students.

average log-likelihood is

$$\frac{1}{n} \log \mathbb{P}[D|M] = \frac{1}{n} \log \prod_{i=1}^n p_i = \frac{1}{n} \sum_{i=1}^n \log p_i .$$

Because both the mixture model and Knowledge Tracing are fit using maximum likelihood, it is natural to compare them in terms of this objective function.

2. *AUC*. AUC evaluates the accuracy of the model’s predictions when they are converted from probabilities to binary values by applying a threshold. It can be defined as the probability that  $p > q$ , where  $p$  is the model’s prediction for a randomly-selected positive example and  $q$  is the model’s prediction for a randomly-selected negative example. This is equivalent to the area under the ROC curve, which plots true positive rate against false positive rate (both of which vary as a function of the chosen threshold).

Figure 2 presents accuracy on the *una* dataset as a function of the number of components in the mixture model, both on training and held-out test data. To make relative improvements clearer, we plot the optimality gap rather than the raw value of the prediction accuracy metric. For example, the optimality gap for test set log likelihood is the difference between the optimal log likelihood on the test data (which can be computed in closed form) and the model’s log likelihood on the test data.

For both AUC and log-likelihood, the improvement in accuracy is largest when going from one component to two, and there are diminishing returns to additional components,

particularly in terms of performance on held-out test data. With more than 5 components, log-likelihood on test data gets slightly worse due to overfitting, while performance on training data improves slightly. In practice, the number of components can be selected using cross-validation.

For both metrics, Knowledge Tracing is similar to the one-component model but significantly worse than the two component model in terms of accuracy, both on training and test data. Furthermore, all mixture models with two or more components outperform Knowledge Tracing by an *order of magnitude* in terms of the optimality gap for log-likelihood and AUC, both on training and on held-out test data. We observed very similar results for datasets based on other Spanish words, such as *come* (eat), *mujer* (woman), and *hombre* (man).

### 3.3 Learning Curve Mixture Analysis

In this section we examine the learning curves that make up the components of the mixture model fit to Duolingo data. This analysis can be viewed as a more general version of learning curve analysis [11], which examines the population learning curve (this is equivalent to the curve for a one-component mixture model).

Figure 3 presents learning curves for the *una* dataset. The left pane of the figure shows the aggregate learning curve, while the right pane shows the curves for a 3-component mixture model fit using the EM algorithm. Examining the right pane, we see that the mixture model clusters students into three quite different groups.

- Around two-thirds of the students belong to a cluster that in aggregate has an error probability around 5% on the first trial, and this error rate does not change with increased trials.
- A second, smaller cluster contains 30% of the students. These students, in aggregate, have an initial error rate of 33% which decreases to around 11% after 7 trials.
- The third cluster contains only 3% of students. These students have a very high initial error rate of 96%, which declines to about 65% after 7 trials.

The existence of this third, high-error-rate cluster surprised us, so we went back to the log data to examine the behavior of students in this cluster in more detail. It turned out that almost all of these students were simply giving up when presented with a listen challenge (although they correctly answered other types of challenges). Further examination of the log data revealed that some of these students skipped all listen challenges, while others would skip all listen challenges for long stretches of time, then at other times would correctly answer listen challenges. We conjecture that the former set of students are either hearing-impaired or do not have working speakers, while the latter do not want to turn their speakers on at certain times, for example because they are in a public place. Duolingo attempts to accommodate such students by offering a setting that disables listen challenges, but not all students realize this is available. As a result of these insights, Duolingo is now exploring user interface changes that will actively detect students that fall into this cluster and make it easier for them to temporarily disable listen challenges.

This analysis shows how mixture modeling can produce valuable insights that are not apparent from examination of the population learning curve alone. We hope this will inspire the use of mixture modeling more broadly as a general-purpose diagnostic tool for intelligent tutoring systems.

## 4. GENERAL MIXTURE MODEL

The single-task model is appropriate for datasets where there is a single knowledge component (KC) and many students. In an actual intelligent tutoring system, a student will learn many KCs, and prediction accuracy can be improved by using student performance on one KC to help predict performance on other, not yet seen KCs. In this section we present a more general mixture model that accomplishes this.

In this more general model, student performance is again modeled as a mixture of  $K$  learning curves. However, instead of treating each point on the learning curve as a separate parameter, we let it be the output of a generalized linear model with features that depend on the student, task, and trial number. In particular, for a student  $s$  and task  $i$ , the probability of a performance vector  $v_1, v_2, \dots, v_T$  is

$$\sum_{j=1}^k p^j \prod_{t=1}^T \mathcal{B}(q^j(s, i, t; \beta^j), v_t)$$

where

$$q^j(s, i, t; \beta^j) = g^{-1}(\phi_{s,i,t} \cdot \beta^j),$$

where  $\phi_{s,i,t}$  is the feature vector for student  $s$ , task  $i$ , trial  $t$ , and  $g$  is the link function for the generalized linear model [12]. Our experiments use logistic regression, for which the link function is  $g(p) = \text{logit}(p)$ .

Note that this model generalizes the single-task mixture model presented in §2. In particular, the single-task model with curve  $q^j(t)$  is recovered by setting  $\phi_{s,i,t} = e_t$ , an indicator vector for trial  $t$ , and setting  $\beta_t^j = g(q^j(t))$ .

As with the single-task model, we can estimate the parameters of this model using the EM algorithm. The main difference is that the maximization step no longer has a closed form solution. However, it is a convex optimization and can still be solved exactly using a number of algorithms, for example stochastic gradient descent.

To define the EM algorithm, first define the likelihood function

$$L_{s,i}^j(\beta) = \prod_{t=1}^T \mathcal{B}(q^j(s, i, t; \beta), v_t).$$

For the E step, we define hidden variables  $z_{s,i}^j$ , which give the probability that the data for student  $s$  and task  $i$  follows curve  $j$ .

$$z_{s,i}^j = \frac{p^j L_{s,i}^j}{\sum_{j'} p^{j'} L_{s,i}^{j'}(\beta)}.$$

For the M step, we optimize the coefficient vector for each component  $j$  so as to maximize expected log-likelihood.

$$\beta^j = \text{argmax}_{\beta} \left\{ \sum_s \sum_i z_{s,i}^j \log(L_{s,i}^j(\beta)) \right\}.$$

When performing inference for a new student, we solve a similar optimization problem, but we only update the coefficients for that particular student.

### 4.1 Relationship to Other Models

This mixture model is quite general, and with appropriate choices for the feature function  $\phi$  can recover many previously-studied models. In particular, any modeling approach that is based on a logistic regression using features that depend only on the student, task, and trial number can be recovered by using a single component ( $K = 1$ ), choosing  $g = \text{logit}$ , and defining  $\phi$  to include the appropriate features. This includes both Additive Factor Models [4] and Performance Factors Analysis [16]. By choosing a larger  $K$ , we immediately obtain generalizations of each of these methods that have the potential to more accurately model the behavior of individual clusters of students. Because the trial number (together with the student and task) identifies a unique learning event, we can also include features that depend on the trial type, elapsed time, and previous learning history, as in learning decomposition [1].

Note that for the mixture model to add value over a simple regression, we must define “task” in such a way that we observe multiple trials for a given (student, task) pair. For datasets where each item requires the use of multiple KCs,

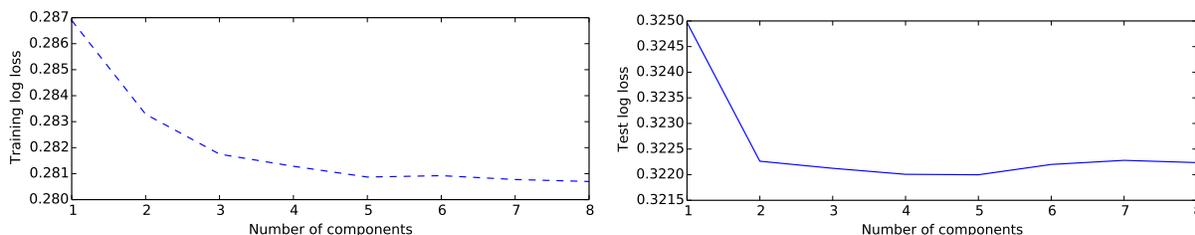


Figure 4: Performance of a mixture of Additive Factor Models on training data (left) and test data (right), as a function of the number of components in the mixture model.

Table 1: Performance on Duolingo dataset

Method	Training log loss	Test log loss	Training AUC loss	Test AUC loss
Knowledge Tracing	0.3429	0.3441	0.3406	0.3460
Performance Factors Analysis	0.3248	0.3285	0.2774	0.2865
Additive Factor Model	0.2869	0.3250	0.1629	0.2789
A.F.M. Mixture (3 components)	<b>0.2818</b>	<b>0.3220</b>	<b>0.1598</b>	<b>0.2760</b>

this entails either (a) defining a task for each combination of KCs, or (b) using error attribution to create a dataset in which each example involves only a single KC, and having one task per KC. We use the latter approach in our experiments in §5. This approach is different from the one taken by algorithms such as LR-DBN [17], which make predictions on multiple-KC items directly.

## 4.2 Parameter Sharing

To make more efficient use of available data when fitting this generalized mixture model, it can be useful for certain coefficient values to be shared across components of the mixture model. To illustrate this issue, consider fitting a mixture of Additive Factor Models. In this case,  $\phi$  includes an indicator feature for each student. If we fit a  $K$  component mixture, we must estimate  $K$  separate coefficient values for each student, which increases the variance of the estimates compared to the basic Additive Factor Model. For students for whom we do not yet have much data, this can result in larger values of  $K$  giving worse performance.

To overcome this difficulty, we allow certain coefficients to be shared across all components of the mixture model, while others have a separate value for each component. This requires only minor changes to the M step of the EM algorithm. Instead of solving  $K$  separate optimization problems, we solve a single larger optimization problem of the form:

$$\operatorname{argmax}_{\beta^1, \beta^2, \dots, \beta^j} \left\{ \sum_j \sum_s \sum_i Z_{s,i}^j \log(L_{s,i}^j(\beta^j)) \right\}$$

subject to

$$\beta_z^1 = \beta_z^2 = \dots = \beta_z^j \text{ for all shared } z.$$

Again, for  $g = \text{logit}$ , this is a weighted logistic regression problem that can be solved using a variety of standard algorithms.

## 5. EXPERIMENTS WITH GENERALIZED MODEL

In this section, we demonstrate the potential of the generalized mixture model by using it to learn a mixture of Additive Factor Models which models student performance on Duolingo listen challenges.

For these experiments, we use the same Duolingo dataset described in §3.1, but with all knowledge components included (i.e., every time student  $s$  completes a listen challenge, there is an example for each word  $w$  in the challenge, and the label for the example indicates whether the student included word  $w$  in their response). Each KC (i.e., each word) is considered a separate task. Note that although each listen challenge involves multiple KCs, we are using error attribution to create a dataset in which each example involves only a single KC. There is nothing about our methodology that requires this, but it mirrors the problem we wish to solve at Duolingo, and also allows for a cleaner comparison with Knowledge Tracing.

When splitting the data into training and test sets, we put each (student, KC) pair into one of the two groups uniformly at random. When fitting a mixture of Additive Factor Models, we use parameter sharing (see §4.2) for the student and KC indicator features, while allowing the times-seen feature to vary across components.

Figure 4 shows how performance on training and test data varies as a function of the number of components in the mixture model. The leftmost point ( $K = 1$ ) corresponds to a regular Additive Factor Model, which can be fit by running a single logistic regression. Other points correspond to mixture models fit using the EM algorithm, in which each iteration entails solving a weighted logistic regression problem. As can be seen, using more than one component in the mixture model improves accuracy on both training and held-out test data.

Table 1 compares the performance of the Additive Factor

Model, the 3-component mixture of Additive Factor Models, Knowledge Tracing, and Performance Factors Analysis [16] on the same dataset. In this table, we present accuracy in terms of losses (log loss is -1 times log-likelihood, while AUC loss is one minus AUC), so lower values are better. As can be seen, the 3-component mixture gives the best performance of all the methods we considered in terms of both metrics, both on training and test data.

## 6. CONCLUSIONS

In this work we explored the use of mixture models to predict how students' error rates change as they learn. This led to order-of-magnitude improvements over Knowledge Tracing in terms of prediction accuracy on single-task datasets from Duolingo, as measured by the optimality gaps for both log-likelihood and AUC. Furthermore, examining the curves in the mixture model led us to uncover surprising facts about different groups of students.

We then generalized this mixture model to the multi-task setting, by learning a mixture of generalized linear models. This generalized mixture model offered state of the art performance on a large Duolingo dataset, outperforming Performance Factors Analysis, Additive Factor Models, and Knowledge Tracing on the same data.

There are several ways in which this work could be extended:

1. *Finding a good prior over learning curves.* In the single-task setting, we simply placed a Beta prior over each point on each learning curve. Though this worked well on the Duolingo dataset we considered (which contained around 15,000 data points), it may not give the best bias/variance tradeoff for smaller datasets. A natural way to constrain the algorithm would be to require error probability to be non-increasing as a function of trial number. Restricting to a particular family of curves such as exponentials or APEX functions [10], which generalize power laws and exponentials, may also be reasonable.
2. *Accounting for forgetting.* We have assumed that performance depends only on the trial number, and not on the amount of time elapsed since a particular knowledge component was last seen. For this reason, our model has no way to capture the benefit of spaced repetition [9] over massed practice, which is important for practice scheduling in the context of language learning [15].
3. *Feature exploration in the multi-task setting.* The generalized mixture model from §4 can be used with any set of features  $\phi$ , but our experiments in §4 considered only a few possible choices. It would be interesting to explore other feature sets, and to see whether the features that work best in the usual regression setting ( $K = 1$ ) are also best for larger  $K$ .

## 7. REFERENCES

- [1] J. E. Beck. Using learning decomposition to analyze student fluency development. In *ITS 2006*, pages 21–28, 2006.
- [2] J. E. Beck and K. Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146. Springer, 2007.
- [3] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, pages 4–16, 1984.
- [4] H. Cen, K. Koedinger, and B. Junker. Learning Factors Analysis – A general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.
- [5] K. Chrysafiadi and M. Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715–4729, 2013.
- [6] A. Corbett. Cognitive computer tutors: Solving the two-sigma problem. In *User Modeling 2001*, pages 137–147. Springer, 2001.
- [7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [8] M. C. Desmarais and R. S. J. d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.
- [9] H. Ebbinghaus. *Memory: A contribution to experimental psychology*. Dover, New York, 1885.
- [10] A. Heathcote, S. Brown, and D. Mewhort. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7(2):185–207, 2000.
- [11] B. Martin, A. Mitrovic, K. R. Koedinger, and S. Mathan. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3):249–283, 2011.
- [12] P. McCullagh, J. A. Nelder, and P. McCullagh. *Generalized linear models*. Chapman and Hall London, 1989.
- [13] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive skills and their acquisition*, pages 1–55. Erlbaum, Hillsdale, NJ, 1983.
- [14] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.
- [15] P. I. Pavlik and J. R. Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.
- [16] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis – A new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pages 531–538, 2009.
- [17] Y. Xu and J. Mostow. Comparison of methods to trace multiple subskills: Is LR-DBN best? In *EDM*, pages 41–48, 2012.

# Accounting for Slipping and Other False Negatives in Logistic Models of Student Learning

Christopher J. MacLellan  
Human-Computer Interaction  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
cmaclell@cs.cmu.edu

Ran Liu  
Human-Computer Interaction  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
ranliu@andrew.cmu.edu

Kenneth R. Koedinger  
Human-Computer Interaction  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
koedinger@cmu.edu

## ABSTRACT

Additive Factors Model (AFM) and Performance Factors Analysis (PFA) are two popular models of student learning that employ logistic regression to estimate parameters and predict performance. This is in contrast to Bayesian Knowledge Tracing (BKT) which uses a Hidden Markov Model formalism. While all three models tend to make similar predictions, they differ in their parameterization of student learning. One key difference is that BKT has parameters for the slipping rates of learned skills, whereas the logistic models do not. Thus, the logistic models assume that as students get more practice their probability of correctly answering monotonically converges to 100%, whereas BKT allows monotonic convergence to lower probabilities. In this paper, we present a novel modification of logistic regression that allows it to account for situations resulting in false negative student actions (e.g., slipping on known skills). We apply this new regression approach to create two new methods AFM+Slip and PFA+Slip and compare the performance of these new models to traditional AFM, PFA, and BKT. We find that across five datasets the new slipping models have the highest accuracy on 10-fold cross validation. We also find evidence that the slip parameters better enable the logistic models to fit steep learning rates, rather than better fitting the tail of learning curves as we expected. Lastly, we explore the use of high slip values as an indicator of skills that might benefit from skill label refinement. We find that after refining the skill model for one dataset using this approach the traditional model fit improved to be on par with the slip model.

## Keywords

Cognitive Modeling, Statistical Models of Learning, Additive Factors Model, Performance Factors Analysis, Knowledge Tracing

## 1. INTRODUCTION

Statistical models of student learning make it possible for Intelligent Tutoring Systems [18] to be adaptive. These models estimate students' latent skill knowledge, so that tutors can use these estimates to intelligently select problems that give students more practice on skills that need it. Prior work has shown that even minor improvements in the predictive fit of latent knowledge models can result in less "wasted" student time, with more time on effective practice [22].

Two popular models of student learning are the Additive Factors Model (AFM) [4] and Performance Factors Analysis (PFA) [16]. Both are extensions of traditional Item Response Theory models [8]. While the two models differ in their parameterization of student learning, they both utilize logistic regression to estimate parameters and predict student performance. These models stand in contrast to other popular approaches like Bayesian Knowledge Tracing (BKT) [7], which uses Hidden Markov Modeling.

The BKT model is used both for "online" knowledge estimation within Intelligent Tutoring Systems (e.g., in Carnegie Learning's Cognitive tutor) to adaptively selecting practice items and for "offline" educational data modeling. The logistic models, on the other hand, have mainly been used in the context of offline data modeling. For example, DataShop, the largest open repository of educational data [12], uses AFM to fit student performance within existing datasets and to generate predicted learning curves. Data-driven cognitive task analyses, i.e., discovering and testing new mappings of tutor items to skills (or knowledge components), have used AFM as the core statistical model [17]. Novel knowledge component models can be discovered, evaluated in conjunction with AFM as a statistical model, validated on novel datasets [14], and used to guide tutor redesign efforts [13].

Despite the success of approaches like AFM, its lack of slip parameters has been emphasized as a key reason for favoring knowledge tracing over logistic models [10]. But knowledge tracing models tend to suffer from identifiability problems [1, 2]; e.g., the same performance data can be fit equally well by different parameters values, with different implications for system behavior. Furthermore, the actual effect of slip parameters on model predictions is complicated. The guess and slip parameters in BKT serve the dual purpose of modeling both noise, and the upper and lower bounds, in student performance. Without slip parameters, if a student gets an answer wrong, then BKT must assume that the student has not yet learned the skill. In contrast, the logistic models just model noise in the observations, so as long as the average student success rate converges to 100% then both models should perform similarly (assuming all other parameters are comparable across models). These approaches should only differ in situations where student performance converges to lower probabilities at higher opportunities; i.e., where false negatives such as slipping are actually occurring.

To investigate false negative phenomena, we augmented the logistic regression formalism to support slipping parameters. Using this new approach, which we call Bounded Logistic Regression, we produce two new student learning models: Additive Factors Model + Slip (AFM+Slip) and Performance Factors Analysis + Slip (PFA+Slip). These models are identical to their traditional counterparts but have additional parameters to model the false negative rates for each skill. We compare these models to their traditional counterparts and to BKT on five datasets across the domains of Geometry, Equation Solving, Writing, and Number Line Estimation. In all cases, the slip models have higher predictive accuracy (based on 10-fold cross validation) than their traditional counterparts.

We then move beyond comparing the predictive accuracies of the models to investigate how these parameters affect the predictions of the models and *why* these models are more accurate. Our analyses suggest that slipping parameters are not used to capture actual student "slipping" behavior (i.e., non-zero base rates for true student errors) but, rather, make the logistic models more flexible and allow better modeling of steeper learning rates while still predicting performance accurately at high opportunity counts (in the learning curve tail).

Lastly, we use AFM+Slip to perform data-driven refinement of the knowledge component (KC) model for a Geometry dataset. We identified a KC with a high false negative, or slip, rate and searched for ways to refine it. Using domain expertise, we refined the underlying KC model and showed that the traditional model (AFM) with the new KC model performed as well as the comparable slip model (AFM+Slip) did with the original KC model. This suggests that slip parameters allow the model to compensate for, and identify, an underspecified KC model.

## 2. STATISTICAL MODELS OF LEARNING

### 2.1 Logistic Models

The models in this class use logistic regression to estimate student and item parameters and to predict student performance. Thus, they model the probability that a student will get an step  $i$  correct using the following logistic function:

$$p_i = \frac{1}{1 + e^{-z_i}}$$

where  $z_i$  is some linear function of student and item parameters for step  $i$ . The likelihood function for these models has been shown to be convex (i.e., no local maximums), so optimal parameter values can be efficiently computed and issues of identifiability only occur when there are limited amounts of data for each parameter. There are many possible logistic student learning models; in fact, most Item Response Theory models are in this class. For this paper, we will focus on two popular models in the educational data mining community: Additive Factors Model [4] and Performance Factors Analysis [16].

#### 2.1.1 Additive Factors Model

This model utilizes individual parameters for each student's baseline ability level, each knowledge component's baseline difficulty, and the learning rate for each knowledge com-

ponent (i.e., how much improvement occurs with each additional practice opportunity). The standard equation for this model is shown here:

$$z_i = \alpha_{student(i)} + \sum_{k \in KCs(i)} (\beta_k + \gamma_k \times opp(k, i))$$

where  $\alpha_{student(i)}$  represents the prior knowledge of the student performing step  $i$ , the  $\beta$ s and  $\gamma$ s represents the difficulty and learning rate of the KCs needed to solve step  $i$ , and  $opp(k, i)$  represents the number of prior opportunities a student has had to practice skill  $k$  before step  $i$ . In the traditional formulation, the learning rates ( $\gamma$ s) are bounded to be positive, so practicing KCs never decreases performance. To prevent the model from overfitting, the student parameters ( $\alpha$ s) are typically  $L_2$  regularized; i.e., they are given a normal prior with mean 0. Regularization decreases the model fit to the training data (i.e., the log-likelihood, AIC, and BIC) but improves the predictive accuracy on unseen data. Thus, when comparing regularized models to other approaches it should primarily be compared on measures that use held out data, such as cross validation.

#### 2.1.2 Performance Factors Analysis

There are two key differences between this model and AFM. First, PFA does not have individual student parameters [16] (later variants have explored the addition of student parameters [6], but we base our current analysis on the original formulation). This usually substantially reduces the number of parameters of the model relative to AFM, particularly in datasets with a large number of unique students. Second, the model takes into account students' actual performance (not just opportunities completed) by splitting the learning rate for each skill into two learning rates: a rate for successful practice and a rate for unsuccessful practice. The standard equation based on these changes is the following:

$$z_i = \sum_{k \in KCs(i)} (\beta_k + \gamma_k success(i, k) + \rho_k failure(i, k))$$

where the  $\beta$ s represent the difficulty of the KCs,  $\gamma$ s and  $\rho$ s represent the learning rates for successful and unsuccessful practice on the KCs,  $success(i, k)$  represents the number of successful applications of a skill  $k$  for the given student prior to step  $i$ , and  $failure(i, k)$  represents the number of unsuccessful applications of a skill  $k$  for the given student prior to step  $i$ . Similar to AFM it is typical to restrict the learning rates (i.e.,  $\gamma$ s and  $\rho$ s) to be positive [9]. One complication when comparing this model to other approaches using held out data (i.e., cross validation) is that the success and failure counts potentially contain additional information about the test data (i.e., performance on held out practice opportunities). Thus, we argue that comparing AFM to PFA using cross validation is usually not a fair comparison. Bearing this in mind, in the current analysis we were more interested in comparing AFM+Slip and PFA+Slip to their respective baseline models than to each other. To this end, we utilized cross validation as the primary measure of predictive accuracy for reasons previously discussed.

### 2.2 Bayesian Knowledge Tracing

There are many different models in the knowledge tracing family [10], but for this paper we focus on traditional 4-parameter BKT [7]. In contrast to the logistic approaches,

BKT utilizes a Hidden Markov Model to estimate latent parameters and predict student performance. This model has four parameters for each skill: the initial probability that the skill is known  $p(L_0)$ , the probability that the skill will transition from an unlearned to a learned state  $p(T)$ , the probability of an error given that the skill is learned  $p(Slip)$ , and the probability of a success when the skill is not learned  $p(Guess)$ . Unlike the logistic models, the estimation of these parameters can sometimes be difficult due to issues of identifiability [2] (e.g., there are many parameter values that yield the same likelihood) so these parameters are typically bounded to be within reasonable ranges; e.g., guess is typically bounded to be between 0 and 0.3 and slip is bounded to be between 0 and 0.1 [1]. Prior research has produced toolkits that can efficiently estimate these parameters using different approaches. For the comparisons in this paper we use the toolkit created by Yudelson et al. [23] and we use the gradient descent method.

One of the core differences between the logistic models and BKT is how they parameterize false negative student actions (i.e., slipping behavior). The logistic models do not have slip parameters and so they model student success as converging monotonically to 100% success (i.e., learning rates are bounded to be positive). In contrast, the BKT model explicitly models false negatives and allows monotonic convergence (under the typical assumption that the probability of forgetting is zero) to lower success rates. The slip parameters in BKT also serve the purpose of accounting for noise in student performance, and it is unclear whether these parameters account for true slipping behavior (i.e., non-zero base rate error) or just general noise in the student actions. Since the logistic models can already handle noise in the data, it remains to be seen what would happen if slip parameters were added to these models. That is the focus of this paper's investigation.

### 3. BOUNDED LOGISTIC REGRESSION

There is no trivial approach to incorporating explicit slip parameters into the logistic models; e.g., the prediction probability cannot be bounded by an additional linear term to the logistic function. In order to add these parameters we modified the underlying logistic model to have the following form:

$$p_i = \frac{1}{1 + e^{-s_i}} \times \frac{1}{1 + e^{-z_i}}$$

where  $z_i$  is the same as that used in standard logistic regression and  $s_i$  is a linear function of the parameters that impose an upper bound on the success probability for the step  $i$ . For modeling a slip rate for each skill we use the following equation:

$$s_i = \tau + \sum_{k \in KC_{s(i)}} \delta_k$$

where  $\tau$  is the parameter corresponding to the average slip rate across all items and students and  $\delta_k$  is the change in the average slip rate for each skill  $k$ . We also apply an  $L_2$  regularization to the  $\delta$  parameters to prevent overfitting. To fit the parameters we used the sequential quadratic programming package in Octave, which uses an approach similar to Newton-Raphson but properly accounts for parameter con-

straints (e.g., positive learning rates). For details on parameter estimation see Appendix A.

This formulation is a generalization of Item Response Theory approaches that model item slip (e.g., [21]). In particular, it supports slipping with multiple KC labels per an item by using a logistic function to map the sum of slip parameters to a value between 0 and 1. For items with a single KC label, the  $\frac{1}{1+e^{-s_i}}$  term reduces to the slip probability for that KC. For multi-KC items, this term models slipping as the linear combination of the individual KC slipping parameters in logit space. This approach mirrors that taken by AFM and PFA for modeling KC difficulty and learning rates in situations with multiple KC labels. In these situations, prior work has shown that the logit approach gives a good approximation of both conjunctive and disjunctive KC behavior [4].

During early model exploration we used Markov Chain Monte Carlo methods to compare this formulation with a more complex formulation that had parameters for both guessing and slipping. Our preliminary results showed that AFM with slip parameters outperformed the guess-and-slip variation for the 'Geometry Area (1996-97)' [11] and the 'Self Explanation sch\_a3329ee9 Winter 2008 (CL)' [3] datasets (accessed via DataShop [12]) in terms of deviance information criterion (a generalization of AIC for sampled data). Further analysis showed that there was little data to estimate the guessing portion of the logistic curve. This is because the average student error rate in these datasets starts off at less than 50% and only gets lower with practice. This is typical of many of the available tutor datasets, so for our Bounded Logistic Regression approach we decided it would be sufficient to model the slipping parameters.

## 4. EVALUATION

### 4.1 Method

We used bounded logistic regression to add slip parameters to AFM and PFA, thus creating two new student learning models: AFM + Slip and PFA + Slip. We were interested in how these approaches compared with their traditional counterparts and to Bayesian Knowledge Tracing, which parameterizes guess and slip. Furthermore, we were interested in how these different approaches compared across different datasets spanning distinct domains. To perform this evaluation we fit each of the five models to five datasets we downloaded from DataShop [12]: Geometry Area (1996-97) [11], Self Explanation sch\_a3329ee9 Winter 2008 (CL)[3], IWT Self-Explanation Study 1 (Spring 2009) (tutors only) [19], IWT Self-Explanation Study 2 (Fall 2009) (tutors only) [20], and Digital Games for Improving Number Sense - Study 1 [15]. These datasets cover the domains of geometry, equation solving, writing, and number line estimation. We selected these datasets because they have undergone extensive KC model refinement, including both manually created models by domain experts and automatically-refined models using Learning Factors Analysis [5]. For all datasets we used the best fitting KC model, based on unstratified cross validation.

In addition to comparing the different statistical models' predictive accuracies, we were interested in understanding

**Table 1: In all five datasets the slip models outperform their non-slip counterparts in terms of log-likelihood and cross validation. In four out of the five datasets, the PFA+Slip model outperforms the AFM+Slip model in terms of log-likelihood and cross validation performance. In this table “Par.” represents the number of parameters in the model and the CV RMSE values are the averages of 10 runs of 10-fold un-stratified cross validation.**

Dataset	Model	Par.	LL	AIC	BIC	CV RMSE
Geometry	AFM	95	-2399.7	4989.4	5610.5	0.396
	AFM+Slip	114	-2377.0	4982.0	5727.3	0.395
	PFA	54	-2374.9	4857.8	<b>5210.8</b>	0.389
	PFA+Slip	73	<b>-2298.3</b>	<b>4742.6</b>	5219.8	<b>0.383</b>
	BKT	72	-2460.8	5065.7	5536.5	0.396
Equation Solving	AFM	106	3011.6	6235.2	6953.9	0.390
	AFM+Slip	125	<b>-2992.5</b>	<b>6235.0</b>	7082.54	<b>0.388</b>
	PFA	48	-3205.2	6506.4	6831.8	0.400
	PFA+Slip	67	-3088.9	6311.8	<b>6766.0</b>	0.392
	BKT	72	-3202.7	6549.5	7037.7	0.426
Writing 1	AFM	169	-3214.6	6767.2	7916.1	0.406
	AFM+Slip	196	-3214.6	6821.2	8153.6	0.406
	PFA	72	-3212.0	6568.0	<b>7057.4</b>	0.401
	PFA+Slip	99	<b>-3158.0</b>	<b>6514.0</b>	7187.0	<b>0.398</b>
	BKT	104	-3480.2	7168.5	7875.6	0.419
Writing 2	AFM	129	-2976.4	6210.8	7096.6	0.375
	AFM+Slip	145	-2962.8	6215.6	7211.3	0.373
	PFA	45	-2994.7	6079.4	<b>6388.4</b>	0.373
	PFA+Slip	61	<b>2965.7</b>	<b>6053.4</b>	6472.2	<b>0.371</b>
	BKT	60	-3177.1	6474.2	6886.2	0.384
Number Line	AFM	93	-2352.7	4891.4	5484.0	0.433
	AFM+Slip	115	-2356.3	4942.6	5675.4	0.432
	PFA	62	-2337.5	<b>4799.0</b>	<b>5194.1</b>	0.430
	PFA+Slip	84	<b>-2318.9</b>	4805.8	5341.1	<b>0.428</b>
	BKT	84	-2548.7	5265.4	5800.7	0.451

and interpreting the situations in which slip parameters improve model fit. Prior to analysis we hypothesized that slipping parameters might have three potential effects on the model fit: (1) enabling the model to capture true student slipping behavior; i.e., KCs that have a non-zero base-rate error, (2) enabling the model to fit steeper initial learning rates while still making correct predictions at higher opportunity counts, and (3) enabling the model to compensate for an underspecified knowledge component model. We focused in on one dataset, Geometry Area (1996-97), to explore these possibilities. Within this dataset we conducted a residual analysis to explore possibilities (1) and (2). We then refined the geometry KC model for a specific KC that the slip model identified as having a high false negative rate (i.e., slip value) to explore possibility (3). For brevity we only show the results of AFM and AFM+Slip in these analyses, but similar trends hold for PFA and PFA+Slip.

## 4.2 Results

### 4.2.1 Model Fits for Five Datasets

We fit each of the five models to the five datasets. Table 1 shows the resulting model fit statistics and the number of parameters used in each model. AFM has 1 parameter per student and 2 parameters per skill, PFA has 3 parameters

for each skill, and BKT has 4 parameters for each skill. The slip variations have an additional parameter for each skill, plus a parameter for the average slip rate. When using the PFA models in practice many of the KCs never had any unsuccessful practice (i.e., their failure count was always 0). In these situations we removed the parameters for the failure learning rates because they have no effect on the model behavior. Thus, in some situations, the number of parameters in each model might differ from the general trends. All of the cross validation results are the average of 10 runs of 10-fold unstratified cross validation, where the cross validated RMSE was computed using the predicted probability of a correct response (rather than discrete correct/incorrect predictions).

All of the slip models have better log-likelihood and cross validation performance than their respective baseline models (AFM and PFM). Furthermore, in four out of the five datasets, PFA+Slip has better cross validation performance than AFM+Slip, even though it does not have individual student parameters. Finally, all of the logistic models outperformed traditional four-parameter BKT. Based on prior work [16] we expected this last result, but we included BKT as a comparison model that supports slipping. In particular,

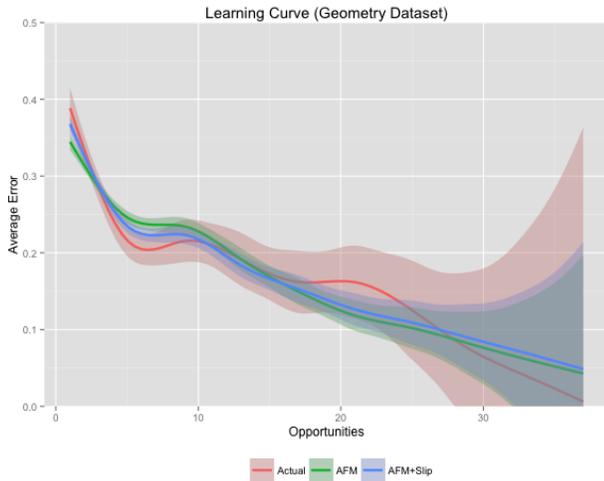


Figure 1: The AFM+Slip model better fits the steeper learning rate of the Geometry dataset than the AFM model, but both models fit the tail of the learning curve reasonably well and the actual student error appears to be converging to 0%. The shaded regions denote the 95% confidence intervals for the respective values.

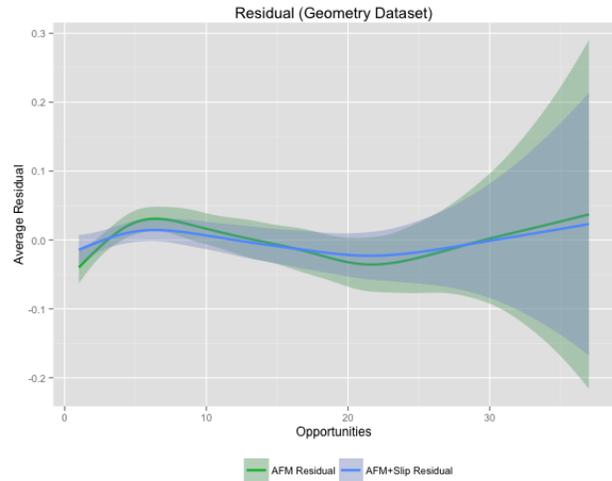


Figure 2: The 95% confidence intervals (shaded regions) for the residuals of the AFM model do not include zero for lower opportunity counts, the model first overpredicts and then underpredicts success. In contrast the 95% confidence intervals for residuals of the AFM+Slip model always include zero indicating a better model fit.

Figure 3 shows an example of how the AFM+Slip model fits the data more like the BKT model than the AFM model for a KC with a high slip rate.

#### 4.2.2 Residual Analysis

To investigate how the predictions of the slip models differ from that of the traditional models we analyzed the residuals for the AFM and AFM+Slip models on the Geometry dataset. Figure 1 shows the actual and predicted error rates for the two models on this dataset and Figure 2 shows the model residuals plotted by opportunity count. Investigating patterns in residual error across opportunity counts is a useful way of assessing systematic discrepancies between a given model’s predicted learning curves and students’ actual learning curves.

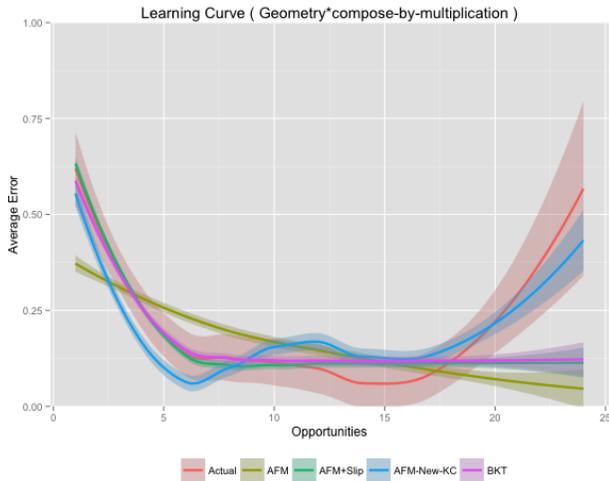
Although both models fit the data reasonably well, the slip model better models the steepness at the beginning of the learning curve. At low opportunity counts, AFM without slip typically predicts a substantially flatter learning curve compared to the actual data. The residual plot mirrors this finding; the 95% confidence interval for the AFM residuals does not include zero for earlier opportunities and the model flips from over-predicting success to under-predicting it. The AFM+Slip model, in contrast, better models the initial steepness of the learning curve. The 95% confidence interval for the AFM+Slip model residuals always includes zero. Finally, we see no evidence of actual slipping behavior in the tail of the learning curve: the 95% confidence intervals for residuals in both models include zero for higher opportunity counts. If true student slipping were occurring, we would expect the traditional AFM model to overpredict success in the tail, but we do not observe this.

#### 4.2.3 KC Refinement Based on False Negatives

In order to explore the hypothesis that a high false negative, or slip, rate on a skill is indicative of a underspecified knowledge component model, we analyzed a KC on which the slip parameter was high and on which AFM and AFM+Slip differed substantially in their predictions. One KC, “geometry\*compose-by-multiplication,” fit this criteria. Figure 3 shows the learning curve with model predictions for this KC. AFM+Slip makes predictions that are nearly identical to BKT and seems to better fit the actual student learning curve. Upon further investigation, we found that many of the items labeled with this skill were on the same problems. Within these problems, we noticed that the later problem steps (items) might actually have been solved by applying the “arithmetic” skill to the result of an earlier application of the “compose-by-multiplication” skill. We generated a new knowledge component model to reflect these findings and re-fit the model using AFM. The predictions of this new model (AFM-New-KC) are also shown in Figure 3. For the AFM-New-KC plot, we plotted the observations with the opportunity counts from the original KC model (x-axis) but with predicted errors from the new KC model (y-axis). This was necessary for the purposes of comparison to the original KC model predictions. Once the knowledge component model was refined based on the insights provided by fitting AFM+Slip, standard AFM improved. Furthermore, based on this change the overall AFM model fit improved to be on par with AFM+Slip in terms of log-likelihood, AIC, and cross validation (LL = -2378.8, AIC = 4947.6, BIC = 5568.6, and CV RMSE = 0.395).

## 5. DISCUSSION

Our model fit results show that the slip models have better predictive accuracy (i.e., cross validation performance) and



**Figure 3: AFM+Slip looks much more like BKT for this KC and seems to model the data better (the overlapping purple and green lines). We took the high false negative rate (i.e., the sharp floor in the predicted error at approx. 11%) as an indicator that the KC model might benefit from refinement. Refitting the regular AFM model with a refined KC model (AFM-new-KC) shows a better fit to the actual data. Shaded regions denote the 95% confidence intervals for the respective values.**

log-likelihood fits than their traditional counterparts across all five datasets. Furthermore, the AIC scores generally mirror this finding. These results suggest that the addition of the slip parameters to the logistic model formalism results in an improved model fit and an increased ability to predict behavior on unseen data.

In four of the five datasets, PFA + Slip best fit the data in terms of both log-likelihood and cross validation. In one sense, its superior cross-validation performance is surprising because the PFA models (as implemented here) have no student intercept parameters. However, they have an advantage for the cross validation statistic because they get success and failure counts that include information about performance on held out data, essentially giving these models an advantage over the other models. The better log-likelihood (and often AIC) scores are indicative of a better ability to fit the data that doesn't suffer from this discrepancy. However, PFA models have an advantage over AFM for this statistic because AFM uses regularization, which intentionally worsens the fit of the model to the data in an effort to improve predictive accuracy. To test if regularizing student parameters was causing PFA and PFA + Slip to outperform AFM and AFM + Slip we refit the AFM models to the Geometry dataset with student parameter regularization disabled and found that, at least for the Geometry dataset, the PFA models still outperforms the AFM models in terms of log-likelihood, AIC, BIC, and CV RMSE. These findings suggest that the PFA models better fits the data than the AFM models, but more work is needed to explore how best to compare these two approaches and to determine when

one approach is preferable to another.

Lastly, the logistic models consistently outperform traditional four-parameter BKT. This is somewhat unsurprising because BKT does not have individual student parameters or separate learning rates for success and failure. However, we still included traditional BKT as a baseline model that is widely used and has explicit parameters for guess and slip. In particular, Figure 3 shows that for a KCs with high slip rate the AFM+Slip model performs more like BKT than AFM, suggesting that the new model is able to fit slipping and other false negative student behavior.

Given the finding that the slip models have better predictive accuracy and log-likelihood fits than their traditional counterparts, we investigated how the addition of slip parameters changed the model predictions. Residual analyses on the Geometry dataset showed that both AFM and AFM+Slip had similar fits to the data, but AFM+Slip better fit the initial steepness of the learning curve while maintaining a good fit in the tail. This intuition is confirmed in the residual by opportunity plot, which shows that the 95% confidence intervals for the residuals from AFM exclude zero at low opportunity counts, first overpredicting success and then underpredicting it. In contrast, the 95% confidence interval for the residuals from AFM+Slip include zero at these same low opportunity counts. This evidence supports the hypothesis that adding slip parameters enables the model to better accommodate steeper learning rates. In contrast, we find no evidence to support the hypothesis that adding slipping parameters enables the model to better fit non-zero base rate error; i.e., true student slipping. If this were the case, then we would expect AFM to overpredict success in the tail (i.e., for the residuals to be non-zero at higher opportunity counts), but we found no evidence that this occurred.

Finally, we demonstrated that high false negative, or slip, rates can serve as detectors of KCs that might benefit from further refinement. We identified a KC in the Geometry dataset that had a high slip rate and that differed from the traditional model: the “geometry\*compose-by-multiplication” KC. We found that this KC could be further refined and showed that AFM with the refined KC model performed on par with AFM+Slip in terms of log-likelihood and cross validation. This suggests that adding slip parameters to a model can enable it to compensate for an underspecified KC model but, more importantly, can help identify these poorly specified KCs. The newly discovered KC model better fit the student data than the previous best model, which was the result of years of hand and automated KC model refinement.

## 6. CONCLUSIONS

Logistic models of learning, such as AFM and PFA, are popular approaches for modeling educational data. However, unlike models in the knowledge tracing family, they do not have the ability to explicitly model guessing and slipping rates on KCs. In this work we augmented traditional logistic regression to support slipping rates using an approach that we call Bounded Logistic Regression. We then used this approach to create two new student models: AFM + Slip and PFA + Slip. We then compared the performance of these new models in relation to their traditional counterparts. Furthermore, for AFM we explored how the addi-

tion of slip parameters changed the predictions made by the model. We explored three possibilities: (1) they might enable the model to capture true student slipping behavior (i.e., non-zero base-rate error), (2) they might enable the model to accommodate steeper learning rates while still effectively predicting performance at higher opportunity counts, and (3) they might enable the model to compensate for an underspecified knowledge component model.

To explore the first two possibilities, we conducted a residual analysis and found that the slip parameters appear to help the model fit steeper learning rates, rather than improving model fit in the tail. To explore the third possibility, we used a high false negative, or slip, rate as an indicator of where the given KC model might benefit from refinement. We found that after refining a KC model using this approach AFM performance (e.g., CV, LL, AIC) improved to be on par with AFM-Slip. This suggests that the slip parameters enable the model to compensate for underspecified KC models and that high slip values can be used to identify KCs that might benefit from further KC label refinement.

## 7. LIMITATIONS AND FUTURE WORK

One key limitation of the current work is that we did not explore issues of identifiability in the Bounded Logistic Regression model. In particular, we have not yet demonstrated that the log-likelihood for models using this formalism are convex. In the current formulation we only model slip parameters (not guess parameters), so we expect identifiability to be less of an issue. In line with this intuition we found that the current approach returned reasonable parameter values and consistently improved model fit across the five data sets we explored. However, we recognize that the model would benefit from a more rigorous analysis of the quality of estimated parameters and acknowledge that this would be an important direction for future work.

Finally, the current work focuses on comparing the slip models to their traditional counterparts, but future work might explore how different models (e.g., AFM+Slip, PFA+Slip, and BKT) compare to one another. In the current work we purposefully avoided making conclusions about how these models compare because there is some ambiguity in how different approaches are evaluated. For example, Yudelson's Bayesian Knowledge Tracing toolkit [23] performs incremental prediction during cross validation (i.e., predicting student performance on a step and then "showing" the model the actual performance before moving on to the next step). While this approach aligns well with the actual use of the BKT model it gives an unfair advantage when comparing it to cross validated AFM, which gets no information about test data when making predictions. A similar complication exists for PFA, which gets information about the performance of unseen steps from the success and failure counts. A more equivalent comparison would be to perform an incremental prediction using AFM and PFA, but this was beyond the scope of the current paper and represents an open area for future work.

## 8. ACKNOWLEDGEMENTS

We thank Erik Harpstead, Michael Yudelson, and Rony Patel for their thoughts and comments when developing this work. This work was supported in part by the Department

of Education (#R305B090023 and #R305B110003) and by the National Science Foundation (#SBE-0836012). Finally, we thank Carnegie Learning and all other data providers for making their data available on DataShop.

## 9. REFERENCES

- [1] R. S. J. d. Baker, A. T. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In B. P. Woolf, E. Aimeur, R. Nkambou, and L. S, editors, *ITS '08*, pages 406–415, 2008.
- [2] J. E. Beck and K.-M. Chang. Identifiability: A Fundamental Problem of Student Modeling. In C. Conati, K. McCoy, and G. Paliouras, editors, *UM '07*, pages 137–146, 2007.
- [3] J. Booth and S. Ritter. Self Explanation sch\_a3329ee9 Winter 2008 (CL). [pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=293](http://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=293).
- [4] H. Cen. *Generalized Learning Factors Analysis: Improving cognitive Models with Machine Learning*. PhD thesis, Carnegie Mellon University, 2009.
- [5] H. Cen, K. R. Koedinger, and B. Junker. Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement. In M. Ikeda, K. Ashlay, and T.-W. Chan, editors, *ITS '06*, pages 164–175, 2006.
- [6] M. Chi, K. R. Koedinger, G. Gordon, P. Jordan, and K. Vanlehn. Instructional Factors Analysis: A Cognitive Model For Multiple Instructional Interventions. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, and J. Stamper, editors, *EDM '11*, pages 61–70, 2011.
- [7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 4(4):253–278, 1995.
- [8] K. L. Draney, P. Pirolli, and M. Wilson. A measurement model for a complex cognitive skill. In P. N, S. Chipman, and R. Brennan, editors, *Cognitively diagnostic assessment*, pages 103–125. Lawrence Erlbaum Associates Inc., 1995.
- [9] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In V. Aleven, J. Kay, and J. Mostow, editors, *ITS '10*, pages 35–44, 2010.
- [10] G.-B. Jose, H. Yun, and P. Brusilovsky. General Features in Knowledge Tracing: Applications to Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren, editors, *EDM '14*, pages 84–91, 2014.
- [11] K. Koedinger. Geometry Area 1996-1997. [pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=76](http://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=76).
- [12] K. R. Koedinger, R. S. J. d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A Data Repository for the EDM community: The PSLC DataShop. In C. Romero, S. Ventura, M. Pechenizkiy, and R. S. J. d. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.
- [13] K. R. Koedinger, J. Stamper, E. McLaughlin, and

- T. Nixon. Using Data-Driven Discovery of Better Student Models to Improve Student Learning. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *AIED '13*, pages 421–430, 2013.
- [14] R. Liu, K. R. Koedinger, and E. A. McLaughlin. Interpreting Model Discovery and Testing Generalization to a New Dataset. In *EDM '14*, pages 107–113, 2014.
- [15] D. Lomas. Digital Games for Improving Number Sense - Study 1. [pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=445](http://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=445).
- [16] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance Factors Analysis –A New Alternative to Knowledge Tracing. In V. Dimitrova and R. Mizoguchi, editors, *AIED '09*, pages 531–538, 2009.
- [17] J. C. Stamper and K. R. Koedinger. Human-machine student model discovery and improvement using Data. In J. Kay, S. Bull, and G. Biswas, editors, *AIED '11*, pages 353–360, 2011.
- [18] K. Vanlehn. The Behavior of Tutoring Systems. *IJAIED*, 16(3):227–265, 2006.
- [19] R. Wylie. IWT Self-Explanation Study 1 (Spring 2009) (tutors only). [pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=313](http://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=313).
- [20] R. Wylie. IWT Self-Explanation Study 2 (Spring 2009) (tutors only). [pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=372](http://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=372).
- [21] Y. C. Yen, R. G. Ho, W. W. Laio, L. J. Chen, and C. C. Kuo. An Empirical Evaluation of the Slip Correction in the Four Parameter Logistic Models With Computerized Adaptive Testing. *APM*, 36(2):75–87, 2012.
- [22] M. V. Yudelson and K. R. Koedinger. Estimating the benefits of student model improvements on a substantive scale. In S. K. D’Mello, R. A. Calvo, and A. Olney, editors, *EDM '13*, 2013.
- [23] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian Knowledge Tracing Models. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *AIED '13*, pages 171–180, 2013.

## APPENDIX

### A. PARAMETER ESTIMATION

Similar to standard logistic regression we assume the data follows a binomial distribution. Thus, the likelihood and log-likelihood are as follows:

$$\begin{aligned} \text{Likelihood}(\text{data}) &= \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{(1-y_i)} \\ \ell(\text{data}) &= \sum_{i=1}^n y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \end{aligned}$$

where  $y_i$  is 0 or 1 depending on if the given step  $i$  was correct. As mentioned earlier,  $p_i$  is defined as:

$$p_i = \frac{1}{1 + e^{-s_i}} \times \frac{1}{1 + e^{-z_i}}$$

where  $s_i$  is the linear combination of the slip parameters and  $z_i$  is the linear combination of the student and item parameters.

To estimate the parameters values for bounded logistic regression, we maximize the conditional maximum likelihood of the data using sequential quadratic programming (specifically the `sqp` package in Octave). This approach reduces to applying the Newton-Raphson method, but properly accounts for situations when the parameter values are constrained, such as the positive bound for the learning rates in AFM and PFA. To apply this method, we needed to compute the gradient and hessian for the likelihood of the data given the model.

To compute the gradient we took the derivative with respect to the student and item parameters ( $w$ 's) and slip parameters ( $sp$ 's). For the student and item parameters the gradient is the following:

$$\frac{dll}{dw_a} = \sum_{i=1}^n \frac{x_{ia}}{1 + e^{z_i}} \frac{(y_i - p_i)}{(1 - p_i)}$$

where  $x_{ia}$  is the value of the student or item feature that is being weighted by parameter  $w_a$  for step  $i$ .

Similarly, for the slip parameters the gradient is the following:

$$\frac{dll}{dsp_a} = \sum_{i=1}^n \frac{q_{ia}}{1 + e^{s_i}} \frac{(y_i - p_i)}{(1 - p_i)}$$

where  $q_{ia}$  is the value of the slip feature (in AFM and PFA these are the 0 or 1 entries from the Q-matrix) that is being weighted by parameter  $sp_a$  for step  $i$ .

Given these gradients we have a hessian matrix with values for the interactions of the  $w$ s with each other, the  $w$ s with the  $sp$ s, and the  $sp$ s with each other. These values are defined as the following:

$$\begin{aligned} \frac{d^2ll}{dw_a dw_b} &= \sum_{i=1}^n \frac{x_{ia} x_{ib}}{(1 + e^{z_i})^2 (1 - p_i)^2} [p_i (y_i - 1) \\ &\quad + e^{z_i} (p_i - y_i) (1 - p_i)] \\ \frac{d^2ll}{dsp_a dsp_b} &= \sum_{i=1}^n \frac{q_{ia} q_{ib}}{(1 + e^{s_i})^2 (1 - p_i)^2} [p_i (y_i - 1) \\ &\quad + e^{s_i} (p_i - y_i) (1 - p_i)] \\ \frac{d^2ll}{dw_a dsp_b} &= \sum_{i=1}^n \frac{x_{ia}}{1 + e^{z_i}} \left[ \frac{(p_i - 1) + (y_i - p_i)}{(1 - p_i)^2} \right] \end{aligned}$$

Finally, in our formulation we applied an  $L_2$  regularization to all of the parameter values (i.e., a normal prior with mean 0), where the  $\lambda$  parameter of the regularization could be set individually for each model parameter. For the AFM models we set  $\lambda$  to 1 for the student parameters. For all of the slip models we  $\lambda$  to 1 for the KC slip parameters (i.e.,  $\delta$ s). For all other parameters we turned regularization off ( $\lambda = 0$ ).

# Learning Environments and Inquiry Behaviors in Science Inquiry Learning: How their Interplay Affects the Development of Conceptual Understanding in Physics

Engin Bumbacher\*  
buben@stanford.edu

Shima Salehi\*  
salehi@stanford.edu

Miriam Wierzchula  
miriamw1989@gmail.com

Paulo Blikstein\*  
paulob@stanford.edu

\* Stanford University, CERAS 102, 520 Galvez Mall, Stanford, CA, 94305

## ABSTRACT

Studies comparing virtual and physical manipulative environments (VME and PME) in inquiry-based science learning have mostly focused on students' learning outcomes but not on the actual processes they engage in during the learning activities. In this paper, we examined experimentation strategies in an inquiry activity and their relation to conceptual learning outcomes. We assigned college students to either use VME or PME for a goal-directed physics inquiry task on mass-spring systems. Our analysis showed that the best predictors of learning outcomes were experimental manipulations that followed a control of variable (CV) strategy, with a delay between manipulations ("systematic inquiry"). Cluster analysis of the prevalence of these manipulations per participant revealed two distinct clusters of participants, systematic inquiry or not. The systematic inquiry cluster had significantly higher learning outcomes than the less systematic one. Furthermore, the majority of the participants using the PME belonged to the more systematic cluster, while most of the participants using the VME fell into the non-systematic cluster, likely because of the specific affordances of the real and virtual equipment they were using. However, beyond this impact on inquiry process, condition had little effect. In light of these results, we argue that investigating processes displayed during learning activities, in addition to outcomes, enables us to properly evaluate the strengths and weaknesses of different learning environments for inquiry-based learning.

## Keywords

Science Discovery Learning, Computer Simulations, Real Laboratories, Inquiry Learning, Cluster Analysis, Virtual and Physical Science Laboratories

## 1. Introduction

Over the past decades, the science teaching community has adopted the view that "students cannot fully understand scientific and engineering ideas without engaging in the practices of inquiry and the discourses by which such ideas are developed and refined" (NRC, 2012, p.218). Inquiry-based instruction requires students to model the practices of scientific inquiry to actively develop their conceptual understanding [1,2]. While physical laboratories were the traditional environments for such inquiry-based learning, there is accumulating evidence that virtual laboratories are similarly well suited to meet the goals of science investigation [3,4]. In particular, they are considered to be at least equally conducive to active manipulations for experimentation [2,3], which is seen as the crucial aspect of inquiry learning [5,6,7].

A major limitation of the research comparing physical and virtual manipulative environments (PME and VME) for science learning was the predominant focus on the learning *outcomes* rather than the learning *processes* when students engage in inquiry activities.

This has not changed with recent work that shifted from treating the environments as two competing entities to examining how to best combine them for increased learning benefits [4]. We argue that research on how learners engage with these manipulative environments could provide a more comprehensive understanding of how the interaction of a learner with an environment impacts the learners' construction of knowledge, and in turn what design features of these environments foster desired manipulative behaviors in the context of science inquiry learning.

The present study lies at the intersection of research on learning environments and research on inquiry behaviors in order to study the characteristics of productive experimentation strategies in open-ended science investigation tasks, and how such strategy use might be influenced by the different affordances of the learning environments. For this purpose we encoded the actual experiments students ran, which allows us to basically replay their processes. This allows us to explore customized operationalizations of inquiry behaviors of interest. This approach integrates data-driven methods with relevant theoretical concepts. As a result, we found a robust characterization of experimentation strategies that meaningfully predicts learning outcomes, and show how participants' strategy use differs between the learning environments. This study is part of a larger research project with the goal of developing automated detectors of systematic inquiry in open-ended science investigation activities for formative assessment and for the design of productive learning environments.

## 2. Inquiry Behaviors

### 2.1. Control of Variable Strategy

Scientific learning through self-directed inquiry activities depends on the actual inquiry behaviors employed [8,9]. In particular, adequate experimentation strategies are required that result in interpretable observations, i.e. evidence that facilitates drawing valid inferences. Research has particularly focused on the abilities to systematically combine variables and to design unconfounded experiments, i.e. experiments that modify variables such that competing hypothesis can be ruled out. The design of unconfounded experiments requires the ability to employ the *control of variables strategy* (CVS), that is, to create experiments with a single contrast between experimental conditions [10]. This is in contrast to inadequate strategies such as changing multiple variables at the same time, which hampers valid inferences and subsequent knowledge [11].

Previous research has examined a host of individual and contextual factors of strategy use [8]. However, only a very small number of studies have explicitly examined the impact of affordances of learning environments on strategy use in experimentation activities [2,12]. While Triona & Klahr [2] focused on the impact of physicality of manipulatives alone on

learning outcomes, Renken & Nunez [12] had students engage in an inquiry activity on pendulum motion using either a PME or a VME that differed in both ease of manipulation and freedom of choice: while the PME provided participants with only three different levels for either pendulum length or mass, the VME allowed participants to modify the variables smoothly by means of continuous valued sliders. Even if there was no difference in conceptual understanding between the VME and PME conditions, participants using the computer simulation ran more trials and were less likely to control variables. Renken and Nunez [12] argued that the additional flexibility and breadth of choice in experimentation in VME was detrimental to participants' use of adequate experimentation strategy.

While this study suggests that indeed strategy use in inquiry-based learning activities is influenced by affordances of the learning environments, it is difficult to generalize these results to less structured and scaffolded inquiry activities.

## 2.2. Operationalization of Inquiry Strategies

As most studies cited mainly focused on CV strategy, they used highly structured tasks where either variables were dichotomous, or there was only one outcome variable, or the activity was restricted. In order to develop a more nuanced characterization of inquiry strategies, we need more complex inquiry tasks. Data mining techniques employed in such contexts have been successful at discovering groups of similar users [13,14,15]. Most of these data-mined systems are based on the user interaction logs [16]. While they achieve good predictive power, such machine-learned detectors of interaction behaviors often come at the cost of interpretability [17]. However, it is crucial to develop data-mined models of inquiry strategies that are interpretable in order to advance our understanding of learning processes through inquiry activities. We apply a different approach, where we do not use labelled action logs but code the actual experiment configurations of each participant. Based on video data, we extract each configuration a participant tried and feed it into a database of experiments of all participants. This allows us to quickly extract and explore relevant variables of inquiry, such as the number of spring-only or mass-only changes, the number of unique configurations, repetitions, etc. That way, we can integrate relevant theoretical concepts into the operationalization of inquiry behaviors.

In the context of this study, we focused on experimentation strategies only. We collected data on the number of experiment trials, the experiment configurations, and the time between manipulations, and coded the type of manipulation per experiment. Particular focus is given to “*control of variable*” manipulations, “*deliberate*” manipulations, and “*deliberate control*” of variable manipulations. Deliberate manipulations (DM) are manipulations into which a participant has put some thought, as measured by *dwelt time* between two consecutive manipulations. We assume that participants who are cognitively engaged – reflecting on evidence from a preceding manipulation, trying to make sense of it in the context of previous observations, or taking notes or planning the next manipulation(s) – will spend more time before executing the next change than those who are cognitively less engaged.

For this reason, we include the third category of manipulations that lies at the intersection of the prior two categories, *deliberate control of variable manipulations* (DCVM). As prior research on

experimentation strategies in inquiry-based activities characterized them as solely CVS or not, activities were designed such that controlling variables in an experiment had to be a deliberate choice of participants [19,20,21]. However, in less structured, open-ended inquiry like those used in this study, it is possible in some cases to manipulate variables according to a CV strategy without the deliberate intention to do so. For example in the computer simulation for our mass & spring activity, one could change the value of the spring constant continuously by means of a slider, without having to interrupt an ongoing experiment. Inherently, this corresponds to a control of variable manipulation (CVM) but not necessarily to a *deliberate* control of variable manipulation (DCVM).

## 3. Present Study

The study reported here was part of a larger study examining participants' inquiry behaviors in different scientific domains using either PME or VME as learning environments. Participants engaged in two activities; the first one was on mass and spring oscillation (see Figure 1), and the second one on basic electric circuits. The current paper presents analysis of the first inquiry activity. During the first activity, participants were either asked to simply think-aloud while engaging in the inquiry or were trained to implement the Predict-Observe-Explain framework (POE) [18]. The training session of the POE framework was highly structured and guided: During the entire activity, before each intended manipulation, participants were asked to predict its result, then observe the actual results of the manipulation, and finally explain their observation in light of the initial prediction. On the other hand, the think-aloud group did not receive any scaffolds or guidance by the experimenter. Therefore, for the purposes of this paper, we report only data for the participants in the think-aloud condition, as the difference in guidance might have altered the nature of the activity, and masked the effect of medium on inquiry processes of the participants.

The main research questions that guided the present study were:

- How can we operationalize inquiry strategies in less well-structured and more complex activities?
- What inquiry strategies are related to better learning outcomes?
- How does strategy use differ between participants using either the physical or the simulation environment?

### 3.1. Sample

For Mass and spring activity in think-aloud condition, we had 36 community college students (24 female, 12 male, average age=20.5, SD=3.6).

### 3.2. Design

The study reported here is a between subject design with two levels. We randomly assigned participants to use either *physical* (PHY) or *computer simulation* (SIM) to engage in an inquiry-based activity on mass and spring oscillation ( $n_{PHY}=18$ ,  $n_{SIM}=18$ ). The task was to discover how the mass and the spring constant affect both the amplitude and the frequency of oscillation of a mass-spring system. We administrated a conceptual test before and after the activity. The post-test scores were the dependent measures of the experiment, while the pre-test scores were used as covariates in the corresponding statistical analyses. The relevant

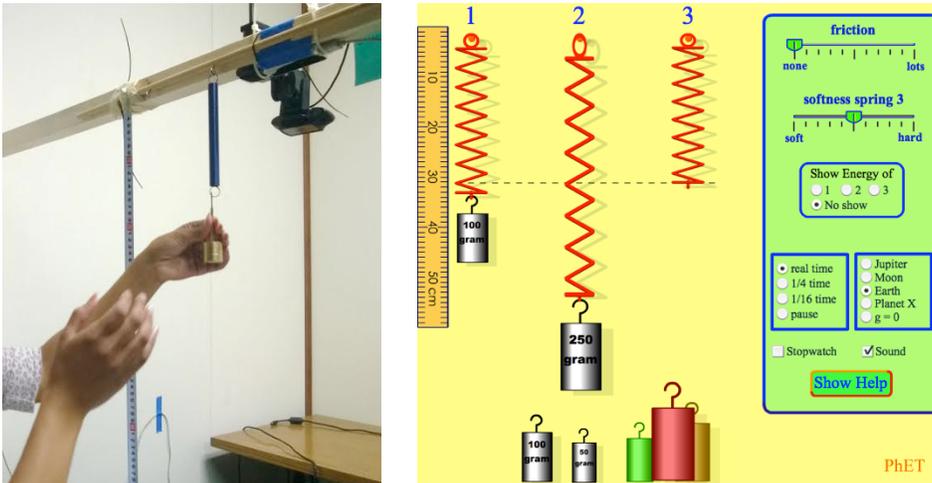


Figure 1. Experimental Setup: Left: Physical toolkit in action: The first hook is just next to the measure tape. Right: Computer Simulation: Participants were only allowed to change the “softness spring 3”.

behavioral measures were treated as independent within-subject variables since they were expected to predict learning outcomes.

### 3.3. Materials

#### 3.3.1. Learning Environment

**Physical Learning Environment.** The physical toolkit consisted of the PASCO<sup>1</sup> Demonstration Spring Set and Mass and Hanger Set. There are four pairs of springs, each with a spring constant between 4 N/m and 14 N/m. The masses consist of hangers to which slices of weights can be attached, ranging from 5 to 20 g. The environment consisted of two hooks, each being able to hold one spring, see Figure 1. For measuring extensions and duration, we provided a measuring tape and a stopwatch.

**Simulation Learning Environment.** The computer simulation we used was created by PhET [22], see Figure 1. It consists of three springs, two of which have a fixed and equal spring constant. The spring constant of the third spring can be changed continuously by means of a slider. It further entails seven weights, four of which are 50g, 100g, 100g and 250g respectively. The other three have no indication of their actual weight. The weights can be attached to and removed from the springs by simple drag-and-drop. The simulation comes with a displaceable measuring tape as well as a stopwatch.

**Differences in Learning Environment.** Instead of designing the learning environments ourselves, we selected the ones that we considered as state of the art of their respective domains. This prevented us from setting up the necessary control of the differences in affordances of the environments for making causal claims about the relation of learning environment and experimentation strategies. However, we can reason about the potentially relevant differences based on the specific user interfaces and interaction designs. The main differences are the following ones: 1. The VME allows participants to use up to three

springs, compared to two in the PME; 2. In the PME, participants could change the spring constant of both springs if needed, while the VME allowed to change the spring constant of only the third spring; 3. In the VME, manipulating the spring constant is easier as it requires only changing the value of a continuous valued slider. Participants could change its value on the fly, without interrupting an ongoing experiment. In order to change the spring constants in the PME, participants had to stop an experiment, and physically replace a spring with another one.

#### 3.3.2. Subject Knowledge Assessment Questionnaire

The pre-test and the post-test consisted of four qualitative questions, each with two sub-questions. The first two questions addressed the impact of changing either the spring constant or the mass on the amplitude and frequency of oscillation. The third question targeted the understanding of force and speed in an oscillating spring-mass system. The fourth question was a near-transfer question inspired by the generalization questions of Renken & Nunez [12].

#### 3.3.3. Procedure.

Students participated individually in the study. They were assigned randomly to either the PHY or the SIM condition. Prior to taking the pre-test, each participant was introduced to the nature and goal of the activity, and to definitions of relevant variables. Possible experiments were restricted only by the given set of weights and springs. The definition sheet contained basic definitions, both verbal and visual, of relevant concepts of harmonic oscillation of mass-spring systems. After the pre-test, the experimenter explained how to manipulate the variables and how to perform measurements, depending on condition using either the physical toolkit or the computer simulation. Participants were instructed to adjust only the settings related to the two variables of interest. They were further asked to think-aloud during the activity. The maximal duration of the inquiry task was 10 minutes. Participants then completed the post-test. Both pre-test and post-test took 5 minutes each.

### 3.4. Coding

#### 3.4.1. Conceptual Tests

Pre-test and post-test items received a score of 1 if they were correctly answered, and 0 otherwise. Questions that required participants to explain their reasoning were given 0.5 for partially correct answers. The maximum score was 8. Besides the overall aggregate score, we calculated also sub-scores for the two conceptual categories, spring constant (two items) and mass dependence (two items).

#### 3.4.2. Inquiry Behaviors

In a first pass, we extracted every experiment a participant ran from the corresponding video records of the experiment. This was done manually. Once the database was established, we could code every experiment computationally based on customized rules for

<sup>1</sup> PASCO scientific, 10101 Foothills Boulevard, P O Box 619011, Roseville, Ca 95678-9011, USA. Web: <http://www.pasco.com>. E-mail: [sales@pasco.com](mailto:sales@pasco.com). National representatives of PASCO can be reached through the USA office.

extracting relevant variables such as number of manipulated objects, etc. Even if the initial step was done by hand, the extraction procedure was operationalized such that we can automatize this process for future iterations: An experiment was characterized by the state of each relevant variable. A new experiment started when either one or more variables of the system were manipulated, or when a current experimental setup was re-initiated, either by touching a mass-spring system with the hand or with the mouse. The type of performed manipulation was then extracted from the contrast between two experiments. All variables representing inquiry behaviors are coded proportionally, relative to the total number of experiments run per activity.

An experiment consisted of the number of springs used, their spring constants, and the weights attached to the springs. The possible manipulations were (1) change of the spring constant, (2) change of the weight, (3) change both, (4) repeat an experiment, and (5) start a new experiment by changing the number of springs used. Changing either the mass only or the spring only corresponded to a *control of variables manipulation* (CVM), while a *confounded manipulation* referred to changing both variables at the same time. In cases participants used only one mass-spring configuration, we defined an experimental comparison through the contrast set up by the configurations in two consecutive runs. When two configurations were used simultaneously, the experimental comparison was defined by the contrast of those two sets of masses and springs. When participants in the SIM condition used all three springs, we defined the experimental comparison by the *most optimal* contrast out of the three possible pairwise combinations (optimal being the mass-spring configurations that differ only in one independent variable).

**Table 1. Regression Models of Post-Test Scores**

<i>Variables / Models</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
(Intercept)	3.79*** (0.68)	3.12** (0.24)	1.09 (1.38)	2.07* (0.16)	2.01* (0.96)
Pre-test Scores	0.32† (0.17)	0.32† (0.17)	0.29† (0.16)	0.32† (0.16)	0.34* (0.16)
Condition	0.33 (0.33)	0.49 (0.44)	-0.05 (0.35)	0.38 (0.37)	0.36 (0.37)
% Control of Variable		0.89 (1.60)			
% Confounded		1.28 (2.17)			
% Delib. Manip.			3.33* (1.50)		
% Delib. CV				3.17* (1.44)	
% Delib. Confounded				3.21 (2.09)	3.29 (2.06)
% Delib. Spring-Only					3.95** (1.52)
% Delib. Mass-Only					1.46 (1.86)
$R^2$	0.113	0.127	0.238	0.254	0.304
<i>adj. R</i> <sup>2</sup>	0.056	0.007	0.162	0.151	0.179
<i>N</i>	34	34	34	34	34

Note: Standard error are in parentheses; † ( $p \leq 0.1$ ), \* ( $p \leq 0.05$ ), \*\* ( $p \leq 0.01$ ), \*\*\* ( $p \leq 0.001$ ); each model regresses post-test scores on the given independent variables.

As explained before, just looking at whether an experiment was unconfounded or not misses out on other relevant aspects. In particular, such a perspective does not provide any insights into how deliberately or considered participants executed and reflected on an experiment. Therefore, we additionally captured the duration of each experiment as the *dwell time* between two succeeding experimental manipulations. Based on the dwell time, we developed a *measure of deliberateness*; any manipulation that had a dwell time bigger than first quartile of all dwell times of all participants was coded as a *deliberate manipulation*.

### 3.5. Data Analysis.

#### 3.5.1. Analysis of Learning Outcomes

In order to analyze the relation between inquiry behaviors and learning outcomes, we ran multiple linear regressions on post-test scores, with condition as independent factor, pre-test scores as covariate, and the corresponding measures of inquiry behavior as independent variables. For pairwise comparisons between variables within the same category that violated the normality assumptions, we report results from the nonparametric Mann-Whitney-Wilcoxon test.

#### 3.5.2. Analysis of Inquiry Behaviors

We applied a cluster method on all experimental manipulation variables to group participants by their inquiry behaviors. We used portioning around medoids (PAM) as the clustering algorithm, which is a more robust version of the standard k-means clustering algorithm, as it minimizes a sum of dissimilarities instead of a sum of squared Euclidian distances [23]. The quality of the clustering result was evaluated based on the silhouette score [24], a measure of similarity between points and the clusters they are assigned to. The larger the silhouette value, the better the clustering. However, instead of selecting the clusters that maximize the silhouette score, we have to make a trade-off between silhouette score and number of clusters in order to have theoretically relevant results. Ideally, we could set the number of clusters to 2, as we were interested in analysis of behaviors with respect to condition.

## 4. Results

### 4.1. Baseline Knowledge

Participants in the two conditions did not differ significantly in pre-test scores,  $t(32) = 1.49$ ,  $p = 0.15$  (PHY:  $M = 3.53$ ,  $SD = 1.59$ ; SIM:  $M = 4.23$ ,  $SD = 1.15$ ). However, the high overall pre-test score average of about 52.5% of the maximal possible score indicates that participants had relevant prior knowledge with regards to the subject. We excluded two participants who scored perfectly on the pre-test. In terms of prior knowledge related to impact of the spring constant versus the mass on harmonic oscillations, there were no significant differences in pre-test scores on the corresponding subcategories (Spring constant:  $M = 41.2\%$ ,  $SD = 31.3\%$ ; Mass:  $M = 52.9\%$ ,  $SD = 30.0\%$ ), paired  $t(33) = -1.54$ ,  $d = 0.38$ ,  $p = 0.13$ . However, as the trend in data nevertheless points in the expected direction, we classify experiments that involve spring manipulations as less familiar than those involving mass manipulations.

### 4.2. Effect of Condition on Learning Gain

The two conditions were not significantly different in terms of average learning outcomes as condition was not a significant

factor for post-test scores, controlling for pre-test scores,  $\beta = 0.33$ ,  $t(32) = 1.01$ ,  $p = 0.32$ ,  $\eta_p^2 = 0.03$  (see Figure 2.B.).

### 4.3. Learning Outcome by Inquiry Behaviors

We examined how various measures of inquiry behaviors related to learning outcomes by multiple linear regression analysis. The baseline variables of each regression model were condition as independent factor, and pre-test score as covariate. All the corresponding regression models are shown in Table 1.

#### 4.3.1. Time on Task and Number of Experiments

While time on task was the same across conditions,  $t(32) = 0.28$ ,  $p > 0.5$ , the total number of experiments per participant was higher for the SIM condition ( $M = 18.7$ ,  $SD = 8.3$ ) than for the PHY condition ( $M = 13.7$ ,  $SD = 7.3$ ),  $d = 0.64$ ,  $t(32) = 1.87$ ,  $p = 0.07$ . Additionally, pre-test scores were not correlated with number of experiments,  $r(32) = -0.05$ ,  $p > 0.5$ . An ANCOVA suggests that the number of experiments was not a significant factor for post-test scores, controlling for pre-test scores,  $F(1, 30) = 0.02$ ,  $p > 0.5$ ,  $\eta_p^2 < 0.01$ . Overall, participants performed 533 different experiments, based on which we built the database.

#### 4.3.2. Control of Variables Manipulations

We did not find a significant effect for overall CVM on post-test scores,  $\beta = 0.89$ ,  $t(31) = 0.33$ ,  $p > 0.5$  (see model 2 in Table 1). Even when looking at mass-only or spring-only manipulations, the respective regression coefficients are not significantly different from zero. These results indicate that performing control of variable manipulations of either the spring or the mass does not necessarily lead to better learning outcomes per se, which is in contrast to the prior literature [8]. We find that control of variable manipulations alone cannot explain the variability in learning outcomes both within and across conditions.

#### 4.3.3. Deliberate Manipulations

We coded the deliberateness of an experimental manipulation by means of the time spend on an experiment. We extracted the duration between manipulations across all participants, and defined the cut-off value between a *rapid* and a *deliberate* manipulation as the 25<sup>th</sup> percentile of the duration histogram ( $Mdn = 20$  seconds). This was at 11 seconds.

Overall deliberate manipulations was a relevant positive predictor of post-test scores,  $\beta = 3.33$ ,  $t(31) = 2.21$ ,  $p = 0.03$ ,  $\eta_p^2 = 0.14$  (model 3 in Table 1). While CVM was not relevant for learning outcomes, deliberate control of variable manipulations (DCVM) was a significant factor in the regression model 4 in Table 1,  $\beta = 3.17$ ,  $t(31) = 2.19$ ,  $p = 0.04$ ,  $\eta_p^2 = 0.15$ . This effect was mainly driven by deliberate spring-only manipulations (see model 5 in Table 1). On the other hand, deliberate

confounded manipulations had a comparably high coefficient value, even if it was not significant. With an adjusted  $R^2 = 0.18$ ,  $F(5,28) = 2.44$ ,  $p = 0.06$ , model 5 did not explain a higher proportion of variance than model 3,  $F(1,2) = 1.32$ ,  $p = 0.28$ .

None of the manipulation types correlated with pre-test scores (all correlation coefficients were lower than 0.1 in absolute value). The lack of correlation supports the claim that the manipulations were context-dependent variables of inquiry behavior.

### 4.4. Inquiry Behavior by Condition

#### 4.4.1. Control of Variables Manipulations and Deliberate Manipulations

The physical and the simulation condition did not differ in terms of control of variables manipulations,  $d = 0.14$ ,  $t(32) = -0.08$ ,  $p = 0.94$  (SIM:  $M = 0.51$ ,  $SD = 0.13$ ; PHY:  $M = 0.53$ ,  $SD = 0.16$ ). In contrast to that, the two conditions differed significantly in the amount of deliberate control variable manipulations (DCV),  $d = 0.77$ ,  $t(32) = 2.23$ ,  $p = 0.033$  (SIM:  $M = 0.35$ ,  $SD = 0.15$ ; PHY:  $M = 0.47$ ,  $SD = 0.18$ ). There is a significant drop in CV when considering the deliberate manipulations for the SIM condition only. In line with the hypothesis that the simulation environment was easier to manipulate, there were significantly more rapid manipulations in the SIM condition ( $Mdn = 17.6\%$ ,  $CI_{95} = \pm 24.5\%$ ) than in the PHY condition ( $Mdn = 0\%$ ,  $CI_{95} = \pm 12.9\%$ ),  $U = 219.5$ ,  $r = 0.46$ ,  $p = 0.007$ .

#### 4.4.2. Cluster Analysis of Inquiry Behaviors

Overall, DCV manipulations were a significant predictor for learning outcomes, in particular the deliberate spring-only manipulations. However, even if there was a significant difference in the amount of these manipulations between the PHY and SIM conditions, learning outcomes did not differ significantly by

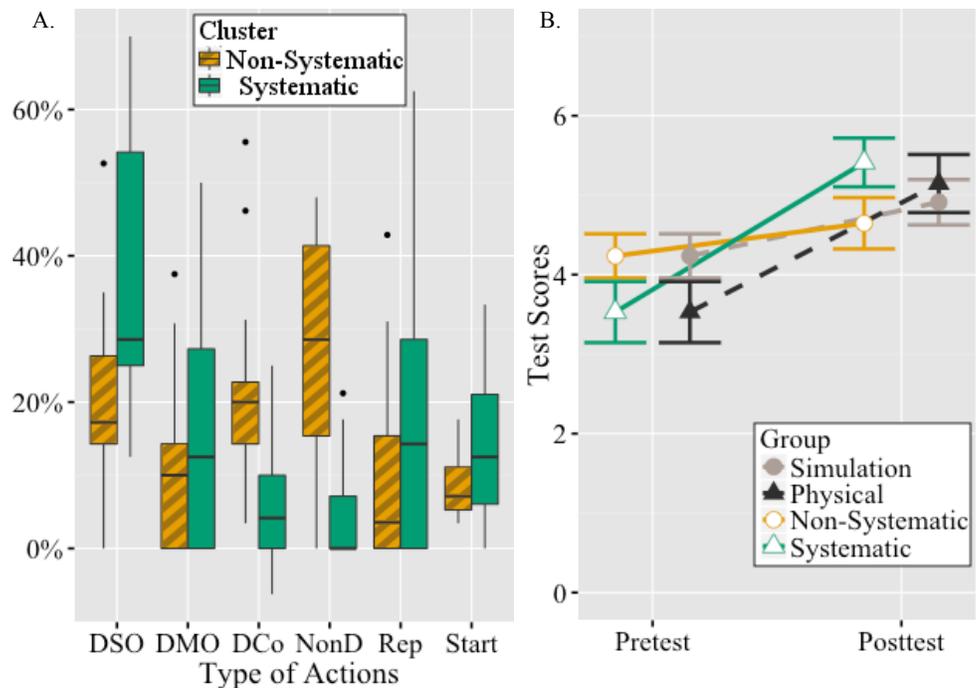


Figure 2. A. Boxplot of proportions of deliberate spring-only (DSO), deliberate mass-only (DMO), deliberate confounded (DCo), and non-deliberate (NonD) manipulations, repetitions (REP) and start of new experiments (Start). B. Comparison of pre-test and post-test scores by cluster as well as condition. Bars indicate standard errors.

condition. It appears that individual differences in inquiry strategies of participants within each condition washed out the actual impact of the learning environment on average post-test scores. There might be people in the physical and the simulation condition that deviated from the average inquiry behaviors for the condition towards the other condition's characteristics. We address this question by grouping all participants by considering all inquiry variables simultaneously instead of grouping them by condition, and then see how the groups distribute across the conditions. This can be done by means of cluster analysis.

Clustering was performed on the 6 possible manipulation types (see Figure 2.A) of the entire sample, which resulted in 2 clusters with 17 participants in each cluster. The average silhouette score was 0.30. While this score is not high enough to exclude the possibility of artificial data structures, an examination of the clusters in terms of variables confirms the clusters reasonably distinguish people by the level of systematicity of their inquiry behaviors: Generally, the participants of *Cluster 1* (“non-systematic”) were less strategic and less deliberate in their manipulations than *Cluster 2* (“systematic”) (see Figure 2.A). Cluster 2 had a higher proportion of deliberate spring-only manipulations than Cluster 1,  $U = 58, r = 0.51, p = 0.002$ , a lower proportion of non-deliberate manipulations than Cluster 1,  $U = 262.5, r = 0.72, p < 0.001$ , and a lower proportion of confounded manipulations,  $U = 240.0, r = 0.57, p < 0.001$ . There was no significant difference in the other variables. Additionally, even if the clustering was not performed on overall DCV, there is a large difference between the clusters; participants in the systematic cluster proportionally performed significantly more DCV (Mdn = 49.8%,  $CI_{.95} = \pm 16.3\%$ ) than in the non-systematic cluster (Mdn = 30.7%,  $CI_{.95} = \pm 12.1\%$ ),  $d = 1.33, t(32) = 3.89, p < 0.001$ .

The two clusters meaningfully differ in learning outcomes, as indicated by a regression of post-test scores on the cluster variable, with pre-test scores as covariates, which revealed a significant main effect of cluster,  $\beta = 1.03, t(31) = 2.39, p = 0.015, \eta_p^2 = 0.16$ . As expected, participants in the systematic scored higher than those in the non-systematic cluster (see Figure 2.B). The regression model explained a significant proportion of variance, adjusted  $R^2 = 0.18, F(2,31) = 4.55, p = 0.02$ .

**Table 2. Conditions distributed across clusters**

Condition	Non-Systematic	Systematic
	(n = 17)	(n = 17)
Physical (n = 17)	3 (17.6%)	14 (82.4%)
Simulation (n = 17)	14 (82.4%)	3 (17.6%)

Finally, Table 2 shows that the majority of participants in the systematic cluster used the physical toolkit, while the majority of participants that belonged to the non-systematic cluster were in the simulation condition, as confirmed by a Fisher's exact test,  $p < 0.0001$ .

## 5. DISCUSSION

Considerable attention has been given separately to research on the impact of virtual and physical learning environment [4] and of inquiry behaviors on the learning outcomes in science discovery activities [8,9]. The aim of the present study was to link these two realms by (1) studying the relation of strategy use and learning outcomes, and (2) comparing strategy use between learning

environments in order to shed light on how different affordances of the learning environments might influence strategy use.

### 5.1. Nuanced View of Experimentation Strategies in Open-Ended Inquiry Tasks

One main finding from this study was that one of the strongest predictors for learning outcomes when controlling for prior knowledge was the manipulation type that (a) created a single contrast in experiment conditions, (b) targeted the problem type that participants generally were less familiar with, and (c) was deliberate. In the context of the mass and spring activity, these were deliberate manipulations that changed only the spring constant from one mass-spring system to the other.

Importantly, this further implies that the control of variables (CV) in experiment design was a necessary but not sufficient condition for developing conceptual understanding through experimentation. This is in contrast to prior research that has predominantly focused on the ability to design unconfounded experiments as the main factor of knowledge acquisition in inquiry learning [2,10,12]. Using control of variable strategy as an important factor for characterizing experimentation strategies works when the student has to make a conscious decision to actually apply this strategy. It fails if the affordances of the user interface do not require that. In the computer simulation, one could change the spring constant continuously using a slider, even during an ongoing experiment. In the physical condition however, an experiment had to be interrupted in order to change either the mass or the spring, which required the participant to deliberately decide what to manipulate, but both changes are coded as CV manipulations. As a consequence, we not only found that there was no difference in CV manipulations between conditions, but also that these manipulations did not have predictive value for learning outcomes.

This picture changed when accounting for the *deliberateness* of experimental manipulations. It turned out to that in contrast to CV manipulations, the percentage of deliberate CV manipulations significantly predicted learning outcomes, as well as differed between conditions. The drop from CV to deliberate CV manipulations was significant only for the SIM condition. This is in line with our reasoning that the user interface for the computer simulation did not make the control of variables a deliberate choice. Even by itself, deliberate manipulations were among the strongest predictor for post-test scores. We suggest that time between manipulations as a measure of deliberateness is not just reflective of the ease of manipulation in a learning environment, but also of the level of cognitive engagement of a participant with an experiment.

Finally, only manipulations targeting the less familiar concept (spring) contributed to conceptual learning, while those targeting the more familiar one (mass) did not seem to impact the learning outcomes, which seems reasonable given that the participants tended to know less about the springs' role in the harmonic oscillation. However, contrary to previous studies [12] that consider confounded manipulations as detrimental to developing conceptual understanding, we found a relatively large though insignificant positive regression coefficient for confounded manipulations on post-test scores. At this point, we can only speculate as to why this is the case; for example, it could be that people with low prior knowledge ran preliminary experiments to get a sense of the physical phenomenon. Further investigation is needed to understand this process.

## 5.2. Differences in Inquiry Behaviors by Learning Environment

We found that conditions did not differ in terms of learning outcomes. In line with previous research that showed equal knowledge gains for virtual and physical manipulative environments [2, 3, 5, 7], we could have argued that there is no difference in benefits of learning environments for developing conceptual understanding in inquiry tasks on mass-spring systems. However, as indicated by the results of the cluster analysis of inquiry behaviors, this would have been the wrong conclusion. The cluster analysis revealed that participants across both conditions could be grouped into two clusters according to how systematic their inquiry behavior was, and that the more systematic cluster had significantly higher learning outcomes than the less systematic cluster. Importantly, almost all of the participants in the physical condition belonged to the more systematic cluster, while most of the participants in the simulation condition fell into the less systematic cluster. This suggests that the learning environments did differ in terms of benefits for developing conceptual understanding. It is important to note that this is not in contradiction to the multiple regression models that show no significant effect for condition. Both analyses show that inquiry strategies had a strong influence on learning outcomes. However, enough participants deviated from their peers in the same condition in terms of inquiry behaviors such that the overall differences in learning outcomes between conditions were canceled. By using more than one variable of inquiry behavior for grouping participants, cluster analysis better accounts for between subject differences in overall inquiry behaviour in each condition. Thus, at least for activities that span a short period of time, we think that measures of experimentation strategies have to be incorporated in studies of the impact of learning environments on learning outcomes in open-ended science inquiry learning.

A possible explanation for these differences in experimental manipulations between conditions is that the ability to employ systematic experimentation strategies is not necessarily a stable domain-general skill but a context-dependent behavior. It is likely that specific affordances of the two learning environments are related to these differences in experimentation strategies, such as the need to pause the experiment to change the spring constant in the real but not virtual environment. While there is consensus on the impact of different affordances of virtual and physical environments on learning outcomes [4], we argue in light of these results that we also need to study the impact of these affordances on the experimentation processes during science inquiry activities. However, as we did not manipulate the specific affordances in the learning environments, we can currently only make educated guesses.

For example, the fact that participants in the SIM condition ran more experiments than in PHY, while spending the same amount of time at the task, supports the claim that it was easier to manipulate variables in the computer simulation than in the physical setup. As argued by Renken and Nunez [12], it might be that systems that enable quick changes with various options prompt participants to get into “play” mode, in which they revert to simple heuristic methods such as trial-and-error and spend less effort on setting up valid experiments. This could explain why proportion of deliberate manipulations was higher for participants using the physical systems.

Another difference in affordances is that in the computer simulation, participants could change the spring constant even as experiments were running, which led to short perturbations in the

oscillations that were due to the change, and not necessarily due to the actual spring-mass configurations. Especially in cases “non-deliberate” manipulations that were too short for the perturbations to vanish, participants might have wrongly interpreted these fluctuations.

## 5.3. Limitations and Future Directions

While the study provided evidence that an investigation of inquiry strategies is more informative than merely looking at outcomes, it only offered hints as to what determines the use of those strategies. These appear to be influenced by the different affordances of a learning environment, but studies with longer interaction times, and a greater range and control of environments is needed to understand the characteristics of these relationships in more detail. Future studies should better control and match the virtual and physical environments in order to focus on one or two specific affordances. Studies that manipulate design features *within* a learning environment to assess its impact on inquiry processes are also needed.

Further studies should incorporate the assessment of hypothesis generation and inference processes to examine the impact of affordances of learning environments not just on experimentation strategies, but on these other critical inquiry behaviors as well.

We found that time between manipulations was an important correlate of learning outcomes; however, with the current study, we can make only educated guesses as to what cognitive processes longer dwell times correspond to. Dwell time could signify the time spent on comparing the current with the prior experiment configuration, on reflecting on existing confusions, on planning the next steps to be taken, or it could just represent the time it takes to perform a manipulation in the learning environment.

Additionally, the lack of difference on learning outcomes between media seems to contradict prior research on virtual versus physical learning environments in comparable inquiry tasks [12]. However, as the tendency of the data goes into the expected direction, we believe that a larger sample size would provide the required power to detect the learning outcome differences.

We currently did not employ automated tracking of participants’ behaviors to extract their experiment configurations. However, novel computer vision algorithms, as well as logging systems would address this limitation. Our data organization scheme can be easily integrated with automatized tracking systems.

## 6. CONCLUSION

Drawing on work on scientific reasoning and inquiry, we developed a novel operationalization of systematic experimentation strategies that predict learning outcomes in open-ended inquiry-based learning activities. We further showed that strategy use is context-dependent, in that participants using the physical system went about the inquiry activity differently than participants using the computer simulation.

These findings suggest that we have to broaden the notion of what counts as “systematic experimentation” from mainly consisting of the design of unconfounded experiments and the performance of optimal heuristic search to a more comprehensive views that integrates contextual and cognitive factors (e.g. deliberateness). Data mining algorithms are particularly well suited for exploring such behaviors. However, it is crucial to develop data-mined models of inquiry strategies that are interpretable in order to advance our understanding of learning processes in more complex

inquiry activities. We suggest that any machine-learned model of inquiry behaviors should incorporate semantic representations of what participants' actually explore in inquiry activities, in order to meaningfully extend the data from interaction logs of users engaging in the learning environment.

A further implication of our results is that research on learning environments for science inquiry learning should focus on developing a broader framework that focuses on the affordances as relevant dimensions, irrespective of medium and examines how under what circumstances they benefit learning.

## 7. ACKNOWLEDGEMENT

We would like to thank Prof. Carl Wieman and Eric Kuo, PhD, for their guidance and strong support in this research, as well as members of the AAALab at the Stanford University for their insightful feedback.

## 8. REFERENCES

- [1] van Joolingen, W., & Zacharia, Z. (2009). Developments in inquiry learning. In *Technology-enhanced learning*. Netherlands: Springer.
- [2] Triona, L., & Klahr, D. (2003). Point and click or grab and heft: Comparing the influence of physical and virtual instructional materials on elementary school students' ability to design experiments. *Cognition and Instruction, 21*, 149-173.
- [3] Zacharia, Z., & Olympiou, G. (2011). Physical versus virtual manipulative experimentation in physics learning. *Learning and Instruction, 21*, 317-331.
- [4] de Jong, T., Linn, M., & Zacharia, Z. (2013). Physical and virtual laboratories in science and engineering education. *Science, 340*, 305-308.
- [5] Klahr, D., Triona, L., & Williams, C. (2007). Hands on what? The relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. *Journal of Research in Science Teaching, 44*, 183-203.
- [6] Zacharia, Z., & Constantinou, C. (2008). Comparing the influence of physical and virtual manipulatives in the context of the physics by inquiry curriculum: The case of undergraduate students' conceptual understanding of heat and temperature. *American Journal of Physics, 76*, 425-430.
- [7] Pyatt, K., & Sims, R. (2012). Virtual and physical experimentation in inquiry-based science labs: Attitudes, performance and access. *Journal of Science Education and Technology, 21* (1), 133-147.
- [8] Zimmerman, C. (2000). The development of scientific reasoning skills. *Developmental Review, 20*, 99-149.
- [9] Zimmerman, C. (2007). The development of scientific thinking skills in elementary and middle school. *Developmental Review, 27*, 172-223.
- [10] Chen, Z., & Klahr, D. (1999). All other things being equal: Acquisition and transfer of the control of variables strategy. *Child Development, 70*, 1098-1120.
- [11] Klahr, D. (2000). *Exploring science: The cognition and development of discovery processes*. Cambridge: MIT Press.
- [12] Renken, M., & Nunez, N. (2013). Computer simulations and clear observations do not guarantee conceptual understanding. *Learning and Instruction, 23*, 10-23.
- [13] Shih, B., Koedinger, K., & Scheines, R. (2010). Unsupervised Discovery of Student Strategies. *Proceedings of the 3rd Intl. Conf. on Educational Data Mining*, (pp. 201-210).
- [14] Kardan, S., & Conati, C. (2011). A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. *Proceedings of the 4th Intl. Conf. on Educational Data Mining*, (pp. 159-168). Eindhoven, the Netherlands.
- [15] Kardan, S., Roll, I., & Conati, C. (2014). The usefulness of log based clustering in a complex simulation environment. *Intelligent Tutoring Systems* (pp. 168-177). Springer International Publishing.
- [16] Sao Pedro, M., Baker, R., Gobert, J., Montalvo, O., & Nakama, A. (2013). Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction, 23* (1), 1-39.
- [17] Sao Pedro, M., Baker, R., & Gobert, J. (2012). Improving construct validity yields better models of systematic inquiry, even with less information. *User Modeling, Adaptation, and Personalization* (pp. 249-260). Berlin Heidelberg: Springer.
- [18] Palmer, D. (1995). *The POE in the primary school: An evaluation*. *Research in Science Education, 25* (3), 323-332.
- [19] Penner, D., & Klahr, D. (1996). The interaction of domain-specific knowledge and domain-general discovery strategies: A study with sinking objects. *Child Development, 67*, 2709-2727.
- [20] Schauble, L. (1996). The development of scientific reasoning in knowledge-rich contexts. *Developmental Psychology, 32*, 102-119.
- [21] Garcia-Mila, M., & Andersen, C. (2007). Developmental change in notetaking during scientific inquiry. *International Journal of Science Education, 29* (8), 1035-1058.
- [22] Perkins, K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., Wieman, C., & LeMaster, R. (2006). PhET: Interactive simulations for teaching and learning physics. *The Physics Teacher, 44*(1), 18-23.
- [23] Reynolds, A., Richards, G., de la Iglesia, B., & Rayward-Smith, V. (1992, 5). Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms, 475-504*.
- [24] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics, 20*, 53-65.

# Toward a Real-time (Day) Dreamcatcher: Sensor-Free Detection of Mind Wandering During Online Reading

Caitlin Mills  
University of Notre Dame  
Department of Psychology  
Notre Dame, IN 46556  
cmills4@nd.edu

Sidney D'Mello  
University of Notre Dame  
Department of Psychology  
Department of Computer Science  
Notre Dame, IN 46556  
sdmello@nd.edu

## ABSTRACT

This paper reports the results from a sensor-free detector of mind wandering during an online reading task. Features consisted of reading behaviors (e.g., reading time) and textual features (e.g., level of difficulty) extracted from self-paced reading log files. Supervised machine learning was applied to two datasets in order to predict if participants were mind wandering as they navigated from one screen of text to the next. Mind wandering was detected with an accuracy of 20% above chance (Cohen's kappa = .207; AUC = .609), which was obtained via leave-one-participant-out cross-validation. Similar to actual rates of mind wandering, predicted rates of mind wandering were negatively related to posttest performance, thus providing some evidence for the predictive validity of the detector. Applications of the detector to attention-aware educational interfaces are discussed.

## Keywords

Mind wandering, attention, machine learning, reading

## 1. INTRODUCTION

It is not uncommon to experience looking up from a book only to realize you have no idea what you just read. In fact, it has been documented that people can read up to 17 words of gibberish before even realizing that they have zoned out [32]. Since students often have trouble realizing when they have zoned out themselves, it can be especially difficult to determine when someone is not paying attention through observation. For example, a student who is deeply engaged in learning can often look quite similar to another student who is thinking about something else completely.

This phenomenon, known as *mind wandering*, is an involuntary shift in attention away from the external task towards task-unrelated thoughts [36]. Mind wandering is detrimental during learning, as learning requires consolidating external information into mental structures. During episodes of mind wandering, however, students are unable to integrate external information with their existing internal representations. Thus, missed information is not processed and mental models are not updated, limiting overall understanding. Given the negative impact of mind

wandering on learning [14, 30, 32, 33], it is important to develop systems that can reorient attention when students mind wander in order to facilitate engagement and learning. Building detectors of mind wandering is an essential first step towards this goal and is the focus of the present paper.

## 1.1 Related Work

One of the first known studies related to mind wandering detection was conducted by Drummond and Litman [13]. In their study, students read a paragraph about biology aloud then performed a learning task (i.e., paraphrase or self-explanation). Students periodically self-reported how frequently they were thinking about off-task thoughts on a scale from 1 (all the time) to 7 (not at all). Supervised machine learning trained on acoustic-prosodic features was used to classify whether students were "high" in zoning out (1-3 on the scale) versus "low" in zoning out (5-7 on the scale). Results indicated an accuracy of 64% in discriminating "low" versus "high" zone outs. This pivotal study on mind wandering was innovative with respect to automatically detecting zone outs during a learning task. However, they used a leave-one-instance-out cross-validation method (rather than a leave-one-participant-out cross-validation method), so generalizability of the detector to new students is unclear.

Recent research has also attempted to detect mind wandering during online reading using both gaze [5] and peripheral physiology [6]. In both of these studies, mind wandering was collected via thought probes that occurred on pseudo-random pages (i.e., computer screens) during reading. Students responded either "yes" or "no" about whether they were mind wandering at the time of the probe. In the first study, a detector of mind wandering achieved an accuracy of 72% (Cohen's kappa = .28) using features extracted from gaze data collected with a Tobii eye tracker [5]. In the second study, a detector of mind wandering built using physiological features (i.e., skin conductance and temperature) achieved an accuracy of 74% (Cohen's kappa = .22). Both of these detectors used a leave-several-subjects-out validation method to ensure generalizability to new students.

These detectors display impressive results given the elusive nature of mind wandering. However, the equipment and sensors required for eye-gaze and physiology tracking might impair scalability. In particular, one issue faced by online learning environments is that sensors are not readily available. For example, students using an ITS deployed online from their home computer would not have access to an eye tracker or a way to measure skin conductance at their convenience. A key question then is how to detect mind wandering based on information that is readily available, for example, in interaction log files. Along these lines, the aim of the current study is to identify a set of features that 1) are theoretically

Copyright space

‘  
‘  
‘  
‘  
‘  
‘  
‘

related to mind wandering, and 2) can be extracted from log files during online learning.

Interaction-based detectors trained from interaction log files have been used to successfully build detectors of other “off-task” states, such as gaming the system and off-task conversation [4, 7–9]. While mind wandering is related to other forms of “off-task” states, such as boredom, behavioral disengagement, and distractions [1, 3, 4, 8, 9, 26, 42], it is inherently distinct because it is involuntary and involves internal thoughts rather than overt expressive behaviors. The involuntary, unconscious nature of mind wandering makes detection particularly difficult. First, whereas other off-task states often involve some behavioral markers to denote disengagement, mind wandering is a completely internal state that can look similar to on-task states. Second, the onset and duration of mind wandering episodes cannot be precisely measured because people are often unaware their attention has been directed away from the external task. Thus, finding features that will pick up on subtle differences in attention is extremely difficult.

To date, one study has attempted sensor-free mind wandering detection (see Table 1 for a summary of mind wandering detectors). Franklin et al. [15] attempted to classify if readers were “mindlessly reading” using two criterion: (1) difficulty and (2) reading time. For the first criterion, readers could only be classified as “mind wandering” while reading difficult text. To establish the level of difficulty, each word was assigned a difficulty score based on the average of three binary ratings: (1) length (at least four letters = 1, less than four letters = 0), (2) syllables (at least two syllables = 1, under two syllables = 0), and (3) familiarity (based on a psycholinguistic database where above average = 1, below average = 0). Then, the average difficulty across a running window of 10 words had to be above a threshold set at .45 for a reader to be classified as “mindless reading.” The second criterion was based on reading time. Participants read one word on a screen at a time. Using a running window of 10 words, a specific threshold (based on pilot data) was applied to determine when readers were reading either too fast or too slow.

**Table 1. Overview of Previous Mind Wandering Detectors**

	<b>Key Features</b>	<b>Classification Accuracy</b>	<b>Validation Method</b>
Bixler et al. (2014)	Eye Gaze	72% correct	leave-several-subjects-out
Blanchard et al. (2014)	Physiology	74% correct	leave-several-subjects-out
Drummond et al. (2010)	Prosodic/ Lexical	64% correct	leave-one-instance-out
Franklin et al. (2011)	Difficulty/ Reading Time	72% correct	thresholds derived from pilot data

This study provided some evidence that reading time, combined with textual features such as difficulty, might be indicative of mind wandering (accuracy = 72%). However, since reading times were collected by presenting one word on the screen at a time, their methods and predetermined thresholds for fast and slow

reading may not be generalizable to other, more natural, reading contexts. Additionally, mind wandering was never predicted to occur during “easy” portions of the text, which may not accurately reflect the real-life occurrence of this phenomenon. For example, mind wandering still occurs around 20% during easy texts [27], even though it is more frequent during difficult texts. Furthermore, their method relied on a number of pre-set thresholds with little information on how these thresholds were established, thereby complicating attempts to replicate their results.

## 1.2 Current Study

This paper reports a person-independent detector of mind wandering during a more natural, computerized self-paced reading task using basic information that can be extracted from reading logs. In an attempt to provide a foundation for an easily-scalable way to capture when mind wandering occurs, the detector is completely sensor-free.

The mind wandering detector was trained on two unpublished datasets in which participants attempted to learn about scientific research methods by reading texts presented online. Participants completed a posttest after reading in order to assess learning. Importantly, these datasets include diversity with respect to population, methods, and level of text difficulty. For example, dataset 1 was collected via Mechanical Turk, a validated online data collection platform [23], and had an average age of 35 years. Dataset 2 was collected from a Midwestern university subject pool and had an average age of 19 years. Therefore, building a detector of mind wandering using more than one dataset with varying conditions will increase our confidence in its relative generalizability.

## 2. DATASETS

The datasets were originally collected to investigate mind wandering under various conditions, such as varying levels of difficulty and text presentations. In addition, a posttest was completed after reading in order to assess how mind wandering relates to learning. In both datasets, participants were instructed to read the text carefully and notified that they would be asked to answer questions about content from the text after reading. Dataset 1 ( $N = 177$ ) was collected on Amazon’s Mechanical Turk, an online data collection platform that has been validated for high quality data [23, 28]. Participants were compensated \$2.50 after completing the experiment. Dataset 2 ( $N = 141$ ) was collected via an online subject pool at a Midwestern university in the United States. Participants received class credit after completing the study.

Table 2 provides an overview of the experimental designs and manipulations used in each dataset. The Text Difficulty manipulation involved participants reading texts that were experimentally manipulated to be either “easy” or “difficult” (see section 2.1 for manipulation details). The Text Presentation manipulation involved participants reading either one sentence or one paragraph at a time on the screen.

### 2.1 Reading Materials

The two texts used in the existing datasets were adapted from texts used in the serious game, *Operation ARA!* [25]. Each text focused on a concept related to research methods: (1) the dependent variable and (2) making causal claims, both of which are key concepts relevant to understanding the scientific method. In the existing datasets, easy and difficult versions of each text

were used in order to investigate the effect of text difficulty on mind wandering.

Easy versions of the text were more narrative in nature, and consisted of shorter sentences and fewer low frequency words (average Flesh-Kincaid Grade Level = 9). Difficult versions of the text consisted of longer, more complex sentences with more low frequency words (average Flesh-Kincaid Grade Level = 13). Both versions had the same conceptual content and were approximately 1500 words in length. An example of an easy sentence is, “People who know about the scientific method do not fall for unsupported claims like this one.” The difficult version of the same sentence was, “So many citizens fall for these dubious claims, but people who comprehend the scientific method are not victimized by these unsupported claims.”

## 2.2 Procedure

Participants first completed an electronic consent form. They were then given instructions for the self-paced learning task. Participants pressed the space bar to move through each screen of the text. Texts were presented on screen either one sentence at a time or one paragraph at a time based on experimental manipulation (see Table 2).

Mind wandering was tracked via auditory thought probes in both datasets. A standard description of mind wandering [36] was employed: “At some point during reading the texts, you may realize that you have no idea what you just read. Not only were you not thinking about the text, you were thinking about something else altogether.” The probe consisted of an auditory beep that occurred on pseudo-random screens throughout each text. Probes were triggered when participants pressed the space bar to advance to the next portion of the text. Participants were instructed to press the “Y” key if they were mind wandering or the “N” key if they were not. Participants were not able to advance to the next screen until they had responded to the mind wandering probe. A total of six auditory mind wandering probes were inserted in each text. Probes were placed in an identical location with respect to content within each text. That is, regardless of whether the text was presented one sentence or paragraph at a time, the probe would occur after reading identical content.

**Table 2. Overview of Two Datasets**

	Dataset 1	Dataset 2
Sample	Mechanical Turk	University subject pool
# Texts	1	2
# Participants	177	141
<b>Manipulations:</b>		
Text Difficulty	Easy/Difficult	Difficult only
Text Presentation	Par/Sen	Par/Sen

*Notes.* Par = Paragraph-by-paragraph; Sen = sentence-by-sentence

Participants completed a posttest after reading each topic. Posttests consisted of four-alternative multiple-choice questions that tapped two levels of comprehension: (1) surface level, and (2) inference level. Surface level questions were based on factual or text level characteristics of the text. Inference questions were designed to elicit patterns of reasoning and require participants to use inference or apply a learned concept to a novel example in

order to answer the question correctly [19]. For dataset 2, participants answered an 18-item posttest that covered both topics, which included six inference and 12 surface level multiple-choice questions. Since only one text was read during dataset 1, the posttest was limited to the 9 corresponding questions (3 inference and 6 surface level questions).

## 2.3 Mind Wandering Reports

Every screen of text where a probe was triggered was classified as either “Mind Wandering” or “Not Mind Wandering” based on participants’ response to the probe. The two datasets were pooled in order to maximize training and validation data. In total, there were 2754 probe screens that were used to build the models. Participants indicated they were mind wandering in response to 31.3% of all the probes. Thus, our data set contained 861 instances of Mind Wandering and 1893 instances of Not Mind Wandering.

## 3. MODEL BUILDING

### 3.1 Feature Engineering

A considerable amount of empirical research has been dedicated to understanding mind wandering through experimental manipulations, such as comparing mind wandering across various conditions. Other studies have focused on explaining the behavioral correlates and temporal patterns of mind wandering [14, 16, 16, 27, 34, 38, 40]. The features in the current research were informed by the following discoveries about mind wandering: First, mind wandering is affected by the difficulty of a task [14, 27]. Second, mind wandering is related to response times and lexical features [15, 29]. Third, mind wandering rates vary as a function of time on task [30, 40]. In line with these findings, a total of 13 features were computed based on information that can be found in log files. The 13 features can be subdivided into three categories: (1) Reading Behavior Features (2 features), (2) Textual Features (8 features), and (3) Context Features (3 features).

**Reading Time Features.** Participants’ reading time (i.e. how long they spent on each screen) was collected during the reading task. Importantly, the thought-probe was triggered as participants attempted to move on to the next screen. Therefore, we can use reading behaviors from the current screen of text (screen K) to detect whether they are mind wandering or not before they moved on to the next screen (K+1).

The first reading behavior feature was *Reading Time*, which was simply the amount of time spent reading a given paragraph before pressing the space bar to advance onto the next screen. Reading Time was computed at the paragraph level in order to account for differences in reading times across the Text Presentation manipulation. When texts were presented one paragraph at a time, *Reading Time* was simply how long they spent on the screen leading up to the thought-probe. When texts were presented one sentence at a time, sentences were aligned with the content from the paragraph presentation condition. Thus, *Reading Time* was calculated as the amount of time spent reading identical content before the thought-probe regardless of presentation style.

The second reading behavior feature was called *Decoupling* [41]. *Decoupling* is a theoretically-driven metric based on the idea that reading times should increase with more complex text characteristics, such as sentence length and other discourse features [18]. If participants are not appropriately allocating resources (i.e., increasing reading times when text complexity increases) to meet the current task demands, then we might expect deviation from this linear relationship thus indicating decoupling

from the reading task. *Decoupling* was computed on the alignment (or misalignment) of reading times and text complexity. Text complexity was assessed using Flesh-Kincaid Grade Level (FKGL; [22]). The formula used to calculate decoupling was:  $|z\text{-score standardized reading times} - z\text{-score standardized FKGL}|$ . It is important to point out that decoupling was computed using the absolute value of the difference between reading time and text complexity, such that higher values would occur both when reading times were both over and under appropriated relative to text complexity. Thus, we are primarily interested in how well the overall magnitude of deviation in the relationship between reading time and text complexity can predict mind wandering.

**Textual features.** Eight textual features were computed in total. The first feature was simply the *Number of Characters* in the current paragraph. The second feature was the *Number of Words* in the current paragraph. Both features were used because they may differ notably between easy and difficult conditions, as easy texts were specifically manipulated to contain shorter words. Regardless of whether the screen was being presented one paragraph at a time or one sentence at a time, these features were used to represent the length of the current unit of text being processed. Longer paragraphs may require increased cognitive resources (related to mind wandering [24]) when a single idea must be kept in working memory across larger amounts of text. The third feature was *FKGL* [22], an indicator of reading level that is derived from the number of syllables and word length in a sentence. The current FKGL was also computed based on the current paragraph being read, as this metric is not reliable for extremely small portions of text, such as a single sentence.

The remaining five textual features were computed using Coh-Metrix, a program that analyzes texts across multiple levels of cognition and comprehension [17, 18]. We used five different features from Coh-Metrix: (1) Narrativity, (2) Deep Cohesion, (3) Referential Cohesion, (4) Syntactic Simplicity, and (5) Word Concreteness. *Narrativity* is computed based on how well the text aligns with the narrative genre, by conveying a story, procedure, or sequence of actions. *Deep Cohesion* is computed based on how well different ideas in the text are cohesively tied together in order to signify causality or intentionality. *Referential Cohesion* is based on how words and ideas are connected to each other across the span of the story or text. *Syntactic Simplicity* is computed based on the simplicity of the syntactic structures in the text. Lastly, *Word Concreteness* is based on the degree to which context words evoke concrete mental images, rather than abstract or conceptual representations.

**Context features.** Three context features were also computed based on the context of the reading task. *Current Paragraph Number* is the number of paragraphs read from the beginning of the text. *Current Difficulty* is whether the text was experimentally manipulated as easy or difficult. *Current Presentation* is whether the text was being presented one sentence at a time or one paragraph at a time.

### 3.2 Supervised Classification and Validation

We used supervised machine learning to build detectors of mind wandering for each screen that included a thought-probe. The goal of the paper was to create a detector that would accurately predict whether participants responded “yes” or “no” to the mind wandering probes. RapidMiner, a popular machine learning tool, was used to train binary classifiers to make this distinction. In total, four binary classifiers provided in RapidMiner were used, including Naïve Bayes, Bayes Net, RIPPER (JRip implementation), and C4.5 (J48 implementation). Down-sampling

was used to create equal classes for the training data only. This was achieved by randomly selecting 45.4% of the Not Mind Wandering instances and 100% percent of the Mind Wandering instances for training. The original distributions were not changed in the testing data to preserve the validity of the results.

Manual feature selection was applied by removing one feature at a time and assessing performance on held-out testing data (see below). If model performance decreased after a feature was removed, it was preserved for the final model<sup>1</sup>.

All models were evaluated using leave-one-participant-out cross-validation, in which  $k-1$  participants are used in the training data set. The model was then tested on the participant who was not used in the training data. This process was repeated  $k$  times until every participant was used as the testing set once. Cross-validating at the participant level increases confidence that models will be more generalizable when applied to new participants because the testing and training sets are independent.

Classification accuracy was evaluated using two metrics: (1) Area Under the ROC Curve (AUC), and (2) Cohen’s kappa. AUC is statistically similar to  $A'$  [21] and ranges from 0 to 1, where 0.5 is chance level of accuracy and 1 is perfect accuracy. Cohen’s kappa [10] indicates the degree to which the model is better than chance (kappa of 0) at correctly predicting Mind Wandering or Not Mind Wandering. A kappa of 1 indicates the detector performs perfectly. We also report percent correctly classified (accuracy), but note that this should be interpreted cautiously since class imbalance tends to inflate accuracy.

## 4. RESULTS

### 4.1 Classification Accuracy

Four classification algorithms (J48, JRIP, Naïve Bayes, and Bayes Net) were applied to the two combined datasets. The final models reported in this section were selected based on the highest AUC achieved after testing all four classification algorithms. A final combined feature model (combined model) was achieved with the J48 decision tree classifier using six features from the feature subtypes: *Reading Time*, *Decoupling*, *Number of Characters*, *Number of Words*, *FKGL*, and *Referential Cohesion*. Importantly, the combined model performed at rates above chance (AUC = .609; kappa = .207; accuracy = 63%). Despite using information solely obtained from log files and text characteristics, these accuracy rates are only slightly lower than the sensor-based detectors of mind wandering reported in Table 1.

We also examined the confusion matrix for the final combined model (see Table 3). The model had a relatively high rate of misses (.427), where actual instances of Mind Wandering were predicted as Not Mind Wandering. However, the model also displayed more correct rejections (.653), such that Not Mind Wandering instances were accurately classified as Not Mind Wandering. This was complemented by a low rate of false alarms as well (.347).

We were also interested in exploring how each of the three feature subtypes (i.e., reading behaviors, textual, and context features) were able to predict mind wandering independently. Each group of feature subtypes was therefore tested independently using the same four classification algorithms (J48, JRIP, Naïve Bayes, and Bayes Net). A summary of the classification accuracies for the

---

<sup>1</sup> We also tested models using all 13 features, which exhibited lower performance (assessed via AUC) than the combined model using feature selection.

best performing models (selected based on highest AUC) can be found in Table 4.

**Table 3. Confusion Matrices of Combined Model**

	Pred. MW	Pred. Not MW	Priors
<b>Actual MW</b>	.573 (hit)	.427 (miss)	.313
<b>Actual Not MW</b>	.347 (false alarm)	.653 (correct rejection)	.687

Note. Pred. = Predicted; MW = Mind Wandering

All three models built from the feature subtypes performed above chance levels (AUC > .5). However, none of these models performed as well as the combined model. For example, the Textual Features Only model did not perform as well in the absence of reading time behaviors and vice versa. This suggests that using a range of feature types might help with classification accuracies rather than a subset of features.

Based on the confusion matrices, it appears that the three feature subtype models exhibited different patterns of classification (see Table 5). Although the Reading Behaviors Only model (*Reading Time* and *Decoupling*) displayed the lowest hit rates (.439), this model also had the highest rate of correct rejections. Conversely, the Textual Features Only (five Coh-Matrix dimensions, *Number of Characters*, and *Number of Words*) and the Context Features Only (*Current Presentation*, *Current Difficulty*, and *Current Paragraph Number*) models had similar higher hit rates, but fewer correct rejections compared to the Reading Behaviors Only Model.

**Table 4. Performance Metrics**

Features in model	AUC	Kappa	Classifier
Combined Model	.609	.207	J48
Reading Behaviors Only	.560	.122	J48
Textual Features Only	.591	.115	Bayes Net
Context Features Only	.542	.104	JRIP

It is important to point out that the combined model's confusion matrix also shared some similarities with the feature subtype models. The Reading Behavior Only model had the highest correct rejections (.687), which were on par with the combined model (.653). Similarly, the Textual Features Only and Context Features Only models had the best hit rates (.554 and .557), which were also on par with the hit rates in the combined model (.573). Thus, the combined model appears to strike a balance between hits and correct rejection, which is why it yields the highest AUC compared to the individual models.

## 4.2 Feature Analysis

Since our features were modeled after empirically-supported relationships of mind wandering (see Section 3.1), we explored how our features related to the model's predictions of mind wandering. For each participant, we computed the mean of each feature as well as the proportion of predicted mind wandering (based on the combined model's predictions). As an additional step, the averages were z-score standardized across the two datasets to account for the differences in methods. Predicted mind wandering was then regressed on each of the six features included in the combined model,  $F(6,317) = 35.5, p < .001, R^2_{adjusted} = .395$ . The regression allowed us to examine the relationship between

each of the features and predicted mind wandering while controlling for the other features in the model. Table 6 presents a summary of the features used the combined model, as well as the standardized regression coefficient ( $\beta$ ) for each feature.

**Table 5. Confusion Matrices for Each Feature Set Separately**

<b>Reading Behavior</b>	<i>Pred. MW</i>	<i>Pred. Not MW</i>
<i>Actual MW</i>	.439 (hit)	.561 (miss)
<i>Actual Not MW</i>	.313 (false alarm)	.687 (correct rejection)
<b>Textual Features</b>	<i>Pred. MW</i>	<i>Pred. Not MW</i>
<i>Actual MW</i>	.554 (hit)	.446 (miss)
<i>Actual Not MW</i>	.424 (false alarm)	.576 (correct rejection)
<b>Context Features</b>	<i>Pred. MW</i>	<i>Pred. Not MW</i>
<i>Actual MW</i>	.557 (hit)	.443 (miss)
<i>Actual Not MW</i>	.432 (false alarm)	.568 (correct rejection)

Note. Pred. = Predicted; MW = Mind Wandering

*Reading Time* was negatively related to predicted mind wandering, indicating that mind wandering predictions were associated with faster reading times. The second reading behavior feature, *Decoupling*, was positively related to predicted mind wandering. Mind wandering was more likely to be predicted when decoupling scores were higher, since higher decoupling scores indicate a misalignment between reading times compared to text complexity.

*Number of Characters* and *Number of Words* were both positively related to predicted mind wandering, suggesting that more content in general is associated with greater predictions of mind wandering. This is also related to the idea that longer paragraphs may have demanded increased cognitive resources, which is theoretically related to episodes of mind wandering [24].

**Table 6. Standardized coefficients for regressing predicted mind wandering on features in the combined model ( $\beta$ )**

Features Included in Combined Model	Standardized Coefficient ( $\beta$ )
<b>Reading Behavior Features</b>	
Reading Time	<b>-.750</b>
Decoupling	<b>.493</b>
<b>Textual Features</b>	
Number of Characters	<b>.139</b>
Number Words	.099
Referential Cohesion	<b>-.139</b>
FKGL	<b>.239</b>

Notes. Bold = significant at  $p < .05$ ; FKGL = Flesch Kincaid Grade Level.

*Referential Cohesion* was also negatively related to predicted mind wandering. This relationship is theoretically plausible, as

breakdowns in *Referential Cohesion* are indicative of increased difficulty [20]. Indeed, difficulty has been found to be related to mind wandering during reading [14, 27].

None of the Context features were included in the combined model. This was an unexpected result, since time on task has previously been correlated to mind wandering [40] and the previous detectors of mind wandering have utilized context features [5, 6]. It is possible that one of the Context Features, *Current Difficulty*, may not have been useful in the combined model due, in part, to the fact that the textual features were essentially more sensitive measures of difficulty. For example, FKGL and Referential Cohesion may be more sensitive measures of *Current Difficulty*.

### 4.3 Predictive Validity

In order to establish predictive validity for the detector, we ascertained if *predicted* mind wandering relates to learning similar to actual (self-reported) mind wandering rates? Based on previous research, we expect a negative relationship between actual mind wandering and learning [11, 32, 39]. To address this question, posttest performance was first correlated with *actual* rates of mind wandering (i.e., responses to the thought probes). Participants' posttest performance was calculated as the proportion of correct responses for the surface- and inference-level questions separately. The variables were standardized across the two datasets to account for any differences in populations. Indeed, *actual* mind wandering was negatively related to both surface (Spearman's  $\rho = -.338, p < .001$ ) and inference level ( $\rho = -.288, p < .001$ ) comprehension on the posttest.

To establish the predictive validity of the detector, we ascertained if *predicted* mind wandering was related to posttest performance similar to actual mind wandering. *Predicted* mind wandering rates (from the combined detector) was negatively correlated with surface level ( $\rho = -.294, p < .001$ ) as well as inference level performance on the posttest ( $\rho = -.193, p = .008$ ). The negative correlations with both types of posttest performance gives us some confidence in our model's predictive validity, since predicted mind wandering shows similar relationships with learning as actual self-reported mind wandering. This finding is notable since the model predicted mind wandering correctly around 20% above chance ( $\kappa = .207$ ), yet *predicted* mind wandering related almost as well to posttest scores as *actual* rates of mind wandering.

## 5. GENERAL DISCUSSION

Mind wandering is a ubiquitous phenomenon that is negatively related to learning [11, 32, 39]. Mind wandering can have a detrimental impact on comprehension when pieces of information are not accurately integrated into a learner's mental model of the instructional texts. Over time, information missed during episodes of mind wandering can accumulate, leaving deficits in the learner's overall understanding of a text. The development of attention-aware systems may provide opportunities to restore learners' attention in real-time to facilitate learning. However, we must first be able to detect mind wandering in order to respond to its occurrence.

We attempted to address this issue by developing a participant-independent detector of mind wandering through analyzing log files and textual characteristics collected during an online reading task. Two diverse datasets were used to ensure further generalizability. The detector was able to accurately classify mind wandering 20% above chance ( $\kappa = .207$ ;  $AUC = .609$ ). Given that mind wandering is an elusive internal state of attention and we used completely sensor-free data, modest classification

accuracies are to be expected. Additionally, the classification accuracy found in this study (63%) is only slightly lower than those reported for previous detectors built using sensor-based approaches including eye gaze and physiology (See Table 1; [5, 6]).

Three types of features were used to build the mind wandering detector: (1) reading behaviors, (2) textual features, and (3) context features. An independent model was built for each subtype of features, which allowed us to better understand how the subtypes of feature perform independently. Each set of features was able to correctly classify mind wandering independently at levels above chance, though performance varied across models. None of these models outperformed the combined model, so we conclude that combining different types of features was optimal in the current detector. Thus, future research may consider using one or more of these subtypes of features, as they are relatively easy to extract from log files.

Many of the features were included based on previous psychological and educational research on mind wandering. The relationships between the features and predicted rates of mind wandering were revealing in a number of ways. For example, a negative relationship between Referential Cohesion and predicted mind wandering directly supports the situation model view of text comprehension [14, 35]. This view posits that reading involves the construction of a *situation model*, which is a constantly-updated mental representation of a text's meaning [18, 43]. Situation models are harder to construct during difficult texts due to inconsistencies or lack of cohesion. Poorly constructed situation models consume fewer attentional resources, leaving extra resources available for off-task thoughts. Therefore, this theory would predict a negative relationship between mind wandering Referential Cohesion, which is what we find.

Response times as well as reading time information have been utilized in previous detectors of off-task states like disengagement [4, 7, 8]. Thus, it is not surprising that both reading time behavior features were related to predicted mind wandering. A negative relationship with Reading Time indicates that shorter reading times were indicative of increased mind wandering predictions. It is also worth noting that Decoupling, which is derived from a theoretically-supported relationship between reading time and text complexity, was positively related to predicting mind wandering. Indeed, these relationships suggest features based on reading times may be used a behavioral indices of attention during reading.

Our detector also showed some evidence for predictive validity. Predicted mind wandering was negatively related to posttest performance, similar to actual mind wandering. Future work should explore other avenues of establishing validity using other online measures of engagement and comprehension. Similar to [15], another method of validation would be to trigger thought probes on the pages where mind wandering is predicted in real-time. We could then evaluate responses to the predicted episodes of mind wandering in order to determine how accurate the model performs in a real-time detection setting.

It is important to note that these models are not without limitations. First, these models were built in the context of an instructional reading task, which may not generalize to other learning environments. Second, although two independent datasets were used, our results cannot currently be generalized beyond the current sample. Third, although self-reports of mind wandering using a thought-probe method have been validated in previous studies [35, 36], they depend on participants accurate

and honest responses. Additionally, given the internal nature of mind wandering, external coders are not a viable option. Therefore, future work may consider using a different method of probing, where participants might self-monitor and report instances of mind wandering at any point during reading [31] (as opposed to only at times when thought-probes occur). Finally, there is no known research establishing a way to determine the onset of mind wandering in real-time [37]. Thus, while detectors to date are able to predict instances of self-reported mind wandering (which is inherently realized), no method has been established to indicate how long the episode lasts or when it began.

Future work may include attempts to improve these models using additional features. For example, additional sensor-free features, such as trait-based features like prior knowledge and interest might further improve prediction rates. In addition, combining features developed here with previous detectors of mind wandering may also improve prediction rates (e.g., eye gaze). It is possible that combining multiple channels of data may have an additive effect to improve prediction rates.

In summary, this paper provides some initial evidence for a sensor-free detector of mind wandering during online instructional reading. A sensor-free detector of mind wandering may open up new avenues for interventions and instructional designs in order to facilitate attention. Previous detectors for disengagement behaviors, such as gaming the system and Gaze Tutor, have been used in the design of interventions, such as reintroducing the content that is missed due to gaming [2] and providing engaging dialogue to redirect students' attention [12]. The detector presented in this paper is an initial step for interventions that can occur when the mind wanders away from the current task. We believe further development of these types of models is promising for creating an attention-aware system that can respond in real-time.

## ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF; DRL 1235958). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

## REFERENCES

[1] Arroyo, I. et al. 2007. Repairing disengagement with non-invasive interventions. *AIED* (2007), 195–202.

[2] Baker, R.S.J. et al. 2006. Adapting to when students game an intelligent tutoring system. *Intelligent Tutoring Systems* (2006), 392–401.

[3] Baker, R.S.J. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), 1059–1068.

[4] Beck, J.E. 2004. Using response times to model student disengagement. *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments* (2004), 13–20.

[5] Bixler, R. and D'Mello, S. 2014. Toward Fully Automated Person-Independent Detection of Mind Wandering. *User Modeling, Adaptation, and Personalization*. Springer. 37–48.

[6] Blanchard, N. et al. 2014. Automated Physiological-Based Detection of Mind Wandering during Learning. *Intelligent Tutoring Systems* (2014), 55–60.

[7] Cocea, M. and Weibelzahl, S. 2006. Can Log Files Analysis Estimate LearnersLevel of Motivation?. *LWA* (2006), 32–35.

[8] Cocea, M. and Weibelzahl, S. 2007. Cross-system validation of engagement prediction from log files. *Creating New Learning Experiences on a Global Scale*. Springer. 14–25.

[9] Cocea, M. and Weibelzahl, S. 2011. Disengagement Detection in Online Learning: Validation Studies and Perspectives. *IEEE Transactions on Learning Technologies*. 4, 2 (Apr. 2011), 114–124.

[10] Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*. 20, 1 (1960), 37–46.

[11] Dixon, P. and Bortolussi, M. 2013. Construction, integration, and mind wandering in reading. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*. 67, 1 (2013), 1.

[12] D'Mello, S. et al. 2012. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies*. 70, 5 (2012), 377–398.

[13] Drummond, J. and Litman, D. 2010. In the zone: Towards detecting student zoning out using supervised machine learning. *Intelligent Tutoring Systems* (2010), 306–308.

[14] Feng, S. et al. 2013. Mind wandering while reading easy and difficult texts. *Psychonomic bulletin & review*. (2013), 1–7.

[15] Franklin, M.S. et al. 2011. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin & Review*. 18, 5 (Oct. 2011), 992–997.

[16] Franklin, M.S. et al. 2013. Thinking one thing, saying another: The behavioral correlates of mind-wandering while reading aloud. *Psychonomic Bulletin & Review*. (Jun. 2013).

[17] Graesser, A.C. et al. 2004. Coh-Matrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*. 36, 2 (2004), 193–202.

[18] Graesser, A.C. et al. 2011. Coh-Matrix Providing Multilevel Analyses of Text Characteristics. *Educational Researcher*. 40, 5 (2011), 223–234.

[19] Graesser, A.C. et al. 2010. What is a good question? *Bringing reading research to life*. Guilford Press. 112–141.

[20] Graesser, A.C. and McNamara, D.S. 2011. Computational analyses of multilevel discourse comprehension. *Topics in Cognitive Science*. 3, 2 (2011), 371–398.

[21] Hanley, J.A. and McNeil, B.J. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*. 143, 1 (1982), 29–36.

[22] Klare, G.R. 1974. Assessing Readability. *Reading Research Quarterly*. 10, 1 (Jan. 1974), 62–102.

[23] Mason, W. and Suri, S. 2012. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*. 44, 1 (2012), 1–23.

[24] McVay, J.C. and Kane, M.J. 2010. Does mind wandering reflect executive function or executive failure? Comment on Smallwood and Schooler (2006) and Watkins (2008). (2010).

[25] Millis, K. et al. 2011. Operation ARIES!: A serious game for teaching scientific inquiry. *Serious games and edutainment applications*. (2011), 169–195.

[26] Mills, C. et al. 2014. To Quit or Not to Quit: Predicting Future Behavioral Disengagement from Reading Patterns. *Intelligent Tutoring Systems* (2014), 19–28.

- [27] Mills, C. et al. 2013. What Makes Learning Fun? Exploring the Influence of Choice and Difficulty on Mind Wandering and Engagement during Learning. *Artificial Intelligence in Education* (2013), 71–80.
- [28] Rand, D.G. 2012. The promise of Mechanical Turk: How online labor markets can help theorists run behavioral experiments. *Journal of theoretical biology*. 299, (2012), 172–179.
- [29] Reichle, E.D. et al. 2010. Eye Movements During Mindless Reading. *Psychological Science*. 21, 9 (Sep. 2010), 1300–1310.
- [30] Risko, E.F. et al. 2012. Everyday attention: variation in mind wandering and memory in a lecture. *Applied Cognitive Psychology*. 26, 2 (2012), 234–242.
- [31] Schooler, J.W. et al. 2011. Meta-awareness, perceptual decoupling and the wandering mind. *Trends in cognitive sciences*. 15, 7 (2011), 319–326.
- [32] Smallwood, J. et al. 2007. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin & Review*. 14, 2 (2007), 230–236.
- [33] Smallwood, J. 2011. Mind-wandering while reading: Attentional decoupling, mindless reading and the cascade model of inattention. *Language and Linguistics Compass*. 5, 2 (2011), 63–77.
- [34] Smallwood, J. et al. 2009. Shifting moods, wandering minds: negative moods lead the mind to wander. *Emotion*. 9, 2 (2009), 271.
- [35] Smallwood, J. et al. 2008. When attention matters: The curious incident of the wandering mind. *Memory & Cognition*. 36, 6 (2008), 1144–1150.
- [36] Smallwood, J. and Schooler, J.W. 2006. The restless mind. *Psychological bulletin*. 132, 6 (2006), 946.
- [37] Smallwood, J. and Schooler, J.W. 2015. The science of mind wandering: empirically navigating the stream of consciousness. *Annual review of psychology*. 66, (2015), 487–518.
- [38] Szpunar, K.K. et al. 2013. Interpolated memory tests reduce mind wandering and improve learning of online lectures. *Proceedings of the National Academy of Sciences*. 110, 16 (2013), 6313–6317.
- [39] Szpunar, K.K. et al. 2013. Mind wandering and education: from the classroom to online learning. *Frontiers in psychology*. 4, (2013).
- [40] Thomson, D.R. et al. 2014. On the link between mind wandering and task performance over time. *Consciousness and cognition*. 27, (2014), 14–26.
- [41] Vega, B. et al. 2013. Reading into the Text: Investigating the Influence of Text Complexity on Cognitive Engagement. *Proceedings of the 6th international conference on educational data mining* (2013), 296–299.
- [42] Wixon, M. et al. 2012. WTF? detecting students who are conducting inquiry without thinking fastidiously. *User Modeling, Adaptation, and Personalization*. Springer. 286–296.
- [43] Zwaan, R.A. and Radvansky, G.A. 1998. Situation models in language comprehension and memory. *Psychological bulletin*. 123, 2 (1998), 162.

# A Comparison of Video-based and Interaction-based Affect Detectors in Physics Playground

Shiming Kai<sup>1</sup>, Luc Paquette<sup>1</sup>, Ryan S. Baker<sup>1</sup>, Nigel Bosch<sup>2</sup>, Sidney D'Mello<sup>2</sup>, Jaclyn Ocumpaugh<sup>1</sup>, Valerie Shute<sup>3</sup>, Matthew Ventura<sup>3</sup>

<sup>1</sup>Teachers College Columbia University, 525 W 120<sup>th</sup> St. New York, NY 10027

<sup>2</sup>University of Notre Dame, 384 Fitzpatrick Hall, Notre Dame, IN 46556

<sup>3</sup>Florida State University, 3205G Stone Building, 1114 West Call Street, Tallahassee, FL 32306

{smk2184, paquette}@tc.columbia.edu, baker2@exchange.tc.columbia.edu, jo2424@tc.columbia.edu, {pbosch, sdmello}@nd.edu, {vshute, mventura}@fsu.edu

## ABSTRACT

Increased attention to the relationships between affect and learning has led to the development of machine-learned models that are able to identify students' affective states in computerized learning environments. Data for these affect detectors have been collected from multiple modalities including physical sensors, dialogue logs, and logs of students' interactions with the learning environment. While researchers have successfully developed detectors based on each of these sources, little work has been done to compare the performance of these detectors. In this paper, we address this issue by comparing interaction-based and video-based affect detectors for a physics game called Physics Playground. Specifically, we report on the development and detection accuracy of two suites of affect and behavioral detectors. The first suite of detectors applies facial expression recognition to video data collected with webcams, while the second focuses on students' interactions with the game as recorded in log-files. Ground-truth affect and behavior annotations for both face- and interaction-based detectors were obtained via live field observations during game-play. We first compare the performance of these detectors in predicting students' affective states and off-task behaviors, and then proceed to outline the strengths and weakness of each approach.

## Keywords

Video-based detectors, interaction-based detectors, affect, behavior, Physics Playground

## 1. INTRODUCTION

The development of models that can automatically detect student affect now constitutes a considerable body of research [12,31], particularly in computerized learning contexts [1,34,35], where researchers have successfully built affect-sensitive learning systems that aim to significantly enhance learning outcomes [4,21,30]. In general, researchers attempting to develop affect detectors have developed systems falling into two categories: interaction-based detectors [9] and physical sensor-based detectors [12]. Many successful efforts to detect student affect in intelligent tutoring systems have used visual, audio or physiological sensors, such as webcams, pressure sensitive seat or

back pads, and pressure-sensing keyboards and mice [3,28,37,41].

The development of sensor-based detectors has progressed significantly over the last decade, but one limitation to this research is that much of it has taken place in laboratory conditions, which may not generalize well to real-world settings [9]. While efforts are being made to address this issue [4], there are often serious obstacles to using sensors in regular classrooms. For example, sensor equipment may be bulky or otherwise obtrusive, distracting students from their primary tasks (learning); sensors may also be expensive and prone to malfunction, making large-scale implementation impractical, particularly for schools that are already financially strained. On the other hand, because physical sensors are external to specific learning systems, their use in affect detection creates the opportunity for them to be applied to entirely new learning systems, though this possibility has yet to be empirically tested.

Interaction-based detection [9] has also improved over the last decade. Unlike sensor-based detectors, which rely upon the physical reactions of the student, these detectors infer affective states from students' interactions with computerized learning systems [5,7,9,14,29,30]. The fact that interaction-based affect detectors rely on student interactions makes it possible for them to run in the background in real time at no extra cost to a school that is using the learning system. Their unobtrusive and cost-efficient nature also makes it feasible to apply interaction-based detectors at scale, leading to a growing field of research regarding discovery with models [8]. For example, interaction-based affect detection has been useful in predicting student long-term outcomes, including standardized exam scores [30] and college attendance [36]. Basing affect detection on student interactions with the system, however, give rise to issues with generalizing such detectors across populations [26] and learning systems. Because interaction-based detectors are highly dependent on the computation of features that captures the student's interactions with the specific learning platform, the type of features generated is contingent on the learning system itself, making it difficult to apply the same sets of features across different systems.

It has become clear that each modeling approach has its own utility; researchers have thus begun to speculate on effectiveness across the various approaches and the possible applications of multimodal detectors. However, the body of research that addresses this question is currently quite limited. Arroyo and colleagues [4] applied sensor-based detectors in a classroom setting, and compared performances between interaction-only detectors and detectors using both interaction and sensor data, in predicting student affect. They found that the inclusion of sensor data in the detectors improved performance and accuracy in

identifying student affect. However, a direct comparison between the two types of detectors was not made. Furthermore, the sample size tested was relatively small (26-30 instances depending on model), and the data was not cross-validated. Comparisons between types of detectors were made in D’Mello and Graesser’s study [18], which compared interaction, sensor and face-based detectors in an automated tutor. They found face-based detectors to perform better than interaction and posture-based detectors at predicting spontaneous affective states. However, the study was conducted in a controlled laboratory setting, and the facial features recorded were manually annotated.

In this paper, we build detectors of student affect in classroom settings, using both sensor-based and interaction-based approaches. For feasibility of scaling, we limit physical sensors to webcams. For feasibility of comparison, the two types of detectors are built in comparable fashions, using the same ground truth data obtained from field observations that were conducted during the study. We conduct this comparison in the context of 8<sup>th</sup> and 9<sup>th</sup> grade students playing an educational game, Physics Playground, in the Southeastern United States. Different approaches were used to build each suite of detectors in order to capitalize on the affordances of each modality. However, the methods and metrics to establish accuracy were held constant in order to render the comparison meaningful.

## 2. PHYSICS PLAYGROUND

Physics Playground (formerly, Newton’s Playground, see [39]) is a 2-dimensional physics game where students apply various Newtonian principles as they create and guide a ball to a red balloon placed on screen [38]. It offers an exploratory and open-ended game-like interface that allows students to move at their own pace. Thus, Physics Playground encourages conceptual learning of the relevant physics concepts through experimentation and exploration. All objects in the game obey the basic laws of physics, (i.e., gravity and Newton’s basic laws of motion).



Figure 1: Screenshot of Physics Playground

Students can choose to enter one of seven different playgrounds, and then play any of the 10 or so levels within that playground. Each level consists of various obstacles scattered around the space, as well as a balloon positioned at different locations within the space (see Figure 1). Students can nudge the ball left and right, but will need to create simple machines (called “agents of force and motion” in the game) on-screen in order to solve the problems presented in the playgrounds. There are four possible agents that may be created: ramps, pendulums, levers and springboards. Students can also create fixed points along a line drawing to create pivots for the agents they create. Students use the mouse to draw agents that come to life after being drawn, and use them to propel the ball to the red balloon. Students control the weight and

density of objects through their drawings, making an object denser, for example, by filling it with more lines.

Each level allows multiple solutions, encouraging students to experiment with various methods to achieve the goal and guide the ball towards the balloon. Trophies are awarded both for achieving the goal objective and for solutions deemed particularly elegant or creative, encouraging students to attempt each playground more than once. This unstructured game-like environment provides us with a rich setting in which to examine the patterns of students’ affect and behavior as they interact with the game platform.

## 3. DATA COLLECTION

Students in the 8<sup>th</sup> and 9<sup>th</sup> grade were selected due to the alignment of the curriculum in Physics Playground to the state standards at those grade levels. The student sample consisted of 137 students (57 male, 80 female) who were enrolled in a public school in the Southeastern U.S. Each group of about 20 students used Physics Playground during 55-minute class periods over the course of four days.

An online physics pretest (administered at the start of day 1) and posttest (administered at the end of day 4), measured student knowledge and skills related to Newtonian physics. In this paper, our focus is on data collected during days 2 and 3, during which time students were participating in two full sessions of game play.

The study was conducted in a computer-enabled classroom with 30 desktop computers. Inexpensive webcams (\$30 each) were affixed at the top of each computer monitor. At the beginning of each session, the webcam software displayed an interface that allowed students to position their faces in the center of the camera’s view by adjusting the camera angle up or down. This process was guided by on-screen instructions and verbal instructions from the experimenters, who were available to answer any additional questions and to troubleshoot any problems.

### 3.1 Field Observations

Students were observed by two BROMP-certified observers while using the Physics Playground software. The Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP 2.0) is a momentary time sampling system that has been used to study behavioral and affective indicators of student engagement in a number of learning environments [9]. BROMP coders observe each student individually, in a predetermined order. They record only the first predominant behavior and affect that the student displays, but they have up to 20 seconds to determine what that might be.

In this study, BROMP coding was done by the 6<sup>th</sup> author and the 4<sup>th</sup> author. The 6<sup>th</sup> author, a co-developer of BROMP, has been validated to achieve acceptable inter-rater reliability ( $\kappa \geq 0.60$ ) with over a dozen other BROMP-certified coders. The 4<sup>th</sup> author achieved sufficient inter-rater reliability ( $\kappa \geq 0.60$ ) with the 6<sup>th</sup> author on the first day of this study.

The coding process was implemented using the Human Affect Recording Tool (HART) application for Android devices [6], which enforces the protocol while facilitating data collection. The study used coding schema that had previously been used in several other studies of student engagement [e.g. 17], and included *boredom*, *confusion*, *engaged concentration*, and *frustration* (affective states) as well as *on task*, *on-task conversation*, and *off-task* (behavioral states). Consistent with previous BROMP research, “?” was recorded when a student could not be coded, when an observer was unable to identify the

student's behavior or affective state, or when the affect/behavior of the student was clearly a construct outside of the coding scheme (such as *anger*).

Modifications to the affective coding scheme were made on the third day of the study, with the addition of *delight* and *dejection*. *Delight* was defined as a state of strong positive affect, often indicated by broad smiling or a student bouncing in his/her chair. This affective state had been coded in previous studies (see [9]), and was used to construct detectors. *Dejection*, defined as a state of being saddened, distressed, or embarrassed by failure [9], is likely the affect that corresponds with the experience of *stuck* [11,20]. Because it had not been coded in previous research, and because it was still quite rare in Physics Playground, it was not modeled for this study.

### 3.2 Affect and Behavior Incidence

An initial number of 2,374 observations were made across all 137 students during the course of the study, culminating in 17.3 observations made per student across the second and third days of the study. Only affect observations on the second and third days were used in the construction of the detectors, since the first and last days mostly consisted of pretests and posttests. Other observations were dropped as a result of two students who switched computers halfway through data collection, resulting in each student being logged under the other student's ID for part of the study. The remaining 2,087 observations recorded during the second and third days were used in the construction of both detectors. An additional 214 were removed prior to the construction of the interaction-based detectors and 863 were removed prior to the construction of the video-based detectors. Because the criteria for these exclusions were methodologically based, further details are provided in the sections describing the construction of each detector.

Within the field observations, the most common affective state observed was *engaged concentration* with 1293 instances (62.0%), followed by *frustration* with 235 instances (11.3%). *Boredom and confusion* were far less frequent despite being observed across both second and third days of observation: 66 instances (3.2%) for *boredom* and 38 instances (1.8%) for *confusion*. *Delight* was only coded on the third day, and was also rare (45 instances), but it still comprised 2.2% of the total observations.

The frequency of off-task behavior observations was 4.0% (84 instances), which was unusually low compared to prior classroom research in the USA using the same method with other educational technologies [27,33]. On-task conversation was seen 18.6% of the time (388 instances).

## 4. INTERACTION-BASED DETECTORS

To create interaction affect detectors, BROMP affect observations were synchronized to the log files of student interactions with the software. Features were then generated and a 10-fold student-level cross validation process was applied for machine learning, using five classification algorithms.

### 4.1 Feature Engineering

The feature engineering process for this study was based largely on previous research on student engagement, learning, and persistence. The initial set of features comprised 76 gameplay attributes that potentially contain evidence for specific affective states and behavior. Some attributes included:

- The total number of springboard structures created in a level

- The total number of freeform objects drawn in a level
- The amount of time between start to end of a level
- The average number of gold and silver trophies obtained in a level
- The number of stacking events (gaming behavior) in a level

Features created may be grouped into two broad categories. Time-based features focus on the amount of time elapsed between specific student actions, such as starting and pausing a level, as well as the time it takes for a variety of events to occur within each playground level. Other features take into account the number of specific objects drawn or actions and events occurring during gameplay, given various conditions.

Missing values were present at certain points in the dataset when a particular interaction was not logged. For example, a feature specifying the amount of time between the student beginning a level and his/her first restart of the level, would contain a missing value if the student manages to complete a level without having to restart it. A variety of data imputation approaches were used in these situations to fill in the missing values so that we could retain the full sample size. We used single, average and zero imputation methods to fill in the missing data, and ran the new datasets through the machine learning process to identify the best data imputation strategy for each affect detector. Zero imputations were performed where the missing values were replaced by the value 0, while average data imputations took place when the average value for the particular feature was computed, and the missing values replaced by this average value. In single data imputation, we used RapidMiner to build an M5' model [32], a tree-based decision model, to predict the values for each feature, and applied the model to compute a prediction of the missing value. We also ran the original dataset without any imputation through any of the classification algorithms that allowed it.

Of the 2087 BROMP field observations that were collected, 214 instances were removed as most of these instances corresponded to times when the student was inactive. Additional instances were removed where the observer recorded a ?, the code used when BROMP observers cannot identify a specific affect or behavior or when students are not at their workstation. In total, 171 instances of affect and 63 instances of behavior were coded as ?. As a result, these instances did not contribute to the building of the respective affect and behavior detectors.

### 4.2 Machine Learning

Data collection was followed by a multi-step process to develop interaction-based detectors of each affect. A two-class approach was used for each affective state, where that affective state was discriminated from all others. For example, engaged concentration was discriminated from all frustrated, bored, delighted, and confused instances combined (referred to as "all other"). Behaviors were grouped into two classes: 1) off task, and 2) both on task behaviors and on task conversation related to the game.

#### 4.2.1 Resampling of Data

Because observations of several of the constructs included in this study were infrequent, (< 5.0% of the total number of observations), there were large class imbalances in our data distributions. To correct for this, we used the *cloning* method for resampling, generating copies of respective positive affect on the training data, in order to make class frequency more balanced for detector development.

#### 4.2.2 Feature Selection and Cross-Validation

Correlation-based filtering was used to remove features that had very low correlation with the predicted affect and behavior constructs (correlation coefficient > 0.04) from the initial feature set. Feature selection for each detector was then conducted using forward selection.

Detectors for each construct were built in the RapidMiner 5.3 data-mining software, using common classification algorithms that have been previously shown to be successful in building affect detectors: JRip, J48 decision trees, KStar, Naïve-Bayes, step and logistic regression. Models were validated using 10-fold student-level batch cross-validation. The performance metric of A' was computed on the original, non-resampled, datasets.

### 4.3 Selected Features

From the forward selection process, a combination of features was selected in each of the affect and behavior detectors that provide some insight into the type of student interactions that predict the particular affective state or behavior.

The features for *boredom* involve a student spending more time between actions on average. A bored student would also expend less effort to guide the ball object to move in the right direction, as indicated by fewer nudges made on the ball object to move it, and more ball objects being lost from the screen.

The features that predict *confusion* are characterized by a student spending more time before his/her first nudge to make the ball object move, and drawing fewer objects in a playground level. A student who is confused may not have known how to draw and move the ball object towards the balloon, thus spending a long time within a certain level and resulting in a lower number of levels attempted in total.

From the features selected, *delight* appears to ensue from some indicator of success, such as a student who is able to achieve a silver trophy earlier on during gameplay, and who completes more levels in total. We can also portray the student who experiences *delight* as someone who was able to achieve the objective without having to make multiple attempts to draw the relevant simple machines (such as springboards and pendulums).

The features for *engaged concentration* would describe a student who is able to complete a level in fewer attempts but erases the ball object more often during each attempt, indicating that the student was putting in more effort to refine his/her strategies within a single attempt at the level. *Engaged concentration* would also depict a student who has experienced success during gameplay and achieved a silver trophy in a shorter than average time, perhaps because of his/her focused efforts during each attempt.

**Table 1. Features in the final interaction-based detectors of each construct**

Affect/ Behavior	Selected features
<b>Boredom</b>	Time between actions within a level
	Total number of objects that were “lost” (i.e. Moved off the screen)
	Total number of nudges made on the ball object to move it
<b>Confusion</b>	Amount of time spent before the ball object was nudged to move

	Total number of levels attempted
	Total number of objects drawn within the level
<b>Delight</b>	Number of silver trophies achieved
	Consecutive number of pendulums and springboards created
	Total number of levels attempted
	Total number of levels completed successfully
<b>Engaged Concentration</b>	Total number of silver trophies achieved in under the average time
	Total number of level re-starts within a playground
	Total number of times a ball object was erased consecutively
<b>Frustration</b>	Total number of silver trophies achieved in under the average time
	Total number of level re-starts within a playground
	Total number of levels completed successfully
	Total number of levels attempted
<b>Off-task Behavior</b>	Time spent in between each student action
	Total number of pauses made within a level
	Total number of times a student quits a level without completing the objective and obtaining a trophy

Unlike *engaged concentration*, a student who experiences *frustration* failed to achieve the objective and achieved fewer silver trophies within the average time taken. Student *frustration*, as seen in the features, would also result in the student having to make more attempts at a level due to repeated failure, thus resulting in fewer levels attempted in total.

Lastly, behavior that is *off-task* involves a student who spends more time pausing the level or between actions as a whole. It is also apparent in a student who draws fewer objects and quits more levels without completing them, implying that he or she did not put in much effort to complete the playground levels.

## 5. VIDEO-BASED DETECTORS

The video-based detectors have been reported in a recent publication [10]. In the interest of completeness, the main approach is re-presented here. There are also small differences in the results reported here due to a different validation approach that was used to make meaningful comparisons with interaction-based detectors.

Video-based affect detectors were constructed using FACET (no longer available as standalone software), a commercialized version of the Computer Expression Recognition Toolbox (CERT) software [25]. FACET is a computer vision tool used to automatically detect Action Units (AUs), which are labels for specific facial muscle activations (e.g. lowered brow). AUs provide a small set of features for use in affect detection efforts. A large database of AU-labeled data can be used to train AU

detectors, which can then be applied to new data to generate AU labels.

## 5.1 Feature Engineering

FACET provides estimates of the likelihood estimates for the presence of nineteen AUs as well as head pose (orientation) and position information detected from video. Data from FACET was temporally aligned with affect observations in small windows. We tested five different window sizes (3, 6, 9, 12, and 20 seconds) for creation of features. Features were created by aggregating values obtained from FACET (AUs, orientation and position of the face) in a window of time leading up to each observation using maximum, median, and standard deviation. For example, with a six-second window we created three features from the AU4 channel (brow lowered) by taking the maximum, median, and standard deviation of AU4 likelihood within the six seconds leading up to an affect observation. In all there were 78 facial features.

We used features computed from gross body movement present in the videos as well. Body movement was calculated by measuring the proportion of pixels in each video frame that differed from a continuously updated estimate of the background image generated from the four previous frames. Previous work has shown that features derived using this technique correlate with relevant affective states including boredom, confusion, and frustration [17]. We created three body movement features using the maximum, median, and standard deviation of the proportion of different pixels within the window of time leading up to an observation, similar to the method used to create FACET features.

Of the initial 2087 instances available for us to train our video-based detectors on, about a quarter (25%) were discarded because FACET was not able to register the face and thus could not estimate the presence of AUs and computation of features. Poor lighting, extreme head pose or position, occlusions from hand-to-face gestures, and rapid movements can all cause face registration errors; these issues were not uncommon due to the game-like nature of the software and the active behaviors of the young students in this study. We also removed 9% of instances because the window of time leading up to the observation contained less than one second (13 frames) of data in which the face could be detected, culminating in 1224 instances where we had sufficient video data to train our affect models on.

## 5.2 Machine Learning

We also built separate detectors for each affective state similar to the interaction-based detectors. Building individual detectors for each state allows the parameters (e.g., window size, features used) to be optimized for that particular affective state.

### 5.2.1 Resampling of Data

Like the interaction-based detectors, there were large class imbalances in the affective and behavior distributions. Two sampling techniques, different from the one used in the building of interaction-based detectors, were used on the training data to compensate for this imbalance. These two techniques included downsampling (removal of random instances from the majority class) and synthetic oversampling (with SMOTE; [13]) to create equal class sizes. SMOTE creates synthetic training data by interpolating feature values between an instance and randomly chosen nearest neighbors. The distributions in the testing data were not changed, to preserve the validity of the results.

### 5.2.2 Feature Selection and Cross-Validation

We used tolerance analysis to eliminate features with high multicollinearity (variance inflation factor > 5) [2]) for video-based detectors. Feature selection was then used to obtain a more diagnostic set of features for classification. RELIEF-F [24] was run on the training data in order to rank features. A proportion of the highest ranked features were then used in the models (.1, .2, .3, .4, .5, and .75 proportions were tested). A detailed analysis or table of the features selected for the video-based detectors is not included because of the large number of features utilized by these detectors.

We then built classification models using 14 different classifiers including support vector machines, C4.5 trees, Bayesian classifiers, and others in the Waikato Environment for Knowledge Analysis (WEKA), a machine learning tool [23].

## 6. RESULTS

We evaluated the extent to which the detectors for each construct are able to identify their respective affect. Both detectors were evaluated using a 10-fold student-level batch cross-validation. In this process, students in the training dataset are randomly divided into ten groups of approximately equal size. A detector is built using data from all possible combinations of 9 out of the overall 10 groups, and finally tested on the last group. Cross-validation at this level increases the confidence that the affect and behavior

**Table 2. A' performance values for affect and behavior using video-based and interaction-based detectors**

Affect/Behavior Construct	Interaction-Based Detectors				Video-Based Detectors		
	Classifier	Data Imputation Scheme	A'	No. Instances	Classifier	A'	No. Instances
Boredom	Logistic regression	Zero	0.629	1732	Classification via Clustering	0.617	1305
Confusion	Step regression	Average	0.588	1732	Bayes Net	0.622	1293
Delight	Logistic regression	None	0.679	1732	Updateable Naïve Bayes	0.860	1003
Engaged Concentration	Naïve Bayes	Zero	0.586	1732	Bayes Net	0.658	1228
Frustration	Logistic regression	Average	0.559	1732	Bayes Net	0.632	1132
Off-Task behavior	Step regression	Zero	0.765	1829	Logistic Regression	0.780	1381

detectors will be more accurate for new students. To ensure comparability between the two sets of detectors, the cross-validation process was carried out with the same randomly selected groups of students.

Detector performance was assessed using  $A'$  values that were computed as the Wilcoxon statistic [22].  $A'$  is the probability that the given algorithm will correctly identify whether an observation is an example of a specific affective state.  $A'$  can be approximated by the Wilcoxon statistic and is equivalent to the area under the Receiver Operating Characteristic (ROC) curve in signal detection theory. A detector with a performance of  $A' = 0.5$  is performing at chance, while a model with a performance of  $A' = 1.0$  is performing with perfect accuracy.

Table 2 shows the performance of the two detector suites. Both interaction-based and video-based detectors' performance over all six affective and behavior constructs was better than chance ( $A' = 0.50$ ). On average, the interaction-based detectors yielded an  $A'$  of 0.634 while the video-based detectors had an average  $A'$  of 0.695. This difference can be mainly attributed to the detection of delight, which was much more successful for the video-based detectors. Accuracy of the two detector suites was much more comparable for the other constructs, though the video-based detectors showed some advantages for engaged concentration and frustration, and were higher for 5 of the 6 constructs.

The majority of the video-based detectors performed the best when using the Bayes Net classifier, except for *boredom*, *delight* and *off-task behavior*. In comparison, logistic and step regression composed the classifiers that produced the best performance for most of the interaction-based detectors, with the exception of *engaged concentration*.

## 7. DISCUSSION

Affect detection is becoming an important component in educational software, which aims to improve student outcomes by dynamically responding to student affect. Affect detectors have been successfully built and implemented via different modalities [3,16,41], and each have their own advantages and disadvantages when implemented in a noisy classroom environment. This study is an extension of previous research conducted on both video-based and interaction-based detectors. Having been mostly built in controlled laboratory settings [12], we now test the performance for video-based detectors within an uncontrolled computer-enabled classroom environment that is more representative of an authentic educational setting. Although interaction-based detectors have been built to some degree of success in whole classroom settings [5,7,29], we now test the performance of these affect detectors in an open-ended and exploratory educational game platform.

In this paper, we compared the performances of six video-based and interaction-based detectors on student affect and behavior in the game-based software. We will discuss the implications of these comparisons in this section, as well as future work.

### 7.1 Main Findings

The performances of both detectors in the six affects and off-task behavior appear to be at similar levels above chance for five of the constructs, with video-based detectors performing slightly better than interaction-based detectors on the whole, and with video-based detector showing a stronger advantage for delight. Several factors may have help to explain the relative performances.

Performance of video detectors could be influenced by the uncontrolled whole-classroom setting in which video data is collected, where there are higher chances of video data being absent or compromised due to unpredictable student movement. While there were initially 2,087 instances of affect and behavior observed and coded, a moderate proportion of facial data instances were dropped from the final dataset when building the models. There were 44 instances of affect observation that were dropped either because the video was corrupted or incomplete, or because no video was recorded at all. In addition, there were 520 instances where video was recorded, but facial data were not detected for some reason, perhaps because the student had left the workstation, or when the face could not be detected in the video. An additional 211 instances were removed even though facial data was detected, because the facial data recorded was present for less than 1 second, such that no features could be calculated.

For interaction-based detectors, the exploratory and open-ended user-interface [40] constitutes a unique challenge in creating accurate models for student affect and behavior. The open-ended interface included multiple goals and several possible solutions that students could come up with to successfully complete each level. During gameplay, there are also multiple factors that could contribute to a student's failure to complete a level, such as conceptual knowledge as well as implementation of appropriate objects. A student with accurate conceptual knowledge of simple machines and Newtonian physics may still fail the level because of problems implementing the actions needed to guide the ball to the target. On the other hand, a student with misconceptions about the relevant physics topics may nevertheless be able to complete the level successfully through systematic experimentation. The possible combinations of student actions that result in failure or success in a playground level would hence contribute to the lower accuracy of interaction-based detectors on identifying students' affect based on their interactions with the software.

Another issue with the Physics Playground software could be that there are fewer indicators of success per unit of time, as compared to other learning software that have been studied previously, such as the Cognitive Tutors [e.g. 5]. During gameplay, the system is able to recognize when combinations of objects the student draws forms an eligible agent. However, this indicator of success or failure is not apparent to the student until after he or she creates the ball object and applies a relevant force to trigger a simulation. Since students often spend at least several minutes building agents and ball objects, this results in coarser-grained indicators and evaluations of success and failure. This is in comparison to affect detectors created in previous studies for the Cognitive Tutor software, in which there was regular evaluation of each question attempted, thus resulting in more indicators of success over a given time period. The combination of open-endedness and lack of success indicators per unit of time consequently leads to greater difficulty translating the semantics of student-software interactions into accurate affect predictions.

When comparing between the two sets of detectors, physical detectors make direct use of students' facial features and bodily movements captured by webcams and constitute embodied representations of students' affective states. On the other hand, interaction detectors were built based on student actions within the software, which serves as an indirect proxy of the students' actual affective states. These detectors rely, therefore on the degree to which student interactions with the software are influenced (or not) by the affective states they experience. Perhaps not surprisingly, video-based detectors perform somewhat better

in predicting some affective states (e.g., delight, engaged concentration, and frustration). Although the video detectors are limited by missing data, interaction-based detectors can only detect something that causes students to change their behaviors within the software, which can be challenging given the issues arising from the open-ended game platform. Simply put, face-based affect detectors appear to provide more accurate affect estimates but in fewer situations, while interaction-based affect detectors provide less accurate estimates, but are applicable in more situations. The two approaches thus appear to be quite complementary.

## 7.2 Limitations

In comparing the performances between interaction and video-based detectors, there exist several limitations in ensuring an equivalent set of methods for a fair comparison to be made.

Although both types of detectors were built based on the same ground truth data, varying sets of limitations exist that are unique to each set of detectors. A smaller proportion of instances were retained to build video-based detectors due to missing video data, which may influence performance comparison. Interaction-based detectors, on the other hand, are relatively more sensitive to the type of educational platform it is built upon, as compared to video-based detectors. The type of learning platform thus affects the variety of features that are relevant and useful in building the affect and behavior detectors, which in turn impacts its performance relative to previous work.

For both detectors, the sample size available for some of the affective states was quite limited, which made it necessary to oversample the training data in order to compensate for the class imbalances. However, because each detector was built on different platforms, different methods were used in oversampling the datasets. The need to conduct data imputations was also unique to interaction-based detectors due to the nature of some of the computed features, and not required for video-based detectors. The difference in these methods may in turn affect performance comparison between the two types of detectors.

## 7.3 Concluding Remarks

Given the various advantages and limitations to each type of detector in accurately predicting student affect, it may be beneficial for affect detection strategies to include a combination of video-based and interaction-based detectors. While video-based detectors provide more direct measures of student affect, practical issues may lead to video data being absent or unusable in detecting affect, simply because there is no facial data available to detect affect in. These situations may be alleviated by the presence of interaction data that are recorded automatically during students' use of the software. On the other hand, video-based facial data would be able to provide support to interaction data and boost the accuracy in which affective states are detected among students. This form of late-fusion or decision-level fusion can also be complemented by early-fusion or feature-level fusion, where features from both modalities are combined prior to classification. Whether this leads to improved accuracy, as routinely documented in the literature on multimodal affect detection [15,16] awaits future work.

## 8. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF) (DRL 1235958) and the Bill & Melinda Gates Foundation. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not

necessarily reflect the views of the NSF or the Bill & Melinda Gates Foundation.

## 9. REFERENCES

- [1] Ai, H., Litman, D.J., Forbes-Riley, K., Rotaru, M., Tetreault, J., and Purandare, A. 2006. Using system and user performance features to improve emotion detection in spoken tutoring dialogs. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 797–800.
- [2] Allison, P.D. 1999. *Multiple regression: A primer*. Pine Forge Press.
- [3] AlZoubi, O., Calvo, R. a., and Stevens, R.H. 2009. Classification of EEG for affect recognition: An adaptive approach. *Lecture Notes in Computer Science 5866 LNAI*, 52–61.
- [4] Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., and Christopherson, R. 2009. Emotion sensors go to school. *Frontiers in Artificial Intelligence and Applications*, 17–24.
- [5] Baker, R.D., Gowda, S., and Wixon, M. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *Proceedings of the 5th International Conference on Educational Data Mining*, 126–133.
- [6] Baker, R.D., Gowda, S., Wixon, M., et al. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *Proceedings of the 5th International Conference on Educational Data Mining*, 126–133.
- [7] Baker, R.S., Ocumpaugh, J., Gowda, S.M., Kamarainen, A.M., and Metcalf, S.J. 2014. Extending Log-Based Affect Detection to a Multi-User Virtual Environment for Science. *22nd Conference on User Modeling, Adaptation and Personalization (UMAP 2014)*, 290–300.
- [8] Baker, R.S.J.D. and Yacef, K. 2009. The State of Educational Data Mining in 2009 : A Review and Future Visions. *Journal of Educational Data Mining 1*, 1, 3–16.
- [9] Baker, R.S.; Ocumpaugh, J. 2015. Interaction-Based Affect Detection in Educational Software. In R.A. Calvo, S.K. D’Mello, J. Gratch and A. Kappas, eds., *Handbook of Affective Computing*. Oxford University Press, Oxford, UK, 233–245.
- [10] Bosch, N., Mello, S.D., Baker, R., et al. Automatic Detection of Learning - Centered Affective States in the Wild. In *Proceedings of the 2015 International Conference on Intelligent User Interfaces (IUI 2015)*. New York, NY, USA: ACM.
- [11] Burleson, W. and Picard, R.W. 2004. Affective agents: Sustaining motivation to learn through failure and a state of stuck. *Proceedings of the Workshop on Social and Emotional Intelligence in Learning Environments in conjunction with the seventh International Conference on Intelligent Tutoring Systems (ITS)*.
- [12] Calvo, R.A.. and D’Mello, S.K. 2010. Affect Detection: An Interdisciplinary Review of Models, Methods, and their Application to Learning Environments. *IEEE Transactions on Affective Computing 1*, 1, 18–37.
- [13] Chawla, N. V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. 2011. SMOTE: synthetic minority over-sampling

- technique. *Journal of Artificial Intelligence Research* 16, , 321–357.
- [14] D’Mello, S., Jackson, T., Craig, S., et al. 2008. AutoTutor Detects and Responds to Learners Affective and Cognitive States. *Proceedings of the Workshop on Emotional and Cognitive issues in ITS in conjunction with the 9th International Conference on ITS*, 31–43.
- [15] D’Mello, S. and Kory, J. A Review and Meta-Analysis of Multimodal Affect Detection. *ACM Computing Surveys*, .
- [16] D’Mello, S. and Kory, J. 2012. Consistent but Modest: A Meta-Analysis on Unimodal and Multimodal Affect Detection Accuracies from 30 Studies. *ACM International Conference on Multimodal Interaction*, 31–38.
- [17] D’Mello, S. 2011. Dynamical emotions: bodily dynamics of affect during problem solving. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.
- [18] D’Mello, S.K. and Graesser, A. 2010. Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modelling and User-Adapted Interaction* 20, 2, 147–187.
- [19] D’Mello, S.K. 2013. A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology* 105, 4, 1082–1099.
- [20] D’Mello, S.K.; Graesser, A. 2012. Emotions During Learning with AutoTutor. In *Adaptive Technologies for Training and Education*. 169–187.
- [21] Dragon, T., Arroyo, I., Woolf, B.P., Bursleson, W., El Kaliouby, R., and Eydgahi, H. 2008. Viewing student affect and learning through classroom observation and physical sensors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5091 LNCS, 29–39.
- [22] Hanley, J.A. and Mcneil, B.J. 1982. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* 143, 29–36.
- [23] Holmes, G., Donkin, A., and Witten, I.H. 1994. WEKA: a machine learning workbench. *Proceedings of ANZIS ’94 - Australian New Zealand Intelligent Information Systems Conference*, 357–361.
- [24] Kononenko, I. 1994. Estimating attributes: Analysis and extensions of RELIEF. In F. Bergadano and L. De Raedt, eds., *Machine Learning: ECML-94*. Springer, Berlin Heidelberg, 171–182.
- [25] Littlewort, G., Whitehill, J., Wu, T., et al. 2011. The Computer Expression Recognition Toolbox (CERT). *International Conference on IEEE*, 298–305.
- [26] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., and Heffernan, C. 2014. Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology* 45, 3, 487–501.
- [27] Ocumpaugh, J., Baker, R.S.J., Gaudino, S., Labrum, M.J., and Dezdendorf, T. 2013. Field Observations of Engagement in Reasoning Mind. *Proceedings of the 16th International Conference on Artificial Intelligence and Education*, 624–627.
- [28] Pantic, M., Pantic, M., Rothkrantz, L.J.M., and Rothkrantz, L.J.M. 2003. Toward an Affect-Sensitive Multimodal Human Computer Interaction. *Proceedings of the IEEE* 91, 9, 1370–1390.
- [29] Paquette, L., Baker, R.S.J. d., Sao Pedro, M., et al. 2014. Sensor-Free Affect Detection for a Simulation-Based Science Inquiry Learning Environment. *Proceedings of the 12th International Conference on ITS 2014*, 1–10.
- [30] Pardos, Z. a., Baker, R.S.J. d., San Pedro, M.O.C.Z., Gowda, S.M., and Gowda, S.M. 2014. Affective states and state tests: Investigating how affect and engagement during the school year predict end of year learning outcomes. *Journal of Learning Analytics* 1, 1, 107–128.
- [31] Picard, R.W. 1997. *Affective Computing*. MIT Press, Cambridge, MA.
- [32] Quinlan, J.R. 1992. Learning with continuous classes. *Machine Learning* 92, 343–348.
- [33] Rodrigo, M., Baker, R., and Rossi, L. 2013. Student Off-Task Behavior in Computer-Based Learning in the Philippines: Comparison to Prior Research in the USA. *Teachers College Record* 115, 10, 1–27.
- [34] Rodrigo, M.M.T. and Baker, R.S.J. d. 2009. Coarse-grained detection of student frustration in an introductory programming course. *Proceedings of the fifth International Computing Education Research Workshop - ICER 2009*.
- [35] Sabourin, J., Mott, B., and Lester, J. 2011. Modeling Learner Affect with Theoretically Grounded Dynamic Bayesian Networks. *Proceedings of the Fourth International Conference on Affective Computing and Intelligent Interaction*, 286–295.
- [36] San Pedro, M.O.Z., Baker, R.S.J. d., Bowers, A.J., and Heffernan, N.T. 2013. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. *Proceedings of the 6th International Conference on Educational Data Mining*, 177–184.
- [37] Sebe, N., Cohen, I., Gevers, T., and Huang, T.S. 2005. Multimodal Approaches for Emotion Recognition: A Survey. *Proceedings of SPIE – The International Society for Optical Engineering*, 56–67.
- [38] Shute, V., Ventura, M., and Kim, Y.J. 2013. Assessment and Learning of Qualitative Physics in Newton ’ s Playground Newton ’ s Playground. *The Journal of Educational Research* 29, 579–582.
- [39] Shute, V. and Ventura, M. 2013. *Measuring and Supporting Learning in Games Stealth Assessment*. MIT Press, Cambridge, MA.
- [40] Shute, Valerie; Ventura, Matthew; Kim, Y.J. 2013. Assessment and Learning of Qualitative Physics in Newton ’ s Playground. *Journal of Educational Research* 106, 423–430.
- [41] Zeng, Z., Pantic, M., Roisman, G.I., and Huang, T.S. 2009. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 39–58.

# Exploring Dynamical Assessments of Affect, Behavior, and Cognition and Math State Test Achievement

Maria Ofelia Z. San Pedro<sup>1</sup>, Erica L. Snow<sup>2</sup>, Ryan S. Baker<sup>1</sup>, Danielle S. McNamara<sup>2</sup>,  
Neil T. Heffernan<sup>3</sup>

<sup>1</sup>Teachers College Columbia University, 525 W 120<sup>th</sup> St. New York, NY 10027

<sup>2</sup>Arizona State University, Learning Sciences Institute, 1000 S. Forest Mall, Tempe, AZ 85287

<sup>3</sup>Worcester Polytechnic Institute, 100 Institute Rd. Worcester, MA 01609

mzs2106@tc.columbia.edu, Erica.L.Snow@asu.edu, baker2@exchange.tc.columbia.edu,  
Danielle.McNamara@asu.edu, nth@wpi.edu

## ABSTRACT

There is increasing evidence that fine-grained aspects of student performance and interaction within educational software are predictive of long-term learning. Machine learning models have been used to provide assessments of affect, behavior, and cognition based on analyses of system log data, estimating the probability of a student's particular affective state, behavior, and knowledge (cognition). These measures have (in aggregate) successfully predicted outcomes such as performance on standardized exams. In this paper, we employ a different approach of relating interaction patterns to learning outcomes, using dynamical methods that assess patterns of fine-grained measures of affect, behavior, and knowledge as they occur across time. We use Hurst exponents and Entropy scores computed from assessments of affect, behavior, performance, and knowledge acquired from 1,376 middle school students who used a math tutoring system (ASSISTments), and analyze the relations of these dynamical measures to the students' end-of-year state test (MCAS) performance. Our results show that fine-grained changes in affect, behavior, and knowledge are significantly related to and predictive of their eventual MCAS performance, providing a new lens on the dynamic and nuanced nature of student interaction within online learning platforms and how it affects achievement.

## Keywords

Affect Detection, Knowledge Modeling, Educational Data Mining, Hurst, Entropy

## 1. INTRODUCTION

The increasing deployment of educational software in classrooms has provided new opportunities for studying a broad range of student modeling constructs. The ability of these systems to log student interaction in fine-grained detail has led to the development of automated detectors or models of student learning and engagement [1, 4, 5, 6, 10]. It has been demonstrated through *discovery with models* analyses [20] that detector assessments of engagement and learning can be used to predict long-term student outcomes such as performance in end-of-year standardized exams [24], college enrollment [31] and college major choice [33], even several years after the student engages in online learning. The fine-grained measures of learning and engagement at the action level are then aggregated at the student-level in forming a training dataset for the prediction of learning outcomes. However, these assessments often use simple aggregation methods such as student-level averages, whereas it is known that there are complex

patterns in how affect develops over time (e.g. [14]). Hence these simple methods of aggregation may miss fine-grained and nuanced patterns in affect or behaviors that manifest across time.

Indeed, research has also shown that students' learning behaviors are complex and dynamic in nature [19]. Recent work has begun to evaluate *interaction patterns* within learning tasks. This work has revealed that fine-grained pattern analysis can shed light upon various cognitive, behavioral, and learning outcomes [21, 22, 29, 30, 37, 38]. For example, Lee and colleagues [21], and Liu and colleagues [22] evaluated how 3-step sequences of confusion [21, 22] and frustration [22] correlate to learning outcomes. Rodrigo and colleagues [29] also found that 3-step sequences of affective states (boredom, engaged concentration, confusion, and delight) from fine-grained detectors correlated to differences in learning outcomes. Sabourin and colleagues [30] found that the impact of student behavior on learning outcomes depended in part on the affect that preceded the behavior. Results from these studies reveal that fluctuations in students' affect and behavior over time (assessed through automated detectors) play important roles in learning outcomes.

However, much of this work had the limitation of only considering changes over brief periods of time. In this paper, we address this limitation by employing dynamical methodologies to quantify nuanced patterns of student affect, behavior, and learning across time, specifically two academic school years. We utilize fine-grained measures of affect, behavior, and knowledge (cognition) from middle school students who used the ASSISTments systems, and compute dynamical measures (i.e., Hurst and Entropy) of these constructs for each student. These measures (see below for details) characterize the occurrence and type of behavior across time for the constructs of interest (affect, behavior, knowledge) for each student within the ASSISTments environment.

We use two types of dynamical analysis techniques, Entropy and Hurst exponents. Entropy is a statistical measure used to assess the amount of predictability present within a time series [34]. Previously, Entropy has been used in EDM analyses by Snow and colleagues [38], to quantify the amount of randomness in students' interaction patterns within a game-based interface. Using this methodology they found that students who acted in more controlled (and predictable) manners had significantly higher task performance compared to students who acted in more random (or unpredictable) fashions. Hurst exponents are similar to Entropy in that they categorize the amount of order present within a system; however, unlike Entropy, Hurst exponents act as long-term correlations that capture how each moment in a time series

relates to the others. Thus, Hurst provides an even finer-grained look at the emergence of patterns across long periods of time. Recently, Hurst exponents have been used to characterize students' learning behaviors within game-based environments. For instance, Snow and colleagues [36] used this technique to examine nuanced fluctuations in students' choice patterns across time. Using the Hurst exponent, Snow and colleagues again found that students who acted in more deterministic manners (i.e., controlled and planned) were more likely to demonstrate higher learning gains compared to students who acted in more random (or impetuous) manners.

In the current work, we evaluate the degree to which Entropy and Hurst exponent measures based on affect, behavior, and knowledge (cognition) predicts a longer-term outcome, students' end-of-year state exam performance. This research was conducted on a dataset of 1,376 students who used ASSISTments when they were in middle school during the school years of 2004-2005 to 2005-2006 and took the standardized end-of-year state exams. We investigate in particular, the following research questions:

- 1) How are fluctuations in patterns of students' affect, behavior, and knowledge related to their end-of-year state math achievement test scores?
- 2) Are dynamical measures of affect, behavior, and knowledge predictive of student performance outcomes (end-of-year test score, i.e., MCAS)?

## 2. METHODOLOGY

### 2.1 Data Source: The ASSISTments System

This study explores students' learning outcomes and their interaction patterns from their usage of the ASSISTments system [27], a web-based tutoring system for middle-school mathematics, provided to students for free by Worcester Polytechnic Institute (WPI). As of 2013, ASSISTments has been used by over 50,000 students a year as part of their regular mathematics classes. ASSISTments *assesses* a student's knowledge while *assisting* them in learning, providing teachers with formative assessment of students as they progress in their acquisition of specific knowledge components.

Within the system, each problem maps to one or more cognitive skills. When students who are working on an ASSISTments problem answer correctly, they proceed to the next problem. When they answer incorrectly (Figure 1), the system scaffolds instruction by dividing the problem into component parts, stepping students through each before returning them to the original problem (as in Figure 2). Once the correct answer to the original question is provided, the student is prompted to go to the next question. Teachers use ASSISTments in designing problem sets completed by students either during class time or as homework assignments. ASSISTments provides data on student performance that is used by teachers to track misconceptions and discuss them in class.

Problem ID: PRAJUFQ [Comment on this problem](#)

The area of a square is 49 square inches.  
What is the length of one side of the square?

Select one:

- A. 49 inches
- B. 25 inches
- C. 12 inches
- D. 7 inches

✖ Sorry, try again: "C. 12 inches" is not correct

Submit Answer

Original problem

Problem ID: PRAJUFQ - 435860 [Comment on this problem](#)

Let's make sure you understand the question. How do you find area of a square?

Select one:

- Multiply 1/2 by base by height.
- Multiply length by width by height.
- Add up the lengths of the 4 sides of the square.
- Multiply the length of the square by the width.

Submit Answer [Show answer](#)

First scaffolding question

Figure 1. Example of an ASSISTments problem.

Problem ID: PRAJUFQ - 435860 [Comment on this problem](#)

Let's make sure you understand the question. How do you find area of a square?

Select one:

- Multiply 1/2 by base by height.
- Multiply length by width by height.
- Add up the lengths of the 4 sides of the square.
- Multiply the length of the square by the width.

✔ Correct!

Submit Answer [Next step](#) [Show answer](#)

First scaffolding question

Problem ID: PRAJUFQ - 435861 [Comment on this problem](#)

Good, the area of a square is length times width.  
You are given the area of the square and now you need to find the length of one side by solving the following equation:  
 $49 = \text{length} * \text{width}$   
What is the length of one side of the square?

There are 2 unknowns in the equation: length and width. However, since the shape is a square, we know that the length and width are equal. That means there is only one unknown. Let's call it x:  
 $49 = x * x$   
What is x?

What is the square root of 49? In other words, what number multiplied by itself will give you 49?

$7 * 7 = 49$ , so the length of one side of the square is 7 inches. Type in 7.

Type your answer below:  
7

✔ Correct!

Submit Answer [Next Problem](#)

Second scaffolding question

Multi-level hints (with bottom-out hint that gives answer)

Figure 2. Example of Scaffolding and Hints in an ASSISTments Problem.

## 2.2 Data

### 2.2.1 State Exam Scores

Students who used ASSISTments when they were in middle school also took the MCAS (Massachusetts Comprehensive Assessment System) state standardized test near the end of their school years. The test is composed of English Language Arts, Mathematics and Science, and Technology subjects. This study analyzes usage of a tutoring system in mathematics; consequently, we examined the relationship of performance to the MCAS test scores for the math portion. Raw scores for the math portion range from 0 to 54 and are later scaled by the state after all tests have been scored. The scaled scores can be categorized into four groups: Failing, Needs Improvement, Proficient, and Advanced. Students in Massachusetts are required to score above failing to be able to graduate from high school; if students score in the Advanced group, they automatically earn a scholarship to a state college.

### 2.2.2 ASSISTments Data

Interaction log files from ASSISTments were obtained for 1,376 students who used the system when they were in middle school ranging from school years 2004-2005 to 2005-2006 (these school years were used due to the availability of the state exam data for these particular cohorts). These students, diverse in terms of both ethnicity and socio-economic status, were drawn from middle schools in an urban district in New England who used the ASSISTments system systematically during the school years. The 1,376 students generated a total of 830,167 actions within the system (an action may be answering a question, or requesting help), across around 3,700 original and scaffolding problems from ASSISTments, with an average of approximately 220 ASSISTments problems per student. Affect, behavior, and knowledge models were applied to this dataset to evaluate interaction patterns.

## 2.3 Computing Interaction Features

The interaction features used to compute dynamical assessments were generated using automated detectors of student engagement and learning previously developed and validated for ASSISTments. These included existing models of educationally-relevant affective states (boredom, engaged concentration, confusion, frustration), disengaged behaviors (off-task behavior and gaming the system), and student knowledge. Each of the detectors was applied to every action in the existing data set, in the same fashion as in previous publications [24]. We also included in our feature set of interactions, information on student correctness over time within ASSISTments.

### 2.3.1 Affect and Disengaged Behaviors

To obtain assessments of affect and disengaged behaviors, we leveraged existing detectors of student affect and behavior within the ASSISTments system [24]. Detectors of four affective states were utilized: boredom, engaged concentration, confusion, and frustration. Detectors of two disengaged behaviors are utilized: off-task behavior and gaming the system. Because our sample of students came from urban middle schools, their respective data were labeled using models optimized for students in urban schools [23, 24].

The affect and behavior detectors were developed in a two-stage process: first, student affect labels were acquired from field observations conducted using the BROMP protocol and HART Android app (reported in [24]), and then those labels were synchronized with the log files generated by ASSISTments at the

same time. This process resulted in automated detectors that can be applied to log files at scale, specifically the data set used in this project (interaction log files for the 1,376 students). The detectors were constructed using only log data from student actions within the software occurring at the same time as or before the observations. The models performed as well as or better than other published models of sensor-free affect detection in educational software [3, 11, 13, 30]. They were then applied to the data set used in this paper to produce confidence values for each construct over time, which were then used to create dynamical assessments of affect and behavior.

### 2.3.2 Student Knowledge

Corbett and Anderson's [12] Bayesian Knowledge Tracing (BKT) model, a knowledge-estimation model that has been used in a considerable number of online learning systems, was applied to the data for this study. Models were fit by employing brute-force grid search (see [2]). BKT infers students' latent knowledge from their performance on problems that exercise the same set of skills. Each time a student attempts a problem or problem step for the first time, BKT recalculates the estimates of that student's knowledge for the skill (or knowledge component) involved in that problem. Estimations for each skill are made along four parameters: (1)  $L_0$ , the initial probability that the student knows the skill, (2)  $T$ , the probability of learning the skill at each opportunity to use that skill, (3)  $G$ , the probability that the student will give the correct answer despite not knowing the skill, and (4)  $S$ , the probability that the student will give an incorrect answer despite knowing the skill. The estimates obtained via BKT were calculated based on the student's first response to each problem, and were applied to each of the student's subsequent attempts on that problem.

We were able to distill interaction features –affect, behavior and knowledge using these models, as well as correctness – for each student action within the ASSISTments system. Affect and behavior features were initially computed at a 20-second grain-size and then applied to all relevant actions. These action-level features values are then used to compute student-level dynamical measures of Hurst and Entropy scores.

## 2.4 Dynamical Assessments of Student Interaction Features

Variations in students' interaction features (affect, behavior, knowledge, correctness) were assessed using two dynamical methodologies: Entropy analyses and Hurst exponents. These dynamic techniques are used to quantify (in standardized values) variations in students' interaction features and examine how these variations impacted students' year-end standardized test scores (i.e., MCAS). A description and explanation of Entropy analyses and Hurst exponents are described below.

### 2.4.1 Entropy

Entropy analyses were conducted to quantify the degree to which fluctuations in students' affective states were ordered (i.e., predictable) or disordered (i.e., unpredictable). Entropy analysis is a statistical measure that quantifies the overall tendency (i.e., amount of predictability) of a time series [34]. Entropy has been used across a variety of domains to measure random and ordered processes [15, 17, 34, 35, 38]. In the current study, Entropy is used to gain a deeper understanding of how changes in students' affective states across time may reflect ordered and disordered processes. To calculate Entropy, we applied the affect, behavior, and knowledge series produced from the models discussed above,

to data from school years 2004-2005 and 2005-2006. Entropy was then calculated using the following (standard) formula:

$$H(x) = - \sum_{i=0}^N P(x_i) (\log_e P(x_i)) \quad (1)$$

Within the Entropy equation,  $P(x_i)$  represents the probability of a given affective state. For instance, the Entropy for student X is the additive inverse of the sum of products calculated by multiplying the probability of each affect state by the natural log of the probability of that state. This formula affords the ability to capture the degree to which fluctuations in students' affect, behavior, knowledge, and correctness are ordered or disordered.

#### 2.4.2 Hurst

While Entropy provides an overall quantification of a time series, it does not calculate how each moment in the time series may be related to the next. Thus, a more fine-grained analysis is needed to examine how fluctuations in students' affect, behavior, knowledge, and correctness manifest and change across time. To classify the tendency of students' affective states, Hurst exponents were calculated using Detrended Fluctuation Analysis (DFA) [26]. To calculate the Hurst exponent, the DFA integrates the normalized time series and then divides the series into equal intervals of length,  $n$ . Each interval is then fit with a least squares line and the integrated time series is *detrended* by subtracting the local predicted values (i.e., least square lines for each interval) from the integrated time series. The procedure is repeated for intervals of different lengths, increasing exponentially by the power of 2. Finally, each interval size is assigned a characteristic fluctuation,  $F(n)$ , that is calculated as the root mean square deviation of the integrated time series from local least squares lines.  $\log_2 F(n)$  is then regressed onto  $\log_2(n)$ ; which produces the slope of the regression line or Hurst exponent,  $H$ . Hurst exponents range from 0 to 1 and can be interpreted as follows:  $0.5 < H \leq 1$  indicates persistent (controlled) behavior,  $H = 0.5$  signifies random (independent) behavior, and  $0 \leq H < 0.5$  denotes anti-persistent (reversion to the mean) behavior.

### 2.5 Predictive Modeling of State Test Scores

Prior work has shown that student usage choices while receiving tutoring in ASSISTments can predict as much of the variance in students' end-of-year state test scores as student performance can on items designed to assess test-related knowledge [16, 28]. It has also been shown that machine-learned and fine-grained assessments of affect and behavior can improve predictions of test score performance [24]. We extend this further and explore the value of also understanding the role of the degree of order/disorder of interaction (through occurrences of affect, behavior, knowledge, and correctness) in predicting student learning outcomes as reflected by students' end-of-year standardized examination scores.

After obtaining the aggregate student-level Hurst and Entropy scores for each student's patterns of affect, behavior, knowledge, and correctness, we examined how the degree of variation in the students' interaction patterns within ASSISTments was related to their MCAS math performance. We further examined these relations by conducting linear regression analyses on the students' MCAS math performance. We fit a cross-validated (6-fold, student-level) machine-learned model using linear regression with M5' feature selection to examine how students' dynamical assessments of interaction were predictive of their MCAS math scores. We generated reduced linear regression models that used three feature sets: (1) Hurst scores of interaction only, (2) Entropy

scores of interaction only, and (3) both Hurst and Entropy scores of interaction. We then compared their cross-validated model performances and evaluated the features in the model with best performance values.

## 3. RESULTS

### 3.1 Hurst, Entropy, and State Test Scores

We first explore the relations between the MCAS scores for math and students' interaction patterns (i.e., their Hurst and Entropy scores) by examining the graphs of student proficiency (from MCAS performance) and the corresponding trends in Hurst and Entropy values. We grouped the students according to their scaled score groupings of Failing, Needs Improvement, Proficient, and Advanced, then computed for the average values of their Hurst and Entropy scores for affect, behavior, knowledge, and correctness in ASSISTments.

The graph of test proficiency and entropy measures (Figure 3) shows that low-achieving and high-achieving students experience fluctuations in affect, behavior, knowledge, and correctness while using ASSISTments in varying degrees. Students who have higher MCAS scores (i.e., *Advanced*) exhibited less fluctuation (lower entropy score) in their frustration ( $F(3,1372) = 56.009, p < 0.001$ , adjusted  $\alpha = 0.013$ ), engaged concentration ( $F(3,1372) = 27.334, p < 0.001$ , adjusted  $\alpha = 0.023$ ), off-task behavior ( $\chi^2(3) = 64.089, p < 0.001$ , adjusted  $\alpha = 0.030$ ), and gaming the system ( $\chi^2(3) = 238.350, p < 0.001$ , adjusted  $\alpha = 0.007$ ), but more fluctuation (higher entropy score) for boredom ( $\chi^2(3) = 26.999, p < 0.001$ , adjusted  $\alpha = 0.040$ ), confusion ( $\chi^2(3) = 29.759, p < 0.001$ , adjusted  $\alpha = 0.033$ ), correctness ( $\chi^2(3) = 185.310, p < 0.001$ , adjusted  $\alpha = 0.010$ ), and knowledge ( $\chi^2(3) = 639.111, p < 0.001$ , adjusted  $\alpha = 0.003$ ). [We used one-way ANOVA (F-test) for features with equal group variances, and Kruskal-Wallis test ( $\chi^2$  test) for features with unequal group variances.]

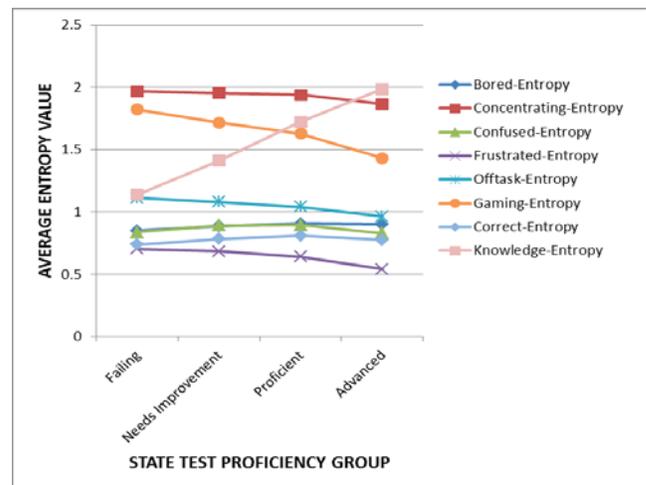


Figure 3. Entropy Scores by MCAS Test Score Category.

These trends suggest that students who performed better in MCAS showed overall consistency across time in exhibiting engaged concentration, frustration, off-task behaviors, and gaming the system, and an overall higher degree of variability across time in exhibiting boredom, confusion, correctness, and knowledge. It is possible that highly successful students may be more aware of their engaged concentration, frustration, off-task, and gaming behaviors within the system, compared to their awareness of the other constructs. Indeed, students who have achieved a higher level of proficiency or mastery of the material may also be more

efficient at controlling and maintaining the negative learning behaviors, and be more engaged. Interestingly, successful students show more variability, indicative of less control, in their boredom, confusion, correctness, and knowledge, possibly due to the nature of the learning task. These successful students may find some problems within ASSISTments too easy or too difficult with respect to their skills, causing them to experience varying degrees of boredom and confusion across time. In other words, the environment may be a major driver of the variability in these constructs. Another possibility comes from results in [24], where more successful students were more likely to be bored or confused when answering original problems, and less bored and confused when answering scaffolding problems. These successful students may also be overconfident in answering problems and become careless [32], exhibiting varying degrees of correctness and knowledge across time.

These relationships suggest that students with higher year-end exam scores were able to control their engagement by becoming less off-task and more consistent in overcoming their frustration and avoiding gaming the system, and be more engaged during their time in ASSISTments. However, a relevant area of future work may be to investigate whether the fluctuations across time for our interaction features are more a function of students' individual differences (e.g. proficiency) and their ability to control their learning behaviors [38], or a function of the learning task (e.g. type of problem, difficulty, etc.) and the learning behaviors it elicits from the students.

While Figure 3 shows the intensity or strength of fluctuations of our constructs across the entirety of student usage of ASSISTments, it does not demonstrate behavior of these fluctuations in fine-grained moments (i.e., persistence or anti-persistence of these constructs; how rapid were the fluctuations?). This is where looking at the Hurst measures of our constructs comes in useful. Figure 4 shows the graph of test proficiency and Hurst measures, where students who have higher MCAS scores achieved lower Hurst scores for engaged concentration ( $\chi^2(3) = 134.719, p < 0.001$ , adjusted  $\alpha = 0.017$ ), frustration ( $F(3,1372) = 27.543, p < 0.001$ , adjusted  $\alpha = 0.020$ ), off-task behavior ( $\chi^2(3) = 70.736, p < 0.001$ , adjusted  $\alpha = 0.027$ ), and confusion ( $F(3,1372) = 9.969, p < 0.001$ , adjusted  $\alpha = 0.037$ ), while higher Hurst scores for knowledge ( $\chi^2(3) = 23.935, p < 0.001$ , adjusted  $\alpha = 0.043$ ) and gaming the system ( $\chi^2(3) = 12.425, p = 0.006$ , adjusted  $\alpha = 0.047$ ).

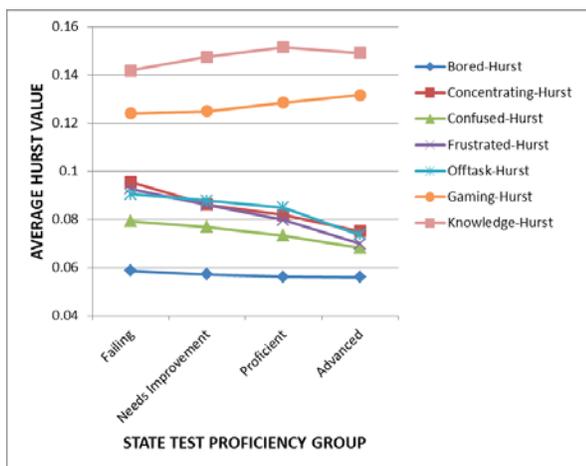


Figure 4. Hurst Scores by MCAS Test Score Category.

This trend in Hurst scores suggests that students who scored high on the MCAS had greater tendency to vary their behaviors, indicative of their actively adapting their learning behaviors. They instead showed regulation strategies in their ability to bounce back from frustration, resolve their confusion, and to re-engage after going off-task. Interestingly, more successful students show more mean reversion in engaged concentration than less successful students. Thus, more successful students were more variable in their engaged concentration (higher probability of concentration at one moment, lower probability of concentration on the next). Along with the Hurst scores for confusion, off-task and frustration, this Hurst trend for engaged concentration may indicate that students who began to feel confused or frustrated switched their focus and went off-task. Conversely, the trend for more successful students showed less variability in their display of knowledge and gaming the system behavior, which would suggest their ability to maintain their high level of knowledge and to not game the system. An understanding of the differences of rate of momentary fluctuations provides a lens on how students who vary in proficiency are able to effectively manage and adjust their affect, behavior, and knowledge within a learning task. It suggests that in the case of ASSISTments, it may be beneficial to teach less successful students strategies for quickly bouncing back from being off-task or ways to resolve their confusion and frustration.

We examine the significance of these differences in trends further by looking at the Pearson correlations between MCAS test scores and student Hurst and Entropy scores for affect, behavior, knowledge, and correctness (Table 1). We also utilize the Benjamini and Hochberg false discovery rate post-hoc correction to adjust the required alpha for significance and to reduce the occurrence of false positives, controlling for inflation of Type 1 error [8].

Table 1. Correlations with MCAS State Test Scores (\*\* - significant,  $p < 0.01$ ; \* - significant,  $p < 0.05$ )

Hurst and Entropy Features	r	p-value	Adjusted $\alpha$
Knowledge-Entropy	.705**	<0.001	0.003
Gaming-Entropy	-.441**	<0.001	0.007
Concentrating- Hurst	-.324**	<0.001	0.010
Frustration-Entropy	-.314**	<0.001	0.013
Correctness-Entropy	.275**	<0.001	0.017
Frustration- Hurst	-.252**	<0.001	0.020
Off-task-Entropy	-.211**	<0.001	0.023
Concentrating-Entropy	-.206**	<0.001	0.027
Off-task- Hurst	-.183**	<0.001	0.030
Confusion- Hurst	-.160**	<0.001	0.033
Bored-Entropy	.139**	<0.001	0.037
Bored-Hurst	-.100**	<0.001	0.040
Knowledge- Hurst	.076**	0.005	0.043
Confusion-Entropy	.076**	0.005	0.047
Gaming- Hurst	.059*	0.029	0.050
Correctness-Hurst	N/A	N/A	N/A

Table 1 shows that there are statistically significant, and reasonably strong relations between MCAS performance and Entropy measures of boredom, engaged concentration, confusion, frustration, off-task, gaming behavior, knowledge and correctness, and Hurst measures of boredom, engaged concentration, confusion, frustration, off-task, gaming behavior and knowledge. Note that for correctness only an Entropy score was calculated as it was a dichotomous measure of a student's answer (1 – correct, 0 – incorrect), and Hurst was not calculated for correctness as Hurst becomes less accurate when the inputs in the time series are discrete rather than continuous (which our other features are).

### 3.2 Prediction of State Test Scores

To examine the relations of these dynamical measures to MCAS performance, we conducted regression analyses to evaluate the predictive power of these measures (Table 2).

**Table 2. State Test Score Model Performance Values Using Different Feature Sets (feature count is after feature selection)**

Feature Set	R	R <sup>2</sup>	RMSE	Number of Features
Hurst Features Only	0.400	0.160	11.251	5
Entropy Features Only	0.762	0.581	7.941	6
Both Hurst and Entropy Features	0.768	0.590	7.862	9

Combined, Hurst and Entropy assessments of affect, behavior, knowledge and correctness within ASSISTments are predictive of long-term performance (end-of-year state test score, MCAS) with reasonably high model performance. This finding shows that when our automated detectors of affect, behavior, and knowledge are applied at scale, the patterns generated are significantly related to learning outcomes. The specific patterns and contexts in which these interactions occur, however, remain to be further analyzed - for example using methods such as sequential pattern mining or recurrence analysis. Moreover, it is also worth noting that despite the interesting findings discussed above, the model created from dynamical assessments of machine-learned measures of interaction is not much better than a model created from just averaging our interaction features per student (for our sample, this model had a cross-validated  $R = 0.764$ ) [24]. This suggests that averaging remains a good tool for predicting standardized exam scores, though it does not shed as much light on the phenomena of interest compared to the approach discussed here.

Optimized for predictor significance and model performance, our final model (Table 3) consists of either Hurst or Entropy scores (or both) of boredom, engaged concentration, confusion, frustration, gaming the system, knowledge, and correctness being predictive of MCAS performance.

Our final model leverages the relationships between MCAS and Hurst and entropy measures previously found. Stronger fluctuations across time for knowledge and correctness (positive coefficient for Entropy), and less persistence or quicker reversions in knowledge and engaged concentration (negative coefficient for Hurst), are associated with higher test scores for students. Furthermore, weaker fluctuations across time for boredom, confusion, gaming the system, and frustration (negative coefficient for Entropy), and more persistence or slow fluctuations for gaming the system (positive coefficient for Hurst), are associated with higher test scores for students. These relationships suggest that students with higher year-end exam scores were able

to control their engagement by resolving their confusion, bouncing back from being bored, overcoming their frustration, and to show active learning, and be more consistent in not gaming the system during their time in ASSISTments.

**Table 3. Final Model of Hurst and Entropy Scores Predicting State Test Scores**

Predictors	B	Std. Error	t	Sig
(Constant)	28.821	3.258	8.845	<0.001
Correctness-Entropy	39.566	4.672	8.469	<0.001
Concentrating-Hurst	-34.185	10.738	-3.183	0.001
Gaming-Hurst	22.952	6.853	3.349	0.001
Knowledge-Hurst	-22.935	4.579	-5.009	<0.001
Bored-Entropy	-21.318	2.773	-7.687	<0.001
Frustration-Entropy	-17.874	1.892	-9.447	<0.001
Knowledge-Entropy	17.463	0.723	24.169	<0.001
Gaming-Entropy	-9.371	1.126	-8.320	<0.001
Confusion-Entropy	-6.157	1.803	-3.416	0.001

## 4. DISCUSSION AND CONCLUSION

In this paper, we utilized dynamical methodologies to investigate how nuanced patterns of affect, behavior, knowledge, and correctness were related to and predictive of students' end-of-year exam scores. Fine-grained models of student affect (boredom, engaged concentration, confusion, frustration) behavior (off-task behavior, gaming the system), and knowledge were applied to data from 1,376 students who used an educational software in mathematics over the course of a year during their middle school to generate interaction features. We then utilized dynamical measures of Hurst exponents and Entropy analysis to quantify the degree of randomness (or non-randomness) present within patterns of these interaction patterns.

Our results show that these dynamical assessments of students' interactions throughout the year (affect, behavior, knowledge, and correctness) are significantly associated with their end-of-year performance in a state test. Entropy scores of students for all of our interaction features showed significant differences between students in varied test proficiencies (as measured by the year-end exam). Across time, the more control a student demonstrated in frustration, engaged concentration, off-task behaviors, and gaming the system behaviors, as well as more flexibility in boredom, confusion, knowledge and correctness, the higher the student scored on the year-end exam. Students' Hurst scores also showed significant relations with the learning outcome, where students with more occurrences of fluctuations for engaged concentration, confusion, frustration, and off-task behaviors, and more persistence for knowledge and gaming the system were likely to perform better. These relations were supported by these dynamical assessments being predictive of performance in the end-of-year state test.

It is notable that most Hurst exponent values fell well below 0.5, indicating that overall, fine-grained machine-learned estimates of affect, behavior, knowledge in the system interaction of the 1,376 students are not random, and according to students' state or the learning task within the system, students show signs of switching between various degrees of affect, behavior, and knowledge over

time. In the future, it may be useful to examine sequential patterns of each interaction feature, looking also at the context and circumstances in the usage of the system that lead to students having increasing or decreasing occurrences (as well as points of inflection) in affect, behavior, and knowledge. The Hurst and Entropy may be able to be used in real-time to capture these affective changes and then provide feedback to a user model (or teacher) about the student. Less successful students may be made aware of their learning behaviors so they may more effectively regulate them, in particular for frustration, confusion, off-task-behavior, and gaming the system. They may also be taught strategies to more quickly bounce back from being off-task or even resolve their frustration and confusion.

Overall, these exploratory findings obtained when we dynamically assess the measures of interaction take a step further in evaluating how fine-grained machine-learned assessments of affect, behavior, and knowledge relate to learning outcomes. Looking at patterns using a combination of machine-learning techniques provides an avenue for observing the degree to which students regulate their actions in a learning task. Self-regulation research shows that when students are motivated to achieve learning goals they are more likely to regulate their behaviors [7]. This current study provides a preliminary lens on how dynamic measures of fine-grained series of distinctive affect (academic emotions) and behavior (engagement) are reflective of students' emotional and motivational regulation within a learning environment [9, 18], as well as the roles of affect and behavior on self-regulated learning [25].

## 5. ACKNOWLEDGMENTS

This research was supported by grants NSF #DRL-1031398, NSF #SBE-0836012, grant #OPP1048577 from the Bill and Melinda Gates Foundation, and grant #R305A130124 from the Institute of Education Sciences.

## 6. REFERENCES

- [1] Baker, R.S.J.d. 2007. Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. In *Proceedings of ACM CHI 2007: Computer-Human Interaction*, 1059-1068.
- [2] Baker R.S.J.d., Corbett A.T., Gowda S.M., Wagner A.Z., MacLaren B.M., Kauffman L.R., Mitchell A.P., and Giguere S. 2010. Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. In *Proc. UMAP 2010*, 52-63.
- [3] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G., Ocumpaugh, J., and Rossi, L. 2012. Towards Sensor-free Affect Detection in Cognitive Tutor Algebra. In *Proceedings of the 5th International Conference on Educational Data Mining*, 126-133.
- [4] Baker, R.S., Corbett, A.T., and Koedinger, K.R. 2004. Detecting Student Misuse of Intelligent Tutoring Systems. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 531-540.
- [5] Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, S.E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., and Beck, J. 2006. Adapting to When Students Game an Intelligent Tutoring System. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401.
- [6] Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., and Graesser, A. C. 2010. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4), 223-241.
- [7] Bandura, A. 1991. Social cognitive theory of self-regulation. *Organizational behavior and human decision processes*, 50, 2, 248-287.
- [8] Benjamini, Y. and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 289-300.
- [9] Bosch, Nigel, and Sidney D'Mello. 2013. Sequential Patterns of Affective States of Novice Programmers. *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*.
- [10] Cocea, M., Hershkovitz, A., and Baker, R.S.J.d. 2009. The Impact of Off-task and Gaming Behaviors on Learning: Immediate or Aggregate? In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, 507-514.
- [11] Conati, C., and Maclaren, H. 2009. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19, 3, 267-303.
- [12] Corbett, A.T., and Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4, 4, 253-278.
- [13] D'Mello, S.K., Craig, S.D., Witherspoon, A. W., McDaniel, B. T., and Graesser, A. C. 2008. Automatic Detection of Learner's Affect from Conversational Cues. *User Modeling and User-Adapted Interaction*, 18 (1-2), 45-80.
- [14] D'Mello, S. K. and Graesser, A. C. 2012. Dynamics of Affective States during Complex Learning. *Learning and Instruction*, 22, 145-157.
- [15] Fasolo, B., Hertwig, R., Huber, M., and Ludwig, M. 2009. Size, entropy, and density: What is the difference that makes the difference between small and large real-world assortments? *Psychology & Marketing*, 26, 3, 254-279.
- [16] Feng, M., Heffernan, N.T., and Koedinger, K.R. 2009. Addressing the assessment challenge in an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI)*, 19, 3, 243-266.
- [17] Grossman, E. R. F. W. 1953. Entropy and choice time: The effect of frequency unbalance on choice-response. *Quarterly Journal of Experimental Psychology*, 41-51.
- [18] Gumora, G. and Arsenio, W. F. 2002. Emotionality, emotion regulation, and school performance in middle school children. *Journal of School Psychology*, 40, 5, 395-413.
- [19] Hadwin, A. F., Nesbit, J. C., Jamieson-Noel, D., Code, J., and Winne, P. H. 2007. Examining trace data to explore self-regulated learning. *Metaknowledge and Learning*, 2, 107-124.
- [20] Hershkovitz, A., Baker, R.S.J.d., Gobert, J., Wixon, M., and Sao Pedro, M. 2013. Discovery with Models: A Case Study on Carelessness in Computer-based Science Inquiry. *American Behavioral Scientist*, 57, 10, 1479-1498.

- [21] Lee, D. M. C., Rodrigo, M. M. T., d Baker, R. S., Sugay, J. O., and Coronel, A. 2011. Exploring the relationship between novice programmer confusion and achievement. In *Proceedings of Affective Computing and Intelligent Interaction*, 175-184.
- [22] Liu, Z., Ocumpaugh, J., and Baker, R. S. 2013. Sequences of Frustration and Confusion, and Learning. In *Proc. Int. Conf. Ed. Data Mining*, 114-120.
- [23] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., and Heffernan, C. 2014. Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*, 45, 3, 487-501.
- [24] Pardos, Z.A., Baker, R.S., San Pedro, M.O.C.Z., Gowda, S.M., and Gowda, S.M. 2014. Affective states and state tests: Investigating how affect and engagement during the school year predict end of year learning outcomes. *Journal of Learning Analytics*, 1, 107-128.
- [25] Pekrun, R., Goetz, T., Titz, W., and Perry, R. P. 2002. Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational Psychologist*, 37, 2, 91-105.
- [26] Peng, C. K., Buldyrev, S. V., Havlin, S., Simons, H. E., Stanley, H. E., and Goldberger, A. L. 1994. Mosaic organization of DNA nucleotides. *Physical Review E*, 49, 1685-1689.
- [27] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar, R., Walonoski, J.A., Macasek, M.A. and Rasmussen, K.P. 2005. The Assistent project: Blending assessment and assisting. In *Proc. AIED 2005*, 555-562.
- [28] Ritter, S., Joshi, A., Fancsali, S. E., and Nixon, T. 2013. Predicting Standardized Test Scores from Cognitive Tutor Interactions. In *Proceedings of the 6th International Conference on Educational Data Mining*, 169-176.
- [29] Rodrigo, M. M. T., Baker, R. S., and Nabos, J. Q. 2010. The relationships between sequences of affective states and learner achievement. In *Proceedings of the 18th International Conference on Computers in Education*, 56-60.
- [30] Sabourin, J., Mott, B., and Lester, J. 2011. Modeling Learner Affect with Theoretically Grounded Dynamic Bayesian Networks. In *Proc. ACII 2011*, 286-295.
- [31] San Pedro, M.O.Z., Baker, R.S.J.d., Bowers, A.J., Heffernan, N.T. 2013. Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In *Proceedings of the 6th International Conference on Educational Data Mining*, 177-184.
- [32] San Pedro, M.O.Z., Baker, R.S.J.d., Rodrigo, Mercedes, M.M.T. 2014. Carelessness and Affect in an Intelligent Tutoring System for Mathematics. *International Journal of Artificial Intelligence in Education*, 24, 189-210.
- [33] San Pedro, M.O.Z., Ocumpaugh, J.L., Baker, R.S., Heffernan, N.T. 2014. Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. In *Proceedings of the 7th International Conference on Educational Data Mining*, 276-279.
- [34] Shannon, C. 1951. Prediction and Entropy of printed English. *Bell Systems Technical Journal*, 27, 50-64.
- [35] Snow, E. L., Allen, L. K., Jacovina, M. E., and McNamara, D. S. 2015. Does agency matter?: Exploring the impact of controlled behaviors within a game-based environment. *Computers and Education*, 26, (2015), 378-392.
- [36] Snow, E. L., Allen L. K., Russell, D. G., and McNamara, D. S. 2014. Who's in control?: Categorizing nuanced patterns of behaviors within a game-based intelligent tutoring system. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining* (London, UK July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, pp. 185-192.
- [37] Snow, E. L., Jackson, G. T., and McNamara, D. S. 2014. Emergent behaviors in computer-based learning environments: Computational signals of catching up. *Computers in Human Behavior*, 41, (2014), 62-70.
- [38] Snow, E. L., Jacovina, M. E., Allen, L. K., Dai, J., and McNamara, D. S. 2014. Entropy: A stealth assessment of agency in learning environments. In J. Stamper, Z. Pardos, M. Mavrikis, B. M. McLaren (eds.), *Proceedings of the 7th International Conference on Educational Data Mining* (London, UK July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, pp. 185-192.

# Sensor-Free or Sensor-Full: A Comparison of Data Modalities in Multi-Channel Affect Detection

Luc Paquette

Teachers College Columbia University  
525 West 120th Street  
New York, NY 10027  
paquette@tc.columbia.edu

Jonathan Rowe

North Carolina State University  
Raleigh, NC 27695  
jprowe@ncsu.edu

Ryan Baker

Teachers College Columbia University  
525 West 120th Street  
New York, NY 10027  
ryanshaunbaker@gmail.com

Bradford Mott

North Carolina State University  
Raleigh, NC 27695  
bwmott@ncsu.edu

James Lester

North Carolina State University  
Raleigh, NC 27695  
lester@ncsu.edu

Jeanine DeFalco

Teachers College Columbia University  
525 West 120th Street  
New York, NY 10027  
jad2234@tc.columbia.edu

Keith Brawner

US Army Research Lab  
Orlando, FL, USA  
keith.w.brawner@mail.mil

Robert Sottilare

US Army Research Lab  
Orlando, FL, USA  
robert.sottilare@us.army.mil

Vasiliki Georgoulas

Teachers College Columbia University  
525 West 120th Street  
New York, NY 10027  
vasiliki.georgoulas@usma.edu

## ABSTRACT

Computational models that automatically detect learners' affective states are powerful tools for investigating the interplay of affect and learning. Over the past decade, affect detectors—which recognize learners' affective states at run-time using behavior logs and sensor data—have advanced substantially across a range of K-12 and postsecondary education settings. Machine learning-based affect detectors can be developed to utilize several types of data, including software logs, video/audio recordings, tutorial dialogues, and physical sensors. However, there has been limited research on how different data modalities combine and complement one another, particularly across different contexts, domains, and populations. In this paper, we describe work using the Generalized Intelligent Framework for Tutoring (GIFT) to build multi-channel affect detection models for a serious game on tactical combat casualty care. We compare the creation and predictive performance of models developed for two different data modalities: 1) software logs of learner interactions with the serious game, and 2) posture data from a Microsoft Kinect sensor. We find that interaction-based detectors outperform posture-based detectors for our population, but show high variability in predictive performance across different affect. Notably, our posture-based detectors largely utilize predictor features drawn from the research literature, but do not replicate prior findings that these features lead to accurate detectors of learner affect.

## Keywords

Affect detection, multimodal interaction, posture, serious games.

## 1. INTRODUCTION

Affect is critical to understanding learning. However, the interplay between affect and learning is complex. Some affective states, such as boredom, have been shown to coincide with reduced learning outcomes ([25]). Other affective states, such as confusion and engaged concentration, have been found to serve beneficial roles ([14], [24]). The ability to detect a learner's affective state while she interacts with an online learning environment is critical for adaptive learning technologies that aim to support and regulate learners' affect ([26]).

Research on affective computing has enabled the development of models that automatically detect learner affect using a wide variety of data modalities (see extensive review in [8]). Many researchers have focused on physical sensors, because of their capacity to capture physiological and behavioral manifestations of emotion, potentially regardless of what learning system is being used. Sensor-based detectors of affect have been developed using a range of physical indicators including facial expressions ([2], [7]), voice [35], posture ([11], [16]), physiological data [22] and EEG [1]. Despite this promise, deploying physical sensors in the classroom is challenging, and sometimes prohibitive [6], and efforts in this area are still ongoing, with some researchers arguing that this type of affect detection has not yet reached its full potential [13].

In recent years, efforts have also been made towards the development of complementary affect detection techniques that recognize affect solely from logs of learner interactions with an online learning environment ([2], [3], [24]). Initial results in this area have shown considerable promise. As both sensor-based and interaction-based affect detectors continue to mature, efforts are needed to compare the relative advantages of each approach. An early comparison was seen in D'Mello et al. [15], but considerable progress has been made in the years since.

In this paper, we compare the performance and the general process of developing models for affect detection using two different data modalities: learner interaction logs and posture data

from a Microsoft Kinect sensor. Ground-truth affect data for detector development was collected through field observation [23] of learners interacting with vMedic, a serious game on tactical combat casualty care, integrated into the General Intelligent Framework for Tutoring (GIFT) [32]. Findings suggest that interaction-based affect detectors outperform posture-based detectors for our population. However, interaction-based detectors show high variability in predictive performance across different emotions. Further, our posture-based detectors, which utilize many of the same predictor features found throughout the research literature, achieve predictive performance that is only slightly better than chance across a range of affective states, a finding that is contrary to prior work on sensor-based affect detection.

## 2. DATA

Three sources of data were used in this work: 1) log file data produced by learners using the vMedic (a.k.a. TC3Sim) serious game, 2) Kinect sensor log data, and 3) quantitative field observations of learner affect using the BROMP 1.0 protocol [23]. This section describes those sources of data, by providing information on the learning environment, study participants, and research study method.

### 2.1 Learning System and Subjects

We modeled learner affect within the context of vMedic, a serious game used to train US Army combat medics and lifesavers on tasks associated with dispensing tactical field care and care under fire (Figure 1). vMedic has been integrated with the Generalized Intelligent Framework for Tutoring (GIFT) [32], a software framework that includes a suite of tools, methods, and standards for research and development on intelligent tutoring systems and affective computing.

Game-based learning environments, such as vMedic, enable learners to interact with virtual worlds, often through an avatar, and place fewer constraints on learner actions than many other types of computer-based learning environments ([3], [19], [24]). Some virtual environments place more constraints on learner behavior than others. For example, learning scenarios in vMedic are structured linearly, presenting a fixed series of events regardless of the learner's actions. In contrast, game-based learning environments such as EcoMUVE [20] and Crystal Island [29] afford learners considerable freedom to explore the virtual world as they please. While vMedic supports a considerable amount of learner control, its training scenarios focus participants' attention on the objectives of the game (e.g., administering care), implicitly guiding learner experiences toward key learning objectives.

To investigate interaction-based and sensor-based affect detectors for vMedic, we utilize data from a study conducted at the United States Military Academy (USMA). There were 119 cadets who participated in the study (83% male, 17% female). The participants were predominantly first-year students. During the data collection, all participants completed the same training module. The training module focused on a subset of skills for tactical combat casualty care: care under fire, hemorrhage control, and tactical field care. The study materials, including pre-tests, training materials, and post-tests, were administered through GIFT. At the onset of each study session, learners completed a content pre-test on tactical combat casualty care. Afterward, participants were presented with a PowerPoint presentation about tactical combat casualty care. After completing the PowerPoint, participants completed a series of training scenarios in the vMedic serious game where they applied skills, procedures, and

knowledge presented in the PowerPoint. In vMedic, the learner adopts the role of a combat medic faced with a situation where one (or several) of her fellow soldiers has been seriously injured. The learner is responsible for properly treating and evacuating the casualty, while following appropriate battlefield doctrine. After the vMedic training scenarios, participants completed a post-test, which included the same series of content assessment items as the pre-test. In addition, participants completed two questionnaires about their experiences in vMedic: the Intrinsic Motivation Inventory (IMI) [30] and Presence Questionnaire [34]. All combined study activities lasted approximately one hour.

During the study, ten separate research stations were configured to collect data simultaneously; each station was used by one cadet at a time. Each station consisted of an Alienware laptop, a Microsoft Kinect for Windows sensor, and an Affectiva Q-Sensor, as well as a mouse and pair of headphones. The study room's layout is shown in Figure 2. In the figure, participant stations are denoted as ovals. Red cones show the locations of Microsoft Kinect sensors, as well as the sensors' approximate fields of view. The dashed line denotes the walking path for the field observers.

Kinect sensors recorded participants' physical behavior during the study, including head movements and posture shifts. Each Kinect sensor was mounted on a tripod and positioned in front of a participant (Figure 2). The Kinect integration with GIFT provided four data channels: skeleton tracking, face tracking, RGB (i.e., color), and depth data. The first two channels leveraged built-in tracking algorithms (which are included with the Microsoft Kinect for Windows SDK) for recognizing a user's skeleton and face, each represented as a collection of 3D vertex coordinates. The RGB channel is a 640x480 color image stream comparable to a standard web camera. The depth channel is a 640x480 IR-based image stream depicting distances between objects and the sensor.

Q-Sensors recorded participants' physiological responses to events during the study. The Q-Sensor is a wearable arm bracelet that measures participants' electrodermal activity (i.e., skin conductance), skin temperature, and its orientation through a built-in 3-axis accelerometer. However, Q-Sensor logs terminated prematurely for a large number of participants, necessitating additional work to determine the subset of field observations that are appropriate to predict with Q-Sensor-based features. Inducing Q-Sensor-based affect detectors will be an area of future work.



Figure 1. vMedic learning environment.

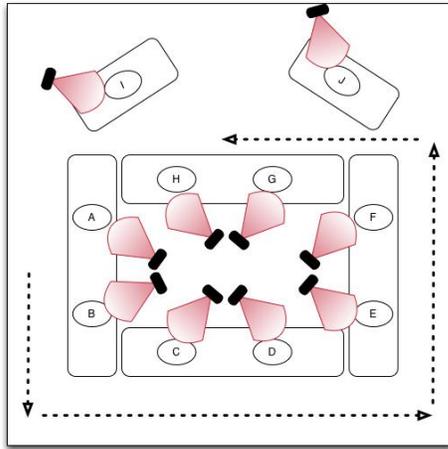


Figure 2. Study room layout.

## 2.2 Quantitative Field Observations (QFOs)

We obtain ground-truth labels of affect using Quantitative Field Observations (QFOs), collected using the Baker-Rodrigo-Ocuppaugh Monitoring Protocol (BROMP) [23]. This is a common practice for interaction-based detection of affect (e.g. [3], [24]). Much of the work to date for video-based affect detection, by contrast, has focused on modeling emotion labels that are based on self-reports ([10], [16]), or labels obtained through retrospective judgments involving freeze-frame video analysis [11]. It has been argued that BROMP data is easier to obtain and maintain reliability for under real-world conditions than these alternate methods [23], being less disruptive than self-report, and easier to gain full context than video data.

To be considered BROMP-certified, a coder must achieve inter-rater reliability of  $Kappa \geq 0.6$  with a previously BROMP-certified coder. BROMP has been used for several years to study behavior and affect in educational settings ([3], [4], [27]), with around 150 BROMP-certified coders as of this writing, and has been used as the basis for successful automated detectors of affect ([3], [24]). Observations in this study were conducted by two BROMP-certified coders, the 2nd and 6th authors of this paper.

Within the BROMP protocol, behavior and affective states are coded separately but simultaneously using the Human Affect Recording Tool (HART), an application developed for the Android platform (and freely available as part of the GIFT distribution). HART enforces a strict coding order determined at the beginning of each session. Learners are coded individually, and coders are trained to rely on peripheral vision and side glances in order to minimize observer effects. The coder has up to 20 seconds to categorize each trainee's behavior and affect, but records only the first thing he or she sees. In situations where the trainee has left the room, the system has crashed, where his or her affect or behavior do not match any of the categories in the current coding scheme, or when the trainee can otherwise not be adequately observed, a '?' is recorded, and that observation is eliminated from the training data used to construct automated detectors.

In this study, the typical coding scheme used by BROMP was modified to accommodate the unique cadet behaviors and affect that was manifest for this specific cadet population and domain. Affective states observed included frustration, confusion, engaged concentration, boredom, surprise and anxiety. Behavioral categories consisted of on-task, off-task behaviors, Without

Thinking Fastidiously behavior [33], and intentional friendly fire (these last two categories will not be discussed in detail, as they were rare).

In total, 3066 BROMP observations were collected by the two coders. Those observations were collected over the full length of the cadets' participation in the study, including when they were answering questionnaires on self-efficacy, completing the pre and post-tests, reviewing PowerPoint presentations, and using vMedic. For this study, we used only the 755 observations that were collected while cadets were using vMedic. Of those 755 observations, 735 (97.35%) were coded as the cadet being on-task, 19 (2.52%) as off-task, 1 (0.13%) as Without Thinking Fastidiously, and 0 as intentional friendly fire. Similarly, 435 (57.62%) of the affect labels were coded as concentrating, 174 (23.05%) as confused, 73 (9.67%) as bored, 32 (4.24%) as frustrated, 29 (3.84%) as surprised and 12 (1.59%) as anxious.

## 3. INTERACTION-BASED DETECTORS

The BROMP observations collected while cadets were using vMedic were used to develop machine-learned models to automatically detect the cadet's affective states. In this section, we discuss our work to develop affect detectors based on cadets' vMedic interactions logs.

### 3.1 Data Integration

In order to generate training data for our interaction-based affect detectors, trainee actions within the software were synchronized to field observations collected using the HART application. During data collections, both the handheld computers and the GIFT server were synchronized to the same internet NTP time server. Timestamps from both the HART observations and the interaction data were used to associate each observation to the actions that occurred during the 20 seconds window prior to data entry by the observer. Those actions were considered as co-occurring with the observation.

### 3.2 Feature Distillation

For each observation, we distilled a set of 38 features that summarized the actions that co-occurred with or preceded that observation. Those features included: changes in the casualty, both recent and since injury, such as changes in blood volume, bleed rate and heart rate; player states in terms of attacker, such as being under cover and being with the unit; the number of time specific actions, such as applying a tourniquet or requesting a security sweep, were executed; and time between actions. (see [5] for a more complete list of features.)

### 3.3 Machine Learning Process

Detectors were built separately for each affective state and behavioral constructs. For example a detector was used to distinguish observations of boredom from observations that were not boredom. It is worth noting that the construct of engaged concentration, was defined during modeling as a learner having the affect of concentration and not being off-task, since concentrating while being off-task reflects concentration with something other than learning within the vMedic game. Only 2 such observations was found amongst the collected observations. Detectors were not developed for off-task behavior, Without Thinking Fastidiously behavior, and anxiety due to the low number of observations for those construct (19, 1 and 12 respectively).

Each detector was validated using 10-fold participant-level cross-validation. In this process, the trainees are randomly separated into 10 groups of equal size and a detector is built using data for

each combination of 9 of the 10 groups before being tested on the 10th group. By cross-validating at this level, we increase confidence that detectors will be accurate for new trainees. Oversampling (through cloning of minority class observations) was used to make the class frequency more balanced during detector development. However, performance calculations were made with reference to the original dataset.

Detectors were fit in RapidMiner 5.3 [21] using six machine learning algorithms that have been successful for building similar detectors in the past ([3], [24]): J48, JRip, NaiveBayes, Step Regression, Logistic Regression and KStar. The detector with the best performance was selected for each affective state. Detector performance was evaluated using two metrics: Cohen's Kappa [9] and  $A'$  computed as the Wilcoxon statistic [18]. Cohen's Kappa assesses the degree to which the detector is better than chance at identifying the modeled construct. A Kappa of 0 indicates that the detector performs at chance, and a Kappa of 1 indicates that the detector performs perfectly.  $A'$  is the probability that the algorithm will correctly identify whether an observation is a positive or a negative example of the construct (e.g. is the learner bored or not?).  $A'$  is equivalent to the area under the ROC curve in signal detection theory [18]. A model with an  $A'$  of 0.5 performs at chance, and a model with an  $A'$  of 1.0 performs perfectly.  $A'$  was computed at the observation level.

When fitting models, feature selection was performed using forward selection on the Kappa metric. Performance was evaluated by repeating the feature selection process on each fold of the trainee-level cross-validation in order to evaluate how well models created using this feature selection procedure perform on new and unseen test data. The final models were obtained by applying the feature selection to the complete dataset.

## 4. POSTURE-BASED DETECTORS

The second set of affect detectors we built were based on learner posture during interactions with vMedic. Kinect sensors produced data streams that were utilized to determine learner posture. Using machine learning algorithms, we trained models to recognize affective states based on postural features.

### 4.1 Data Integration

GIFT has a *sensor module* that is responsible for managing all connected sensors and associated data streams. This includes Kinect sensor data, which is comprised of four complementary data streams: face tracking, skeleton tracking, RGB channel, and depth channel data. Face- and skeleton-tracking data are written to disk in CSV format, with rows denoting time-stamped observations and columns denoting vertex coordinates. RGB and depth channel data are written to disk as compressed binary data files. To analyze data from the RGB and depth channels, one must utilize the *GiftKinectDecoder*, a standalone utility that is packaged with GIFT, to decompress and render the image data into a series of images with timestamp-based file names. Data from all four channels can be accessed and analyzed outside of GIFT. For the present study, we utilized only vertex data to analyze participants' posture. Each observation in the vertex data consisted of a timestamp and a set of 3D coordinates for 91 vertices, each tracking a key point on the learner's face (aka face tracking) or upper body (aka skeletal tracking). The Kinect sensor sampled learners' body position at a frequency of 10-12 Hz.

It was necessary to clean the Kinect sensor data in order to remove anomalies from the face and skeletal tracking. Close examination of the Kinect data revealed periodic, and sudden, jumps in the coordinates of posture-related vertices across frames.

These jumps were much larger than typically observed across successive frames, and they occurred due to an issue with the way GIFT logged tracked skeletons: recording the *most recently* detected skeleton, rather than the *nearest* detected skeleton. This approach to logging skeleton data caused GIFT to occasionally log bystanders standing in the Kinect's field of view rather than the learner using vMedic. In our study, such a situation could occur when a field observer walked behind the trainee.

To identify observations that corresponded to field observers rather than participants, Euclidean distances between subsequent observations of a central vertex were calculated. The distribution of Euclidean distances was plotted to inspect the distribution of between-frame movements of the vertex. If the Kinect tracked field observers, who were physically located several feet behind participants, the distribution was likely to be bimodal. In this case, one cluster would correspond to regular posture shifts of a participant between frames, and the other cluster corresponded to shifts between tracking participants and field observers. This distribution could be used to identify a distance threshold for determining which observations should be thrown out, as they were likely due to tracking field observers rather than participants. Although the filtering process was successful, the need for this process reveals a challenge to the use of BROMP for detectors eventually developed using Kinect or video data.

In addition to cleaning the face and skeleton mesh data, we performed a filtering process to remove data that were unnecessary for the creation of posture-based affect detectors. A majority of the facial vertices recorded by the Kinect sensor were not necessary for investigating trainees' posture. Of the 91 vertices recorded by the Kinect sensor, only three were utilized for posture analysis: *top\_skull*, *head*, and *center\_shoulder*. These vertices were selected based on prior work investigating postural indicators of emotion with Kinect data [16].

Finally, HART observations were synchronized with the data collected from the Kinect sensor. As was the case for our interaction-based sensor, the Kinect data provided by GIFT was synchronized to the same NTP time server as the HART data. This allowed us to associate field observations with observations of face and skeleton data produced by the Kinect sensor.

### 4.2 Feature Distillation

We used the Kinect face and skeleton vertex data to compute a set of predictor features for each field observation. The engineered features were inspired by related work on posture sensors in the affective computing literature, including work with pressure-sensitive chairs ([10], [11]) and, more recently, Kinect sensors [16]. Several research groups have converged on common sets of postural indicators of emotional states. For example, in several cases boredom has been found to be associated with leaning back, as well as increases in posture variance ([10], [11]). Conversely, confusion and flow have been found to be associated with forward-leaning behavior ([10], [11]).

We computed a set of 73 posture-related features. The feature set was designed to emulate the posture-related features that had previously been utilized in the aforementioned posture-based affect detection work ([10], [11], [16], [17]). For each of three retained skeletal vertices tracked by the Kinect (*head*, *center\_shoulder*, and *top\_skull*), we calculated 18 features based on multiple time window durations. These features are analogous to those described in [16], and were previously found to predict learners' retrospective self-reports of frustration and engagement:

- Most recently observed distance

**Table 1. Performance of each of the interaction-based and posture-based detectors of affect**

Affect	Interaction-Based Detectors			Posture-Based Detectors		
	Classifier	Kappa	A'	Classifier	Kappa	A'
Boredom	Logistic Regression	0.469	0.848	Logistic Regression	0.109	0.528
Confusion	Naïve Bayes	0.056	0.552	JRip	0.062	0.535
Engaged Concentration	Step Regression	0.156	0.590	J48	0.087	0.532
Frustration	Logistic Regression	0.105	0.692	Support Vec. Machine	0.061	0.518
Surprise	KStar	0.081	0.698	Logistic Regression	-0.001	0.493

- Most recently observed depth (Z coordinate)
- Minimum observed distance observed thus far
- Maximum observed distance observed thus far
- Median observed distance observed thus far
- Variance in distance observed thus far
- Minimum observed distance during past 5 seconds
- Maximum observed distance during past 5 seconds
- Median observed distance during past 5 seconds
- Variance in distance during past 5 seconds
- Minimum observed distance during past 10 seconds
- Maximum observed distance during past 10 seconds
- Median observed distance during past 10 seconds
- Variance in distance during past 10 seconds
- Minimum observed distance during past 20 seconds
- Maximum observed distance during past 20 seconds
- Median observed distance during past 20 seconds
- Variance in distance during past 20 seconds

We also induced several *net\_change* features, which are analogous to those reported in [11] and [10] using pressure-sensitive seat data:

$$net\_dist\_change[t] = \begin{cases} head\_dist[t] - head\_dist[t-1] + \\ cen\_shldr\_dist[t] - cen\_shldr\_dist[t-1] + \\ top\_skull\_dist[t] - top\_skull\_dist[t-1] \end{cases} \quad (1)$$

$$net\_pos\_change[t] = \begin{cases} head\_pos[t] - head\_pos[t-1] + \\ cen\_shldr\_pos[t] - cen\_shldr\_pos[t-1] + \\ top\_skull\_pos[t] - top\_skull\_pos[t-1] \end{cases} \quad (2)$$

These features were calculated from Kinect vertex tracking data, as opposed to seat pressure data. Specifically, the *net\_dist\_change* feature was calculated as each vertex's net change in distance (from the Kinect sensor) over a given time window, and then summed together. The *net\_pos\_change* feature was calculated as the Euclidean distance between each vertex's change in position over a given time window, and then summed together. Both the *net\_dist\_change* feature and *net\_pos\_change* feature were calculated for 3 second and 20 second time windows.

We also calculated several *sit\_forward*, *sit\_back*, and *sit\_mid* features analogous to [10] and [17]. To compute these features, we first calculated the average median distance of participants' *head* vertex from each Kinect sensor. This provided a median distance for each of the 10 study stations (see Figure 1). We also calculated the average standard deviation of *head* distance from each sensor. Then, based on the station-specific medians and standard deviations, we calculated the following features for each participant:

$$sit\_forward = \begin{cases} 1 & \text{if } head\_dist \leq median\_dist - st\_dev \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$sit\_back = \begin{cases} 1 & \text{if } head\_dist \geq median\_dist + st\_dev \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The *sit\_mid* feature was the logical complement of *sit\_forward* and *sit\_back*; if a learner was neither sitting forward, nor sitting back, they were considered to be in the *sit\_mid* state. We also computed predictor features that characterized the proportion of observations in which the learner was in a *sit\_forward*, *sit\_back*, or *sit\_mid* state over a window of time. Specifically, we calculated these features for 5, 10, and 20 second time windows, as well as over the entire session to-date.

### 4.3 Machine Learning

Posture-based detectors of affect were built using a process analogous to the one used to build our interaction-based detectors. As such, separate detectors were, once again, built for each individual affective state and behavioral construct. All observations labeled as '?' were removed from the training set as they represent observations where the cadet's affective state or behavior could not be determined.

Each detector was validated using 10-fold participant-level cross-validation. Oversampling was used to balance class frequency by cloning minority class instances, as was the case when training our interaction-based detectors. RapidMiner 5.3 was used to train the detectors using multiple different classification algorithms: J48 decision trees, naïve Bayes, support vector machines, logistic regression, and JRip. When fitting posture-based affect detection models, feature selection was, once again, performed through forward selection using a process analogous to the one used for our interaction-based detectors.

## 5. RESULTS

As discussed above, each of the interaction-based and posture-based detectors of affect were cross-validated at the participant level (10 folds) and performance was evaluated using both Kappa and A'. Table 1 summarizes the performance achieved by each detector for both the Kappa and A' metrics.

Performance of our interaction-based detectors was highly variable across affective states. The detector of boredom achieved, by far, the highest performance (Kappa = 0.469, A' = 0.848) while some of the other detectors achieved very low performance. This was the case for the confusion detector that performed barely above chance level (Kappa = 0.056, A' = 0.552). Detectors of

frustration and surprise achieved relatively low Kappa (0.105 and 0.081 respectively), but good A' (0.692 and 0.698 respectively). Performance for engaged concentration achieved a Kappa closer to the average (0.156), but below average A' (0.590).

In general, posture-based detectors performed only slightly better than chance, with the exception of the surprise detector, which actually performed worse than chance. The boredom detector, induced as a logistic regression model, achieved the highest predictive performance (Kappa = 0.109, A' = 0.528), induced as a logistic regression model.

## 6. DISCUSSION

Across affective states, the posture-based detectors achieved lower predictive performance than the interaction-based detectors. In fact, the posture-based detectors performed only slightly better than chance, and in the case of some algorithms and emotions, worse than chance. This finding is notable, given that our distilled posture features were inspired largely from the research literature, where these types of features have been shown to predict learner emotions effectively in other contexts ([10], [11], [16], [17]). For example, D'Mello and Graesser found machine-learned classifiers discriminating affective states from neutral yielded kappa values of 0.17, on average [10]. Their work utilized posture features distilled from pressure seat data, including several features analogous to those used in our work. Grafsgaard et al. found that Pearson correlation analyses with retrospective self-reports of affect revealed significant relationships between posture and emotion, including frustration, focused attention, involvement, and overall engagement. Reported correlation coefficients ranged in magnitude from 0.35 to 0.56, which are generally considered moderate to large effects [19]. Cooper et al. found that posture seat-based features were particularly effective for predicting excitement in stepwise regression analyses ( $R = 0.56$ ), and provided predictive benefits beyond log-based models across a range of emotions [10]. While the methods employed in each of these studies differ from our own, and thus the empirical results are not directly comparable, the qualitative difference in the predictive value of postural features is notable.

There are several possible explanations for why our posture-based predictors were not more effective. First, our use of BROMP to generate affect labels distinguishes our work from prior efforts, which used self-reports ([10], [16], [17]) or retrospective video freeze-frame analyses [11]. It is possible that BROMP-based labels of affect present distinct challenges for posture-based affect detection. BROMP labels are based on holistic judgments of affect, and pertain to 20-second intervals of time, which may be ill matched for methods that depend upon low-level postural features to predict emotion. Similarly, much of the work on posture-based affect detection has taken place in laboratory settings involving a single participant at a time [11], especially prior work using Kinect sensors ([16], [17]). In contrast, our study was performed with up to 10 simultaneous participants (see Figure 2), introducing potential variations in sensor positions and orientations. This variation may have introduced noise to our posture data, making the task of inducing population-general affect detectors more challenging than in settings where data is collected from a single sensor. If correct, this explanation underscores the challenges inherent in scaling and generalizing sensor-based affect detectors.

The study room's setup also limited how sensors could be positioned and oriented relative to participants. For example, it was not possible to orient Kinect cameras to the sides of participants, capturing participants' profiles, which would have made it easier to detect forward-leaning and backward-leaning

postures. This approach has shown promise in other work, but was not a viable option in our study [31]. Had the Kinect sensors been positioned in this manner, the video streams would have been disrupted by other participants' presence in the cameras' fields of view.

Another possible explanation has to do with the population of learners that was involved in the study: U.S. Military Academy (USMA) cadets. Both BROMP observers noted that the population's affective expressiveness was generally different in kind and magnitude than the K-12 and civilian academic populations they were more accustomed to studying. Specifically, they indicated that the USMA population's facial and behavioral expressions of affect were relatively subdued, perhaps due to military cultural norms. As such, displays of affect via movement and body language may have been more difficult to recognize than would have otherwise been encountered in other populations.

In general, we consider the study population, BROMP affect labels, and naturalistic research setup to be strengths of the study. Indeed, despite the difference in how military display affect compared to the K-12 and civilian academic population, human observers were able to achieve the inter-rater reliability required by BROMP (Kappa  $\geq 0.6$ ) [23]. Thus we do not have plans to change these components in future work. Instead, we will likely seek to revise and enhance the data mining techniques that we employ to recognize learner affect, as well as the predictor features engineered from raw posture data. In addition, we plan to explore the predictive utility of untapped data streams (e.g., Q-Sensor data, video data).

It is notable that our interaction-based detectors had a more varied performance than had been seen in prior studies using this methodology; the detectors were excellent for boredom, and varied from good to just above chance for other constructs. It is possible that this too is due to the population studied, but may also be due to the nature of the features that were distilled in order to build the models. For example, the high performance of our detector of boredom can be attributed to the fact that one feature, whether the student executed any meaningful actions in the 20 second observation window, very closely matched the trainees' manifestation of this affective state. In fact, a logistic regression detector trained using this feature alone achieved higher performance than our detectors for any of the other affective state (Kappa = 0.362, A' = 0.680). It can be difficult to predict, a priori, which features will most contribute to the detection of a specific affective state. It is also possible that some of the affective states for which interaction-based detection was less effective (e.g., confusion) simply did not manifest consistently in the interactions with the learning environment across different trainees. It is thus difficult to determine whether poor performance of detectors for some constructs, such as our confusion detector, is due to insufficient feature engineering or inconsistent behaviors by the trainee. As such, the creation of interaction-based detectors is an iterative process, where features are engineered, and models are induced and refined, until performance reaches an acceptable level, or no improvement in performance is observed, despite repeated knowledge-engineering efforts.

We aim to identify methods to improve the predictive accuracy of posture-based detectors in future work. One advantage they possess relative to interaction-based detectors is that posture-based detectors may be more generalizable, since they pertain to aspects of learner behavior that are outside of the software itself. By contrast, much of the effort invested in the creation of interaction-based detectors is specific to the system for which the

detectors are created. Features are built to summarize the learner's interaction in the learning environment and, as such, are dependent on the system's user interface. Much of the creation of interaction-based detectors must hence be replicated for new learning environments, though there have been some attempts to build toolkits that can replicate features seen across many environments, such as unitizing the time between actions by the type of action or problem step (e.g. [28]).

On the other hand, posture-based detectors are built upon a set of features that are more independent of the system for which the detectors are designed. The process of creating the features itself requires considerable effort when compared with building a set of features for interaction-based detectors, such as elaborate efforts to adequately clean the data, but at least in principle, it is only necessary to develop the methods for doing so once. The same data cleaning and feature distillation procedures can be repeated for subsequent systems. This is especially useful in the context of a generalized, multi-system tutoring framework such as GIFT [32]. Although different posture-based affect detectors might need to be created for different tutoring systems—due to differences in the postures associated with affect for different populations of learners, environments and contexts—the posture features we computed from the data provided by Kinect sensors will ultimately become available for re-use by any tutor created using GIFT. This has the potential to considerably reduce the time required to build future posture-based affect detectors for learning environments integrated with the GIFT architecture.

## 7. CONCLUSION

Interaction-based and posture-based detectors of affect show considerable promise for adaptive computer-based learning environments. We have investigated their creation and predictive performance in the context of military cadets using the vMedic serious game for tactical combat casualty care. Interaction-based and posture-based detectors capture distinct aspects of learners' affect. Whereas interaction-based detectors capture the relationship between affect and its impact on the trainee's action in the learning environment, posture-based detectors capture learners' physical expressions of emotion.

In our study, we found that interaction-based detectors achieved overall higher performance than posture-based detectors. We speculate that the relatively weak predictive performance of our posture-based affect detectors may be due to some combination of the following: the interplay of high-level BROMP affect labels and low-level postural features, the challenges inherent in running sensor-based affect studies with multiple simultaneous participants, and population-specific idiosyncrasies in USMA cadets' affective expressiveness compared to other populations. The relative advantages and limitations of both interaction-based and posture-based detectors point toward the need for continued research on both types. Each type of detector captures different aspects of learners' manifestations of affective state, and many open questions remain about feature engineering and the predictive ability of each type of detector.

An important direction for future work will be the integration and combination of the two types of detectors presented here. In multiple cases, the combination of data modalities for the creation of affect detectors has been shown to produce detectors with better performance than single-modality detectors ([12], [13], [17]). As such, future work will focus on the study of how these two channels of information can be combined to produce more effective and robust detectors of affect.

Further research on effective, generalizable predictor features for posture-based affect detectors is also needed, as shown by the relatively weak predictive performance of existing features observed in this study. Complementarily, investigating the application of other machine learning algorithms, including temporal models, is likely to prove important, given the complex temporal dynamics of affect during learning. These directions are essential for developing an enhanced understanding of the interplay between affect detector architectures, learning environments, student populations, and methods for determining ground truth affect labels. While significant progress has been made toward realizing the vision of robust, generalizable affect-sensitive learning environments, these findings point toward the need for continued empirical research, as well as advances in educational data mining methods applicable to affective computing.

## 8. ACKNOWLEDGMENTS

We thank our research colleagues, COL James Ness and Dr. Michael Matthews in the Behavioral Science & Leadership Department at the United States Military Academy for their assistance in conducting the study. This research is supported by cooperative agreement #W911NF-13-2-0008 between the U.S. Army Research Laboratory, Teachers College Columbia University, and North Carolina State University. Any opinions, findings, and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the U.S. Army Research Laboratory.

## 9. REFERENCES

- [1] AlZoubi, O., Calvo, R.A., and Stevens, R.H. 2009. Classification of EEG for Emotion Recognition: An Adaptive Approach. *Proc. of the 22nd Australian Joint Conference on Artificial Intelligence*, 52-61.
- [2] Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., and Christopherson, R. 2009. Emotion Sensors Go to School. *Proc. of the 14th Int'l Conf. on Artificial Intelligence in Education*, 17-24.
- [3] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G.W., Ocumpaugh, J., and Rossi, L. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *Proc. of the 5th Int'l Conf. on Educational Data Mining*, 126-133.
- [4] Baker, R., D'Mello, S., Rodrigo, M.M.T., and Graesser, A. 2010. Better to be Frustrated than Bored: The Incidence and Persistence of Affect During Interactions with Three Different Computer-Based Learning Environments. *Int'l J. of Human-Computer Studies*, 68 (4), 223-241.
- [5] Baker, R.S., DeFalco, J.A., Ocumpaugh, J., and Paquette, L. 2014. Towards Detection of Engagement and Affect in a Simulation-Based Combat Medic Training Environment. *2nd Annual GIFT User Symposium (GIFTSym2)*.
- [6] Baker, R.S., and Ocumpaugh, J. 2015. Interaction-Based Affect Detection in Educational Software. *The Oxford Handbook of Affective Computing*, 233-245.
- [7] Bosch, N., D'Mello, S., Baker, R., Ocumpaugh, J., Shute, V., Ventura, M., Wang, L., and Zhao, W. 2015. Automatic Detection of Learning-Centered Affective States in the Wild. *Proc. of the 2015 Int'l Conf. on Intelligent User Interfaces*.
- [8] Calvo, R.A., and D'Mello, S. 2010. Affect Detection: An Interdisciplinary Review of Models, Methods, and their

- Applications. *IEEE transactions on Affective Computing*, 1 (1), 18-37.
- [9] Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational Psychological Measurement*, 20 (1), 37-46.
- [10] Cooper, D.G., Arroyo, I., Woolf, B.P., Muldner, K., Burleson, W., and Christopherson, R. 2009. Sensors Model Student Self Concept in the Classroom. *Proc. of the 17th Int'l Conf. on User Modeling, Adaption, and Personalization*, 30-41.
- [11] D'Mello, S., and Graesser, A. 2009. Automatic detection of learners' affect from gross body language. *Applied Artificial Intelligence*, 23, 2, 123-150.
- [12] D'Mello, S., Kory, J. 2012. Consistent but Modest: Comparing Multimodal and Unimodal Affect Detection Accuracies from 30 Studies. *Proc. of the 14th ACM International Conf. on Multimodal Interaction*, 31-38.
- [13] D'Mello, S.K., Kory, J. in press. A Review and Meta-Analysis of Multimodal Affect Detection Systems. *ACM Computing Surveys*.
- [14] D'Mello, S., Lehman, B., Pekrun, R., and Graesser, A. 2014. Confusion can be Beneficial for Learning. *Learning and Instruction*, 29, 153-170.
- [15] D'Mello, S., Jackson, T., Craig, S., Morgan, B., Chipman, P., White, H., Person, N., Kort, B., el Kaliouby, R., Picard, R., and Graesser, A. 2008. AutoTutor Detects and Responds to Learners Affective and Cognitive States. *Workshop on Emotional and Cognitive Issues at the 9th Int'l Conf. on Intelligent Tutoring Systems*.
- [16] Grafsgaard, J., Boyer, K., Wiebe, E., and Lester, J. 2012. Analyzing Posture and Affect in Task-Oriented Tutoring. *Proc. of the 25th Florida Artificial Intelligence Research Society Conference*, 438-443.
- [17] Grafsgaard, J., Wiggins, J., Boyer, K.E., Wiebe, E., and Lester, J. 2014. Predicting Learning and Affect from Multimodal Data Streams in Task-Oriented Tutorial Dialogue. *Proc. of the 7<sup>th</sup> Int'l Conf. on Educational Data Mining*, 122-129.
- [18] Hanley, J., and McNeil, B. 1982. The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143, 29-36.
- [19] Litman, D.J., and Forbes-Riley K. 2006. Recognizing Student Emotions and Attitudes on the Basis of Utterances in Spoken and Tutoring Dialogue with Both Humans and Computer-Tutors. *Speech Communication*, 48, 559-590.
- [20] Metcalf, S., Kamarainen, A., Tutwiler, M.S., Grotzer, T., and Dede, C. 2011. Ecosystem Science Learning via Multi-User Virtual Environments. *Int'l J. of Gaming and Computer-Mediated Simulations*, 3 (1), 86-90.
- [21] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., and Euler, T. 2006. YALE: Rapid Prototyping for Complex Data Mining Tasks. *Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*, 935-940.
- [22] Nasoz, F., Alvarez, K., Lisetti, C.L., and Finkelstein, N. 2004. Emotion from Physiological Signals Using Wireless Sensors for Presence Technologies. *Cognition, Technology and Work*, 6, 4-14.
- [23] Ocumpaugh, J., Baker, R.S.J.d, and Rodrigo, M.M.T. 2012. *Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual version 1.0*. Technical Report.
- [24] Pardos, Z., Baker, R.S.J.d., San Pedro, M.O.Z., Gowda, S.M., and Gowda, S. 2014. Affective States and State Tests: Investigating how Affect and Engagement During the School Year Predict End of Year Learning Outcomes. *J. of Learning Analytics*, 1 (1), 107-128.
- [25] Pekrun, R., Goetz, T., Daniels, L.M., Stupnisky, R.H., and Perry, R.H. 2010. Boredom in Achievement Settings: Exploring Control-Value Antecedents and Performance Outcomes of a Neglected Emotion. *J. of Educational Psychology*, 102, 531-549.
- [26] Rai, D., Arroyo, I., Stephens, L., Lozano, C., Burleson, W., Woolf, B.P., and Beck, J.E. 2013: Repairing Deactivating Emotions with Student Progress Pages. *Proc. of the 16th Int'l Conf. on Artificial Intelligence in Education*, 795-798.
- [27] Rodrigo, M.M.T., and Baker, R.S.J.d. 2009. Coarse-Grained Detection of Student Frustration in an Introductory Programming Course. *Proc. of the 5th Int'l Workshop on Computing Education Research Workshop*, 75-80.
- [28] Rodrigo, M.M.T., Baker, R.S.J.d., McLaren, B., Jayme, A., and Dy, T. 2012. Development of a Workbench to Address the Educational Data Mining Bottleneck. *Proc. of the 5<sup>th</sup> Int'l Conf. on Educational Data Mining*, 152-155.
- [29] Rowe, J., Mott, B., McQuiggan, J., Robison, S., and Lester, J. 2009. Crystal Island: A Narrative-Centered Learning Environment for Eighth Grade Microbiology. *Workshop on Educational Games at the 14th Int'l Conf. on Artificial Intelligence in Education*, 11-20.
- [30] Ryan, R.M. 1982. Control and Information in the Intrapersonal Sphere: An Extension of Cognitive Evaluation Theory. *J. of Personality and Social Psychology*, 43, 450-461.
- [31] Sanghvi, J., Castellano, G., Leite, I., Pereria, A., McOwan, P., and Paiva, A. 2011. Automatic Analysis of Affective Postures and Body Motion to Detect Engagement with a Game Companion. *Proc. of the 6<sup>th</sup> ACM/IEEE Int'l Conf. on Human-Robot Interaction*, 305-311.
- [32] Sottolare, R.A., Golberg, B. and Holden, H. 2012. *The Generalized Intelligent Framework for Tutoring (GIFT)*.
- [33] Wixon, M., Baker, R.S.J.d., Gobert, J., Ocumpaugh, J., and Bachmann, M. 2012. WTF? Detecting Students who are Conducting Inquiry Without Thinking Fastidiously. *Proc. of the 20th Int'l Conf. on User Modeling, Adaptation and Personalization*, 286-298.
- [34] Witmer, B.G., Jerome, C. J., & Singer, M. J. (2005, June). The factor structure of the presence questionnaire. *Presence*, Vol. 14(3), 298-312. MIT Press, Cambridge MA.
- [35] Zeng, Z., Pantic, M., Roisman, G.I., and Huang, T.S. 2009. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (1), 39-58.

# Machine beats experts: Automatic discovery of skill models for data-driven online course refinement

Noboru Matsuda<sup>1</sup>    Tadanobu Furukawa<sup>2</sup>    Norman Bier<sup>3</sup>    Christos Faloutsos<sup>2</sup>  
mazda@cs.cmu.edu    tfuru@cs.cmu.edu    nbier@cmu.edu    christos@cs.cmu.edu  
<sup>1</sup>Human-Computer Interaction Institute    <sup>2</sup>Computer Science Department    <sup>3</sup>Open Learning Initiative  
Carnegie Mellon University  
5000 Forbes Ave., Pittsburgh PA 15213, USA

## ABSTRACT

How can we automatically determine which skills must be mastered for the successful completion of an online course? Large-scale online courses (e.g., MOOCs) often contain a broad range of contents frequently intended to be a semester’s worth of materials; this breadth often makes it difficult to articulate an accurate set of skills and knowledge (i.e., a skill model, or the Q-Matrix). We have developed an innovative method to discover skill models from the data of online courses. Our method assumes that online courses have a pre-defined skill map for which skills are associated with formative assessment items embedded throughout the online course. Our method carefully exploits correlations between various parts of student performance, as well as in the text of assessment items, to build a superior statistical model that even outperforms human experts. To evaluate our method, we compare our method with existing methods (LFA) and human engineered skill models on three Open Learning Initiative (OLI) courses at Carnegie Mellon University. The results show that (1) our method outperforms human-engineered skill models, (2) skill models discovered by our method are interpretable, and (3) our method is remarkably faster than existing methods. These results suggest that our method provides a significant contribution to the evidence-based, iterative refinement of online courses with a promising scalability.

## Keywords

Online course refinement, skill model discovery, evidence-base course engineering, MOOC, Q-matrix

## 1. INTRODUCTION

When designing and implementing large-scale online courses (aka MOOCs), defining a set of skills to be learned and having individual skills associated with particular part of course contents often becomes quite challenging. Making an effective course with explicit associations between a necessary set of skills and course contents requires intensive cognitive task analysis and time-consuming evidence-based iterative engineering [1]. Studies show

how important it is to have data-analytics feedback for course improvement and theory development [2-5]. However, cognitive task analysis driven by human experts has an issue in its accuracy and scalability; applying it for a large-scale online course is often impractical.

Research shows the potential for advanced technologies to automatically and semi-automatically discover a set of skills for online courses. Learning Factor Analysis (LFA), for example, semi-automatically refines a given skill set [6]. However, LFA works only when meaningful “features” are given, which (usually) requires cognitive task analysis by subject domain experts. Other studies apply matrix factorization methods for automatic skill set (aka Q-matrix) discovery from students’ response data [7, 8]. However, these methods often face the issue of interpretability—i.e., providing meaningful feedback to course designers and developers based on the machine-generated skill set is often troublesome.

We developed an efficient, practical, and scalable method that we call eEPIPHANY, to fully and automatically discover skill sets from online course data, which are the combination of the assessment item text data (i.e., problem and feedback text sentences for assessment items) and student learning interaction data. eEPIPHANY is a collection of data-mining techniques to automatically refine (or rebuild) a human-crafted set of skills, initially given by course designers and developers.

The most important goal of eEPIPHANY is to provide constructive feedback to online course designers and developers for iterative course improvement. We assume that our target online courses have occasional formative assessments to probe students’ competency towards learning objectives. We hypothesize that students’ response data and assessment item text data both reflect latent skills to be learned, and assessment items can be clustered based on those latent skills. To test these hypotheses, we implemented eEPIPHANY as a combination of the matrix factorization to analyze students’ response data and bag-of-words techniques to analyze course content data.

The contributions of this work are the following: (1) *A new problem formulation*—We show how to integrate diversified information such as student performance and assessment item text data. (2) *A new algorithm*—Our solution, the eEPIPHANY algorithm, is scalable and effective for practical use for large-scale online course engineering. (3) *Evaluation*—eEPIPHANY outperforms past competitors, including *human experts*, on several, real online course datasets.

The goal of this paper is to introduce the eEPIPHANY method (section 3) and provide empirical evaluation for its effectiveness (section 4). We discuss implications for the application of eEPIPHANY to evidence-based online course refinement (section 5.3). To begin, the next section provides a standard structure of our target online courses and various definitions for later discussions.

## 2. SKILL MODEL FOR ONLINE COURSES

We assume that our target online courses have occasional low-stake assessments throughout the course—aka formative assessments—to assess students’ competency on target *skills*. We assume that each formative assessment has multiple *assessment items* (i.e., problems to answer), each of which is associated with one or more skills.

We assume that online courses have a pre-defined *skill map* (often called Q-matrix [9, 10]) that shows one-to-many mapping between individual skills and one or more assessment items. In this paper a mapping between a single skill and multiple assessment items in the skill map is called a *skill-item association*.

We call a set of skills a *skill model*. The terms “skill model” and “skill map” will be used interchangeably in this paper. The pre-defined skill model is therefore called the “*default*” *skill model*—a human-developed model that is initially guided by authors’ intuition in the absence of data, or a human-developed model that has been refined based on student data.

The Open Learning Initiative (OLI) at Carnegie Mellon University [11] is an example of an online course platform that meets the above-mentioned criteria [12]. OLI courses all have a *human-crafted* “default” skill model that is often recognized as semi-optimal, and could always be improved.

To improve skill models to refine online courses, it becomes crucial that the machine-discovered skill models have high interpretability so that course designers and developers can make sense of the proposed skill model improvements. Our proposed method, eEPIPHANY, discovers accurate and interpretable skill models from learning data and assessment item text data. The next section describes details of the eEPIPHANY method.

## 3. eEPIPHANY

eEPIPHANY is a collection of data mining techniques for automatic discovery of skill models from online course data. The primary input to eEPIPHANY is a matrix representing a chronological record of students’ responses to assessment items, called an A-matrix (Figure 6-a). The A-matrix is a three-dimensional matrix showing a history of attempts on individual assessment items made by individual students. Each attempt is a vector of binary values representing the correctness of a student’s response—0 indicates incorrect and 1 indicates correct. The A-matrix contains at most one correct response per student per assessment item.

The goal of eEPIPHANY is to find a skill model (Q-matrix) that produces the best prediction of the A-matrix. The predictive power is measured by cross-validation. eEPIPHANY can either find a Q-matrix by itself or refine a given Q-matrix by the following steps: (1) clustering assessment items with latent features that would best characterize the similarity in the difficulties of assessment items (section 3.1), (2) proposing a new skill model by assuming that the above-mentioned cluster of assessment items provides a hint for new skills (section 3.2), and

(3) searching for the best skill model by comparing multiple skill model candidates (section 3.3).

## 3.1 Feature Extraction

We have developed two latent-feature extraction strategies: (1) the Matrix Factorization (MF) strategy, and (2) the Bag-of-Words (BoW) strategy. The goal of feature extraction, regardless of the strategy difference, is to generate a two-dimensional matrix, the P-Matrix, showing a mapping between assessment items and “skill candidates” (Figure 6-d. Also see below).

### 3.1.1 Matrix factorization (MF) strategy

For the MF strategy, the A-matrix is first transformed into the difficulty matrix (D-matrix), which is a two-dimensional matrix representing an individual student’s difficulty for each assessment item. We hypothesize that the record of individual students’ performance on assessment items reflect their “difficulties” in answering assessment items, and that those students who show a similar distribution pattern of difficulties share a similar competency on latent skills.

The item difficulty  $id$ , by definition, is computed as  $id = 1 - 1/d$  where  $d$  is the number of attempts made on an assessment item. We only include attempts until the first correct attempt is made, i.e.,  $id$  is the length of the vector of attempts in the A-matrix (Figure 6-a). We hypothesize that students would more likely skip items that look too easy for them hence no difficulties at all. Therefore, we defined  $id$  as 0 for missing data in the A-Matrix (i.e., skipped items).

The D-matrix is then factorized into U and V matrices (i.e.,  $D = U \times V$ ) by the Non-Negative Matrix Factorization method [13]. The V-matrix is a two-dimensional (assessment item by latent feature) matrix. It is therefore a collection of *feature vectors*, each corresponding to an assessment item (Figure 6-b).

Assessment items in the V-matrix are then clustered by the k-means method [14], resulting in an F-matrix (Figure 6-c). We hypothesize that each cluster in the F-matrix represents a “skill candidate” that can be used to construct the P-Matrix (Figure 6-d).

The P-Matrix is a two-dimensional binary matrix showing which assessment item belongs to which skill candidate. The P-matrix represents the association of each assessment item to a skill candidate. By its nature, in the current eEPIPHANY algorithm, each assessment item has an association to at most one skill candidate (if any).

### 3.1.2 Bag-of-words (BoW) strategy

The BoW strategy creates the F-matrix directly from a collection of *item stems* (i.e., assessment item text data showing problem and feedback texts) for assessment items. That is, the assessment items are clustered by the bag-of-words method using item stems.

We first transform each assessment item into a set of component words from a collection of item stems using a part-of-speech tagger, TreeTagger<sup>1</sup>. We then apply the Latent Dirichlet Allocation model (LDA) [15] to cluster assessment items. Assessment items are clustered based on the probability of topic distribution—i.e., individual assessment items are assigned to the topic with the highest topic probability, which results into the F-Matrix from which the P-Matrix is generated as mentioned above.

<sup>1</sup> [www.cis.uni-muenchen.de/~schmid/tools/TreeTagger](http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger)

## 3.2 Skill Model Construction

eEPIPHANY refines a given “default” skill model by either modifying it or replacing it with a new skill model. In either case, eEPIPHANY first proposes candidates for new skills, and then finds the best way to refine the default skill model in terms of the accuracy of the data fit. This subsection describes the former step, whereas the latter step is described in section 3.3.

Given a P-matrix, there are three strategies to refine the “default” skill model: (1) Replacing the entire “default” skill model with an entirely new skill model, (2) appending new skill-item associations to the “default” skill model, (3) splitting given a skill-item association(s) in the “default” skill model into multiple skill-item associations.

### 3.2.1 Replace Strategy

To replace the default skill model with an entirely new skill model, the P-matrix is straightforwardly converted into the Q-matrix. Namely, each skill candidate becomes a new skill. Assessment items that are associated with the skill candidate become members of the skill-item association for the newly defined skill.

### 3.2.2 Append Strategy

The *append* strategy adds more skill-item associations to the default skill model, while the original skill-item associations in the default skill model remain intact. Skill-item associations that are being newly added are the same set of skill-item associations proposed by the *replace* strategy. The following example illustrates this process (Figure 1):

Assume that there is a skill-item association  $a_i$  for a skill  $s_i$  with assessment items  $q^i_1 \dots q^i_5$  in the default skill model. Also, assume that there is a skill candidate  $c_1$  and  $c_2$  in the P-matrix where  $c_1$  has a skill-item association with assessment items  $q^i_1, q^i_2$ , and  $q^i_3$ ; and  $c_2$  has a skill-item association with assessment items  $q^i_4$  and  $q^i_5$ . The *append* strategy enters two new skill-item associations, one for  $c_1$  and another one for  $c_2$  into the default skill model. As a consequence, the assessment item  $q^i_1$ , for example, is now associated with two skills,  $s_i$  and  $c_1$ .

It is worth noting that the skill model produced by the *replace* strategy is the proper subset of the skill model produced by the *append* strategy. The number of skills in the skill model produced by the *append* strategy is the sum of the number of skills in the default skill model and the number of skills in the skill model produced by the *replace* strategy.

### 3.2.3 Split Strategy

The *split* strategy refines the default skill model by individually splitting skill-item associations into multiple new skill-item associations. These splits are based on skill-item associations in

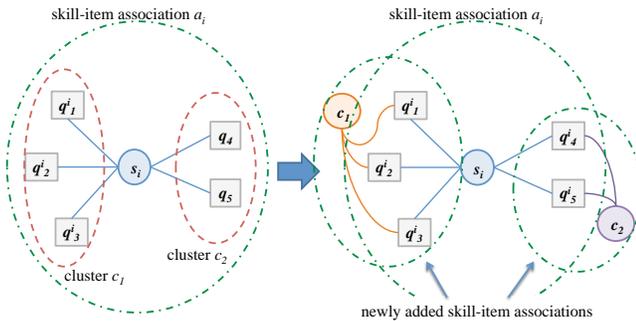


Figure 1. The *append* strategy appends new skill-item associations to the default skill model

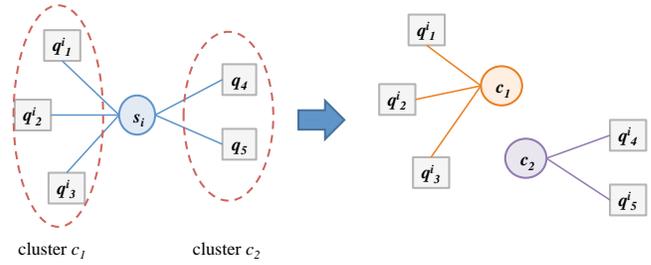


Figure 2. The *split* strategy breaks given skill-item associations into new ones with newly discovered skills

the P-Matrix. The following example illustrates this process (Figure 2):

Assume the same situation as mentioned above for the *append* strategy. That is, there is a skill-item association  $a_i$  for a skill  $s_i$  with assessment items  $q^i_1 \dots q^i_5$  in the default skill model. Also, assume that there is a skill candidate  $c_1$  and  $c_2$  in the P-matrix where  $c_1$  has a skill-item association with  $q^i_1, q^i_2$ , and  $q^i_3$ ; and  $c_2$  has a skill-item association with  $q^i_4$  and  $q^i_5$ . The *split* strategy then replaces the original skill-item association  $a_i$  with two new skill associations  $a_{i-1}$  and  $a_{i-2}$ , where  $a_{i-1}$  has  $c_1$  as a skill and  $q^i_1, q^i_2$ , and  $q^i_3$  as associated assessment-items, while  $a_{i-2}$  has  $c_2$  as a skill and  $q^i_4$  and  $q^i_5$  as associated assessment-items.

## 3.3 Model Search

We hypothesize that two different types of feature-extraction strategies (section 3.1) present pros and cons for our purposes. For example, the item stem (i.e., problem sentences and feedback messages) might reflect skills necessary to answer the assessment item correctly. On the other hand, the student response data might reflect skills that students have actually acquired. The BoW strategy might provide better interpretability, but the student response data might provide more accurate skill models. The BoW strategy can be applied even before the course has been used (i.e., before student data is available).

With the lack of a predictive theory of parameter selection to compute the best skill model, eEPIPHANY exhaustively searches for the best skill model by comparing all possible skill models with different combinations of the following four parameters. The comparison is done by the model fit using the Bayesian Knowledge Tracing as a predictor:

- (1) The number of components used for the Matrix Factorization ( $N_C$ )—This determines a dimension of the V-matrix.  $N_C$  reflects the variance in the pattern of student competency over the latent features. Although, the greater  $N_C$  value would result in the smaller reconstruction error (i.e.,  $\|D-U*V\|$ ), it might also result in the over fit to the data (which is penalized in the AIC and BIC scores).  $N_C$  varies from 10 to the number of students, increased by 10 during the model search.
- (2) The number of clusters in k-means ( $N_k$ )—We hypothesize that each feature is shared by at least five assessment items. Therefore,  $N_k$  varies from 25 to  $N_Q/5$  where  $N_Q$  is the number of assessment items; increased by 25 during the model search.
- (3) The number of topics used for LDA (section 3.1.2) to compute the bag-of-words clustering ( $N_T$ )—Here again, applying the same hypotheses as for  $N_k$ .  $N_T$  varies from 25 to  $N_Q/5$ , increased by 25 during the model search.
- (4) The threshold used for the split strategy ( $\beta$ )—Assume that skill  $s$  is associated with  $n$  assessment items,  $q_i \dots q_n$ . Also assume

that in the P-matrix, these  $n$  assessment items are associated with  $k$  skill candidates,  $C = \langle c_1, \dots, c_k \rangle$ . The skill-item association for  $s$  will be split into new skill-item associations with skill candidate  $c$  in  $C$ , if the number of assessment items associate with the skill candidate  $c$  is greater than  $n \times \beta$ .  $\beta$  is set to 0.05, 0.25, and 0.5 in this order during the model search.

### 3.4 Model Interpretation: The DoE Analysis

The most important goal of the skill-model discovery and refinement proposed in the current paper is to improve online courses. Providing *interpretable* feedback based on a machine-discovered skill model and model refinement is therefore crucial. We hypothesize that to achieve this goal, two subgoals must be met: (1) to identify what part of the default skill model has been improved the most, and (2) to understand the improvement from a domain perspective.

To identify the part of the skill model that has been improved most, we analyze the *degree of enhancement* (DoE) of the proposed change in skill models. We hypothesize that the DoE would be maximized among a skill(s) for which the accuracy of students' performance prediction improved the most [16]. The accuracy of student performance prediction is operationalized as the root mean squared error (RMSE) in cross-validation for the model-fit evaluation.

Based on this hypothesis, we identify skills with the most DoE in the default skill model  $M_D$  relative to a refined (i.e., machine-discovered) skill model  $M_R$  as follows:

- (1) For each skill  $s_i$  in the default skill model  $M_D$ , let  $I_D^i$  be a set of assessment items associated with  $s_i$ .
- (2) Find all skills  $c_j^i$  ( $j=1, \dots, n_i$ ) in the refined skill model  $M_R$  that are associated with any assessment items in  $I_D^i$ .
- (3) Compute  $xI_D^i$ , the extended version of  $I_D^i$ , by adding all assessment items associated with any of  $c_j^i$  to  $I_D^i$ .
- (4) Compute  $RMSE_{s_i}$  that is an RMSE in predicting student performance on assessment items in  $xI_D^i$  using corresponding  $s_i$  in  $M_D$  as the predictor.
- (5) Compute  $RMSE_{c_i}$  that is an RMSE in predicting student performance on assessment items in  $xI_D^i$  using corresponding  $c_j^i$  in  $M_R$  as a predictor.
- (6) Let  $d_i = RMSE_{s_i} - RMSE_{c_i}$  be the *DoE score* of skill  $s_i$  relative to  $c_j^i$ .
- (7) Find a skill  $s$  in  $M_D$  with the largest DoE score. The skill  $s$  has the largest error reduction from  $M_D$  to  $M_R$ .

Once the skill with the largest error reduction is found, the next step is to understand what the improvement is about, that is, to interpret the machine-discovered model refinement with the focus on the skill with the largest error reduction.

To interpret the proposed model refinement, we use the bag-of-words analysis in combination with manual inspection of the assessment item text. For each skill-item association in the refined skill model, a set of keywords is extracted from the item stem (i.e., the combination of text sentences for the items and their feedback messages). The  $\chi^2$  value is computed for individual word  $w$  appearing in the item stem for a skill-item association  $k$  as follows [17]:  $\chi^2(k, w) = (\text{aic}(k, w) - \text{aict}(k, w))^2 / \text{aict}(k, w)$  where  $\text{aic}(k, w)$  is the number of assessment items that contains  $w$  in  $k$ , and  $\text{aict}(k, w)$  is a theoretical implication for  $\text{aic}(k, w)$ , i.e.,  $\text{aict}(k, w) = \text{aic}(k, *) \times \text{aic}(*, w) / \text{aic}(*, *)$ . The word  $w$  is considered as a keyword only when  $\text{aict}(k, w) < \text{aic}(k, w)$ .

Table 1. Three OLI datasets used for the evaluation

	Statistics	Biology	C@CM
#Students	1,013	481	100
#Transactions	538,062	418,344	94,612
#Unique Items	1,791	916	912

## 4. EVALUATION

To evaluate the efficiency and effectiveness of the eEPIPHANY method, we applied it to actual online course data.

### 4.1 Data

Three OLI courses—Computing@CarnegieMellon (C@CM), Biology, and Statistics—were used for evaluation. All three courses are actively used at Carnegie Mellon University and other educational institutions for registered, academic students and in open sections for independent learners. Table 1 shows the number of students, transactions (i.e., students' responses to assessment items), and unique items; these datasets represent use in academic contexts. All these OLI data are available on DataShop [18]. It turned out that the C@CM data only contains randomly selected students' data from a larger pool of the OLI data that contains more than 1300 academic students enrolled.

### 4.2 Method

For each of the three OLI datasets, we applied eEPIPHANY and had it search the best skill model by finding the optimal clustering parameters (section 3.3). During the search we recorded the model-fit for three feature-extraction strategies (matrix factorization, bag-of-words, and their combination as described in section 3.1) crossed over three skill-model construction strategies (*split*, *add*, and *replace* as in section 3.2). The model-fit was computing by cross-validation using the Bayesian Knowledge Tracing technique.

### 4.3 Results

#### 4.3.1 Comparison of feature extraction and refinement strategies

Table 2 shows the best skill models, annotated with the strategies and parameters used to discover them. As the table shows, *the matrix factorization (MF) strategy always outperformed the BoW strategy for the three datasets used in the study*. When the MF strategy is used, *replacing the default skill model with a completely new skill model discovered by eEPIPHANY yielded the best skill model* for all dataset.

To understand how the size of cluster impacts the quality of the resultant skill model, we compared different skill models with different sizes measured as the number of skills. Figure 3 plots the

Table 2. eEPIPHANY always found better skill model than experts. FS: Feature Extraction Strategy, SC: Skill Construction Strategy, #S: Number of items

FS	SC	#S	AIC	BIC	RMSE
<b>Statistics</b>					
MF	Replace	63	307730	310731	0.447
BoW	Append	143	317808	323802	0.456
<b>Biology</b>					
MF	Replace	86	224944	228514	0.389
BoW	Split	187	228597	236360	0.393
<b>C@CM</b>					
MF	Replace	41	59497	60998	0.364
BoW	Split	137	61648	66661	0.371

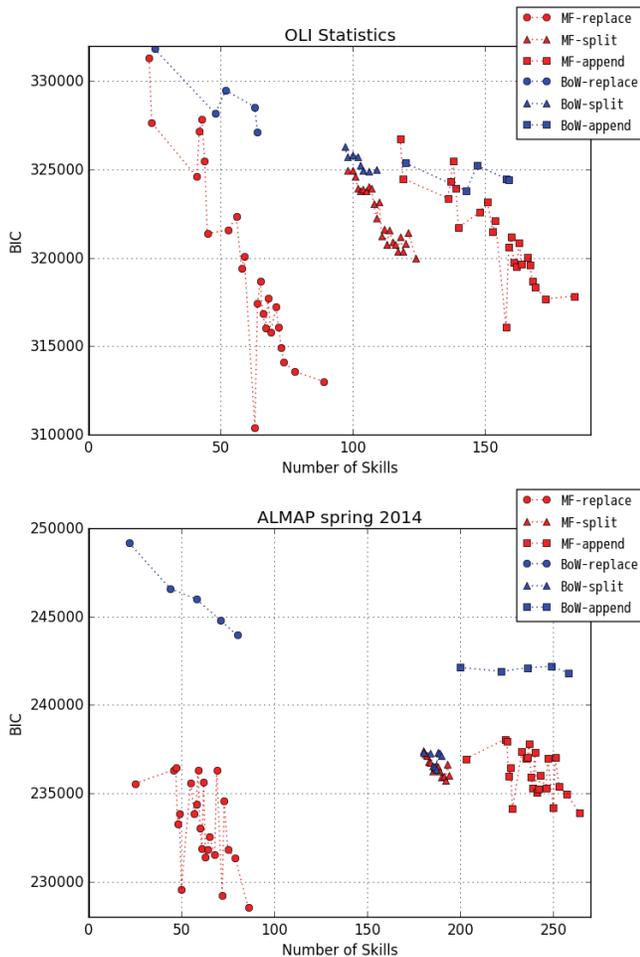


Figure 3. MF-replace wins or ties with MF-append: Comparison of skill models with different size. OLI Statistics (top) and Biology (bottom)

BIC (Y-axis) against a number of skills (X-axis). In the figure, two feature extraction strategies—MF and BoW—are crossed three skill-model construction strategies—replace, split, and append.

As the figure shows, it turned out that *for any strategy combination, the bigger the size of the model (i.e., the number of the clusters) the better the model*. It can be also seen that the *replace strategy is relatively better than other two skill-model construction strategies* (as depicted by more dots towards the bottom).

#### 4.3.2 Comparison with other methods

Table 3 shows the comparison of the model-fit between skill models discovered by LFA, an OLI course designer (OLI), and eEPIPHANY (eEPI) on the OLI Statistics course. In DataShop, skill models discovered by LFA and human expert only contain data from Unit 1. Therefore, for this analysis, we applied eEPIPHANY only to the OLI data from Unit 1.

The table shows the number of skills (#S) and the number of assessment items (Obs.). The model fit was evaluated by AIC, BIC, and RMSE scores computed by using Additive Factor Model (AFM) [19] and Bayesian Knowledge Tracing (BKT). As shown in the table, eEPIPHANY outperformed human expert (OLI),

Table 3. eEPIPHANY beats human expert on OLI Statistics. The analysis contains data only from Unit 1.

Method	#S	Obs.	AIC	BIC	RMSE
<b>AFM</b>					
eEPI	22	75955	72125	80901	0.412
LFA	28	75955	69108	77984	0.404
OLI	19	75955	74787	83507	0.418
<b>BKT</b>					
eEPI	22	75955	74560	75373	0.407
LFA	28	75955	74343	75378	0.404
OLI	19	75955	77405	78107	0.414

Table 4. Assessment items involved in the most beneficial skill model refinement

ID(Skill)	Assessment item (item stem)
Q881(c31)	The ability or tendency of organisms and cells to maintain stable internal conditions is called homeostasis (value:A) metabolism (value:B) evolution (value:C) emergent property (value:D)
Q885(c31)	Why do organisms maintain fairly steady conditions within their cells and bodies? They need to keep conditions stable so that they can obtain food. (value: A) Organisms just change along with whatever is happening in the outside world, which is usually quite steady. (value: B) They must maintain stable conditions to keep their enzymes working and generally to enable the chemical reactions of life. (value: C) Unstable conditions will destroy the DNA in cells; this is the most important risk for a cell facing physical or chemical stress. (value: D)
Q901(c31)	An organism or cell exhibits _____ when it maintains steady internal conditions despite changes in the outer environment. homeostasis (value: A) evolution (value: B) natural selection (value: C) balance (value: D)
Q717(c3)	Humans maintain a blood pH between 7.35 and 7.45. In order to maintain homeostasis, how will your body respond if your blood pH drops to 7.0? If your blood pH is 7.0, your body will raise your pH. (value: A) If your blood pH is 7.0, your body will lower your pH. (value: B) A blood pH of 7.0 is close enough to 7.35. Your body won't do anything. (value: C)

and arguably tied with LFA. We will further discuss this result in section 5.3.

#### 4.3.3 Model interpretation

**Figure 5** shows the skill *k153* with the largest DoE score (section 3.4) in the OLI Biology course. In the figure, the skill *k153* in the default skill model was associated with four assessment items. In the discovered skill model, these 4 assessment items are associated with two skills—*c31* and *c3*. The newly constructed skills *c31* and *c3* have 16 and 19 assessment items associated respectively. The RMSE is computed for those 35 steps using skills in the default skill model. The RMSE is then re-computed using *c31* and *c3*. According to the DoE analysis, splitting skill *k153* into two skills *c3* and *c31* yields the biggest DoE score. This addressed the first subgoal of the model interpretation.

To interpret model improvement, we investigated four assessment items associated with *k153* in the default skill model to see why they were split into two groups. Table 4 shows four assessment

Table 5. Bag of words for a skill (k153) split into two new skills (c31 and c3)

Skill	Bag of Words
k153	homeostasis range internal maintain steady condition narrow tendency metabolism raise optimal entity exhibit sensitive balance chemistry drop world despite happening
c31	steady homeostasis evolutionary stress valid theme progress favor module tree ancestor selection adapt internal evolution ancestry natural conclusion environmental whale
c3	hazy fundamentally matter space play concept structure yet mass nutrient exchange determine sometimes dramatically biology rule ability quite period peanut

items and their skill association in the refined skill model. Table 5 shows the bag-of-words associated with each skill cluster.

In the default skill model, the skill *k153* is to “Define homeostasis and explain its role in maintaining life.” All four assessment-items related to *k153* in the default skill model mention “homeostasis” and “sustainable life.” However, a closer look shows that this skill is most appropriate for the three out of four assessment items—Q881, Q885, and Q901. In the refined skill model, these three assessment items are correctly tagged as one skill *c31*.

Although the fourth assessment item Q717 relates to homeostasis, a closer look shows that learners are being asked to engage in a more sophisticated task—i.e., determine (or predict) necessary action to achieve homeostasis, which results in a separate association with skill *c3*.

For those four rows, the machine-generated split is very coherent from a subject-matter expert’s perspective. This satisfies the second subgoal of the model interpretation.

#### 4.3.4 Efficiency

One of the notable strengths of the eEPIPHANY method is its efficiency. As described in section 3.3, eEPIPHANY searches the best skill model by a brute-force search by merely changing the number of clusters, which takes linear time  $O(n)$ . This linear computation must be repeated nine times for three different feature-extraction strategies crossed with three different skill-model construction strategies, which still takes  $O(n)$ .

The Learning Factor Analysis (LFA) method [6] requires an intensive search for each skill ( $s$ ) over multiple difficulty factors ( $d$ ) that takes  $O(s^d)$ .

During the evaluation study that used three real OLI course data, eEPIPHANY found the best model in 2 to 3 hours per dataset running on a single-core personal computer, showing its practical potential for actual application to large-scale online course improvement.

## 5. DISCUSSION

### 5.1 Strategy comparison

Our study showed that using student response data (i.e., the number of attempts made on assessment items before a student finally made their first correct response) always yields a better skill model than using the bag-of-words with item stems. We also found that *even only using the bag-of-words, eEPIPHANY always yields a better skill model than the default skill model that is hand-crafted by human experts.*

As for the skill-model construction strategy, the *replace* strategy always discovers the best skill model in our study, suggesting that

*the Matrix Factorization strategy efficiently discovers a latent skill model from the student learning data.* On the other hand, the *split* strategy always resulted in producing an inferior skill model in our study; suggesting that *the split strategy hardly improves on the human-crafted skill.*

The above observation also implies that *eEPIPHANY can actually find a better skill model completely automatically without human interaction (which is what the replace strategy does) from real online course data.*

### 5.2 Interpretability

To interpret skill models proposed by the Matrix Factorization (MF) strategy is to interpret clusters of assessment items, which is often quite challenging. For the purpose of course refinement however, interpretability becomes crucial.

To overcome this issue, while still taking the advantage of the MF strategy to produce high-quality skill models, we applied the degree of enhancement (DoE) analysis to identify the instance of refinement that received the most benefit—i.e., identifying the skill that received the largest benefit from skill decomposition. We also combined the bag-of-words technique with manual inspection. Our study demonstrated that *this hybrid technique allows course designers to make meaningful interpretations of the proposed refinements of the skill model.*

Yet the obvious limitation of the current technique is its dependence on manual inspection. We hypothesize that one idea to overcome this issue is to combine MF and BoW, namely, to expand the V-matrix (Figure 6-b) by adding the bag-of-words keyword information as a latent feature, and then applying k-mean clustering. The resulting clusters (i.e., the skill candidates) would have better interpretability supported by the bag-of-words keyword information. Testing this hypothesis is an important future study.

### 5.3 Implication for evidence-based online course refinement

Our study demonstrated that eEPIPHANY discovers skill models that reflect student learning more accurately than human-crafted skill models on all three OLI course data. Even though eEPIPHANY requires human labor to interpret the discovered skill models (with the aid of DoE), it is arguably still less time consuming than creating skill models by hand. Figure 4 depicts this argument as a two-dimensional plot.

We also argue that eEPIPHANY is less labor intensive than LFA, because LFA requires human experts to generate the P-Matrix, which usually requires time-consuming cognitive task analysis. The high demand on human labor might not practical and hence might not scale up to apply to large online courses such as OLI. In fact, as far as we know, there has been no actual application of LFA with human-crafted P-Matrix to OLI courses. In the comparison in Table 3, the data for LFA is taken from DataShop [18], but LFA for these skill models used other existing skill models as P-Matrix (personal communication), therefore, it is not actually a fair comparison—LFA shows in this paper does not use the P-Matrix created by human experts. On the other hand, eEPIPHANY automatically discover the P-Matrix from data.

Nonetheless, as our study has shown, eEPIPHANY and LFA discovered equally accurate skill models. We also found that different evaluation criteria (i.e., AFM vs. BKT in Table 3) show different favors on different search algorithm. LFA uses AFM and eEPIPHANY uses BKT as a search bias, and that might have affected the results. We have yet to investigate this issue.

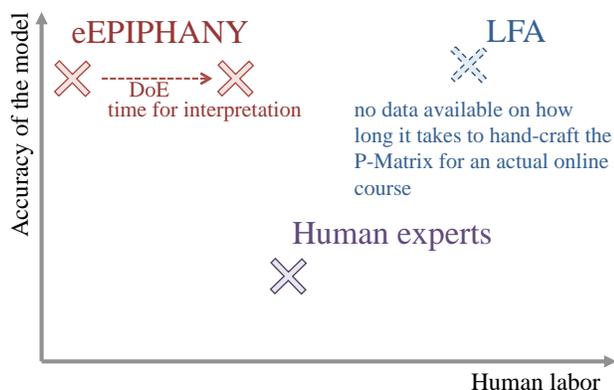


Figure 4. eEPIPHANY discovers skill models better than human experts and quicker than LFA

For our core goal—to provide evidence-based feedback for online course refinement—our study also suggests that eEPIPHANY can be used for a dual purposes with regard to skill model improvement: (1) When the online course is initially implemented, we should apply eEPIPHANY with the bag-of-words strategy. (2) When the online course is actually used and student learning data are collected, then we should apply eEPIPHANY with the student data to further improve the course.

The above observations further suggest that *authors of online courses would not need to create a default skill model at all—eEPIPHANY can find the default model by itself using the bag-of-words method*. This rather strong argument must be investigated as future research.

## 6. CONCLUSION

We found that eEPIPHANY is an efficient, practical, and quick method to automatically discover skill models from online course data without human interaction. Our empirical study showed that eEPIPHANY always finds skill models that are better than human-crafted skill models used in actual online courses. We also demonstrated that eEPIPHANY-crafted skill models have reasonable interpretability with the added help of the text analysis technique.

Creating effective online courses often requires intensive, iterative system engineering. Studying techniques for automatic skill model refinement and its application for evidence-based course refinement therefore is a critical research agenda for the successful future of online education.

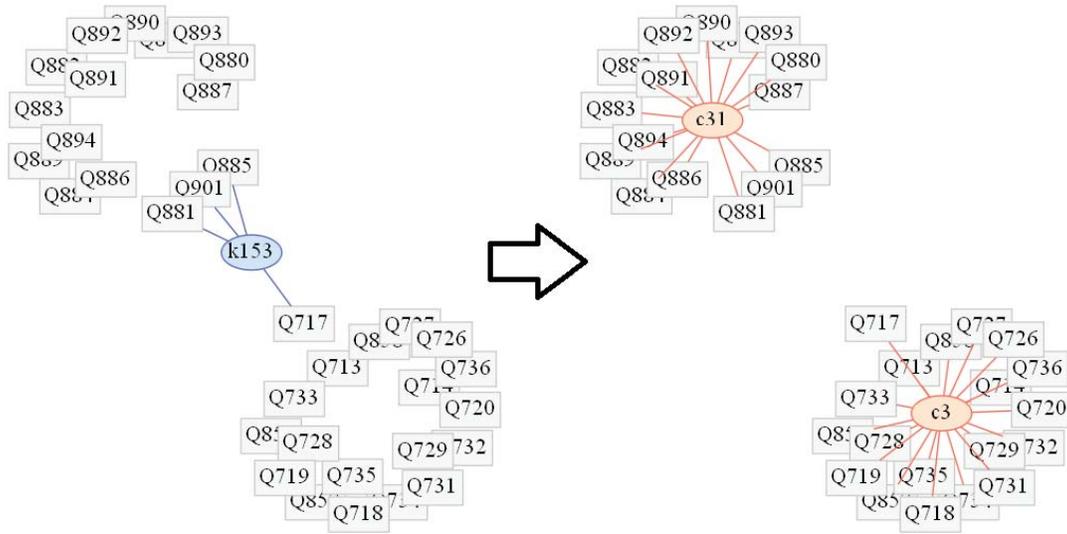
## ACKNOWLEDGEMENT

The research reported here was supported by National Science Foundation Awards No.1418244.

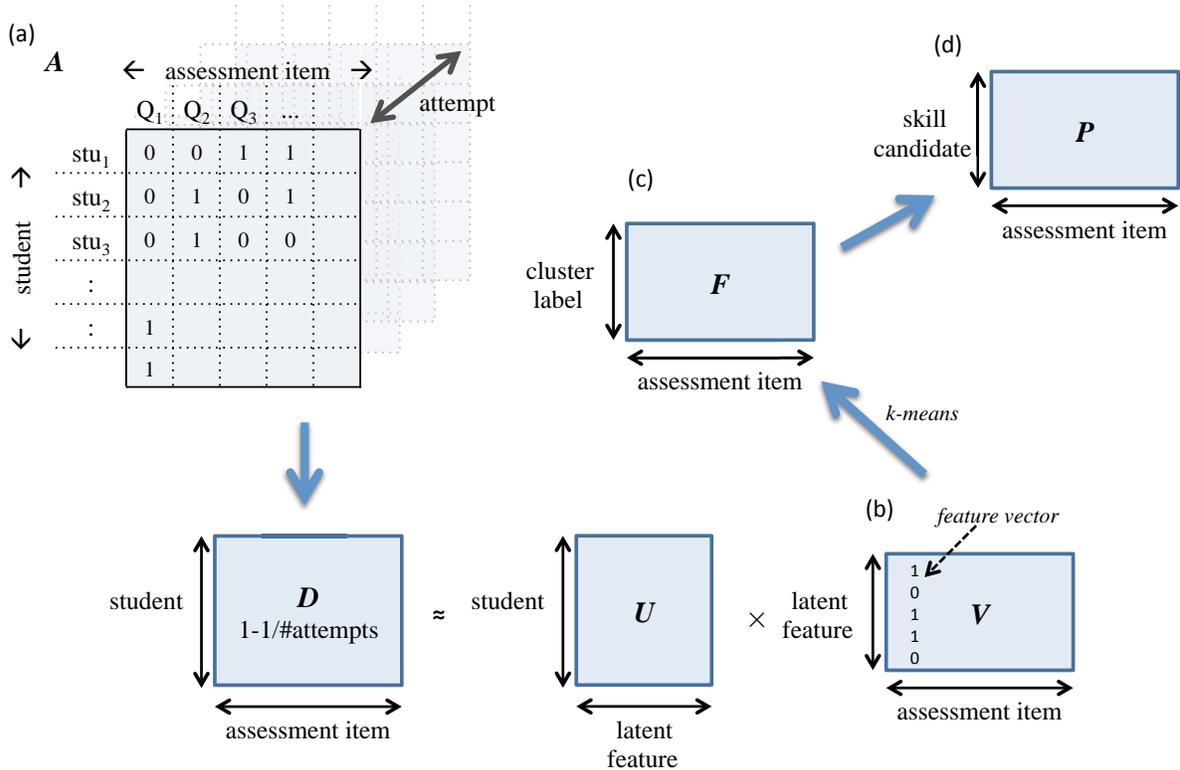
## 7. REFERENCES

1. Fishman, B., et al., *Creating a Framework for Research on Systemic Technology Innovations*. The Journal of the Learning Sciences, 2004. **13**(1): p. 43-76.
2. Stamper, J.C. and K.R. Koedinger, *Human-machine student model discovery and improvement using data*, in *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, S.B. G. Biswas, J. Kay, & A. Mitrovic, Editor. 2011, Springer: Berlin. p. 353-360.
3. Koedinger, K.R., et al., *Using Data-Driven Discovery of Better Student Models to Improve Student Learning*, in *Proceedings of the International Conference on Artificial*

4. Velmahos, G.C., et al., *Cognitive task analysis for teaching technical skills in an inanimate surgical skills laboratory*. Am. J. Surg, 2004. **18**: p. 114–119.
5. Koedinger, K.R. and E.A. McLaughlin, *Seeing language learning inside the math: Cognitive analysis yields transfer*, in *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, S. Ohlsson and R. Catrambone, Editors. 2010, Cognitive Science Society: Austin, TX.
6. Cen, H., K. Koedinger, and B. Junker, *Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement*, in *Intelligent Tutoring Systems*, M. Ikeda, K. Ashley, and T.-W. Chan, Editors. 2006, Springer Berlin Heidelberg. p. 164-175.
7. Desmarais, M.C., *Mapping Question Items to Skills with Non-negative Matrix Factorization*. SIGKDD Explor. NewsL., 2012. **13**(2): p. 30–36.
8. Sun, Y., et al., *Alternating Recursive Method for Q-matrix Learning*, in *Proceedings of the International Conference on Educational Data Mining*, J. Stamper, et al., Editors. 2014. p. 14-20.
9. Barnes, T., *The Q-matrix Method: Mining Student Response Data for Knowledge*, in *Proceedings of AAAI 2005 Educational Data Mining Workshop*. 2005.
10. Tatsuoka, C., et al., *Developing Workable Attributes for Psychometric Models Based on the Q-Matrix*. Journal for Research in Mathematics Education, accepted.
11. Lovett, M., O. Meyer, and C. Thille, *The Open Learning Initiative: Measuring the effectiveness of the OLI statistics course in accelerating student learning*. Journal of Interactive Media in Education, 2008.
12. Bier, N., R. Strader, and D. Zimmaro, *An Approach to Skill Mapping in Online Courses*, in *Learning with MOOCs2014*: Cambridge, MA.
13. Lee, D.D. and H.S. Seung, *Learning the parts of objects by non-negative matrix factorization*. Nature, 1999. **401**.
14. MacQueen, J., *Some methods for classification and analysis of multivariate observations*, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967.
15. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent Dirichlet Allocation*. The Journal of Machine Learning Research, 2003. **3**.
16. Koedinger, K.R., E.A. McLaughlin, and J.C. Stamper, *Automated student model improvement*, in *Proceedings of the 5th International Conference on Educational Data Mining*, K. Yacef, et al., Editors. 2012. p. 17-24.
17. Manning, C.D., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. 2008, New York, NY: Cambridge University Press.
18. Koedinger, K.R., et al., *A Data Repository for the EDM community: The PSLC DataShop*, in *Handbook of Educational Data Mining*, C. Romero, et al., Editors. 2010, CRC Press: Boca Raton, FL.
19. Cen, H., K.R. Koedinger, and B. Junker, *Is over practice necessary? – improving learning efficiency with the Cognitive Tutor through educational data mining*, in *Proceedings of 13th International Conference on Artificial Intelligence in Education*, R. Luckin, K.R. Koedinger, and J. Greer, Editors. 2007, IOS Press: Amsterdam. p. 511-- - 518.



**Figure 5. eEPIPHANY agrees with intuition: Assessment items are plotted in a skill-item association. (a) In the default skill model (left), skill  $k153$  are associated with assessment items Q881, Q885, and Q901 in the default skill model). (b) In the refined skill model (right), these three assessment items are associated with two skills ( $c3$  and  $c31$ ) among others. In the figure, those other skills plotted in the “default” skill model are the ones contained in  $\mathbf{xI}_D^i$  (section 3.4).**



**Figure 6. Overview of eEPIPHANY**

# Student Models for Prior Knowledge Estimation

Juraj Nižnan  
Masaryk University Brno  
niznan@mail.muni.cz

Radek Pelánek  
Masaryk University Brno  
xpelanek@mail.muni.cz

Jiří Řihák  
Masaryk University Brno  
thran@mail.muni.cz

## ABSTRACT

Intelligent behavior of adaptive educational systems is based on student models. Most research in student modeling focuses on student learning (acquisition of skills). We focus on prior knowledge, which gets much less attention in modeling and yet can be highly varied and have important consequences for the use of educational systems. We describe several models for prior knowledge estimation – the Elo rating system, its Bayesian extension, a hierarchical model, and a networked model (multivariate Elo). We evaluate their performance on data from application for learning geography, which is a typical case with highly varied prior knowledge. The result show that the basic Elo rating system provides good prediction accuracy. More complex models do improve predictions, but only slightly and their main purpose is in additional information about students and a domain.

## 1. INTRODUCTION

Computerized adaptive practice [14, 22] aims at providing students with practice in an adaptive way according to their skill, i.e., to provide students with tasks that are most useful to them. In this work we focus on the development of adaptive systems for learning of facts, particularly on modeling of prior knowledge of facts.

In student modeling [6] most attention is usually paid to modeling student learning (using models like Bayesian Knowledge Tracing [4] or Performance Factors Analysis [24]). Modeling of prior knowledge was also studied in prior work [22, 23], but it gets relatively little attention. It is, however, very important, particularly in areas where students are expected to have nontrivial and highly varying prior knowledge, e.g., in domains like geography, biology, human anatomy, or foreign language vocabulary. As a specific case study we use application for learning geography, which we developed in previous work [22]. The estimate of prior knowledge is used in models of current knowledge (learning), i.e., it has important impact on the ability of the practice system to ask suitable questions.

We consider several approaches to modeling prior knowledge and explore their trade-offs. The basic approach (described in previous work [22]) is based on a simplifying assumption of homogeneity among students and items. The model uses a global skill for students and a difficulty parameter for items; the prior knowledge of a student for a particular item is simply the difference between skill and difficulty. The model is basically the Rasch model, where the parameter fitting is done using a variant of the Elo rating system [9, 25] in order to be applicable in an online system.

The first extension is to capture the uncertainty in parameter estimates (student skill, item difficulty) by using Bayesian modeling. We propose and evaluate a particle based method for parameter estimation of the model. This approach is further extended to include multiplicative factors (as in collaborative filtering [15]) which allows to better model the heterogeneity among students and items.

The second extension is the hierarchical model which tries to capture more nuances of the domain by dividing items into disjoint subsets called concepts (or knowledge components). The model then computes student skill for each of these concepts. Since these concept skills are related, they are still connected by a global skill. With this model we have to choose an appropriate granularity of used concepts and find an assignment of items to these concepts. We use both manually determined concepts (e.g., “continents” in the case of geography) and concepts learned automatically from the data [19].

The third extension is a networked model, which bypasses the choice of concepts by modeling relations directly on the level of items. This model can be seen as a variation on previously proposed multivariate Elo system [7]. For each item we compute the most similar items (based on students’ answers), e.g., in the geography application, knowledge of Northern European countries is correlated. Prior knowledge of a student for a particular item is in this model estimated based on previous answers to similar items (still using the global skill to some degree).

Extended models are more detailed than the basic model and can potentially capture student knowledge more faithfully. They, however, contain more parameters and the parameter estimation is more susceptible to the noise in data. We compare the described models and analyze their performance on a large data set from application for learning geography [22].

The results show that the studied extensions do bring an improvement in predictive accuracy, but the basic Elo system is surprisingly good. The main point of extension is thus in their additional parameters, which bring an insight into the studied domain. We provide several specific examples of such insight.

## 2. MODELS

Although our focus is on modeling knowledge of facts, in the description of models we use the common general terminology used in student modeling, particularly the notions of *items* and *skills*. In the context of geography application (used for evaluation) items correspond to locations and names of places and skill corresponds to knowledge of these facts.

Our aim is to estimate the probability that a student  $s$  knows an item  $i$  based on previous answers of students  $s$  to questions about different items and previous answers of other students to questions about item  $i$ . As a simplification we use only the first answer about each item for each student.

In all models we use the logistic function  $\sigma(x) = \frac{1}{1+e^{-x}}$  as a link between a skill and a probability that a student answers correctly. In the case of multiple-choice questions the probability can be modeled by a shifted logistic function  $\sigma(x, k) = 1/k + (1 - 1/k) \frac{1}{1+e^{-x}}$ , where  $k$  is the number of options. We restrict our attention to online models (models that are updated after each answer). Such models can adapt to user behavior quickly and therefore are very useful in adaptive practice systems.

### 2.1 Basic Model

The basic model (described in previous work [22] and currently used in the online application) uses a key assumption that both students and studied facts are homogeneous. It assumes that students' prior knowledge in the domain can be modeled by a one-dimensional parameter.

We model the prior knowledge by the Rasch model, i.e., we have a student parameter  $\theta_s$  corresponding to the global knowledge of a student  $s$  of a domain and an item parameter  $d_i$  corresponding to the difficulty of an item  $i$ . The probability that the student answers correctly is estimated using a logistic function of a difference between the global skill and the difficulty:  $P(\text{correct}|\theta_s, d_i) = \sigma(\theta_s - d_i)$ .

A common approach to the parameter estimation for the Rasch model is joint maximum likelihood estimation (JMLE). This is an iterative approach that is slow for large data, particularly it is not suitable for an online application, where we need to adjust estimates of parameters continuously.

In previous work [22, 25] we have shown that the parameter estimation can be done effectively using a variant of the Elo rating system [9]. The Elo rating system was originally devised for chess rating, but we can use it in student modeling by interpreting a student's answer on an item as a "match" between the student and the item. The skill and difficulty estimates are updated as follows:

$$\begin{aligned}\theta_s &:= \theta_s + K \cdot (\text{correct} - P(\text{correct}|\theta_s, d_i)) \\ d_i &:= d_i + K \cdot (P(\text{correct}|\theta_s, d_i) - \text{correct})\end{aligned}$$

where *correct* denotes whether the question was answered correctly and  $K$  is a constant specifying sensitivity of the estimate to the last attempt. An intuitive improvement, which is used in most Elo extensions, is to use an "uncertainty function" instead of a constant  $K$  – the update should get smaller as we have more data about a student or an item. We use an uncertainty function  $U(n) = \alpha/(1 + \beta n)$ , where  $n$  is the number of previous updates to the estimated parameters and  $\alpha, \beta$  are meta-parameters.

### 2.2 Bayesian Model

In the basic model the uncertainty is modeled as a simple function of number of attempts. Such an approach is a simplification since some answers are more informative than others and thus the effect of answers on reduction of uncertainty should be differentiated. This can be done by using a Bayesian modeling approach. For this model we treat  $\theta_s, d_i$  and *correct* as random variables. We can use Bayes' theorem for updating our beliefs about skills and difficulties:

$$P(\theta_s, d_i|\text{correct}) \propto P(\text{correct}|\theta_s, d_i) \cdot P(\theta_s, d_i)$$

We assume that the difficulty of an item is independent of a skill of a student and thus  $P(\theta_s, d_i) = P(\theta_s) \cdot P(d_i)$ . The updated beliefs can be expressed as marginals of the conditional distribution, for example:

$$P(\theta_s|\text{correct}) \propto P(\theta_s) \cdot \int_{-\infty}^{\infty} P(\text{correct}|\theta_s, d_i = y) \cdot P(d_i = y) dy$$

In the context of rating systems for games, the basic Elo system has been extended in this direction, particularly in the Glicko system [11]. It models prior skill by a normal distribution and uses numerical approximation to represent the posterior by a normal distribution and to perform the update of the mean and standard deviation of the skill distribution using a closed form expressions. Another Bayesian extension is TrueSkill [12], which further extends the system to allow team competitions.

This approach is, however, difficult to modify for new situations, e.g., in our case we want to use the shifted logistic function (for modeling answers to multiple-choice questions), which significantly complicates derivation of equations for numerical approximation. Therefore, we use a more flexible particle based method to represent the skill distribution. The skill is represented by a skill vector  $\theta_s$ , which gives the values of skill particles, and probability vector  $\mathbf{p}_s$ , which gives the probabilities of the skill particles (sums to 1). The item difficulty is represented analogically by a difficulty vector  $\mathbf{d}_i$  and a probability vector  $\mathbf{p}_i$ . In the following text the notation  $\mathbf{p}_{s_k}$  stands for the  $k$ -th element of the vector  $\mathbf{p}_s$ .

The skill and difficulty vectors are initialized to contain values that are spread evenly in a specific interval around zero. The probability vectors are initialized to proportionally reflect the probabilities of the particles in the selected prior distribution. During updates, only the probability vectors change, the vectors that contain the values of the particles stay fixed. Particles are updated as follows:

$$\begin{aligned}\mathbf{p}_{s_k} &:= \mathbf{p}_{s_k} \cdot \sum_{l=1}^n P(\text{correct}|\theta_s = \theta_{s_k}, d_i = \mathbf{d}_{i_l}) \cdot \mathbf{p}_{i_l} \\ \mathbf{p}_{i_l} &:= \mathbf{p}_{i_l} \cdot \sum_{k=1}^n P(\text{correct}|\theta_s = \theta_{s_k}, d_i = \mathbf{d}_{i_l}) \cdot \mathbf{p}_{s_k}\end{aligned}$$

After the update, we must normalize the probability vectors so that they sum to one. A reasonable simplification that avoids summing over the particle values is:

$$\begin{aligned} \mathbf{p}_{s_k} &:= \mathbf{p}_{s_k} \cdot P(\text{correct}|\theta_s = \theta_{s_k}, d_i = E[\mathbf{d}_i]) \\ \mathbf{p}_{i_l} &:= \mathbf{p}_{i_l} \cdot P(\text{correct}|\theta_s = E[\theta_s], d_i = \mathbf{d}_{il}) \end{aligned}$$

where  $E[\mathbf{d}_i]$  ( $E[\theta_s]$ ) is the expected difficulty (skill) particle value (i.e.  $E[\mathbf{d}_i] = \mathbf{d}_i^T \cdot \mathbf{p}_i$ ). By setting the number of particles we can trade off between precision on one hand and speed and memory requirements on the other hand.

Using the described particle model in a real-world application would require storing the probabilities for all the particles in a database. If we assume that our beliefs stay normal-like even after many observations then we can approximate each of the posteriors by a normal distribution. This approach is called assumed-density filtering [17]. Consequently, each posterior can be represented by just two numbers, the mean and the standard deviation. In this simplified model, each update requires the generation of new particles. We generate the particles in the interval  $(\mu - 6\sigma, \mu + 6\sigma)$ . Otherwise, the update stays the same as before. After the update is performed, the mean and the standard deviation are estimated in a standard way:  $\mu_{\theta_s} := \theta_s^T \cdot \mathbf{p}_s$ ,  $\sigma_{\theta_s} := \|\theta_s - \mu_{\theta_s}\|_2$ .

The model can be extended to include multiplicative factors for items ( $q_i$ ) and students ( $r_s$ ), similarly to the Q-matrix method [1] or collaborative filtering [15]. Let  $k$  be the number of factors, then  $x$  passed in to the likelihood function  $\sigma(x)$  has the form:  $x = \theta_s - d_i + \sum_{j=1}^k q_{i,j} \cdot r_{s,j}$ . The updates are similar, we only need to track more variables.

### 2.3 Hierarchical Model

In the next model, which we call ‘hierarchical’, we try to capture the domain in more detail by relaxing the assumption of homogeneity. Items are divided into disjoint sets – usually called ‘concepts’ or ‘knowledge components’ (e.g., states into continents). In addition to the global skill  $\theta_s$  the model now uses also the concept skill  $\theta_{sc}$ . We use an extension of the Elo system to estimate the model parameters. Predictions are done in the same way as in the basic Elo system, we just correct the global skill by the concept skill:  $P(\text{correct}|\theta_s, \theta_{sc}, d_i) = \sigma((\theta_s + \theta_{sc}) - d_i)$ . The update of parameters is also analogical ( $U$  is the uncertainty function and  $\gamma$  is a meta-parameter specifying sensitivity of the model to concepts):

$$\begin{aligned} \theta_s &:= \theta_s + U(n_s) \cdot (\text{correct} - P(\text{correct}|\theta_s, \theta_{sc}, d_i)) \\ \theta_{sc} &:= \theta_{sc} + \gamma \cdot U(n_{sc}) \cdot (\text{correct} - P(\text{correct}|\theta_s, \theta_{sc}, d_i)) \\ d_i &:= d_i + U(n_i) \cdot (P(\text{correct}|\theta_s, \theta_{sc}, d_i) - \text{correct}) \end{aligned}$$

This proposed model is related to several student modeling approaches. It can be viewed as a simplified Bayesian network model [3, 13, 16]. In a proper Bayesian network model we would model skills by a probability distribution and update the estimates using Bayes rule; equations in our model correspond to a simplification of this computation using only point skill estimates. Bayesian network model can also model more complex relationships (e.g., prerequisites), which are not necessary for our case (fact learning). Other related modeling approaches are the Q-matrix method [1], which focuses on modeling mapping between skills and items

(mainly using  $N : M$  relations), and models based on knowledge space theory [8]. Both these approaches are more complex than the proposed model. Our aim here is to evaluate whether even a simple concept based model is sensible for modeling factual knowledge.

The advantage of the hierarchical model is that user skill is represented in more detail and the model is thus less sensitive to the assumption of homogeneity among students. However, to use the hierarchical model, we need to determine concepts (mapping of items into groups). This can be done in several ways. Concepts may be specified manually by a domain expert. In the case of geography learning application some groupings are natural (continents, cities). In other cases the construction of concepts is more difficult, e.g., in the case of foreign language vocabulary it is not clear how to determine coherent groups of words. It is also possible to create concepts automatically or to refine expert provided concepts with the use of machine learning techniques [5, 19].

To determine concepts automatically it is possible use classical clustering methods. For our experiments we used spectral clustering method [27] with similarity of items  $i, j$  defined as a Spearman’s correlation coefficient  $c_{ij}$  of correctness of answers (represented as 0 or 1) of shared students  $s$  (those who answered both items). To take into account the use of multiple-choice questions we decrease the binary representation of a response  $r$  by guess factor to  $r - 1/k$  ( $k$  is the number of options). Disadvantages of the automatic concept construction are unknown number of concept, which is a next parameter to fit, and the fact that found concepts are difficult to interpret.

It is also possible to combine the manual and the automatic construction of concepts [19]. With this approach the manually constructed concepts are used as item labels. Items with these labels are used as a training set of a supervised learning method (we used logistic regression with regularization). For the item  $i$ , the vector of correlation with all items  $c_{ij}$  is used as vector of features. Errors of the used classification method are interpreted as ‘corrected’ labels; see [19, 20] for more details.

### 2.4 Networked Model

The hierarchical model enforces hard division of items into groups. With the next model we bypass this division by modeling directly relations among individual items, i.e., we treat items as a network (and hence the name ‘networked model’). For each item we have a local skill  $\theta_{si}$ . For each pair of items we compute the degree to which they are correlated  $c_{ij}$ . This is done from training data or – in the real system – once a certain number of answers is collected. After the answer to the item  $i$  all skill estimates for all other items  $j$  are updated based on  $c_{ij}$ . The model still uses the global skill  $\theta_s$  and makes the final prediction based on the weighted combination of global and local skill:  $P(\text{correct}|\theta_s, \theta_{si}) = \sigma(w_1\theta_s + w_2\theta_{si} - d_i)$ . Parameters are updated as follows:

$$\begin{aligned} \theta_s &:= \theta_s + U(n_s) \cdot (\text{correct} - P(\text{correct}|\theta_s, \theta_{si})) \\ \theta_{sj} &:= \theta_{sj} + c_{ij} \cdot U(n_s) \cdot (\text{correct} - P(\text{correct}|\theta_s, \theta_{si})) \\ &\quad \text{for all items } j \\ d_i &:= d_i + U(n_i) \cdot (P(\text{correct}|\theta_s, \theta_{si}) - \text{correct}) \end{aligned}$$

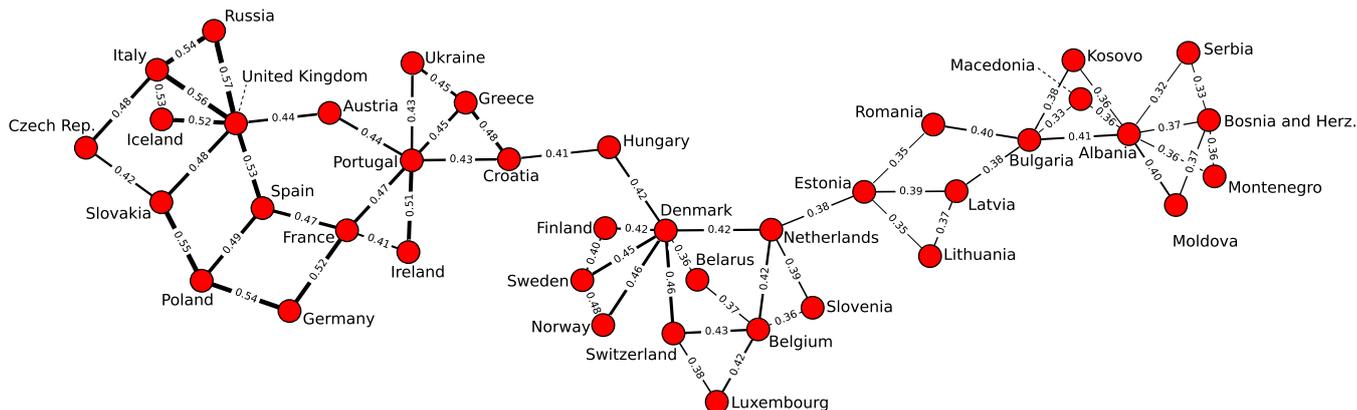


Figure 1: Illustration of the networked model on European countries. Only the most important edges for each country are shown.

This model is closely related to multivariate Elo which was previously proposed in the context of adaptive psychometric experiments [7].

For illustration of the model, Figure 1 shows selection of the most important correlations for European countries. Note that this automatically generated figure contains some natural clusters as Balkan countries (right), Scandinavian countries (middle), and well-known<sup>1</sup> countries (left).

### 3. EVALUATION

We provide evaluation of the above described models over data from an adaptive application for learning facts.

#### 3.1 The Used System and Data

For the analysis we use data from an online adaptive system `slpemapy.cz` for practice of geography facts (e.g., names and location of countries, cities, mountains). The system estimates student knowledge and based on this estimate it adaptively selects questions of suitable difficulty [22]. The system uses a target success rate (e.g., 75 %) and adaptively selects questions in such a way that the students' achieved performance is close to this target [21]. The system uses open questions ("Where is France?") and multiple-choice questions ("What is the name of the highlighted country?") with 2 to 6 options. Students answer questions with the use of an interactive 'outline map'. Students can also access a visualization of their knowledge using an open learner model.

Our aim is to model prior knowledge (not learning during the use of the system), so we selected only the first answers of students to every item. The used data set contains more than 1.8 million answers of 43 thousand students. The system was originally available only in Czech, currently it is available in Czech, English, and Spanish, but students are still mostly from Czech republic (> 85%) and Slovakia (> 10%). The data set was split into train set (30%) and test set (70%) in a student-stratified manner. As a primary metric for model comparison and parameter fitting we use root mean square error (RMSE), since the application works with absolute values of predictions [22] (see [26] for more details on choice of a metric).

<sup>1</sup>By students using our system.

#### 3.2 Model Parameters

The train set was used for finding the values of the meta-parameters of individual models. Grid search was used to search the best parameters of the uncertainty function  $U(n)$ . Left part of Figure 2 shows RMSE of the basic Elo model on training data for various choices of  $\alpha$  and  $\beta$ . We chose  $\alpha = 1$  and  $\beta = 0.06$  and we used these values also for derived models which use the uncertainty function.

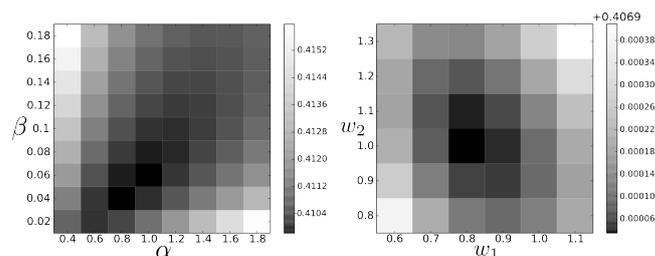


Figure 2: Grid searches for the best uncertainty function parameters  $\alpha, \beta$  (left) and the best parameters  $w_1, w_2$  of the networked model (right). As can be seen from different scales, models are more sensitive to  $\alpha$  and  $\beta$  parameters.

Grid search (Figure 2 right) was used also to find the best parameters  $w_1 = 0.8, w_2 = 1$  of the networked model. The train set was also used for computation of correlations. To avoid spurious high correlations of two items  $i, j$  as consequence of lack of common students we set all  $c_{ij} = 0$  for those pairs  $i, j$  with less than 200 common students. Correlations computed by this method show stability with respect to selection of train set. For two different randomly selected train sets correlation values correlate well (> 0.95). As Figure 1 shows, the resulting correlations are interpretable.

For the particle-based Bayesian model we can tune the performance by setting the number of particles it uses for estimating each distribution. We found out that increasing the number of particles beyond 100 does not increase performance. For the simplified version, only 10 particles are sufficient. This is probably due to the way the algorithm uses the particles (they are discarded after each step).

**Table 1: Comparison of models on the test set.**

Model	RMSE	LL	AUC
Elo ( $\alpha = 1, \beta = 0.06$ )	0.4076	-643179	0.7479
Bayesian model	0.4080	-644362	0.7466
Bayesian model (3 skills)	0.4056	-637576	0.7533
Hierarchical model	0.4053	-636630	0.7552
Networked model	0.4053	-636407	0.7552

### 3.3 Accuracy of Predictions

All the reported models work online. Training of models (parameters  $\theta_s, d_i$ ) continues on the test set but only predictions on this set are used to evaluate models.

Table 1 shows results of model comparison with respect to model performance metrics. In addition to RMSE we also report log-likelihood (LL) and area under the ROC curve (AUC); the main results are not dependent on the choice of metric. In fact, predictions for individual answers are highly correlated. For example for the basic Elo model and hierarchical model most of the predictions (95%) differ by less than 0.1.

The hierarchical model reported in Table 1 uses manually determined concepts based on both location (e.g., continent) and type of place (e.g., country). Both the hierarchical model and the networked model bring an improvement over the basic Elo model. The improvement is statistically significant (as determined by a t-test over results of repeated cross-validation), but it is rather small. Curiously, the Particle Bayes model is slightly worse than the simple Elo system, i.e., the more involved modeling of uncertainty does not improve predictions. The performance improves only when we use the multiple skill extension. We hypothesize that the improvement of the hierarchical (resp. multiple skill) extensions model be more significant for less homogeneous populations of students. Each skill could then be used to represent a different prior knowledge group.

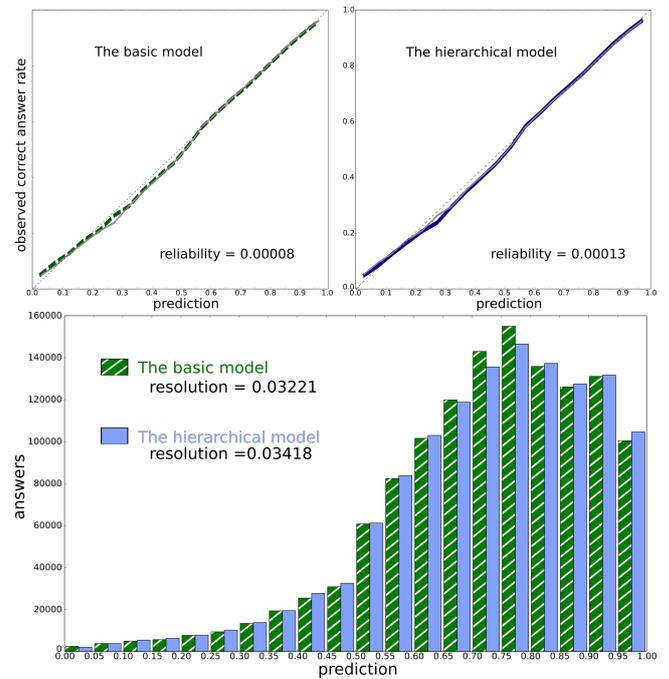
RMSE is closely related to Brier score [26], which provides decomposition [18] to uncertainty (measures the inherent uncertainty in the observed data), reliability (measures how close the predictions are to the true probabilities) and resolution (measures how diverse the predictions are).

This decomposition can be also illustrated graphically. Figure 3 shows comparison of the basic Elo model and the hierarchical model. Both calibration lines (which are near the optimal one) reflect very good reliability. On the other hand, histograms reflect the fact that the hierarchical model gives more divergent predictions and thus has better resolution.

### 3.4 Using Models for Insight

In student modeling we are interested not just in predictions, but also in getting insight into characteristics of the domain or student learning. The advantage of more complex models may lie in additional parameters, which bring or improve such insight.

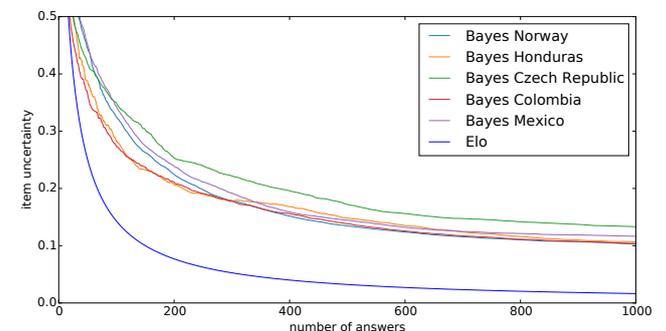
Figure 5 gives comparison of item difficulty for Elo model



**Figure 3: Illustration of the Brier score decomposition for the basic model and the hierarchical model. Top: reliability (calibration curves). Bottom: resolution (histograms of predicted values).**

and Particle Bayes. As we can see, the estimated values of the difficulties are quite similar. The main difference between these models is in estimates of uncertainty. The uncertainty function used in Elo converges to zero faster and its shape is the same for all items. In Particle Bayes, the uncertainty is represented by the standard deviation of the normal distribution. This uncertainty can decrease differently for each item, depending on the amount of surprising evidence the algorithm receives, as is shown in Figure 4. The better grasp of uncertainty can be useful for visualization in an open learner model [2, 10].

Other extensions (networked, hierarchical, Bayesian with multiple skills) bring insight into the domain thanks to the analysis of relations between items, e.g., by identifying most



**Figure 4: Evolution of uncertainties in the Bayes model and Elo.**

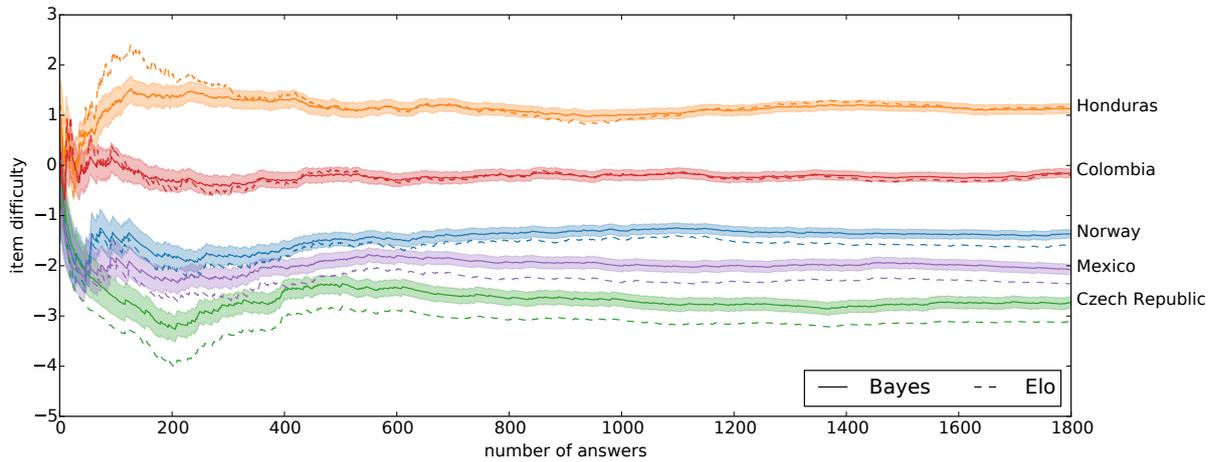


Figure 5: Difficulty of countries – the basic Elo model versus the Bayes model.

useful clusters of items. Such results can be used for improving the behavior of an adaptive educational system. For example, the system can let the user practice items from one concept and after reaching mastery move to the next one. Another possible use of concepts is for automatic construction of multiple-choice questions with good distractors (falling under the same concept).

We performed evaluation of the hierarchical model with different concepts. We used several approaches for specifying the concepts manually: based on type (e.g., countries, cities, rivers), location (e.g., Europe, Africa, Asia) and combination of the two approaches (e.g, European countries, European cities, African countries). Since we have most students' answers for European countries, we also considered a data set containing only answers on European countries. For this data set we used two sets of concepts. The first is the partition to Eastern, Western, Northwestern, Southern, Central and Southeastern Europe, the second concept set is obtained from the first one by union of Central, Western and Southern Europe (countries from these regions are mostly well-known by our Czech students) and union of Southeastern and Eastern Europe.

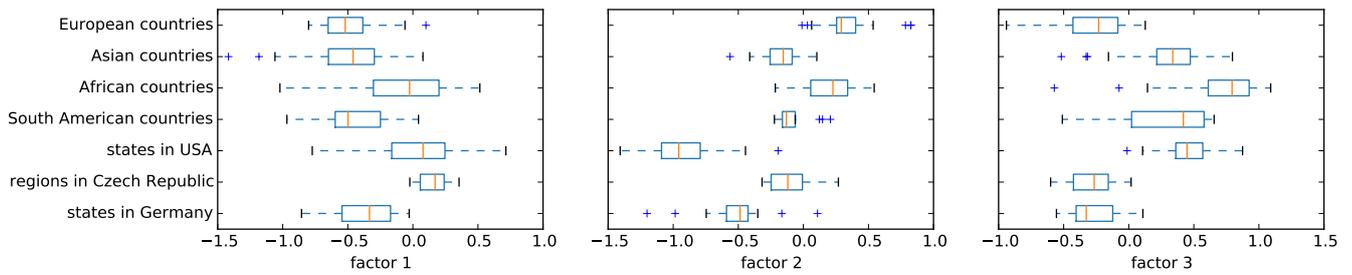
We compared these manually specified concepts with automatically corrected and entirely automatically constructed concepts (as described in Section 2.3; 'corrected' concepts are based on manually specified concepts and are revised based on the data). The quality of concepts was evaluated using prediction accuracy of the hierarchical model which uses these concepts. Table 2 shows the results expressed as RMSE improvement over the basic model. Note that the differences in RMSE are necessarily small, since the used models are very similar and differ only in the allocation of items to concepts. For the whole data set (1368 items) a larger number of concepts brings improvement of performance. The best results are achieved by manually specified concepts (combination of location and type of place), automatic correction does not lead to significantly different performance. For the smaller data set of European countries (39 items) a larger number of (both manual and automatically determined) concepts brings worse performance – a

model with too small concepts suffers from a loss of information. In this case the best result is achieved by a correction of manually specified concepts. The analysis shows that the corrections make intuitive sense, most of them are shifts of well-known and easily recognizable countries as Russia or Iceland to block of well-known countries (union of Central, Western and Southern Europe).

Table 2: Comparison of manual, automatically corrected manual, and automatic concepts. Quality of concepts is expressed as RMSE improvement of the hierarchical model with these concepts over the basic model.

	number of concepts	RMSE improvement
All items		
manual – type	14	0.00132
corrected – type	14	0.00132
manual – location	22	0.00179
corrected – location	22	0.00167
<b>manual – combination</b>	<b>56</b>	<b>0.00235</b>
corrected – combination	56	0.00234
automatic	5	–0.00025
automatic	20	0.00039
automatic	50	0.00057
Europe		
manual	3	0.00003
<b>corrected</b>	<b>3</b>	<b>0.00011</b>
manual	6	–0.00015
corrected	6	0.00003
automatic	2	0.00007
automatic	3	0.00004
automatic	5	–0.00019

Models with multiple skills bring some additional information not just about the domain, but also about students. Correlation of concept skills with the global skill range from -0.1 to 0.5; the most correlated concepts are the ones with large number of answers like European countries (0.48) or



**Figure 6: Boxplots of the item factor values from the Bayesian model (3 factors) grouped by some manually created concepts.**

Asian countries (0.4), since answers on items in these concepts have also large influence on the global skill. Correlation between two clusters skills typically range from -0.1 to 0.1. These low correlation values suggest that concept skills hold interesting additional information about student knowledge.

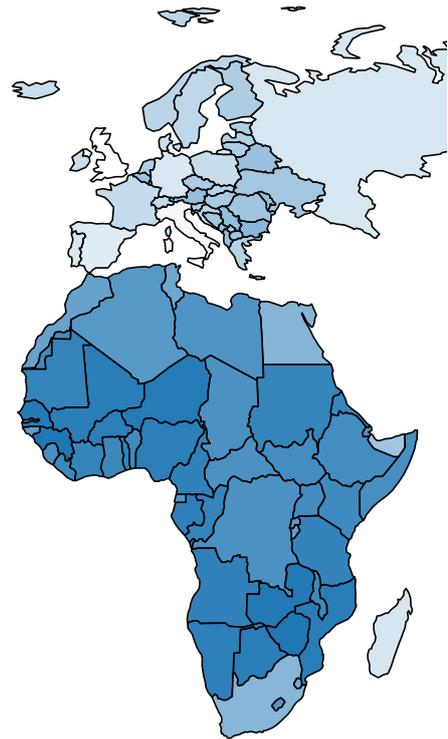
Another view of relations between items is provided by the Bayesian model with multiplicative factors – this model does not provide division of items into disjoint sets, but rather determines for each item a strength of its relation to each factor (based on the data). Figure 6 illustrates how the learned factors relate to some manually specified concepts. Note that the results in Table 1 suggest that most of the improvement in predictive accuracy can be achieved by just these three automatically constructed factors. We can see that *factor 3* discriminates well between countries in Europe and Africa (Figure 7 provides a more detailed visualization). In the case of geography the division of items to concepts can be done in rather natural way and thus the potential application of such automatically determined division is limited and serves mainly as a verification of the method. For other domains (e.g., vocabulary learning) such natural division may not exist and this kind of model output can be very useful.

Also, note that *Factor 2* differentiates between states in USA and countries on other continents and *Factors 1* and *2* have different values for regions in Czech republic and states in Germany. This evidence supports an assumption that the model may be able to recognize students with varied background.

#### 4. DISCUSSION

We have described and compared several student models of prior knowledge. The models were evaluated over extensive data from application for learning geography. The described models should be directly applicable to other online systems for learning facts, e.g., in areas like biology, human anatomy, or foreign language vocabulary. For application in domains which require deeper understanding (e.g., mathematics, physics) it may be necessary to develop extensions of described models (e.g., to capture prerequisite relations among concepts).

The results show that if we are concerned only with the accuracy of predictions, the basic Elo model is a reasonable choice. More complex models do improve predictions in statistically significant way, but the improvement is relatively



**Figure 7: Visualization of the values of the third factor in the Bayesian model with multiple skills.**

small and evenly spread (i.e., individual predictions by different models are very similar).

The improvement in predictions by the hierarchical or networked models may be more pronounced in less homogeneous domains or with less homogeneous populations. Nevertheless, if the main aim of a student model is prediction of future answers (e.g., applied for selection of question), then the basic Elo model seems to be sufficient. Its performance is good and it is very simple to apply. Thus, we believe that it should be used more often both in implementations of educational software and in evaluations of student models.

The more complex models may still be useful, since improved accuracy is not the only purpose of student models. Described models have interpretable parameters – assignment of items to concepts and better quantification of uncertainty

of estimates of knowledge and difficulty. These parameters may be useful by themselves. We can use them to guide the adaptive behavior of educational systems, e.g., the choice of questions can be done in such a way that it respects the determined concepts or at the beginning of the session we can prefer items with low uncertainty (to have high confidence in choosing items with appropriate difficulty). The uncertainty parameter is useful for visualization of student knowledge in open learner models [2, 10]. Automatically determined concepts may also provide useful feedback to system developers, as they suggest potential improvements in user interface, and also to teachers for whom they offer insight into student’s (mis)understanding of target domain. Given the small differences in predictive accuracy, future research into extensions of basic models should probably focus on these potential applications.

## 5. REFERENCES

- [1] Tiffany Barnes. The q-matrix method: Mining student response data for knowledge. In *Educational Data Mining*, 2005.
- [2] Susan Bull. Supporting learning with open learner models. In *Information and Communication Technologies in Education*, 2004.
- [3] Cristina Conati, Abigail Gertner, and Kurt Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction*, 12(4):371–417, 2002.
- [4] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [5] Michel C Desmarais, Behzad Beheshti, and Rhouma Naceur. Item to skills mapping: deriving a conjunctive q-matrix from data. In *Intelligent Tutoring Systems*, pages 454–463. Springer, 2012.
- [6] Michel C Desmarais and Ryan SJ d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.
- [7] Philipp Doebler, Mohsen Alavash, and Carsten Giessing. Adaptive experiments with a multivariate elo-type algorithm. *Behavior Research Methods*, pages 1–11, 2014.
- [8] Jean-Paul Doignon and Jean-Claude Falmagne. *Knowledge spaces*. Springer, 1999.
- [9] Arpad E Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.
- [10] Carrie Demmans Epp, Susan Bull, and Matthew D Johnson. Visualising uncertainty for open learner model users. 2014. to appear.
- [11] Mark E Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.
- [12] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2006.
- [13] Tanja Käser, Severin Klingler, Alexander Gerhard Schwing, and Markus Gross. Beyond knowledge tracing: Modeling skill topologies with bayesian networks. In *Intelligent Tutoring Systems*, pages 188–198. Springer, 2014.
- [14] S Klinckenberg, M Straatemeier, and HLJ Van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.
- [15] Yehuda Koren and Robert Bell. Advances in collaborative filtering. *Recommender Systems Handbook*, pages 145–186, 2011.
- [16] Eva Millán, Tomasz Loboda, and Jose Luis Pérez-de-la Cruz. Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683, 2010.
- [17] Thomas P Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.
- [18] Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.
- [19] Juraž Nižnan, Radek Pelánek, and Jiří Řihák. Using problem solving times and expert opinion to detect skills. In *Educational Data Mining (EDM)*, pages 434–434, 2014.
- [20] Juraž Nižnan, Radek Pelánek, and Jiří Řihák. Mapping problems to skills combining expert opinion and student data. In *Mathematical and Engineering Methods in Computer Science*, pages 113–124. Springer, 2014.
- [21] Jan Papoušek and Radek Pelánek. Impact of adaptive educational system behaviour on student motivation. In *Artificial Intelligence in Education*, 2015.
- [22] Jan Papoušek, Radek Pelánek, and Vít Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining (EDM)*, pages 6–13, 2014.
- [23] Zachary A Pardos and Neil T Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.
- [24] Philip I Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis—a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.
- [25] Radek Pelánek. Time decay functions and elo system in student modeling. In *Educational Data Mining (EDM)*, pages 21–27, 2014.
- [26] Radek Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 2015. To appear.
- [27] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

# Discovering Prerequisite Structure of Skills through Probabilistic Association Rules Mining

Yang Chen, Pierre-Henri Wuillemin, Jean-Marc Labat  
Sorbonne Universites, UPMC, Univ. Paris 6, UMR 7606, LIP6, Paris, France  
CNRS, UMR 7606, CNRS, Paris, France  
4 Place Jussieu, 75005 Paris, France  
{yang.chen, pierre-henri.wuillemin, jean-marc.labat}@lip6.fr

## ABSTRACT

Estimating the prerequisite structure of skills is a crucial issue in domain modeling. Students usually learn skills in sequence since the preliminary skills need to be learned prior to the complex skills. The prerequisite relations between skills underlie the design of learning sequence and adaptation strategies for tutoring systems. The prerequisite structures of skills are usually studied by human experts, but they are seldom tested empirically. Due to plenty of educational data available, in this paper, we intend to discover the prerequisite structure of skills from student performance data. However, it is a challenging task since skills are latent variables. Uncertainty exists in inferring student knowledge of skills from performance data. Probabilistic Association Rules Mining proposed by Sun et al. (2010) is a novel technique to discover association rules from uncertain data. In this paper, we preprocess student performance data by an evidence model. Then the probabilistic knowledge states of students estimated by the evidence model are used by the probabilistic association rules mining to discover the prerequisite structure of skills. We adapt our method to the testing data and the log data with different evidence models. One simulated data set and two real data sets are used to validate our method. The discovered prerequisite structures can be provided to assist human experts in domain modeling or to validate the prerequisite structures of skills from human expertise.

## Keywords

Probabilistic association rules mining, Skill structure, Prerequisite, DINA, BKT

## 1. INTRODUCTION

In most Intelligent Tutoring Systems (ITSs) and other educational environments, learning sequence is an important issue investigated by many educators and researchers. It is widely believed that students should be capable of solving the easier problems before the difficult ones are presented to them, and likewise, some preliminary skills should be learned prior to the learning of the complex skills. The prerequisite relations between problems and between skills underlie the adaptation strategies for tutoring and assessments. Furthermore, improving the accuracy of a student model with the prerequisite structure of skills has been

exemplified by [1, 2]. The prerequisite structures of problems and skills are in accordance with the Knowledge Space Theory [3] and Competence-based Knowledge Space Theory [4]. A student's knowledge state should comply with the prerequisite structure of skills. If a skill is mastered by a student, all the prerequisites of the skill should also be mastered by the student. If any prerequisite of a skill is not mastered by a student, it seems difficult for the student to learn the skill. Therefore, according to the knowledge states of students, we can uncover the prerequisite structure of skills. Most prerequisite structures of skills reported in the student modeling literature are studied by domain or cognition experts. It is a tough and time-consuming task since it is quite likely that the prerequisite structures from different experts on the same set of skills are difficult to come to an agreement. Moreover, the prerequisite structures from domain experts are seldom tested empirically. Nowadays, some prevalent data mining and machine learning techniques have been applied in cognition models, benefiting from large educational data available through online educational systems. Deriving the prerequisite structures of observable variables (e.g. problems) from data has been investigated by some researchers. However, discovering prerequisite structures of skills is still challenging since a student's knowledge of a skill is a latent variable. Uncertainty exists in inferring student knowledge of skills from performance data. This paper aims to discover the prerequisite structures of skills from student performance data.

## 2. RELATED WORK

With the emerging educational data mining techniques, many works have investigated the discovery of the prerequisite structures within domain models from data. The Partial Order Knowledge Structures (POKS) learning algorithm is proposed by Desmarais and his colleagues [5] to learn the item to item knowledge structures (i.e. the prerequisite structure of problems) which are solely composed of the observable nodes, like answers to test questions. The results from the experiments over their three data sets show that the POKS algorithm outperforms the classic BN structure learning algorithms [6] on the predictive ability and the computational efficiency. Pavlik Jr. et al. [7] used the POKS algorithm to analyze the relationships between the observable item-type skills, and the results were used for the hierarchical agglomerative clustering to improve the skill model. Vuong et al. [8] proposed a method to determine the dependency relationships between units in a curriculum with the student performance data that are observed at the unit level (i.e. graduating from a unit or not). They used the statistic binominal test to look for a significant difference between the performance of students who used the potential prerequisite unit and the performance of students who did not. If a significant difference is found, the prerequisite relation is deemed to exist. All these methods above are proposed

to discover prerequisite structures of the observable variables. Tseng et al. [9] proposed to use the frequent association rules mining to discover concept maps. They constructed concept maps by mining frequent association rules on the data of the fuzzy grades from students' testing. They used a deterministic method to transfer frequent association rules on questions to the prerequisite relations between concepts, without considering the uncertainty in the process of transferring students' performance to their knowledge. Deriving the prerequisite structure of skills from noisy observations of student knowledge is considered in the approach of Brunskill [10]. In this approach, the log likelihood is computed for the precondition model and the flat model (skills are independent) on each skill pair to estimate which model better fits the observed student data. Scheines et al. [11] extended causal discovery algorithms to discover the prerequisite structure of skills by performing statistical tests on latent variables. In this paper, we propose to apply a data mining technique, namely the probabilistic association rules mining, to discover prerequisite structures of skills from student performance data.

### 3. METHOD

Association rules mining [12] is a well-known data mining technique for discovering the interesting association rules in a database. Let  $I = \{i_1, i_2, \dots, i_m\}$  be a set of attributes (called items) and  $D = \{r_1, r_2, \dots, r_n\}$  be a set of records (or transactions), i.e. a database. Each record contains the values for all the attributes in  $I$ . A pattern (called itemset) contains the values for some of the attributes in  $I$ . The support count of pattern  $X$  is the number of records in  $D$  that contain  $X$ , denoted by  $\sigma(X)$ . An association rule is an implication of the form  $X \Rightarrow Y$ , where  $X$  and  $Y$  are related to the disjoint sets of attributes. Two measures are commonly used to discover the strong or interesting association rules: the support of rule  $X \Rightarrow Y$  denoted by  $Sup(X \Rightarrow Y)$ , which is the percentage of records in  $D$  that contain  $XUY$ , i.e.  $P(XUY)$ ; the confidence denoted by  $Conf(X \Rightarrow Y)$ , which is the percentage of records in  $D$  containing  $X$  that also contains  $Y$ , i.e.  $P(Y|X)$ . The rule  $X \Rightarrow Y$  is considered strong or interesting if it satisfies the following condition:

$$\begin{aligned} & (Sup(X \Rightarrow Y) \geq minsup) \\ & \wedge (Conf(X \Rightarrow Y) \geq minconf) \end{aligned} \quad (1)$$

where  $minsup$  and  $minconf$  denote the minimum support threshold and the minimum confidence threshold. The support threshold is used to discover frequent patterns in a database, and the confidence threshold is used to discover the association rules within the frequent patterns. The support condition makes sure the coverage of the rule, that is, there are adequate records in the database to which the rule applies. The confidence condition guarantees the accuracy of applying the rule. The rules which do not satisfy the support threshold or the confidence threshold are discarded in consideration of the reliability. Consequently, the strong association rules could be selected by the two thresholds.

To discover the skill structure, a database of students' knowledge states is required. The knowledge state of a student is a record in the database and the mastery of a skill is a binary attribute with the values mastered (1) and non-mastered (0). If skill  $S_i$  is a prerequisite of skill  $S_j$ , it is most likely that  $S_i$  is mastered given that  $S_j$  is mastered, and that skill  $S_j$  is not mastered given that  $S_i$  is not mastered. Thus this prerequisite relation corresponds with the two association rules:  $S_j=1 \Rightarrow S_i=1$  and  $S_i=0 \Rightarrow S_j=0$ . If both the association rules exist in a database,  $S_i$  is deemed a prerequisite of  $S_j$ . To examine if both the association rules exist in a database,

according to condition (1), the following conditions could be used:

$$\begin{aligned} & (Sup(S_j = 1 \Rightarrow S_i = 1) \geq minsup) \\ & \wedge (Conf(S_j = 1 \Rightarrow S_i = 1) \geq minconf) \end{aligned} \quad (2)$$

$$\begin{aligned} & (Sup(S_i = 0 \Rightarrow S_j = 0) \geq minsup) \\ & \wedge (Conf(S_i = 0 \Rightarrow S_j = 0) \geq minconf) \end{aligned} \quad (3)$$

When condition (2) is satisfied, the association rule  $S_j=1 \Rightarrow S_i=1$  is deemed to exist in the database, and when the condition (3) is satisfied, the association rule  $S_i=0 \Rightarrow S_j=0$  is deemed to exist in the database. Theoretically, if skill  $S_i$  is a prerequisite of  $S_j$ , all the records in the database should comply with the two association rules. To be exact, the knowledge state  $\{S_i=0, S_j=1\}$  should be impossible, thereby  $\sigma(S_i=0, S_j=1)$  should be 0. According to the equations (4) and (5), the confidences of the rules in the equations should be 1.0. Since noise always exists in real situations, when the confidence of an association rule is greater than a threshold, the rule is considered to exist if the support condition is also satisfied. We cannot conclude that the prerequisite relation exists if one rule exists but the other not. For instance, the high confidence of the rule  $S_j=1 \Rightarrow S_i=1$  might be caused by the high proportion  $P(S_i=1)$  in the data.

$$\begin{aligned} Conf(S_j = 1 \Rightarrow S_i = 1) &= P(S_i = 1 | S_j = 1) \\ &= \frac{\sigma(S_i = 1, S_j = 1)}{\sigma(S_i = 1, S_j = 1) + \sigma(S_i = 0, S_j = 1)} \rightarrow 1 \end{aligned} \quad (4)$$

$$\begin{aligned} Conf(S_i = 0 \Rightarrow S_j = 0) &= P(S_j = 0 | S_i = 0) \\ &= \frac{\sigma(S_i = 0, S_j = 0)}{\sigma(S_i = 0, S_j = 0) + \sigma(S_i = 0, S_j = 1)} \rightarrow 1 \end{aligned} \quad (5)$$

The discovery of the association rules within a database depends on the support and confidence thresholds. When the support threshold is given a relatively low value, more skill pairs will be considered as frequent patterns. When the confidence threshold is given a relatively low value, the weak association rules within frequent patterns will be deemed to exist. As a result, the weak prerequisite relations will be discovered. It is reasonable that the confidence threshold should be higher than 0.5. The selection of the two thresholds requires human expertise. Given the data about the knowledge states of a sample of students, the frequent association rules mining can be used to discover the prerequisite relations between skills.

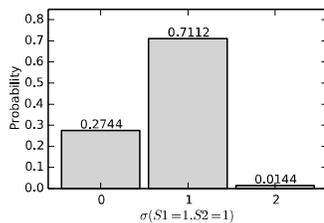
However, a student's knowledge state cannot be directly obtained since student knowledge of a skill is a latent variable. In common scenarios, we collect the performance data of students in assessments or tutoring systems and estimate their knowledge states according to the observed data. The evidence models that transfer the performance data of students to their knowledge states in consideration of the noise have been investigated for several decades. The psychometric models DINA (Deterministic Input Noisy AND) and NIDA (Noisy Input Deterministic AND) [13] have been used to infer the knowledge states of students from their response data on the multi-skill test items. The well-known Bayesian Knowledge Tracing (BKT) model [14] is a Hidden Markov model that has been used to update students' knowledge states according to the log files of their learning in a tutoring system. A Q-matrix which represents the items to skills mapping is required in these models. The Q-matrix is usually created by domain experts, but recently some researchers [15, 16, 17] investigated to extract an optimal Q-matrix from data. Our method

assumes that an accurate Q-matrix is known, like the method in [11]. Since the noise (e.g. slipping and guessing) is considered in the evidence models, the likelihood that a skill is mastered by a student can be estimated. The estimated knowledge state of a student is probabilistic, which incorporates the probability of each skill mastered by the student. Table 1 shows an example of the database consisting of probabilistic knowledge states. For example, the probabilities that skills  $S1$ ,  $S2$  and  $S3$  are mastered by student “st1” are 0.9, 0.8 and 0.9 respectively.

We discover the prerequisite relations between skills from the probabilistic knowledge states of students that are estimated by an evidence model. The frequent association rules mining can no longer be used to discover the prerequisite relations between skills from a probabilistic database. Because any attribute value in a probabilistic database is associated with a probability. A probabilistic database can be interpreted as a set of deterministic instances (named possible worlds) [18], each of which is associated with a probability. We assume that the noise (e.g. slipping, guessing) causing the uncertainty for different skills is mutually independent. In addition, we assume that the knowledge states of different students are observed independently. Under these assumptions, the probability of a possible world in our database is the product of the probabilities of the attribute values over all the records in the possible world [18, 19, 20]. For example, a possible world for the database in Table 1 is that both the knowledge states of the students “st1” and “st2” are  $\{S1=1, S2=0, S3=1\}$ , whose probability is about 0.0233 (i.e.  $0.9 \times 0.2 \times 0.9 \times 0.2 \times 0.9 \times 0.8$ ). The support count of a pattern in a probabilistic database should be computed with all the possible worlds. Thus the support count is no longer a deterministic number but a discrete random variable. Figure 1 depicts the probability mass function (*pmf*) of the support count of pattern  $\{S1=1, S2=1\}$  in the database of Table 1. For instance, the probability of  $\sigma(S1=1, S2=1)=1$  is about 0.7112, which is the sum of the probabilities of all the possible worlds in which only one record contains the pattern  $\{S1=1, S2=1\}$ . Since there are an exponential number of possible worlds in a probabilistic database (e.g.  $2^6$  possible worlds in the database of Table 1), computing the support count of a pattern is expensive. The Dynamic-Programming algorithm proposed by Sun et al. [20] is used to efficiently compute the support count *pmf* of a pattern.

**Table 1. A database of probabilistic knowledge states**

Student ID	Probabilistic Knowledge State
st1	{S1: 0.9, S2: 0.8, S3: 0.9}
st2	{S1: 0.2, S2: 0.1, S3: 0.8}



**Figure 1. The support count *pmf* of the pattern  $\{S1=1, S2=1\}$  in the database of Table 1**

To discover the prerequisite relations between skills from the probabilistic knowledge states of students, the probabilistic association rules mining technique [20] is used in this paper, which is an extension of the frequent association rules mining to discover association rules from uncertain data. Since the support

count of a pattern in a probabilistic database is a random variable, the conditions (2) and (3) are satisfied with a probability. Hence the association rules derived from a probabilistic database are also probabilistic. We use the formula proposed by [20] to compute the probability of an association rule satisfying the two thresholds. It can be also interpreted as the probability of a rule existing in a probabilistic database. For instance, the probability of the association rule  $Sj=1 \Rightarrow Si=1$  existing in a probabilistic database is the probability that the condition (2) is satisfied in the database:

$$\begin{aligned}
 & P(Sj=1 \Rightarrow Si=1) \\
 &= P((Sup(Sj=1 \Rightarrow Si=1) \geq minsup) \wedge (Conf(Sj=1 \Rightarrow Si=1) \geq minconf)) \\
 &= \frac{(1-minconf)^n}{\sum_{m=0}^{minconf} f_{Si=0, Sj=1}[m]} \sum_{n=minsup \times N}^N f_{Si=1, Sj=1}[n]
 \end{aligned} \tag{6}$$

where  $N$  is the number of records in the database and  $f_X$  denotes the support count *pmf* of pattern  $X$ , and  $f_X[k]=P(\sigma(X)=k)$ .

The probability of the rule related to condition (3) is computed similarly. According to formula (6), the probability of an association rule changes with the support and confidence thresholds. Given the two thresholds, the probability of an association rule existing in a probabilistic database can be computed. And if the probability is very close to 1.0, the association rule is considered to exist in the database. If both the association rules related to a prerequisite relation are considered to exist, the prerequisite relation is considered to exist. We can use another threshold, the minimum probability threshold denoted by *minprob*, to select the most possible association rules. Thus, if both  $P(Sj=1 \Rightarrow Si=1) \geq minprob$  and  $P(Si=0 \Rightarrow Sj=0) \geq minprob$  are satisfied,  $Si$  is deemed a prerequisite of  $Sj$ . When a pair of skills are estimated to be the prerequisite of each other, the relation between them are symmetric. It means that the two skills are mastered or not mastered simultaneously. The skill models might be improved by merging the two skills with the symmetric relation between them.

## 4. EVALUATION

We use one simulated data set and two real data sets to validate our method. The prerequisite structure derived from the simulated data is compared with the presupposed structure that is used to generate the data, while the prerequisite structure derived from the real data is compared with the structure investigated by another research on the same dataset or the structure from human expertise. Moreover, we adapt our method to the testing data and the log data. Different evidence models are used to preprocess the two types of data to get the probabilistic knowledge states of students. The DINA model is used for the testing data, whereas the BKT model is used for the log data.

### 4.1 Simulated Testing Data

**Data set.** We use the data simulation tool available via the R package CDM [21] to generate the dichotomous response data according to a cognitive diagnosis model (the DINA model used here). The prerequisite structure of the four skills is presupposed as Figure 3(a). According to this structure, the knowledge space decreases to be composed of six knowledge states, that is  $\emptyset$ ,  $\{S1\}$ ,  $\{S1, S2\}$ ,  $\{S1, S3\}$ ,  $\{S1, S2, S3\}$ ,  $\{S1, S2, S3, S4\}$ . The reduced knowledge space implies the prerequisite structure of the skills. The knowledge states of 1200 students are randomly generated from the reduced knowledge space restricting every knowledge state type in the same proportion (i.e. 200 students per type). The

simulated knowledge states are used as the input of the data simulation tool. There are 10 simulated testing questions, each of which requires one or two of the skills for the correct response. The slip and guess parameters for each question are restricted to be randomly selected in the range of 0.05 and 0.3. According to the DINA model with these specified parameters, the data simulation tool generates the response data. Using the simulated response data as the input of a flat DINA model, the slip and guess parameters of each question in the model are estimated and the probability of each student's knowledge on each skill is computed. The tool for the parameter estimation of DINA model is also available through the R package CDM [21], which is performed by the Expectation Maximization algorithm to maximize the marginal likelihood of data.

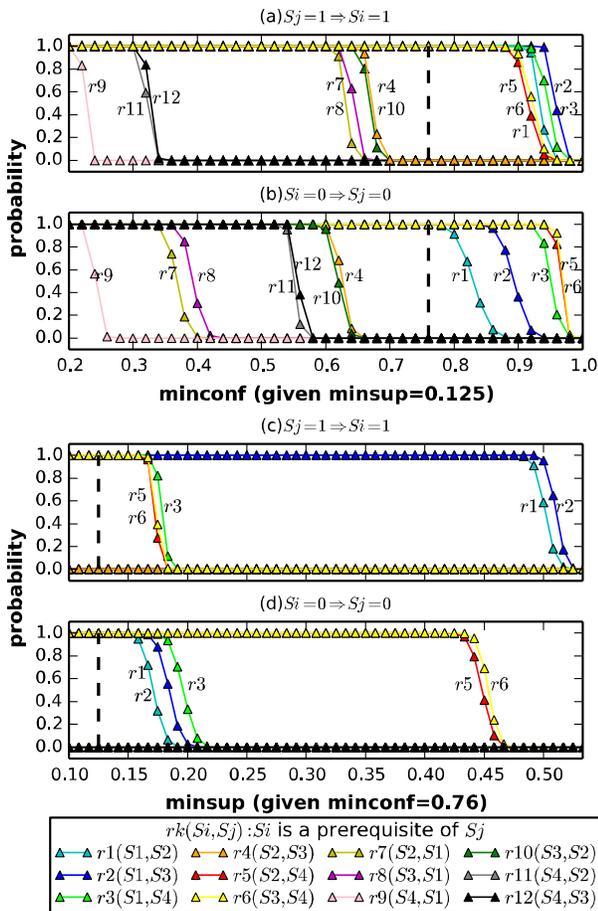


Figure 2. The probabilities of the association rules in the simulated data given different confidence or support thresholds

**Result.** The estimated probabilistic knowledge states of the simulated students are used as the input data to discover the prerequisite relations between skills. For each skill pair, there are two prerequisite relation candidates. For each prerequisite relation candidate, we examine if the two corresponding association rules  $S_j=1 \Rightarrow S_i=1$  and  $S_i=0 \Rightarrow S_j=0$  exist in the database. The probability of an association rule existing in the database is computed according to formula (6), which is jointly affected by the selected support and confidence thresholds. For the sake of clarity, we look into the effect of one threshold leaving the other one unchanged. The joint effect of the two thresholds will be discussed in section

4.4. Giving a small constant to one threshold that all the association rules satisfy (perhaps several trials are needed or simply assign 0.0), we can observe how the probabilities of the association rules change with different values of the other threshold.

Figure 2 (a) and (b) describe how the probabilities of the corresponding association rules in the simulated data change with different confidence thresholds, where the support threshold is given as a constant (0.125 here). When the probability of a rule is close to 1.0, the rule is deemed to satisfy the thresholds. All the association rules satisfy the support threshold since their probabilities are almost 1.0 at first. The rules in the two figures corresponding to the same prerequisite relation candidate are depicted in the same color. In the figures, when the confidence threshold varies from 0.2 to 1.0, the probabilities of the different rules decrease from 1.0 to 0.0 in different intervals of threshold value. When we choose different threshold values, different sets of rules will be discovered. In each figure, there are five rules that can satisfy the significantly higher threshold. Given  $\text{minconf}=0.78$ , the probabilities of these rules are almost 1.0 whereas others are almost 0.0. These rules are very likely to exist. Moreover, the discovered rules in the two figures correspond to the same set of prerequisite relation candidates. Accordingly, these prerequisite relations are very likely to exist. To make sure the coverage of the association rules satisfying the high confidence threshold, it is necessary to know the support distributions of these rules. Figure 2 (c) and (d) illustrate how the probabilities of the corresponding association rules change with different support thresholds. The confidence threshold is given as a constant 0.76. Only on these rules, the effect of different support thresholds can be observed. In each figure, the rules gather in two intervals of threshold value. For example, in Figure 2 (c), to select the rules corresponding to  $r_3$ ,  $r_5$  and  $r_6$ , the highest value for the support threshold is roughly 0.17, while for the other two rules, it is 0.49. If both the confidence threshold and the support threshold are appropriately selected, the most possible association rules will be distinguished from others. As a result, the five prerequisite relations can be discovered in this experiment.

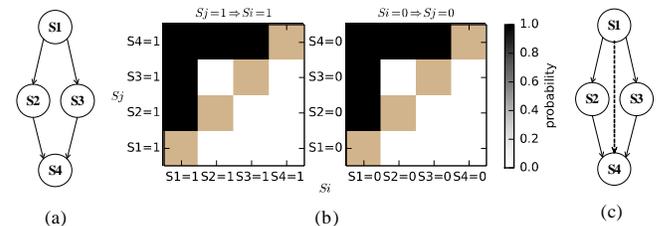


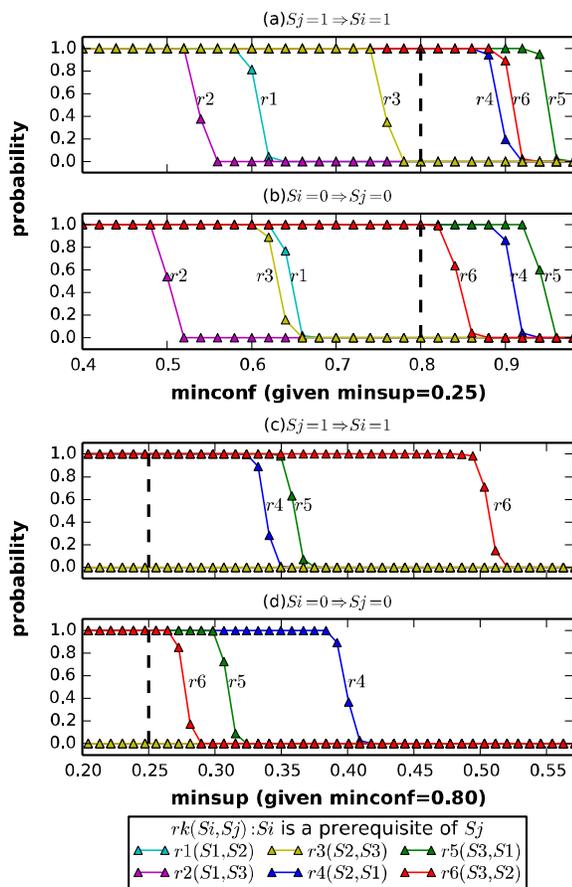
Figure 3. (a) Preresupposed prerequisite structure of the skills in the simulated data; (b) Probabilities of the association rules in the simulated data given  $\text{minconf}=0.76$  and  $\text{minsup}=0.125$ , brown squares denoting impossible rules; (c) Discovered prerequisite structure

Figure 3 (b) illustrates the probabilities of the corresponding association rules in the simulated data given  $\text{minconf}=0.76$  and  $\text{minsup}=0.125$ . A square's color indicates the probability of the corresponding rule. Five association rules in each of the figures whose probabilities are almost 1.0 are deemed to exist. And the prerequisite relations corresponding to the discovered rules are deemed to exist. To qualitatively construct the prerequisite structure of skills, every discovered prerequisite relation is represented by an arc. It should be noted that the arc representing

the relation that  $S1$  is a prerequisite of  $S4$  is not present in Figure 3 (a) due to the transitivity of prerequisite relation. Consequently, the prerequisite structure discovered by our method which is shown in Figure 3 (c), is completely in accordance with the presupposed structure shown in Figure 3 (a).

## 4.2 Real Testing Data

**Data set.** The ECPE (Examination for the Certification of Proficiency in English) data set is available through the R package CDM [21], which comes from a test developed and scored by the English Language Institute of the University of Michigan [22]. A sample of 2933 examinees is tested by 28 items on 3 skills, i.e. Morphosyntactic rules ( $S1$ ), Cohesive rules ( $S2$ ), and Lexical rules ( $S3$ ). The parameter estimation tool in the R package CDM [21] for DINA model is also used in this experiment to estimate the slip and guess parameters of items according to the student response data. And with the estimated slip and guess parameters, the probabilistic knowledge states of students are assessed according to the DINA model, which are the input data for discovering the prerequisite structure of skills.

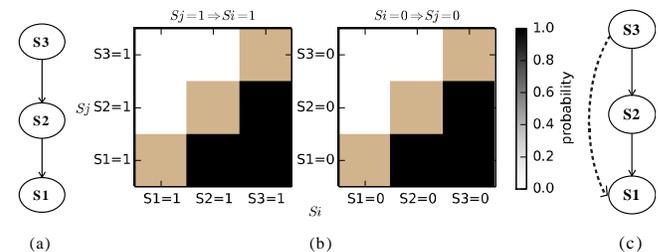


**Figure 4.** The probabilities of the association rules in the ECPE data given different confidence or support thresholds

**Result.** The effect of different confidence thresholds on the association rules in the ECPE data is depicted in Figure 4 (a) and (b) given the support threshold as a constant (0.25 here). In each figure, there are three association rules that can satisfy a significantly higher confidence threshold than others. The maximum value of the confidence threshold for them is roughly 0.82. And these rules in the two figures correspond to the same set of prerequisite relation candidates, that is,  $r4$ ,  $r5$  and  $r6$ . Thus

these candidates are most likely to exist. It can be noticed that in Figure 4 (a) the rule  $S3=1 \Rightarrow S2=1$  can satisfy a relatively high confidence threshold. The maximum threshold value that it can satisfy is roughly 0.74. However, its counterpart in Fig 4 (b), i.e. the rule  $S2=0 \Rightarrow S3=0$ , cannot satisfy a confidence threshold higher than 0.6. When a strong prerequisite relation is required, the relation corresponding to the two rules cannot be selected. Only when both the two types of rules can satisfy a high confidence, the corresponding prerequisite relation is considered strong. Likewise, the effect of different support thresholds is shown in Figure 4 (c) and (d), where the confidence threshold is given as 0.80. And in each figure, only the three association rules which satisfy the confidence threshold are sensitive to different support thresholds. It can also be found that these rules are supported by a considerable proportion of the sample. Even when  $minsup=0.27$ , all the three rules in each figure satisfy it. According to the figures, when the support and confidence thresholds are appropriately selected, these rules can be distinguished from others. Consequently, the strong prerequisite relations can be discovered.

Given the confidence and support thresholds as 0.80 and 0.25 respectively, for instance, the probabilities of the corresponding association rules are illustrated in Figure 5 (b). The rules that satisfy the two thresholds (with a probability of almost 1.0) are deemed to exist, which are evidently distinguished from the rules that do not (with a probability of almost 0.0). Three prerequisite relations shown in Figure 5 (c) are found in terms of the discovered association rules. To validate the result, we compare it with the findings of another research on the same data set. The attribute hierarchy, namely the prerequisite structure of skills, in ECPE data has been investigated by Templin and Bradshaw [22] as Figure 5 (a). Our discovered prerequisite structure totally agrees with their findings.



**Figure 5.** (a) Prerequisite structure of the skills in the ECPE data discovered by Templin and Bradshaw [22]; (b) Probabilities of the association rules in the ECPE data given  $minconf=0.80$  and  $minsup=0.25$ , brown squares denoting impossible rules; (c) Discovered prerequisite structure

## 4.3 Real Log Data

**Data set.** We use the 2006-2007 school year data of the curriculum “Bridge to Algebra” [23] which incorporates the log files of 1146 students collected by Cognitive Tutor, an ITS for mathematics learning. The units in this curriculum involve distinct mathematical topics, while the sections in each unit involve distinct skills on the unit topic. A set of word problems is provided for each section skill. We use the sections in the units “equivalent fractions” and “fraction operations” as the skills (see Table 2). There are 560 students in the data set performing to learn one or several of the item-type skills in these units. The five skills discussed in our experiment are instructed in the given order in Table 2. A student’s knowledge of the prior skills has the potential to affect his learning of the new skill. Hence, it makes sense to estimate whether a skill trained prior to the new skill is a

prerequisite of it. If the prior skill  $S_i$  is a prerequisite of skill  $S_j$ , students who have mastered skill  $S_j$  quite likely have previously mastered skill  $S_i$ , and students not mastering the skill  $S_i$  quite likely learn the skill  $S_j$  with great difficulty. Thus if both the rules  $S_j=1 \Rightarrow S_i=1$  and  $S_i=0 \Rightarrow S_j=0$  exist in the data, the prior skill  $S_i$  is deemed a prerequisite of skill  $S_j$ .

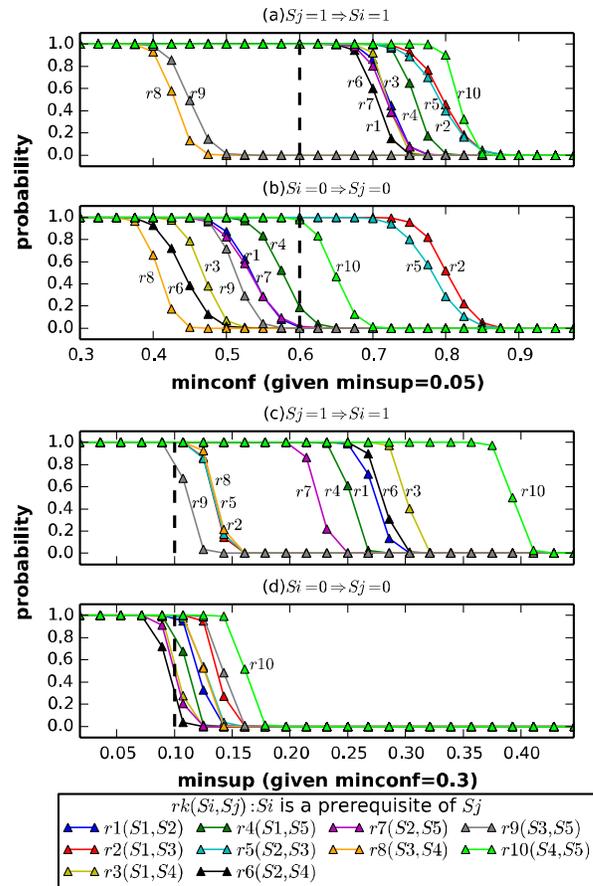
**Table 2. Skills in the curriculum “Bridge to Algebra”**

Skill	Example
S1: Writing equivalent fractions	Fill in the blank: $\frac{2}{3} = \frac{\square}{6}$ .
S2: Simplifying fractions	Write the fraction in simplest form: $\frac{24}{30} = \frac{\square}{\square}$ .
S3: Comparing and ordering fractions	Compare the fractions $\frac{3}{4}$ and $\frac{5}{6}$ .
S4: Adding and subtracting fractions with like denominators	$\frac{2}{10} + \frac{3}{10} =$
S5: Adding and subtracting fractions with unlike denominators	$\frac{2}{3} - \frac{1}{4} =$

To discover the prerequisite relations between skills, firstly we need to estimate the outcomes of student learning according to the log data. A student learns a skill by solving a set of problems that requires applying that skill. At each opportunity, student knowledge of a skill probably transitions from the unlearned to learned state. Thus their knowledge should be updated each time they go through a problem. The BKT model has been widely used to track the dynamic knowledge states of students according to their activities on ITSS. In the standard BKT, four parameters are specified for each skill [14]:  $P(L_0)$  denoting the initial probability of knowing the skill a priori,  $P(T)$  denoting the probability of student’s knowledge of the skill transitioning from the unlearned to the learned state,  $P(S)$  and  $P(G)$  denoting the probabilities of slipping and guessing when applying the skill. We implemented the BKT model by using the Bayes Net Toolbox for Student modeling [24]. The parameter  $P(L_0)$  is initialized to 0.5 while the other three parameters are initialized to 0.1. The four parameters are estimated according to the log data of students, and the probability of a skill to be mastered by a student is estimated each time the student performs to solve a problem on that skill. In the log data, students learned the section skills one by one and no student relearned a prior section skill. If a prior skill  $S_i$  is a prerequisite of skill  $S_j$ , the knowledge state of  $S_i$  after the last opportunity of learning it has an impact on learning  $S_j$ . We use the probabilities about students’ final knowledge state of  $S_i$  and  $S_j$  to analyze whether a prerequisite relation exists between them. Thus students’ final knowledge states on each skill are used as the input data of our method.

**Result.** The probabilities of the association rules in the log data changing with different confidence thresholds are illustrated in Figure 6 (a) and (b) given the support threshold as a small constant (0.05 here). In Figure 6 (a), compared with the rules  $S_4=1 \Rightarrow S_3=1$  and  $S_5=1 \Rightarrow S_3=1$ , all the other association rules can satisfy a significantly higher confidence, while in Figure 6 (b) if given  $minconf=0.6$ , only three rules satisfy it. The effect of different support thresholds on the probabilities of the association rules is depicted in Figure 6 (c) and (d) given the confidence

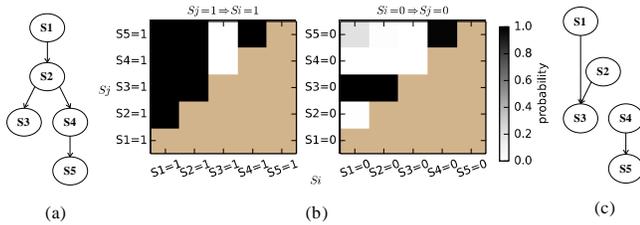
threshold as a constant (0.3 here). All the association rules satisfy the confidence threshold as the probabilities of the rules are almost 1.0 at first. In Figure 6 (c), there are six rules that can satisfy a relatively higher support threshold (e.g.  $minsup=0.2$ ). But in Figure 6 (d), even given  $minsup=0.14$ , only the rule  $S_4=0 \Rightarrow S_5=0$  satisfy it, and the maximum value for the support threshold that all the rules can satisfy is roughly 0.07.



**Figure 6. The Probabilities of the association rules in the “Bridge to Algebra 2006-2007” data given different confidence or support thresholds**

Given the confidence and support thresholds as 0.6 and 0.1 respectively, the probabilities of the association rules in the log data are depicted in Figure 7 (b). There are eight of the rules in the form of  $S_j=1 \Rightarrow S_i=1$  (left) and three of the rules in the form of  $S_i=0 \Rightarrow S_j=0$  (right) discovered, whose probabilities to satisfy the thresholds are almost 1.0. According to the result, only the three prerequisite relations shown in Figure 7 (c), whose corresponding rules both are discovered, are deemed to exist. Figure 7 (a) shows the prerequisite structure of the five skills from the human experts’ opinions. It makes sense that the skills  $S_1$  and  $S_2$  rather than skill  $S_3$  are required for learning the skills  $S_4$  and  $S_5$ . This is supported by the chapter warm-up content in the student textbook of the course [25]. The discovered rules in the form of  $S_j=1 \Rightarrow S_i=1$  completely agree with the structure of human expertise. But the discovered rules in the form of  $S_i=0 \Rightarrow S_j=0$  is inconsistent with it. The counterparts of a large part of the discovered rules  $S_j=1 \Rightarrow S_i=1$  do not satisfy the confidence threshold. Even reducing the confidence threshold to the lowest value, i.e. 0.5, the rules  $S_1=0 \Rightarrow S_4=0$  and  $S_2=0 \Rightarrow S_4=0$  still do not satisfy it (see Figure 6 (b)). It seems that the rules  $S_j=1 \Rightarrow S_i=1$  are more reliable than

$S_i=0 \Rightarrow S_j=0$  since most of the former can satisfy a higher support threshold than the latter (see Figure 6 (c) and (d)). In addition, the log data is very likely to contain much noise. It is possible that some skills could be learned if students take sufficient training, even though some prerequisites are not previously mastered. In this case, the support count  $\sigma(S_i=0, S_j=1)$  would increase. Or perhaps students learned the prerequisite skills by solving the scaffolding questions in the process of learning new skills, even though they performed not mastering the prerequisite skills before. In this case, the observed values of  $\sigma(S_i=0, S_j=1)$  would be higher than the real values. According to the equations (4) and (5), if  $\sigma(S_i=0, S_j=1)$  increases, the confidence of the rules will decrease. And when the noise appears in the data, the confidences of the association rules which are supported by a small proportion of sample will be affected much more than those supported by a large proportion of sample.

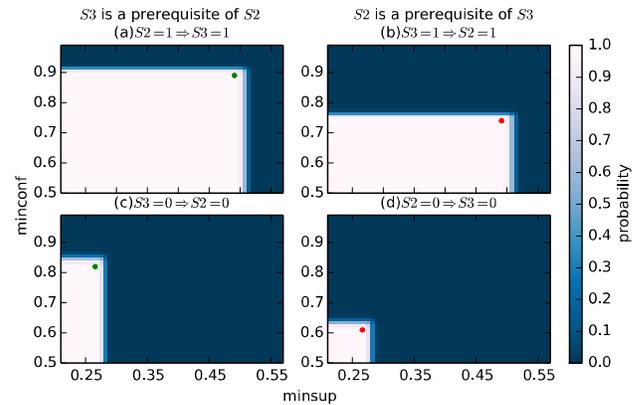


**Figure 7. (a) Prerequisite structure from human expertise; (b) Probabilities of the association rules in the “Bridge to Algebra 2006-2007” data given  $minconf=0.6$  and  $minsup=0.1$ , brown squares denoting impossible rules; (c) Discovered prerequisite structure**

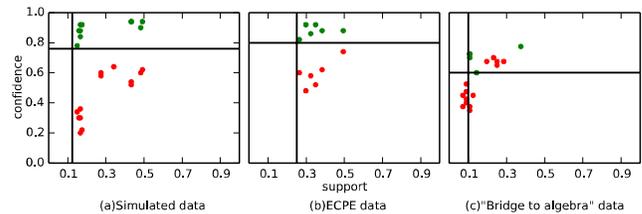
#### 4.4 Joint Effect of thresholds

We have discussed the effect of one threshold on the probability of association rules while eliminating the effect of the other one in the three experiments. To determine the values for the thresholds, we investigate how the two thresholds simultaneously affect the probability of an association rule. Figure 8 depicts how the probabilities of the association rules for the skill pair S2 and S3 in the ECPE data change with different support and confidence thresholds, where (a) and (c) involve one relation candidate while (b) and (d) involve the other one. The figures demonstrate that the probability of a rule decreases almost from 1.0 to 0.0 when the confidence and support thresholds vary from low to high. It can be found that the rules in the left figures can satisfy an evidently higher confidence threshold than those in the right figures, and have the same support distributions with them. If we set  $minconf=0.8$  and  $minsup=0.25$ , only the rules in the left figures satisfy them. Suppose that a rule satisfy the thresholds if its probability is higher than 0.95, i.e.  $minprob=0.95$ . When we change the values of the confidence and support thresholds from 0.0 to 1.0, for each rule, we can find a point whose coordinates consist of the maximum values of the confidence and support thresholds that the rule can satisfy. Finding the optimal point is hard and there are probably several feasible points. To simplify the computation, the thresholds are given by a sequence of discrete values from 0.0 to 1.0. We find the maximum value for each threshold when only one threshold affects the probability of the rule given the other as 0.0. And for each threshold,  $minprob$  is given as 0.97, roughly the square root of the original value. The found maximum values for the two thresholds are the coordinates of the point. The found point is actually an approximately optimal point. For convenience, the point is named maximum threshold point in this paper. The points for all the rules in the three data sets are found by our method as well as plotted in Figure 9 (some

points overlap). When we set certain values to the thresholds, the points located in the upper right area satisfy them and the related rules are deemed to exist. For one prerequisite relation, a couple of related points should be verified. Only when both of them are located in the upper right area, they are considered eligible to uncover the prerequisite relation. The eligible points in Figure 8 and Figure 9 are indicated given the thresholds.



**Figure 8. Probabilities of the association rules within the skill pair S2 and S3 in the ECPE data given different confidence and support thresholds, and their maximum threshold points which are eligible (green) or not (red) given  $minconf=0.8$  and  $minsup=0.25$**



**Figure 9. Maximum threshold points for the association rules in our three experiments, where eligible points are indicated in green given the thresholds**

## 5. CONCLUSION AND DISCUSSION

Discovering the prerequisite structure of skills from data is challenging in domain modeling since skills are the latent variables. In this paper, we propose to apply the probabilistic association rules mining technique to discover the prerequisite structure of skills from student performance data. Student performance data is preprocessed by an evidence model. And then the probabilistic knowledge states of students estimated by the evidence model are used as the input data of probabilistic association rules mining. Prerequisite relations between skills are discovered by estimating the corresponding association rules in the probabilistic database. The confidence condition of an association rule in our method is similar to the statistical hypotheses used in the POKS algorithm for determining the prerequisite relations between observable variables (see the details in [5]). But our method targets on the challenge of discovering the prerequisite relations between latent variables from the noisy observable data. In addition, our method takes the coverage into account (i.e. the support condition), which could strengthen the reliability of the discovered prerequisite relations. Determining the appropriate confidence and support thresholds is a crucial issue in our method. The effect of a single threshold and the joint effect of two thresholds on the probabilities of the rules are

discussed. The maximum threshold points of the probabilistic association rules are proposed for determining the thresholds. We adapt our method to two common types of data, the testing data and the log data, which are preprocessed by different evidence models, the DINA model and the BKT model. An accurate Q-matrix is required for the evidence models, which is a limitation of our method. According to the results of the experiments in this paper, our method performs well to discover the prerequisite structures from a simulated testing data set and a real testing data set. However, applying our method in the log data still needs to be improved. Since much noise exist in the log data, the strategies to reduce the noise need to be applied. The prerequisite structures of skills discovered by our method can be applied to assist human experts in skill modeling or to validate the prerequisite structures of skills from human expertise.

## 6. REFERENCES

- [1] Käser, T., Klinger, S., Schwing, G., Gross, M.: Beyond Knowledge Tracing: Modeling Skill Topologies with Bayesian Networks. In *Proceedings of the 12<sup>th</sup> International Conference on Intelligent Tutoring Systems*, Honolulu, USA, 188-198, 2014
- [2] Chen, Y., WUILLEMIN, P.H., Labat, J.M.: Bayesian Student Modeling Improved by Diagnostic Items. In *Proceedings of the 12<sup>th</sup> International Conference on Intelligent Tutoring Systems*, Honolulu, USA, 144-149, 2014
- [3] Falmagne, J.C., Cosyn, E., Doignon, J.P., Thiéry, N.: The Assessment of Knowledge, in Theory and in Practice. In *Proceedings of the 4<sup>th</sup> International Conference on Formal Concept Analysis*, Dresden, Germany, 61-79, 2006
- [4] Heller, J., Steiner, C., Hockemeyer, C., Albert, D.: Competence-based Knowledge Structures for Personalized Learning. *International Journal on E-learning*, 5, 75-88, 2006
- [5] Desmarais, M.C., Meshkinfam, P., Gagnon, M.: Learned Student Models with Item to Item Knowledge Structures. *User Modeling and User-adapted Interaction*, 16(5), 403-434, 2006
- [6] Spirtes, P., Glymour, C., Scheines, R.: *Causation, Prediction, and Search*. 2<sup>nd</sup> Edition, The MIT Press, Cambridge, 2000
- [7] Pavlik Jr., P.I., Cen, H., Wu, L., Koedinger, K.R.: Using Item-type Performance Covariance to Improve the Skill Model of an Existing Tutor. In *Proceedings of the 1<sup>st</sup> International Conference on Educational Data Mining*, Montreal, Canada, 77-86, 2008
- [8] Vuong, A., Nixon, T., Towle, B.: A Method for Finding Prerequisites within a Curriculum. In *Proceedings of the 4<sup>th</sup> International Conference on Educational Data Mining*, Eindhoven, Netherlands, 211-216, 2011
- [9] Tseng, S.S., Sue, P.C., Su, J.M., Weng, J.F., Tsai, W.N.: A New Approach for Constructing the Concept Map. *Computers & Education*, 49(3), 691-707, 2007
- [10] Brunskill, E.: Estimating Prerequisite Structure from Noisy Data. In *Proceedings of the 4<sup>th</sup> International Conference on Educational Data Mining*, Eindhoven, Netherlands, 217-222, 2011
- [11] Scheines, R., Silver, E., Goldin, I.: Discovering Prerequisite Relationships among Knowledge Components. In *Proceedings of the 7<sup>th</sup> International Conference on Educational Data Mining*, London, UK, 355-356, 2014
- [12] Agrawal, R., Srikant, R.: Fast Algorithm for Mining Association Rules. In *Proceedings of the 20<sup>th</sup> International Conference on Very Large Data Bases*, San Francisco, USA, 487-499, 1994
- [13] Roussos, L.A., Templin, J.L., Henson, R.A.: Skills Diagnosis Using IRT-based Latent Class Models. *Journal of Educational Measurement*, 44(4), 293-311, 2007
- [14] Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253-278, 1995
- [15] Barnes, T.: The Q-matrix Method: Mining Student Response Data for Knowledge. *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, 2005
- [16] Desmarais, M.C., Naceur, R.: A Matrix Factorization Method for Mapping Items to Skills and for Enhancing Expert-Based Q-Matrices. In *Proceedings of the 16<sup>th</sup> International Conference on Artificial Intelligence in Education*, Memphis, USA, 441-450, 2013
- [17] González-Brenes, J.P.: Modeling Skill Acquisition Over Time with Sequence and Topic Modeling. In *Proceedings of the 18<sup>th</sup> International Conference on Artificial Intelligence and Statistics*, San Diego, USA, 296-305, 2015
- [18] Bernecker, T., Kriegel, H.P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic Frequent Itemset Mining in Uncertain Databases. In *Proceedings of the 15<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 119-128, 2009
- [19] Chui, C.K., Kao, B., Hung, E.: Mining Frequent Itemsets from Uncertain Data. In *Proceedings of the 11<sup>th</sup> PAKDD Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Nanjing, China, 47-58, 2007
- [20] Sun, L., Cheng, R., Cheung, D.W., Cheng, J.: Mining Uncertain Data with Probabilistic Guarantees. In *Proceedings of the 16<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 273-282, 2010
- [21] Robitzsch, A., Kiefer, T., George, A.C., Uenlue, A.: Package CDM (Version 3.4-21, 2014), <http://cran.r-project.org/web/packages/CDM/index.html>
- [22] Templin, J., Bradshaw, L.: Hierarchical Diagnostic Classification Models: A Family of Models for Estimating and Testing Attribute Hierarchies. *Psychometrika*, 79, 317-339, 2014
- [23] Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G.J., & Koedinger, K.R.: Bridge to Algebra 2006-2007. Development data set from KDD Cup 2010 Educational Data Mining Challenge. Find it at <http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp>
- [24] Chang, K., Beck, J., Mostow, J., & Corbett, A.: A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. In *Proceedings of the 8<sup>th</sup> International Conference on Intelligent Tutoring Systems*, Jhongli, Taiwan, 104-113, 2006
- [25] Hadley, W.S., Raith, M.L.: *Bridge to Algebra Student Text*. Carnegie Learning. 2008

# Choosing to Interact: Exploring the Relationship Between Learner Personality, Attitudes, and Tutorial Dialogue Participation

Aysu Ezen-Can  
Department of Computer Science  
North Carolina State University  
aezen@ncsu.edu

Kristy Elizabeth Boyer  
Department of Computer Science  
North Carolina State University  
keboyer@ncsu.edu

## ABSTRACT

The tremendous effectiveness of intelligent tutoring systems is due in large part to their interactivity. However, when learners are free to choose the extent to which they interact with a tutoring system, not all learners do so actively. This paper examines a study with a natural language tutorial dialogue system for computer science, in which students interacted with the JavaTutor system through natural language dialogue over the course of problem solving. We explore the relationship between students' level of dialogue interaction and learner characteristics including personality profile and pre-existing attitudes toward the learning task. The results show that these learner characteristics are significant predictors of the extent to which students engage in dialogue with the tutoring system, as well as the number of task actions students make. By identifying students who may not engage with tutoring systems as readily, this work constitutes a step toward building adaptive systems that successfully support a variety of students with different attitudes and personalities.

## Keywords

Learner characteristics, personality, disengagement, tutorial dialogue

## 1. INTRODUCTION

Tutorial dialogue systems effectively support learning through rich natural language dialogue [7,8,14,19]. However, the effectiveness of tutorial dialogue systems, like other adaptive learning environments, depends in large part on students' willingness to interact with them [18]. Interaction varies tremendously across individual students and student populations. We observe various types of *disengagement* including lack of motivation or interest for the learning task [10], as well as *gaming* an intelligent tutor by exploiting properties of the learning environment [2,4].

In addition to these factors, individual differences such as self-reported interest in the task and confidence in learning have been found to be strong predictors of engagement [6]. Similarly, students' hidden attitudes toward learning [1] and motivation for the task

[3] may be highly influential. Boredom, which is associated with reduced motivation to perform the activity [15], has been positively correlated with attention problems and negatively correlated with performance. Students' participation in tutorial dialogue has also been found to be associated with the students' expectations [11], and in human-human tutorial dialogue, student personality traits have recently been found to be significant factors [16]. However, the field is far from a full understanding of the factors that influence students' choices to engage or interact with tutorial dialogue systems.

This paper presents an investigation into the relationship between student characteristics and interactions with a tutorial dialogue system. We hypothesized that students' personality profile, for example their tendencies toward extraversion or openness, would be significantly associated with the level of natural language interaction observed within a tutorial dialogue system. We also hypothesized that students' attitudes toward the learning task would be a significant factor in their interactions with the system. We examine these hypotheses within a data set of 51 university students interacting with the JavaTutor tutorial dialogue system for introductory computer science. Regression models were built that predict both dialogue and task participation by the students, who have the choice to interact with the dialogue system as little or as much as desired over the course of the learning tasks. The models demonstrate that students' attitudes and personalities are significantly predictive of their willingness to interact with the tutorial dialogue system. The findings suggest that some learner characteristics may put students at risk of low participation with a tutorial dialogue system, and constitute a first step toward proactively adapting the systems to benefit these learners.

## 2. TUTORING STUDY

The JavaTutor tutorial dialogue system (Figure 1) supports students in solving introductory computer programming problems in the Java programming language while interacting in textual natural language. Students are provided with a series of learning tasks that build on each other to guide the students through creation of a simple text-based adventure game.<sup>1</sup>

The study reported here was conducted with the JavaTutor tutorial dialogue system in 2014. The students (12 female; 39 male; mean age = 21) were drawn from a university-level engineering class. They interacted with the tutorial dialogue system for one session lasting approximately 45 minutes.

<sup>1</sup>Implementation details of the system are beyond the scope of this paper but are described in [9].

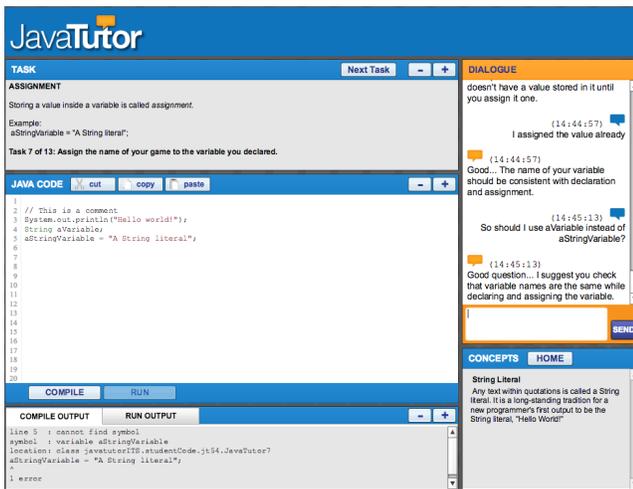


Figure 1: Screenshot from the tutorial dialogue system.

Prior to interacting with JavaTutor, students took a pre-survey that included validated items to measure goal orientation [17], general self-efficacy [5], confidence in learning computer science and programming [13], and personality profile using a concise version of the Big Five Inventory [12]. Students also completed a pre-test and posttest before and after their interaction with JavaTutor.

### 3. ANALYSIS

Students were instructed that they could make comments, pose questions, and request feedback at any time through textual dialogue. Overall, students interacting with JavaTutor achieved significant learning gains from pre-test to posttest (average= 12%, median= 13.4%, stdev = 32%,  $p = 0.001$ ). However, we observed that 58.8% of students never made an utterance. For students who did engage in dialogue with the tutor, the average number of utterances was 5.1 (stdev=7.36, median= 2). Regardless of the extent to which they chose to engage in natural language dialogue, all students received some tutorial dialogue utterances based upon the system’s model of feedback for task events.

Our goal is to identify the factors that may be influential in students’ levels of interaction with the system. To this end, we built multiple regression models. The remainder of this section describes the analysis.

#### 3.1 Response Variables

Based upon the logged interaction traces, we extracted dialogue and task events and used them to compute a numeric representation of the student’s level of interaction with the system. For dialogue interaction we utilized the *number of utterances* written by each student. The range of number of student utterances was between 0 and 33.

We extracted four features that represent interaction of students with the system throughout tutoring. The first of these four features is *number of content changes* which refers to the changes in the student’s programming code, as the code they write is referred to as content pane. We also computed the *number of compile events* and *number of run activities*. The number of compile/run events ranges from 4 to 224, whereas the number of content changes ranges from 88 to 1099 to complete the series of learning tasks.

Finally, we computed the *number of tutor messages* each student received. The tutoring systems provided students with feedback. The number of messages received is closely related to the number of actions that triggered tutor feedback, which is also a measure of participation. The minimum number of tutor messages provided to any student was 8, whereas the largest number of tutor messages to a student during a tutoring session was 121. We built separate multiple regression models to predict level of dialogue interaction and level of task interaction.

#### 3.2 Predictor Variables

We hypothesized that several learner characteristics were significantly associated with level of interaction in the system. We provided these variables for selection within the models (see Table 1). All of the predictors were standardized to a common scale before model building.

Predictor variable	Example survey item/ Description
Computer science confidence	<i>I am sure that I can learn programming.</i>
Perceived computer science usefulness	<i>I'll need programming for my future work.</i>
Motivation toward computer science	<i>Programming is enjoyable and stimulating to me.</i>
General self-efficacy	<i>I will be able to achieve most of the goals that I have set for myself.</i>
Learning goal orientation	<i>I often look for opportunities to develop new skills and knowledge.</i>
Performance demonstration	<i>I like to show that I can perform better than my coworkers.</i>
Failure avoidance	<i>Avoiding a show of low ability is more important to me than learning a new skill.</i>
Achievement goals	<i>It is important for me to do better than other students.</i>
Gender	Male/female
Age	Age of the student
University class standing	The year that the student is in the university
Perception of student’s own computer skill	<i>How skilled are you with computers, compared to the average person?</i>
Extraversion	<i>I see myself as someone who is talkative.</i>
Agreeableness	<i>I see myself as someone who is helpful and unselfish with others.</i>
Conscientiousness	<i>I see myself as someone who does a thorough job.</i>
Neuroticism	<i>I see myself as someone who is depressed, blue.</i>
Openness	<i>I see myself as someone who is original, comes up with new ideas.</i>
Pre-test score	Score showing the performance of the student before tutoring session

Table 1: Predictor variables from pre-survey and pre-test.

#### 3.3 Modeling Level of Participation

We built separate models for each of the response variables (number of utterances, compile/run events, content changes, received tutor messages). For each response variable we used the whole dataset

and selected features via stepwise linear regression. Because the goal was to investigate relationships between pre-measures (student characteristics, attitudes) and level of participation, we conducted descriptive analyses using the entire data set for model building.

The model for number of dialogue utterances (Table 2) revealed that students' failure avoidance characteristic is a significant predictor of tutorial dialogue interactivity. Students who indicated that they tend to avoid tasks in which they may have higher chance of failure wrote fewer utterances to the system.

Number of utterances =	Coefficient	p
Failure Avoidance	-0.3089	0.0274
~1 (intercept)		1
<b>RMSE = 0.961</b>		
$R^2 = 0.0954$		

Table 2: Stepwise linear regression model for the number of utterances.

The model for number of compile/run events during tutoring session showed that students' personality scores, particularly the binary agreeableness score, was a significant predictor of participation from a task-related perspective. The students who were more agreeable (indicated as a 1 for the model, rather than a 0) made more task interactions considering compile/run events as shown in Table 3. The other regression model having the number of content changes as a response variable did not produce significant results.

Number of compile/run =	Coefficient	p
Agreeableness (binarized)	0.2897	0.0392
~1 (intercept)		1
<b>RMSE = 0.967</b>		
$R^2 = 0.0839$		

Table 3: Stepwise linear regression model for number of compile/run events.

Another regression model that showed significant results was the regression model that predicted the number of tutor messages students received. Interestingly, both student perceptions (computer science confidence and motivation) and personality (openness score from Big Five Inventory) were selected by the model as shown in Table 4. There was a negative correlation between computer science confidence and tutor messages, however it was the opposite for computer science motivation. The students who were more motivated to study computer science interacted more with the system, triggering more tutor messages. Also, the students who had low confidence towards programming received less tutor feedback. Figure 2 shows the scatter plots for both computer science motivation and confidence measures.

**Discussion.** Understanding how student characteristics are associated with tutorial dialogue interaction holds great promise for identifying possible disengagement types and taking adaptive action during tutoring sessions to further improve learning effectiveness. The results of the models indicate that as hypothesized, student characteristics such as personality profile were significantly predictive of the student's level of interactivity with the tutorial dialogue system. We found that students' attitudes and personalities have significant relationships with their level of participation in terms of

Number of tutor messages =	Coefficient	p
Age	0.3802	0.0033
Computer science confidence * Openness	-0.5244	0.0008
Computer science motivation Openness	0.5317	0.00006
~1 (intercept)		1
<b>RMSE = 0.739</b>		
$R^2 = 0.52$		

Table 4: Stepwise linear regression model for number of tutor messages received.

both dialogue and task.

Another important finding was that although pre-test was present in all regression models as an independent variable, it was not significantly predictive of either the number of utterances or the task activities. In other words, the level of participation was more correlated to student characteristics than to their incoming knowledge. These results are important for understanding how to better foster interaction with intelligent tutoring systems. If we can identify students who tend to participate less or become disengaged, the system can automatically adapt to these students with scaffolding. For instance, when a student with low motivation toward the task is identified, the tutorial dialogue system might put particular emphasis on moves that are part of "adjacency pairs," such as asking a question and awaiting a student response. Adapting the task may also be appropriate in these cases. By utilizing information that we can glean from quick pre-measures, we may be able to significantly improve the effectiveness of the system.

## 4. CONCLUSION

Adapting to broader populations with varying characteristics is crucial for increasing the use of intelligent tutoring systems and making them more effective. A central challenge is determining the factors that might affect level of participation with intelligent systems. The current literature is far from totally understanding underlying relationships between student characteristics and how they affect system interactions during tutoring. The findings presented here have identified student characteristics such as level of failure avoidance which are particularly strongly associated with low interaction.

Several directions of future work are promising. First, incorporating multiple sources of information such as multimodal features (e.g., posture, gesture, eye gaze) can help us better understand students and respond in real time to engage them in more interactions. Each of these types of features has been shown to contribute to modeling student behavior. Additionally, customizing scaffolding to different learner characteristics is very promising. Modifying the realized utterances delivered to students based on their personality style, gender, and skill are likely to improve interactions with the system. It is important to devise and investigate strategies for learners of all characteristics in order to better engage students and help them learn more.

**Acknowledgments** The authors wish to thank the members of the LearnDialogue group at North Carolina State University for their helpful input. This work is supported in part by the Department of Computer Science at North Carolina State University and the National Science Foundation through Grants IIS-1409639, and the

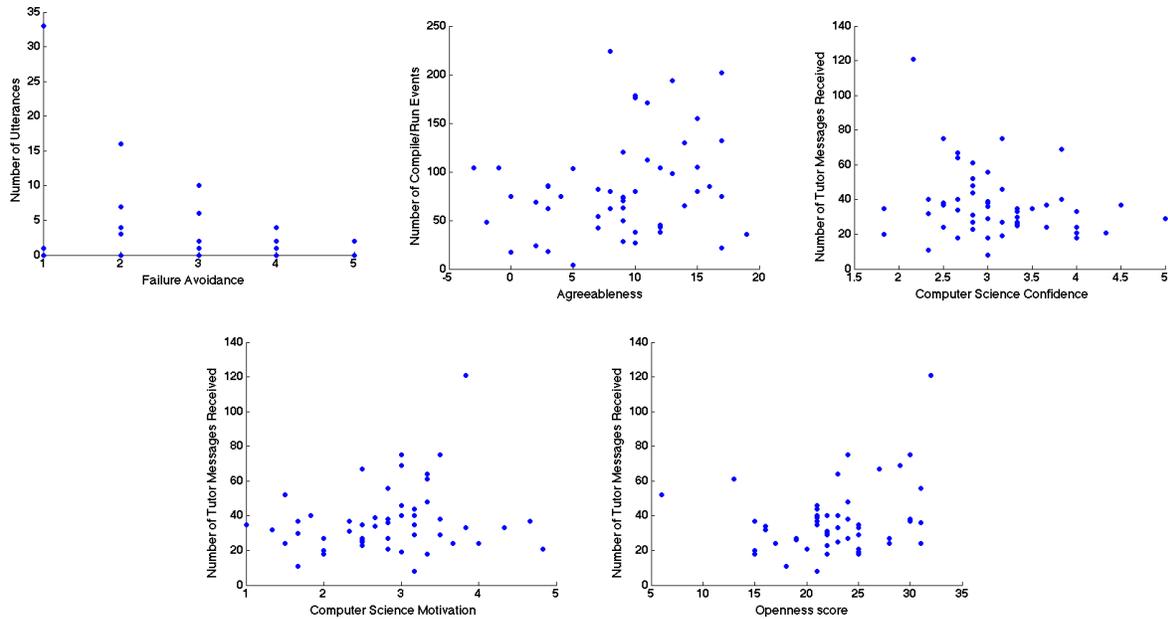


Figure 2: Scatter plots of various predictors and response variables.

STARS Alliance, CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

## 5. REFERENCES

- [1] I. Arroyo and B. P. Woolf. Inferring learning and attitudes from a bayesian network of log file data. In *Proceedings of AIED*, pages 33–40, 2005.
- [2] R. S. Baker, A. de Carvalho, J. Raspat, V. Alevan, A. T. Corbett, and K. R. Koedinger. Educational software features that encourage and discourage “gaming the system”. In *Proceedings of AIED*, pages 475–482, 2009.
- [3] C. R. Beal, L. Qu, and H. Lee. Mathematics motivation and achievement as predictors of high school students’ guessing and help-seeking with instructional software. *Journal of Computer Assisted Learning*, 24(6):507–514, 2008.
- [4] J. E. Beck. Engagement tracing: Using response times to model student disengagement. In *Proceedings of AIED*, pages 88–95, 2005.
- [5] G. Chen, S. M. Gully, and D. Eden. Validation of a new general self-efficacy scale. *Organizational research methods*, 4(1):62–83, 2001.
- [6] S. D’Mello, C. Williams, P. Hays, and A. Olney. Individual differences as predictors of learning and engagement. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, pages 308–313, 2009.
- [7] S. K. D’Mello, B. Lehman, and A. Graesser. A motivationally supportive affect-sensitive AutoTutor. In *New perspectives on affect and learning technologies*, pages 113–126, 2011.
- [8] M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, and G. Campbell. BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *IJAIED*, 24(3):284–332, 2014.
- [9] A. Ezen-Can and K. E. Boyer. A tutorial dialogue system for real-time evaluation of unsupervised dialogue act classifiers: Exploring system outcomes. *To appear*.
- [10] K. Forbes-Riley and D. Litman. When does disengagement correlate with performance in spoken dialog computer tutoring? *IJAIED*, 22(2):19–41, 2008.
- [11] G. T. Jackson, A. C. Graesser, and D. S. McNamara. What students expect may have more impact than what they know or feel. In *Proceedings of AIED*, pages 73–80, 2009.
- [12] O. P. John, L. P. Naumann, and C. J. Soto. Paradigm shift to the integrative Big Five trait taxonomy. *Handbook of personality: Theory and research*, 3:114–158, 2008.
- [13] C. Lee and P. Bobko. Self-efficacy beliefs: Comparison of five measures. *Journal of Applied Psychology*, 79(3):364, 1994.
- [14] D. Litman and S. Silliman. ITSPoke : An Intelligent Tutoring Spoken Dialogue System. In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8, 2004.
- [15] R. Pekrun, T. Goetz, L. M. Daniels, R. H. Stupnisky, and R. P. Perry. Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion. *Journal of Educational Psychology*, 102(3):531, 2010.
- [16] A. K. Vail and K. E. Boyer. Adapting to Personality Over Time: Examining the Effectiveness of Dialogue Policy Progressions in Task-Oriented Interaction. In *Proceedings of the Annual SIGDIAL Meeting*, pages 41–50, 2014.
- [17] D. VandeWalle, W. L. Cron, and J. W. Slocum Jr. The role of goal orientation following performance feedback. *Journal of Applied Psychology*, 86(4):629, 2001.
- [18] K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney, and C. P. Rosé. When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1):3–62, 2007.
- [19] K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, et al. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of ITS*, pages 158–167, 2002.

# Considering the influence of prerequisite performance on wheel spinning

Hao Wan  
Worcester Polytechnic Institute  
100 Institute Road  
Worcester, MA, USA  
hale@wpi.edu

Joseph Barbosa Beck  
Worcester Polytechnic Institute  
100 Institute Road  
Worcester, MA, USA  
josephbeck@wpi.edu

## ABSTRACT

The phenomenon of wheel spinning refers to students attempting to solve problems on a particular skill, but becoming stuck due to an inability to learn the skill. Past research has found that students who do not master a skill quickly tend not to master it at all. One question is why do students wheel spin? A plausible hypothesis is that students become stuck on a skill because they do not understand the necessary prerequisite knowledge, and so are unable to learn the current skill. We analyzed data from the ASSISTments system, and determined the impact of how student performance on prerequisite skills influenced ability to learn postrequisite skills. We found a strong gradient with respect to knowledge of prerequisites: students in the bottom 20% of pre-required knowledge exhibited wheel spinning behavior 50% of the time, while those in the top 20% of pre-required knowledge exhibited wheel spinning behavior only 10% of the time. This information is a statistically reliable predictor, and considering it results in a modest improvement in our ability to detect wheel spinning behaviors: R2 improves from 0.264 to 0.268, and AUC improves from 0.884 to 0.888.

## Keywords

Wheel Spinning; Prerequisite; Student Model.

## 1. INTRODUCTION

Many Intelligence Tutoring Systems (ITS) make use of a mastery learning framework where students continue practicing a skill until they master it. However, some students are unable to achieve mastery despite having numerous opportunities to practice the skill. As a result, these students are stuck in the mastery learning cycle of the ITS and are given additional problems on a topic they are unable to master. We refer to these students as “wheel spinning” on the skill. The term wheel spinning comes from a car that is stuck in snow or mud, and despite rapid movement of the wheels, the car is going nowhere. As defined in [1], a student who takes 10 practice opportunities without mastering a skill is considered to be wheel spinning on this skill. Based on this definition, they also point out that about 31% student-skill pairs in CAT and 38% in ASSISTments are wheel spinning. This earlier work identified the students, but did not provide an explanation for why certain students become stuck. Thus, the next question to address is to

understand why students wheel spin in order to provide effective remediation to those students.

Beck and Gong [1] developed a model, consisting of 8 features, to predict which students will wheel spin on a skill. They found that there is a relationship between wheel spinning and gaming the system [12]. Beck and Rodrigo [2] constructed a causal model (using non-Western students) that situated wheel spinning in the face of affective factors. They found that wheel spinning and gaming were strongly related. This work also presented a path model that found gaming was not causal of wheel spinning, but rather, wheel spinning was related to a lack of prior knowledge, which in turn led to gaming. A more concrete wheel spinning model is developed in [3], in which three aspects of features are considered: student in-tutor performance, the seriousness of the learner, and general factors. However, these models do not provide actionable results for how to make a student less likely to wheel spin on a skill, or how to get an already wheel spinning student unstuck.

A natural question is why are some students able to learn a skill and achieve mastery, while other students fail to do so? One plausible hypothesis of what makes wheel-spinning students different from their peers is a difference in ability to learn the skill. Students certainly differ in cognitive abilities, but addressing such would be beyond the scope of most interventions ITS developers can develop. Another plausible difference in ability to learn the skill is due to differences in student preparation. For example, if students do not understand the concept of equivalent fractions, they will have great difficulty mastering the later skill of addition of fractions, which requires them to solve problems such as  $1/3 + 1/4$ .

We define a skill S's prerequisite skills as those skills necessary to be mastered before studying skill S. This prerequisite structure has been used to improve different student models in many research works. For example, Carmona et al. [4] add a new prerequisite layer into student model based on Bayesian Networks. Their experiments suggest that the prerequisite relationships can improve the model's efficiency in diagnosing students. Botelho et al. use prerequisite structure to estimate students' initial knowledge for subsequent skills [5].

Therefore, in this paper, we incorporate the prerequisite structure into wheel spinning model, in order to check if prerequisite performance has impact in wheel spinning of post-skills. Although prior research has proposed automatic algorithms of adapting prerequisite structures [6] [7] [8], we instead use a prerequisite structure developed by a domain expert.

As an overview, we abstract students' prerequisite performance as a feature, and then add this feature into the wheel-spinning model [1]. Our main points include: 1) determine if there is connection

between the prerequisite performance and the wheel spinning of post-skill; 2) explore how prerequisite factor would affect wheel spinning model; 3) compare the prerequisite factor with another possible effect that could cause wheel spinning – students’ general learning ability. The rest paper is organized as following: Section 2 describes the wheel-spinning model; Section 3 introduces our method of how to represent prerequisite performance; results are shown in Section 4, and further discussion is in Section 5; conclusion and future works are made in Section 6.

## 2. WHEEL SPINNING MODEL

The wheel spinning model used in this work is mainly derived from the one in [1], but there are two differences between them, we will explain later. This model is fitted using logistic regression algorithm in SPSS on the following features:

- a) The number of prior correct responses by the student on this skill. This feature is proved useful in the Performance Factors Analysis model (PFA) [9].
- b) The number of problems in a row correctly responded by the on the skill prior to the current problem. Since for this paper we are operationalizing mastery as 3 correct responses in a row<sup>1</sup>, the number of consecutive correct responses is an important factor. The value of this feature is from 0 to 2.
- c) The exponential mean Z-score of response times on this skill. The response time for each item is transferred into a Z-score, and then exponential mean is calculated for each student by:  $\gamma * \text{prior\_average} + (1 - \gamma) * \text{new\_observation}$ , with  $\gamma = 0.7$  found to work well in practice in prior research, and so we have retained it here.
- d) The exponential mean count of rapid guessing. This measures how often the student was rapidly guessing.
- e) The exponential mean count of rapid response. This measures how often the student took a rapid response. This feature as well as the feature (d) reflects how serious the student is learning the skill through the tutoring system. Similar features related with “gaming” the system were used in gaming detectors as in [10] [11] [12].
- f) Count of bottom-out hint. The number of times the student reached a bottom-out hint on this skill prior to the current problem.
- g) The exponential mean count of 3 consecutive bottom-out-hints. This measures how often the student reached bottom out hints on 3 consecutive problems.
- h) Skill identification.
- i) Prior response count.

As aforementioned, the model in our experiments is different from the Beck and Gong’s model [1] in two places: one is that we use one more feature in the model, the feature b) above; the other is that in some experiments, we treat the last feature – prior response count – as a covariate, not a factor like in their model. We found this parameter’s affect was approximately linear, and thus treating it as a covariate made more sense. We call the model based on these 9 features the baseline model, and compare it with a model that includes the prerequisite performance.

<sup>1</sup> We use this definition for consistency with prior work, and for ease of application across systems. This mastery

## 3. METHOD

### 3.1 Computing Students’ Performance on Skills

In this paper, our goal is to find the influence of students’ prerequisite performance on wheel spinning. So the first step is to choose which measure to represent students’ performance on each skill. In this work, we regard a student’s percentage of correct responses to questions involving a skill to be his performance on that skill.

However, a student could answer correctly, by chance, even though this student does not understand the skill at all. Similarly, a student could give the wrong answer through a careless mistake, as in the guess and slip parameters in the Knowledge Tracing model [13]. These two cases will deviate the student’s performance from his/her “true understanding” on the skill, especially if the student has very few practices. To deal with these cases, we balance the “accidental performance” with student’s overall performance on all skill. The formula for calculating a student’s performance on a skill  $i$  is:

$$P_i = \frac{1}{2^x} * \bar{R} * S_i + \left(1 - \frac{1}{2^x}\right) * C_i$$

- $x$ : The number of practices on this skill;
- $S_i$ : The percent correctness of skill  $i$ ,  $S_i = \frac{\text{\#correct practices}}{\text{\#overall practices}}$  (over all students). This also reflects the hardness of skill  $S_i$ .
- $C_i$ : The student’s percent correctness on skill  $i$ ,  $C_i = \frac{\text{\#correct practices}}{\text{\#overall practices}}$  (over the student  $st_i$ ).
- $R_i = \frac{C_i}{S_i}$ : This represents how well the student  $st_i$  does on skill  $i$  comparing with the other students.
- $\bar{R} = \frac{\sum_{i=1}^m R_i}{m}$ :  $m$  is the number of the student’s started skills.

**Table 1. A small sample of students’ practices.**

Student	Skill	Problem	Correct?
st1	s1	p1	1
st1	s1	p2	0
st1	s2	p3	1
st1	s3	p4	0
st2	s1	p1	1
st2	s1	p2	1
st2	s3	p5	1

**Table 2. Calculated skills’ hardness and students’ performance according to the data in Table 1.**

Skill	Correctness	Student performance		Normalized performance	
		st1	st2	st1	st2
s1	0.75	0.48	1.06	0.45	1
s2	1.0	0.78	1.67	0.47	1
s3	0.5	0.28	0.92	0.3	1

criterion is fairly weak, and presumably underestimates the amount of wheel spinning.

Notice in the formula, the more practices on a skill, the more weight is assigned to the performance on this skill. Take the data in Table 1 as an example. There are in total 4 trials for skill s1, of which 3 are answered correctly, so its correctness is 0.75. The correctness of the other two skills is: s2, 1.0; s3, 0.5. The student, st1, answered two problems of s1, getting one correct and the other incorrect. So this student's correctness of s1 is 0.5, and  $R_1(st1) = \frac{0.5}{0.75} = 0.67$ . We can also get that  $R_2(st1) = 1.0, R_3(st1) = 0$ , then  $\bar{R}(st1) = 0.56$ . Hence, the student st1's estimated understanding on the skill s1 is:  $\frac{1}{2^2} * 0.56 * 0.75 + \left(1 - \frac{1}{2^2}\right) * 0.5 = 0.48$ . All the performance results are shown in Table 2. Sometimes, a student's adjusted performance is larger than 1, as the student st2's performances on skill s1 and s2. This effect can occur by a student doing very well on a very difficult skill. In this paper, we normalize the values to bring them in the range from 0 to 1.

### 3.2 Computing Prerequisite Performance

Once the normalized students' performances have been computed, the next step is to think about how to represent prerequisite performances, and then incorporate it into the wheel-spinning model. If a skill has only one pre-required skill, such a representation is straightforward: the student's adjusted performance on that pre-required skill. But what if a skill has multiple prerequisites? In our data set, 39 out of 128 skills have multiple prerequisites. There are a variety of approaches for handling multiple prerequisites. We chose two different methods to compute the prerequisite performance: weakest link and weighted by hardness.

#### 3.2.1 Weakest Link

This method is based on an assumption that learning a skill requires mastery of all its prerequisites. For example, lack knowledge of square or square root might not solve the Pythagorean equation. Therefore, this method regards the prerequisite skill with the worst performance, called weakest link, as the bottom boundary of estimation of prerequisite knowledge.

In this paper, we use the lowest performance value in all prerequisite skills as the wheel-spinning model's input for prerequisite performance. For example, in Table 1, if skill s1's prerequisite skills are s2 and s3, then the prerequisite performance for student st1 on skill s1 is estimated as 0.3 (normalized).

#### 3.2.2 Weighted by Hardness

This method assumes each prerequisite skill has different importance in affecting learning a post-skill, and this importance is determined by how hard the prerequisite skill is. Thus, we sum up a student's prerequisite performances by assigning a corresponding weight to each prerequisite skill, according to the skill hardness. Here we define a skill's hardness to be  $1/correctness$ . Thus, for a skill, the representation for its prerequisites is calculated as:

$$Pr_i = \frac{\sum_{j=1}^n w_j P_j}{\sum_{j=1}^n w_j}$$

- n: Number of prerequisites.
- $P_j$ : A student's performance on the jth prerequisite.
- $w_j = \frac{1}{S_j}$ : The weight assigned into the jth prerequisite.  $S_j$  is the correctness of this prerequisite.

Suppose we also have the skill s1's prerequisites are s2 and s3, then using the data from Table 1 the student st1's prerequisite performance on skill s1 is:

$$\frac{0.47 * \frac{1}{1} + 0.3 * \frac{1}{0.5}}{\frac{1}{1} + \frac{1}{0.5}} = 0.36$$

Respectively, the student st2's prerequisite representation value for s1 is 1.

### 3.3 Defining General Learning Ability

Our approach is to construct a variable, which we refer to as General Learning Ability (GLA), that encapsulates some of the constructs like diligence, home support, raw ability, and so on. GLA refers to a student's latent ability that affects his ability to learn new skill, similar in spirit to the unidimensional trait in Item Response Theory (IRT) [14]. In IRT, a student's trait is assumed measurable; it is measured through a series of adaptive questions given by a tutoring system.

To simplify our work, we measure student's general learning ability as following steps:

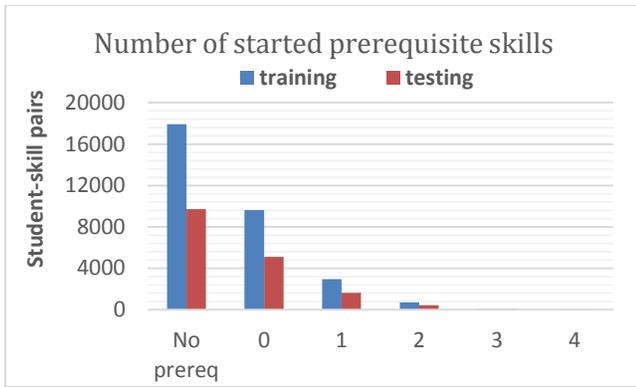
- For each student-skill pair, randomly select the other two started skills. Here a started skill means the student has practiced at least one problem on it;
- Compute the performance values for the two skills, as described in Section 3.1;
- Take the average of those two performance values as the general learning ability for this student-skill pair.

Our intuition in defining GLA in this manner is that if the reason for WH's strong gradient with wheel spinning (Figure 3) is due to the knowledge of the prerequisite being important, we would expect GLA to perform poorly. However, if the power of WH comes not from estimating a particular aspect of student knowledge, but rather than providing a proxy measurement for a student's general ability and willingness to learn, we would expect estimating the student's knowledge of two random skills would work as well. We chose to use two random skills since that was the average number of prerequisites, and we wanted to avoid issues with one measure having lower variability (and hence higher reliability) simply by being an aggregate of more skills. One potential drawback of our approach is that two skills is a small number, and in some cases will certainly provide an over- or under-estimate of knowledge for a particular student. However, since our sample size is large enough, 48256 student-skill pairs in total, this approach is unlikely to produce skewed results.

## 4. RESULTS

### 4.1 Data Set

The data in this work is from ASSISTments. We tracked all ASSISTments students when they used the system to practice Math problems for almost a full year from September 2010 to July 2011. This data set contains 7591 different students, and we randomly select 4976 of the students (about 2/3 of students) to form our training data set, while the other students comprise the testing data. There are 31301 student-skill pairs in the training set and 16955 in the testing set. In this work, we consider students who fail to achieve mastery within 10 practice opportunities for a skill (including indeterminate cases [1]) as wheel spinning, which results in 20.6% instances in the training set as wheel spinning and 19.2% in the testing set.



**Figure 1. Distribution of number of started prerequisite skills in training set and testing set.**

In the training data, there are 177713 problems solved by the students, while 97768 problems in testing data. These problems cover 128 different skills. In the training and testing set, students learn different skills. The maximum number of learned skills by a student is 61, and the average is 6.4. As aforementioned, the prerequisite-to-post skill structure is defined by domain expert as a recommended sequence of topics for instructors. Among the skills in our data set, 66 skills have at least one prerequisite. Some skills have multiple prerequisites, the max number of prerequisites is 8, and the average is 2.4.

However, it is the teacher’s choice which skills and in which order to assign to students. Consequently, the majority of student-skill pairs do not have any started prerequisite skills in our data set, as shown in Figure 1. Apparently (and understandably), teachers are less likely to assign review material than to focus on new topics. The maximum number of started prerequisites is 4, and the average is only 0.37. Thus, our experiments will run over three different data sets:

- D1: the whole data set, as depicted in Figure 1, which is splitted into training and testing set.
- D2: the prerequisite data set. This data set excludes the skills that have no prerequisite skills, as identified by the domain expert, from D1. Thus, it is comprised of the points on the x-axis in Figure 1 corresponding to 0, 1, 2, 3 and 4. It is also splitted into training and testing set, and its training set is constructed from the training set in D1 by removing the non-prerequisite skills, while its testing set from testing set in D1 respectively.
- D3: the *started* prerequisite data set, and includes only student-skill pairs where the student has at least begun one of the prerequisites. This data set excludes the skills that have no started prerequisite skills from D2. Thus, it is comprised of the points on the x-axis in Figure 1 corresponding to 1, 2, 3 and 4. Similarly, its training (testing) set is generated from training (testing) set in D2 by removing non-started-prerequisite skills.

The reason for these three datasets is that they answer different research questions. D1 enables us to investigate the impact of prerequisite performance on wheel spinning in an already-existing system in a real-world deployment. That is, how much benefit would we see in the current usage context of the tutor. Unfortunately, that real-world deployment involves teachers assigning no work on most prerequisites, and thus no information about student prerequisite knowledge is available to the model. D2 enables us to examine where there is at least potential benefit. D3 enables us to answer questions about whether a system that had

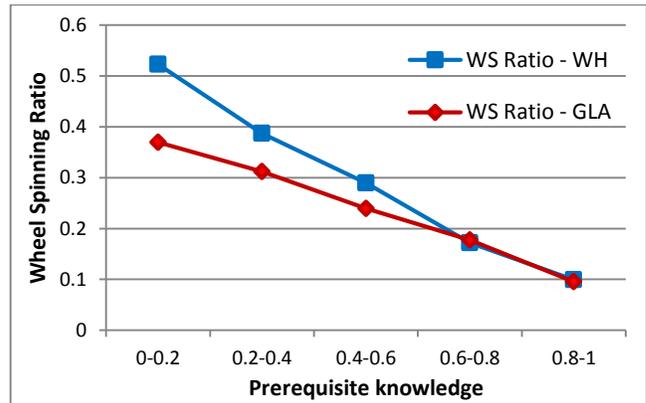
fuller information about prerequisite would perform better at detecting wheel spinning. D3 lets us consider possible changes to policy where teachers are more willing to assign review work, or a system is better able to access past student performance to assess prior knowledge.

## 4.2 Prerequisite Effect on Wheel Spinning

### 4.2.1 The Gradient of the Wheel Spinning Ratio

In order to determine how likely a student will be to wheel spin on a skill based on his corresponding prerequisite performance value, we focus on the training set of D3. We separate D3 into 5 bins according to the prerequisite performance value, calculated by the method weighted by hardness. The wheel spinning ratio in each bin is shown in Figure 2, named WS Ratio - WH.

As observed in the figure, there is a strong gradient with respect to the prerequisite performance: students in the bottom 20% of pre-required knowledge exhibited wheel spinning behavior 50% of the time, while those in the top 20% of pre-required knowledge exhibited wheel spinning behavior only 10% of the time. This expresses strong evidence supporting our hypothesis that student’s wheel spinning on post-skill results from poor preparation for future learning in terms of prerequisite knowledge [15].



**Figure 2. Wheel spinning ratio according with respect to prerequisite knowledge and general learning ability on D3.**

### 4.2.2 Changes in the Model

To test the impact of prerequisite features, we integrated them into the wheel-spinning model described previously. We compare the effects of different factors in the wheel spinning model, Weakest Link (WL), Weighted by Hardness (WH), and General Learning Ability (GLA). Table 3 shows the results of training each model on the training test, and evaluating it on the test set.

In this experiment, we use the Cox and Snell R square [15] and AUC (area under curve) to measure model fit. As we can see, the model does not appreciably change in the data set D1, due to the fact that the part of the data containing started prerequisite skills is such a small component of the data. In D2 and D3, the model is improved slightly by integrating the prerequisite feature, WH or WL. This result supports that prerequisite performance is useful in determining students’ wheel spinning status in postprerequisite-skills. We can also notice that the model with GLA has the similar results with the ones with WH and WL.

Futhermore, to comare the difference between models, a paired t-test is applied on the results at the student’s level of each pair of models, as shown in Table 4. The result shows that adding a

prerequisite factor – WH or WL – into the baseline model makes it performing significantly differently in all data sets, D1, D2, and D3. On the other hand, the model “Baseline+WH” and “Baseline+WL” have the similar results in those three data sets, which also implies these two prerequisite features have similar effect in the wheel spinning model. More interesting, the p-values indicate that the model with GLA is significantly different from the model with WH (or WL respectively) in D1 and D3, but not in D2, and significantly different from the Baseline model in D2, but not in D1 and D3.

**Table 3. Measurements of different models.**

Model	R Square			AUC		
	D1	D2	D3	D1	D2	D3
Baseline	0.285	0.301	0.264	0.879	0.888	0.884
Baseline +WL	0.285	0.302	0.268	0.879	0.889	0.887
Baseline +WH	0.285	0.302	0.268	0.879	0.889	0.888
Baseline +GLA	0.291	0.306	0.268	0.883	0.891	0.887

**Table 4. P-values of paired t-test. In each data set (D1, D2, and D3), we first compute the RMSE for each model predicting over each student. And then the t-test is applied on the RMSE results at the student’s level for each pair of models. The p-values in this table are shown in the order (D1, D2, D3).**

	Baseline	Baseline+WL	Baseline+WH
Baseline +WL	<0.01,<0.01, <0.01		
Baseline +WH	<0.01,<0.01, <0.01	0.62, 0.1, 0.27	
Baseline +GLA	<0.01,<0.01, 0.21	<0.01,0.29, <0.01	<0.01,0.3, <0.01

### 4.2.3 Impact of Prerequisite Effect on the Predictive Model

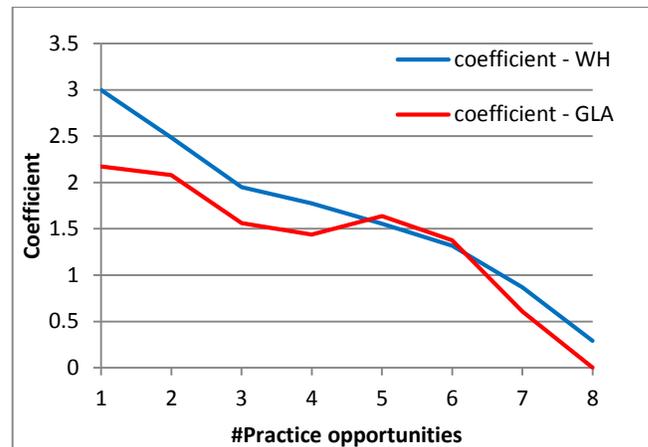
We now move to determining the impact of the prerequisite feature on the predictive model. In our intuition, the prerequisite factor might have strong effect in predicting wheel spinning when a student just starts learning a post-skill, and the effect weakens with time as the student solves problems on the postprerequisite skill

In the logistic regression algorithm, researchers typically use the odds ratio, exponential the coefficient, to represent effect of the corresponding feature [15]. Then the coefficient could be also used to represent the effect on the model. Therefore, in this work, we use the coefficient of prerequisite feature to reflect its effect in predicting students’ wheel spinning on post-skill.

In this experiment, we group the D3 of training set by amount of practice on the skill, and construct a wheel spinning model for each group. The coefficients of prerequisite feature (for the WH model) in the corresponding models are shown in Figure 3. As we can see, the coefficient representing the impact of prerequisite knowledge has the highest value at the beginning, and it decreases in influence as students obtain more practice on the skill. This result support our intuition that the prerequisite factor is a good predictor for wheel spinning only at the beginning stage of learning post-skill.

Thus, prerequisite knowledge is useful for overcoming the cold start problem in student modeling. When a student first starts working on a skill, his performance on that skill provides little basis with whether to classify him as likely to wheel spin or not. In this situation, knowing how he performed on the prerequisite skills provides some information in his ability to master the current material. As the system observes more and more performances on the skill, those performance provide a much more pertinent source of information about the student’s likely trajectory, and the relative importance of prerequisite skills diminishes.

The decrease in in predictive performance for the WH coefficient is monotonic and roughly linear. From a standpoint of statistical significance, the WH coefficient is reliably different than 0 for practice opportunities 1 through 7 ( $p=0.026$  at the 7<sup>th</sup> opportunity). At the 8<sup>th</sup> opportunity, the impact of the WH coefficient has  $p=0.51$ .



**Figure 3. The changes of coefficient with respect to number of practice opportunities on D3.**

## 4.3 Understanding What Prerequisite Performance Really Represents

The performance of the WH feature raises an interesting question: to what does it owe its predictive power. Although we refer to this feature as representing student’s prerequisite knowledge, it captures much more than just knowledge. For example, if one student demonstrates strong performance on prerequisite skills and the other does not, those students probably differ in many dimensions beyond knowledge of the skill: diligence in doing math homework, support at home, raw ability at learning new concepts, and perseverance when stuck. Wrapping this bundle of constructs together and calling it “prerequisite knowledge” certainly simplifies discussion, but does a disservice to accuracy. Therefore, we perform a baseline experiment to investigate what prerequisite knowledge represents.

### 4.3.1 Compare GLA with WH

Since the effects of two prerequisite features, WL and WH, are pretty much the same in the wheel spinning model. Therefore, we will compare only the WH with the GLA. These two features are compared through three different experiments.

The first experiment is to construct wheel spinning ratio gradient for GLA. As we can see in Figure 2, there is the same broad trend for both GLA and WH. For both measures, students with lower general learning ability are more likely to be wheel spinning, which is in accord with our common sense. By comparing the two wheel spinning ratio gradients, we notice that the ratio is the same when the WH and GLA values are high; that is, if a student’s performance

is relative high ( $> 0.6$ ) for WH and GLA, then there is a similar chance the student will wheel spin. However, in the lower range of 0 to 0.6, students are more likely to be wheel spinning according to WH value than the students having the same GLA value. This result suggests that prerequisite factor has stronger correlation with wheel spinning than general learning ability, although general learning ability has strong overlap.

The second experiment is to add the GLA into wheel spinning model and compare the model measurements. According to the results in Table 3, adding the GLA into the baseline model makes more improvement than adding the WH on the data set D1 and D2. This is because the student-skill pairs with pre-required knowledge are very rare in those data sets, while every student-skill pair is assigned with a computed GLA value based on that student's performance on a pair of random skills. The model with GLA and the model with WH on the data set D3 have nearly identical performance.

The third experiment is to compare the effect in the learning procedure. As seen in Figure 3, the GLA coefficient also decreases with respect to the number of practice. But in the first 5 practices, the slope of GLA coefficient is more moderate than the slope of WH coefficient, which defends the statement that the prerequisite factor is useful in predicting wheel spinning at early learning stage. By examine the GLA coefficient Wald statistic p-value, it is also statistically reliable ( $p < 0.05$ ) before the 7<sup>th</sup> practice.

## 5. DISCUSSION AND FUTURE WORK

It should be noticed that even though we found that prerequisite knowledge is related to wheel spinning on post-skills, the general learning ability also has the similar relation. Therefore, it is hard to identify which factor has a stronger connection with wheel spinning in this data set. This is because of two possible reasons: improper prerequisite structure and indirect prerequisite-post relation.

### 5.1 Prerequisite Structure

As aforementioned, the prerequisite structure used in this work is defined by domain experts. Through this structure, the experts suggest a general curriculum over all grades, not specified in a single year or a single class. It is certainly possible that our structure is in error either by missing some links and incorrectly creating others. Such errors would impact the results.

Moreover, in the method of computing prerequisite performance for a post-skill, we assume that the prerequisite skill with the worst performance (or the hardest prerequisite skill) has the strongest influence in learning post-skill. However, this assumption might be inappropriate here. Botelho [5] et al. also illustrate in their experiments that the prerequisite relation in some post-skills are not as stable as expected by domain experts.

Therefore, there are two possible ways of improving our experiments. The first one is to construct a prerequisite structure specifically for the data. Previous works have been focused on this area. For example, Vuong et al. [8] introduce a method for finding prerequisite structure within a curriculum. Their method calculates the overall graduation rate for each unit, and regards Unit A as prerequisite knowledge for Unit B if the experience in Unit A promotes graduation rate in Unit B.

The other possible way is to measure the correlation between each prerequisite skill and a post-skill, and then we can obtain which prerequisite skill is most effective in affecting learning post-skill. Vuong et al. also distinguish the prerequisite relationship between significant and non-significant in their work [8].

## 5.2 Prerequisite-post Relation

Obviously, students' general learning ability influences their performance in both prerequisites and post-skills. Therefore, one might argue that there is no direct causal prerequisite-post relationship. The student who is wheel spun on learning post-skill as well as lack of pre-required knowledge is mainly because he/she has weak learning ability, as shown in Figure 4. In this view, GLA is the primary driver of both prerequisite and postrequisite performance.

According to this argument, a consequent case would be: a student who is wheel spun on a skill, he/she will be wheel spun on every skill, due to the weak learning ability. However, in our data set, the wheel spinning ratio of the students who have at least one wheel spinning case is about 23%. Thus, the GLA is an effective factor in wheel spinning, but not a unique or crucial one. Another drawback of this model is that, for low levels of performance, prerequisite knowledge is more strongly related to wheel spinning than GLA. Therefore, even if GLA is the primary driver, there is apparently some impact of prerequisite knowledge on postrequisite performance, represented by the dotted line in Figure 4.

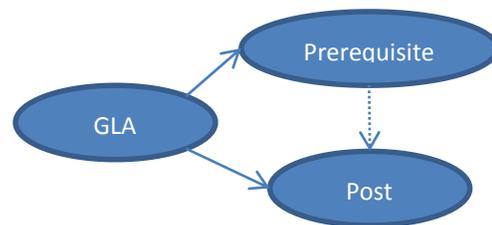


Figure 4. A structure to explain indirect prerequisite-post relationship.

In order to validate the structure in Figure 4, a subtler model should be constructed, in which students' GLA is finely measured. A proper way is to utilize the IRT model to estimate a student's trait; this trait is regarded as the GLA value. And then it is used in predicting if the student will be wheel spinning or not. Meanwhile this trait is updated for each item practiced or for each skill learned. The similar work is in [16], the authors integrate temporal IRT into Knowledge Tracing model, in order to track students' knowledge stage and predict next problem correctness.

## 6. CONTRIBUTIONS AND CONCLUSION

This work makes two contributions. First, it examines the relationship between prerequisite performance and wheel spinning. One plausible hypothesis for why some students are stuck in the mastery learning cycle is due to inadequate preparation in the building block skills. We found such an association, with students who performed less well on the prerequisite skills being more likely to wheel spin. This work represents an advance over what is known about wheel spinning [1][2].

The second contribution of this work is unpacking what is meant by knowledge of prerequisite skills, and discovering that it is not always related to relevant knowledge. Specifically, by showing that two random skills work approximately as well as prerequisite performance, we show that, for this study, the impact is largely due to general properties of the student than the student's knowledge about particular skills. This reasoning is more than a semantic game, as it directly impacts the conclusions we can draw from our data.

Given just the WH line in Figure 2, a reasonable interpretation is that we can reduce wheel spinning by increasing student

prerequisite knowledge, and we could imagine interventions designed to target such. Given the additional context of the results for GLA, we realize that most of the effect attributed to prior knowledge is really just how well the student learns math in general. Unfortunately, interventions to target diligence, grit, math ability, and home support are outside the scope of plausible interventions to deliver with an ITS. However, the difference in the gradients of the two lines suggests there is some benefit from improving student knowledge to at least a moderate level to reduce wheel spinning. This analysis also raises the question of how much work reporting effects related to student prior knowledge is really talking about some other construct than knowledge. Unless the difference in knowledge is caused by a randomized manipulation, differences in knowledge are a proxy for a collection of variables. Hopefully this work will spur EDM researchers to more carefully investigate the meaning of the constructs they are reporting.

In conclusion, this paper investigates the effect of prerequisite performance on wheel spinning and finds that they are related. The addition of prerequisite or GLA features provides a small enhancement in predictive accuracy to our wheel spinning model, improving R<sup>2</sup>, on skills for which we have prerequisite data, from 0.264 to 0.268, and AUC from 0.884 to 0.888. The baseline model results are quite strong for ITS research, so third-decimal improvement in both metrics is fairly good.

This work also found that prerequisite performance and GLA are both effective for overcoming the cold start problem in student modeling. When students begin working on a skill, the tutor has little knowledge of the student's capabilities on that skill. We found that the new factors in our model had the greatest impact when students were first starting to work with a skill, and diminish in importance as we acquire additional data about his knowledge of the skill.

## 7. ACKNOWLEDGEMENTS

We acknowledge funding from NSF (# 1440753, 1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024) grant for ASSISTments and support of the author.

## REFERENCES

- [1] J. E. Beck and Y. Gong, "Wheel-Spinning: Students Who Fail to Master a Skill," in *Proceedings of 16th International Conference, AIED 2013*, Memphis, TN, USA, 2013.
- [2] J. E. Beck and M. M. T. Rodrigo, "Understanding Wheel Spinning in the Context of Affective Factors," in *Proceedings of 12th International Conference, ITS 2014*, Honolulu, HI, USA, 2014.
- [3] Y. Gong and J. Beck, "Towards Detecting Wheel-Spinning: Future Failure in Mastery Learning," in *Learning at Scale 2015*, 2015.
- [4] C. Carmona, E. Millán, J. L. Pérez-de-la-Cruz, M. Trella and R. Conejo, "Introducing Prerequisite Relations in a Multi-layered Bayesian Student Model," in *Proceedings of 10th International Conference, UM 2005*, Edinburgh, Scotland, UK, 2005.
- [5] A. Botelho, H. Wan and N. Heffernan, "The Prediction of Student First Response Using Prerequisite Skills," in *Learning at Scale 2015*, 2015.
- [6] E. Brunskill, "Estimating Prerequisite Structure From Noisy Data," in *EDM*, 2011.
- [7] P. I. Pavlik Jr, H. Cen, L. Wu and K. R. Koedinger, "Using Item-Type Performance Covariance to Improve the Skill Model of an Existing Tutor," in *Proceedings of the 1st International Conference on Educational Data Mining*, Montreal, Canada, 2008.
- [8] A. Vuong, T. Nixon and B. Towle, "A Method for Finding Prerequisites Within a Curriculum," in *EDM*, 2011.
- [9] P. I. Pavlik Jr, H. Cen, L. Wu and K. R. Koedinger, "Performance Factors Analysis - A New Alternative to Knowledge Tracing," in *the 14th International Conference on Artificial Intelligence in Education*, Brighton, UK, 2009.
- [10] I. Arroyo and B. P. Woolf, "Inferring learning and attitudes from a Bayesian Network of log file data," in *Artificial Intelligence in Education*, 2005.
- [11] Y. Gong, J. E. Beck, N. T. Heffernan and E. Forbes-Summers, "The impact of gaming (?) on learning at the fine-grained level," in *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010)*, 2010.
- [12] R. S. J. d. Baker, A. T. Corbett, I. Roll and K. R. Koedinger, "Developing a generalizable detector of when students game the system," *User Modeling and User-Adapted Interaction*, vol. 18, no. 3, pp. 287-314, 2008.
- [13] A. T. Corbett and J. R. Anderson, "Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge," *User Modeling and User-Adapted Interaction*, pp. 253-278, 1995.
- [14] S. E. Embretson and S. P. Reise, *Item Response Theory for Psychologists*, Psychology Press, 2013.
- [15] R. S. J. D. Baker, S. M. Gowda, A. T. Corbett and J. Ocumpaugh, "Towards automatically detecting whether student learning is shallow," in *Intelligent Tutoring Systems*, 2012.
- [16] D. W. Hosmer Jr and S. Lemeshow, *Applied Logistic Regression*, John Wiley & Sons, 2004.
- [17] Y. Huang, J. González-Brenes and P. Brusilovsky, "General Features in Knowledge Tracing to Model Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge," in *Educational Data Mining 2014*, 2014.

# Comparing Novice and Experienced Students within Virtual Performance Assessments

Yang Jiang, Luc Paquette, Ryan S. Baker  
Teachers College, Columbia University  
525 W 120<sup>th</sup> Street  
New York, NY 10027  
yj2211@tc.columbia.edu  
paquette@tc.columbia.edu  
baker2@exchange.tc.columbia.edu

Jody Clarke-Midura  
Utah State University  
2830 Old Main Hill  
Logan, UT 84322  
jody.clarke@usu.edu

## ABSTRACT

Inquiry skills are an important part of science education standards. There has been particular interest in verifying that these skills can transfer across domains and instructional contexts [4,15,16]. In this paper, we study transfer of inquiry skills, and the effects of prior practice of inquiry skills, using data from over 2000 middle school students using an open-ended immersive virtual environment called Virtual Performance Assessments (VPAs) that aims to assess science inquiry skills in multiple virtual scenarios. To this end, we assessed and compared student performance and behavior within VPA between two groups: novice students who had not used VPA previously, and experienced students who had previously completed a different VPA scenario. Our findings suggest that previous experience in a different scenario prepared students to transfer inquiry skills to a new one, leading these experienced students to be more successful at identifying a correct final conclusion to a scientific question, and at designing causal explanations about these conclusions, compared to novice students. On the other hand, a positive effect of novelty was found for motivation. To better understand these results, we examine the differences in student patterns of behavior over time, between novice and experienced students.

## Keywords

Virtual environment, science inquiry, educational data mining, sequential pattern mining, transfer, novelty effect.

## 1. INTRODUCTION

One of the important goals for science education is to help students develop the scientific knowledge and practices needed to actively and effectively engage in science inquiry. As such, science inquiry skills have been a critical component of the K-12 science curriculum standards [18]. It is particularly crucial that students acquire inquiry skills which are not specific to a domain or instructional context, but which can transfer broadly, preparing students for using science and understanding science in their future schooling, and in their lives [4, 15, 16].

With the increasing popularity of online learning systems that engage learners in science inquiry activities [e.g., 7, 21],

Educational Data Mining (EDM) techniques have proven effective in automatically assessing science inquiry skills. Sao Pedro et al. demonstrated that science inquiry skills can be assessed within online learning activities using EDM, predicting future performance not only within the same domain [21], but also across domains [22].

Many studies of student inquiry behavior have been conducted within open-ended online learning environments, such as virtual environments, in which users have the freedom to decide their own inquiry behaviors. This, combined with the fact that these open-ended environments are typically more loosely-scaffolded and coarser-grained than more tightly-scaffolded systems such as intelligent tutoring systems or simulations [e.g., 21], makes the assessment of science inquiry in these contexts challenging. Sequential Pattern Mining [1], a methodology that has been extensively used in EDM [23], has shown potential in discovering complicated patterns of learning behavior within open-ended learning environments. For example, Kinnebrew and colleagues [13] applied sequential pattern mining techniques to log data produced by students engaging in activities within Betty's Brain, an open-ended learning environment for science learning. This allowed them to study the differences in students' productive and unproductive learning behaviors by identifying frequent sequential patterns related to the use of concept maps and determining which sequential patterns were characteristic of high-performing students as compared to low-performing students. Differential pattern mining was also used by Sabourin and colleagues [20] to analyze the differences in inquiry behaviors utilized by learners depending on their level of self-regulation within a virtual environment. In another study, Gutierrez-Santos et al. [10] conducted analysis of student actions to detect repetitive sequences in an open-ended learning environment.

Another EDM approach that has proven useful for the study of inquiry behaviors in open-ended contexts involves in-depth analysis of features distilled from log data. For instance, Baker and Clarke-Midura [2] distilled a set of features related to inquiry behavior from log data in Virtual Performance Assessments (VPAs), an open-ended immersive virtual environment used in the current study, to develop predictive models of student success on two inquiry tasks. The current study combines both sequential pattern mining and analysis of features related to science inquiry to study transfer of inquiry skills. In doing so, we also analyze differences in inquiry behavior between novice students and experienced students.

The degree of student experience with an environment can also be hypothesized to have important impact on their inquiry. Clark [6] argued that novelty effect occurs when new computer programs

are introduced. In those cases, the novel computer programs initially attract student attention, leading to increased efforts invested, persistence, motivation, and achievement gains. Previous studies [e.g., 8, 12, 24] indicated that students showed greater initial enthusiasm and motivation in classrooms when novel educational technologies were introduced. This enthusiasm gradually diminished as students were more familiar with the technologies and the initial novelty effect wore off. Therefore, in our study, we investigate whether relative novelty created by the introduction of a new 3D virtual environment will lead to differences in motivation and learning between novice students and experienced students. We also study the relationship between the potential novelty effect and inquiry skills.

To research these questions, we assess and compare student performance and behavior within VPA between two groups: novice students who had not used VPA previously, and more experienced students who had previously spent one class session completing a different VPA scenario. We compare student performance on two inquiry skills – identifying a correct final claim and designing causal explanations. We also compare student responses to a motivation survey between the two groups. Finally, we analyze the difference in student behavior between the two groups using differential sequence mining.

## 2. VIRTUAL PERFORMANCE ASSESSMENTS

This study was conducted within the context of Virtual Performance Assessments (VPAs; see <http://vpa.gse.harvard.edu>). VPAs are online 3D immersive virtual environments, designed using the Unity game development engine [26] that assess middle school students' science inquiry skills, in line with state and national standards for science content and inquiry processes. Within VPAs, whose interface is similar to that of video games, students engage in authentic inquiry activities and solve scientific problems by navigating around the virtual environment as an avatar, making observations, interacting with non-player characters (NPCs), gathering data, and conducting laboratory experiments. VPAs enable automated and non-intrusive collection of process data (logged actions and behaviors) and product data (student final claims), facilitating the capture and assessment of science inquiry *in situ*.

Multiple VPA assessment scenarios have been developed. In this study, two scenarios were used, the frog scenario and the bee scenario. In the frog scenario (see Figure 1), students are presented with a six-legged frog in the virtual environment and have to collect and reason through evidence to determine what is causing the frog's mutation, selecting from a set of possible causal factors including parasites (the correct causal explanation), pesticides, pollution, genetic mutation, and space aliens. In this scenario, students can talk with NPCs from four virtual farms who provide conflicting opinions, collect items such as frogs, tadpoles, and water samples at each farm, run laboratory experiments on water quality, frog blood and DNA, and read informational pages from a research kiosk. Once students think that they have sufficient data, they submit a final conclusion on the causal factor resulting in the mutation, and justify their final claim with supporting evidence. In the bee scenario, students must determine what causes the death of a local bee population. Similar to the frog scenario, they can talk with NPCs from four different farms, read informational pages at the research kiosk, and conduct tests (e.g., nectar test, protein test, genetic test) on the items they have collected at the farms (e.g., nectar samples, bees, larvae). By the

end of the assessment, students choose a final claim about the cause of the bee deaths from possible hypotheses including genetic mutation (the correct causal factor), parasites, pesticides, pollution, and space aliens, and support their final claim with evidence. The activities in each VPA scenario are deliberately similar, allowing researchers to assess performance of the same inquiry practices in different contexts.



Figure 1. Screenshots of the VPA frog scenario.

## 3. DATA SET

Data for this study was composed of action logs produced by middle school students who used Virtual Performance Assessments within their science classes at the end of the 2011-2012 school year. A total of 2,431 students in grades 7-8 (12-14 years old) from 138 science classrooms (40 teachers) participated in this study. These students were from a diverse range of school districts in the Northeastern and Midwestern United States, and Western Canada. A total of 1,985 students completed the frog scenario and 2,023 students completed the bee scenario, with 1,579 students completing both scenarios. Overall, students completed 423,616 actions within the frog scenario and 396,863 actions within the bee scenario. They spent an average of 30 minutes and 47 seconds ( $SD = 14$  minutes, 6 seconds) in the frog scenario and an average of 26 minutes and 5 seconds ( $SD = 12$  minutes, 27 seconds) in the bee scenario.

The 2,431 students were randomly assigned to begin with either the frog scenario or the bee scenario. Two weeks later, they were assigned to complete the other scenario. Therefore, within each scenario, participants could be put into two groups – novice users who were using VPA for the first time (*novice* group), and experienced users who had previously experienced the other VPA scenario (*experienced* group). Accordingly, among the 1,985 students who completed the frog scenario, 1,232 completed the frog scenario as their first scenario (frog-novice) and 753 had previous experience in the bee scenario (frog-experienced). Among the students who completed the bee scenario, 1,198 students had no previous experience in the frog scenario (bee-novice), whereas 825 had previous experience in the frog scenario (bee-experienced). Student actions and performance in the virtual environment were logged as they worked within each VPA scenario and used for later analyses.

## 4. OVERALL ANALYSIS

In this section, we compare student performance on identifying a correct final claim and constructing causal explanations, the amount of time spent on VPA, and students' motivation level, between the novice group and the experienced group, within each VPA scenario.

### 4.1 CFC and DCE Performance

To explore the transfer of student science inquiry skills between scenarios, two measures of student performance within the VPAs were collected and compared between the two groups of students within each scenario: 1) the correctness of the student's final claim (CFC) on the cause of the six-legged frog or the death of the

bees; and 2) student's success in designing causal explanations (DCE) for why that claim is correct.

In each VPA, students submitted a final claim by choosing from five possible causal factors. A student's final claim was considered correct if the student concluded that the mutation of the six-legged frog was caused by parasites (correct causal factor), or that the bee deaths were caused by genetic mutation (correct choice). Otherwise, if the student selected the other potential hypotheses, the student's final claim was considered incorrect. Overall, 29.6% of students correctly concluded that parasites led the frog to have six legs, and 28.3% of students made a correct claim on what was killing the bee population. In this paper, a chi-square test was conducted to compare student CFC performance between the two groups in each scenario.

In the bee scenario, 34.8% of experienced students who had previously used the frog scenario identified correctly that genetic mutation was killing the bees, while 23.9% of novice students (*without* prior experience in the frog scenario) made the correct final conclusion. This difference was statistically significant according to a chi-square test,  $\chi^2(1, N = 2023) = 28.67$ ,  $p < .001$ . Logistic regression results revealed that the odds of making a correct final claim for experienced students (0.533) was statistically significantly larger than the odds for novice students (0.314) by 70%. This suggested that the students transferred what they learned about how to make a correct final claim from the frog scenario to the bee scenario.

Similarly, in the frog scenario, a statistically significantly higher percentage of experienced students (33.2%) made a correct final claim than the percentage of novice students (27.5%) who made a correct conclusion,  $\chi^2(1, N = 1985) = 7.45$ ,  $p = .006$ . Logistic regression results indicated that previous experience in the bee scenario significantly improved the odds of making a correct final claim in the frog scenario by 31.5% (odds = 0.378 for novice group and 0.497 for experienced group).

The DCE measure evaluates student ability in supporting final conclusions with evidence. By the end of the assessment in each scenario, students needed to select the evidence that supported their claims from the data they had collected within the virtual world and the results of laboratory tests they had conducted. They were then presented with all possible data (including data that the students did not collect/conduct) and asked to identify the evidence supporting their claim. In each VPA scenario, most evidence was consistent with the correct causal claim. However, for the incorrect claims, there was often evidence consistent with those claims along with counter-evidence that conclusively disproved those hypotheses. Therefore, even if students were unsuccessful in identifying the correct final conclusion, partial credit would be awarded to them for the quality and quantity of the causal evidence they identified in support of their claim from the non-causal data and results. Student success in selecting evidence and constructing causal explanations were aggregated into a single composite DCE measure that ranges from 0 to 100%, by averaging across the use of each piece of evidence. The mean DCE score for the frog scenario was 50.0% ( $SD = 23.3\%$ ), and the average DCE score for the bee scenario was 46.1% ( $SD = 21.4\%$ ). A two-tailed Mann-Whitney U test, a nonparametric alternative to t-test, was then conducted to compare student ability in designing causal explanations between the two groups in each scenario.

Results of the Mann-Whitney U test comparing the DCE score between the two groups in the bee scenario suggested that the experienced group had a significantly higher average DCE score

( $M = 48.9\%$ ,  $SD = 19.3\%$ ) than the novice group ( $M = 44.2\%$ ,  $SD = 23.8\%$ ),  $U = 453873$ ,  $Z = -3.12$ ,  $p = .002$ . Further analyses revealed that the difference in DCE performance was dependent on the correctness of final claims. Among students who made a correct final claim in the bee scenario, the experienced group achieved significantly higher DCE scores ( $M = 75.1\%$ ,  $SD = 18.3\%$ ) than the novice group ( $M = 68.1\%$ ,  $SD = 20.5\%$ ),  $U = 32448.5$ ,  $Z = -4.34$ ,  $p < .001$ . However, among students who did not make a correct final claim, the novice group showed higher DCE scores ( $M = 36.7\%$ ,  $SD = 11.2\%$ ) than the experienced group ( $M = 34.9\%$ ,  $SD = 11.4\%$ ),  $U = 223797$ ,  $Z = -2.80$ ,  $p = .005$ .

In the frog scenario, student performance in designing causal explanations for the novice group ( $M = 49.7\%$ ,  $SD = 22.7\%$ ) was not statistically significantly different from the experienced group ( $M = 50.6\%$ ,  $SD = 24.3\%$ ),  $U = 454398$ ,  $Z = -.76$ ,  $p = .446$ .

## 4.2 Time

As each VPA scenario logged the timing of each student starting and exiting the virtual environment, we also compared the total amount of time students spent within VPA recorded by the log data between the novice group and the experienced group, by employing one-way ANOVA.

An analysis of variance showed that, on average, novice students without previous experience in the frog scenario spent significantly more time in the bee scenario ( $M = 27$  minutes, 43 seconds,  $SD = 11$  minutes, 56 seconds) than experienced students who had used the frog scenario ( $M = 23$  minutes, 43 seconds,  $SD = 12$  minutes, 48 seconds),  $F(1, 2021) = 51.64$ ,  $p < .001$ . On the other hand, the total amount of time spent in the frog scenario by novice students ( $M = 30$  minutes, 56 seconds,  $SD = 14$  minutes, 24 seconds) and experienced students ( $M = 30$  minutes, 33 seconds,  $SD = 13$  minutes, 35 seconds) was not statistically significantly different ( $F(1, 1983) = .36$ ,  $p = .548$ ).

## 4.3 Motivation

In this study, students completed an online motivation survey shortly after they finished the VPA assessment for each scenario. Student responses to the survey were analyzed to better understand the impact of experience with the environment on learning and motivation. The survey was adapted from the Intrinsic Motivation Inventory [IMI; 27] and the Player Experience of Need Satisfaction [PENS; 19] survey and was comprised of 27 six-point Likert-type items that aimed to measure seven components related to student motivation, autonomy, and in-game immersion: interest/enjoyment, perceived competence, effort/importance, pressure/tension, value/usefulness, presence/immersion, and autonomy. Items were slightly modified to fit the specific activity in this game-like environment. Student subscale scores were calculated by averaging across all items on each subscale. One-way ANOVA was applied to assess whether there were any systematic differences in student motivation between the novice group and the experienced group within each VPA scenario. Given the substantial number of statistical tests, we controlled for the proportion of false positives by applying Storey's q-value method [25] (calculated using the QVALUE package for R).

Analyses of motivational survey results (see Table 1) indicated that, on average, novice students scored significantly higher on the interest/enjoyment subscale than experienced students in both scenarios ( $F(1, 1800) = 50.02$ ,  $q < .001$  for the frog scenario;  $F(1, 1740) = 27.67$ ,  $q < .001$  for the bee scenario). Similarly, students

in the novice group had a significantly higher level of perceived effort invested to the VPA activity and perceived importance of the activity than students in the experienced group ( $F(1, 1800) = 25.41, q < .001$  for the frog scenario;  $F(1, 1740) = 18.94, q < .001$  for the bee scenario). Novice students also regarded the VPA activity as more useful and valuable than experienced students,  $F(1, 1800) = 19.37, q < .001$  for the frog scenario;  $F(1, 1740) = 4.66, q = .019$  for the bee scenario. Finally, novice students also had significantly higher presence/immersion, autonomy, and tension/pressure subscale scores than the experienced students, indicating that they were more immersed in the virtual environment, and felt a higher sense of autonomy and a higher level of tension/pressure than experienced students. These corresponded to previous findings on novelty effect [8, 12].

**Table 1. Average subscale scores on the motivational survey (standard deviations in parentheses) by condition. Differences that are sig. after post-hoc controls ( $q < 0.05$ ) are marked by \*.**

Subscale	Frog-N	Frog-E	F (q)	Bee-N	Bee-E	F (q)
int/enj	4.47 (1.32)	3.98 (1.55)	50.02* ( $<.001$ )	4.26 (1.42)	3.87 (1.56)	27.67* ( $<.001$ )
comp	4.28 (1.21)	4.23 (1.37)	0.73 (.213)	4.13 (1.27)	4.14 (1.37)	0.006 (.473)
eff/imp	4.38 (1.19)	4.06 (1.44)	25.41* ( $<.001$ )	4.21 (1.30)	3.91 (1.49)	18.94* ( $<.001$ )
val/use	4.07 (1.41)	3.74 (1.62)	19.37* ( $<.001$ )	3.84 (1.51)	3.67 (1.64)	4.66* (.019)
pres/ten	1.86 (1.25)	1.72 (1.39)	4.62* (.019)	1.85 (1.29)	1.69 (1.38)	5.86* (.011)
pres/imm	3.51 (1.36)	3.16 (1.53)	24.72* ( $<.001$ )	3.36 (1.42)	3.13 (1.53)	10.14* (.001)
auto	4.26 (1.29)	3.82 (1.55)	41.12* ( $<.001$ )	4.01 (1.41)	3.76 (1.56)	11.42* (.001)

Note. Frog-N = frog-novice, Frog-E = frog-experienced, Bee-N = bee-novice, Bee-E = bee-experienced. Int/enj=interest/enjoyment, comp=perceived competence, eff/imp=effort/importance, pres/ten=pressure/tension, val/use=value/usefulness, pres/imm=presence/immersion, auto= autonomy.

## 5. USAGE ANALYSIS

In the previous section, differences were found in motivation and learning outcomes between novice and experienced students. In the current section, we aim to go beyond just looking at whether previous experience in VPA improved student inquiry performance, and instead look into whether more experienced students used VPAs differently than less experienced students.

For example, this will allow us to determine whether the higher success for experienced students within VPAs was related to the acquisition and transfer of science inquiry skills, or whether it was merely the result of increased familiarity and proficiency with using the system and tools than novice users.

We studied these questions by investigating the prevalence of specific behaviors between groups, and by applying sequential pattern mining to identify and compare the frequent sequential patterns of student actions between groups.

### 5.1 Comparing Behaviors Between Groups

In order to understand student behavior, and how it differed between groups, a set of 30 semantically meaningful features of student behavior thought to potentially differ between groups were distilled from raw interaction data and were compared between the novice and experienced groups in each scenario. These features were a subset of the 48 features that were used to build models predicting a student's CFC and DCE performance within the frog scenario in [2]. Examples of these features will be given in the following paragraphs.

After distilling the 30 features from each student's interaction logs, t-tests were conducted to compare the value of each feature between the experienced and novice groups, within each scenario. Storey's q-values [25] were calculated to control for multiple comparisons. Table 2 presents the average values of 10 features that strongly differentiated between groups.

According to the results, features representing the maximum or average fullness of a student's backpack in the frog scenario, both including repeats (e.g. picking up two green frogs counts as two objects), and not including repeats (e.g. two green frogs counts as one object), had significantly higher value for the novice group than the experienced group. Similar results were found in terms of the number of times a student went to the lab to run tests, the number of different (types of) non-sick frogs that the student took to the lab at the same time, the number of times that lab water was taken to the lab, and the percentage of time the student spent at farms to collect evidence in the frog scenario. Similarly, novice students in the bee scenario had higher values on all these features than experienced students. This suggested that novice students collected significantly more data for testing and spent a larger proportion of time on collecting evidence in farms than the experienced students in both scenarios. This finding was consistent with the higher motivation level of novice students (in both scenarios) and the longer time they spent working on VPA

**Table 2. Comparisons of features between novice group and experienced group. Sig. differences ( $q < 0.05$ ) are marked by \*.**

Feature	Frog-N	Frog-E	t	q	Bee-N	Bee-E	t	q
The number of times student went to the lab	6.66	5.14	6.81	$<.001^*$	16.37	12.71	8.97	$<.001^*$
Maximum number of items (including repeats) in backpack	7.48	6.69	11.25	$<.001^*$	6.03	4.76	11.57	$<.001^*$
Maximum number of items (not including repeats) in backpack	7.45	6.65	11.68	$<.001^*$	8.54	7.28	12.27	$<.001^*$
Average number of items (including repeats) in backpack	4.77	4.02	11.39	$<.001^*$	3.86	3.06	11.91	$<.001^*$
Average number of items (not including repeats) in backpack	4.75	4.00	11.50	$<.001^*$	6.17	5.14	11.61	$<.001^*$
Number of times that lab water/nectar was taken to the lab	0.42	0.38	2.11	.022*	1.69	0.93	8.31	$<.001^*$
Number of different (types of) non-sick frogs/bees student took to the lab at the same time	1.87	1.70	2.34	.014*	4.32	3.90	4.09	$<.001^*$
How long, on average, did students spend reading information pages? (average per read)	15.28	17.17	-0.72	.146	11.93	13.93	-2.07	.027*
How long, in total, did student spend reading information page on correct hypothesis?	32.33	35.13	-0.70	.146	23.45	27.46	-2.20	.021*
Percentage of time student spent at farms	0.29	0.26	4.43	$<.001^*$	0.34	0.31	5.46	$<.001^*$

(in the bee scenario).

Despite the fact that the novices collected more data and spent more total time within the VPA bee scenario, they spent significantly less time on reading an information page at the research kiosk each time they accessed the page ( $M = 11.93$  seconds,  $SD = 17.69$  seconds) than experienced students ( $M = 13.93$  seconds,  $SD = 25.48$  seconds),  $t(2021) = -2.07$ ,  $q = 0.027$ ,  $Cohen's D = 0.15$ . In specific, experienced students spent more time in total reading the information page on the correct hypothesis – genetic mutation ( $M = 27.46$  seconds,  $SD = 46.51$  seconds) compared to novice students ( $M = 23.45$  seconds,  $SD = 35.46$  seconds),  $t(2021) = -2.20$ ,  $q = 0.021$ ,  $Cohen's D = 0.11$ . Gaining more information about the correct hypothesis might have contributed to the students' domain-specific knowledge base, which had been found to be crucial for problem solving and the development of expertise [5]. However, the corresponding pattern was not statistically significant for the frog scenario, probably due to higher standard deviations.

## 5.2 Sequential Pattern Mining

In this section, we investigate patterns in behavior by the two groups, over time. Prior to performing sequential pattern mining, detailed raw action log data were transformed into more abstract sequences. This involved three steps. First, a set of actions related to science inquiry were identified from the log files, including picking up and inspecting objects (e.g., frogs, tadpoles, bees, larvae, water sample, nectar sample) within VPA (*inspect*), talking with NPCs (*talk*), saving objects to backpack (*save*), discarding objects (*discard*), opening and reading informational pages at the research kiosks (*read*), running laboratory tests (*blood/protein test*, *water/nectar sample test*, *genetic test*), reviewing and looking at test results (*look*), starting to answer final questions (*start final questions*), and submitting a final claim (*final claim*). Some actions that were irrelevant to the inquiry process, such as selecting an avatar, closing the scratchpad, and entering/exiting a specific area were filtered out from the raw interaction data. Second, as in [13], repeated actions that occurred more than once in succession were distinguished from a single action and were labeled as the “action” followed by the “-MULT” suffix. This adjustment prevents frequent patterns from being overlooked merely due to differences in how many times the action is repeated. Last, the actions were represented as sequences of actions for each student in each group.

Simple two-action sequential patterns were identified using the *arules* package [11] within the statistical software program R. *Arules* was used to determine the most frequent short sequences of two actions by selecting the temporal associations of one specific action and a subsequent action with the highest support and confidence. In this study, sequential patterns of consecutive actions were selected with the cut-off thresholds of support = 0.0005 and confidence = 0.1.

In the frog scenario, a total of 51 short sequential patterns (length = 2) were identified that met the minimum support and confidence constraints within the novice group; 54 patterns were identified within the experienced group. In the bee scenario, 55 short sequential patterns met the minimum constraints within the novice group; 59 were selected within the experienced group. These patterns were similar across the 4 conditions, and most had support and confidence considerably higher than the threshold. They were then ordered according to their *Jaccard* similarity coefficient – a measure of the patterns' interestingness [17] that was found to be the most highly correlated with human judgments [3] – to find interesting sequential patterns. According to [3], lower *Jaccard* measures indicated higher interestingness.

To facilitate the comparison of the frequency measures between the novice group and the experienced group, the support and confidence for each pattern were calculated separately for each student. Mann-Whitney U tests that controlled for multiple comparisons were then conducted to compare the metric values between two groups in each scenario.

Table 3 presents the comparison of the support and confidence levels of 9 frequent sequential patterns with low *Jaccard* measure (indicating high interestingness) across conditions that were considered as meaningful due to the nature of the actions they contained. The sequential patterns with the lowest *Jaccard* included patterns related to making final claims (*final claim*) or starting to answer final questions (*start final questions*) and reading informational pages (*read*), such as “*final claim* → *read-MULT*”, “*final claim* → *read*”, “*read-MULT* → *final claim*”, “*start final questions* → *read-MULT*”, and “*start final questions* → *read*”. These patterns indicated that students tended to review research and read informational pages as resources to assist with their decision-making before submitting a final claim, or that they used the research information to check and monitor the claims they had just made. All these 5 patterns appeared to have higher support for experienced students than novice students within each

**Table 3. Comparison of the support and confidence of 9 frequent sequential patterns between novice and experienced conditions. Average support/confidence values, and post-hoc controlled sig. of tests are presented. Sig. differences ( $q < 0.05$ ) are marked by \*.**

Pattern	support			confidence			support			confidence		
	Frog-N	Frog-E	q	Frog-N	Frog-E	q	Bee-N	Bee-E	q	Bee-N	Bee-E	q
final claim → read-MULT	.0033	.0043	.420	.296	.313	.594	.0030	.0036	.619	.326	.298	.420
read-MULT → final claim	.0061	.0074	.584	.114	.109	.619	.0055	.0064	.675	.101	.109	.594
final claim → read	.0020	.0026	.675	.164	.158	.675	.0014	.0024	.018*	.142	.193	.107
start final questions → read-MULT	.0046	.0047	.594	.282	.261	.594	.0044	.0049	.675	.274	.257	.594
start final questions → read	.0029	.0033	.682	.160	.167	.675	.0025	.0027	.675	.147	.142	.675
look-MULT → read-MULT	.0027	.0032	.718	.143	.176	.517	.0028	.0030	.594	.141	.189	.309
look → read	.0025	.0028	.711	.103	.142	.214	.0017	.0021	.675	.080	.107	.361
look → read-MULT	.0027	.0033	.675	.113	.158	.073	.0019	.0027	.594	.105	.155	.018*
look-MULT → read	.0021	.0021	.594	.104	.117	.675	.0021	.0017	.018*	.106	.101	.420

scenario, but most of the differences were not statistically significant. In the bee scenario, the pattern *final claim* → *read* showed significantly higher support and marginally significantly higher confidence for the experienced group than the novice group (for *support*,  $M_s=0.024$  and  $0.014$ ,  $U=474169.5$ ,  $Z=-3.03$ ,  $q=0.018$ ; for *confidence*,  $M_s=0.193$  and  $0.142$ ,  $U=46833.5$ ,  $Z=-2.32$ ,  $q=0.107$ ). This finding indicated that experienced students who had previously used the frog scenario were more likely to review research and read information, possibly to monitor their answers and reflect on previous steps [cf. 15], after submitting a final claim in the bee scenario than novice students. However, this trend was not replicated in the frog scenario (for *support*,  $M_s=0.0026$  and  $0.0020$ ,  $U=462294.5$ ,  $Z=-.23$ ,  $q=.675$ ; for *confidence*,  $M_s=0.158$  and  $0.164$ ,  $U=58423.5$ ,  $Z=-.32$ ,  $q=.675$ ).

Another four interesting sequential patterns corresponded to looking at laboratory test results (once or repeatedly), followed by reading informational pages (once or repeatedly) (i.e., *look-MULT* → *read-MULT*, *look* → *read*, *look* → *read-MULT*, *look-MULT* → *read*). For three out of the four patterns, both the support and the confidence for the experienced group were higher than those for the novice group in both scenarios. For the pattern *look* → *read-MULT*, the confidence for the experienced group was statistically significantly higher than that for the novice group in the bee scenario and marginally higher than confidence for the novice group in the frog scenario (in bee scenario,  $M_s=0.105$  and  $0.155$ ,  $U=94500.5$ ,  $Z=-3.09$ ,  $q=.018$ ; in frog scenario,  $M_s=0.113$  and  $0.158$ ,  $U=111697.5$ ,  $Z=-2.53$ ,  $q=.073$ ). That is, experienced students were more likely to read multiple research information pages on possible causal factors immediately after looking at the results of lab tests. This is consistent with results from previous studies on the development of expertise, where experts were found to be more opportunistic in using resources and exploit more available sources of information than novices [9]. The higher relative frequency of reading research information, which might help experienced students interpret laboratory test results

and facilitate the acquisition of domain-specific knowledge [4], might have contributed to their higher success on making correct final claims than novice students.

In addition to two-action patterns, a differential sequence mining technique developed by Kinnebrew and colleagues [13] was utilized for identifying longer sequential patterns (length > 2) that occurred with significantly different frequencies between the two groups. This methodology used sequence support (*s-support*) and instance support (*i-support*) as frequency measures. S-support is defined as the percentage of sequences in which the pattern occurs [13]. It is different from the standard metric *support* in that s-support measures the percentage of students whose action sequence contained the specific pattern, regardless of the frequency of occurrence within each sequence for each student. The i-support corresponds to the number of times a given pattern occurs, without overlap, within a student's sequence of actions. A set of most frequent sequential patterns that met the s-support threshold was identified within each group by employing Kinnebrew et al.'s sequential pattern mining algorithm [13]. The i-support value of each pre-identified pattern was then calculated for each sequence in each group, after which t-tests comparing the mean i-support between the groups were conducted and q-value post-hoc control [25] was applied to select significantly differentially frequent patterns.

The 25 most differentially frequent long patterns with at least three consecutive actions were identified in the frog scenario and the 32 differentially frequent long patterns were identified in the bee scenario by employing a cutoff s-support of 50% and a cutoff q-value of 0.05 for comparison of pattern usage between two groups. 14 out of the 25 long patterns in the frog scenario and 16 out of 32 long patterns in the bee scenario were common (i.e., met the 50% s-support threshold) for both groups, with relatively higher usage in the novice group. 11 long patterns in frog scenario and 16 in the bee scenario were frequently used only by students in the novice group. All differentially frequent long patterns had a

**Table 4. Top differentially frequent patterns between the novice group (nov) and the experienced group (exp).**

Scenario	Pattern	s-support		i-support			Frequent
		nov	exp	nov	exp	q	
Frog	talk-MULT → inspect → save → inspect → save	0.58	0.36	0.78	0.45	<.001	nov
	talk-MULT → inspect → save → inspect	0.59	0.37	0.79	0.46	<.001	nov
	save → discard → inspect → save	0.53	0.36	0.74	0.48	<.001	nov
	inspect → save → discard → inspect	0.53	0.36	0.75	0.49	<.001	nov
	inspect → save → discard → inspect → save	0.53	0.36	0.74	0.48	<.001	nov
	talk-MULT → inspect → save	0.78	0.53	1.25	0.70	<.001	both
	inspect → save → talk	0.78	0.60	1.50	0.99	<.001	both
	discard → inspect → save	0.82	0.62	1.97	1.31	<.001	both
	inspect → save → discard	0.78	0.60	1.74	1.19	<.001	both
	talk → inspect → save	0.78	0.63	1.56	1.10	<.001	both
Bee	talk-MULT → inspect → save → inspect → save → inspect	0.59	0.27	0.72	0.32	<.001	nov
	talk-MULT → inspect → save → inspect → save → inspect → save	0.59	0.27	0.71	0.32	<.001	nov
	talk-MULT → inspect → save → inspect → save	0.74	0.45	0.99	0.57	<.001	nov
	talk-MULT → inspect → save → inspect	0.74	0.45	0.99	0.57	<.001	nov
	start assessment → talk-MULT → inspect	0.51	0.26	0.51	0.26	<.001	nov
	talk-MULT → inspect → save	0.85	0.62	1.30	0.87	<.001	both
	inspect → save → inspect → save → inspect	0.82	0.60	1.83	1.18	<.001	both
	save → inspect → save → inspect → save	0.82	0.60	1.82	1.18	<.001	both
	inspect → save → inspect → save → inspect → save	0.82	0.59	1.81	1.17	<.001	both
	save → inspect → save → inspect	0.83	0.60	1.99	1.31	<.001	both

higher s-support and a significantly higher average i-support for novice students than experienced students.

Table 4 presents the top five differentially frequent long patterns that were common to both groups and the top five that were frequently used only by the novice group within each scenario. Most of these long sequential patterns entailed the repetition and combination of actions including inspecting objects, saving objects to backpack, discarding objects, and talking with NPCs. It seemed that novice students who had not used VPA before executed more sequences comprised of exploratory behaviors such as talking with NPCs and collecting data, while more experienced students focused primarily on what was necessary to answer the core inquiry question.

## 6. DISCUSSION AND CONCLUSION

This paper investigates the transfer of student science inquiry skills across two Virtual Performance Assessment scenarios, and the impact of the novelty of the immersive virtual environment on motivation and learning. We do so by comparing performance and behaviors between novice students and experienced students. A novelty effect was found as novice students who engaged in VPA for the first time showed significantly higher scores on motivational survey subscales such as interest/enjoyment, effort/importance, pressure/tension, value/usefulness, presence/immersion, and autonomy than more experienced students. As these students were first introduced to the novel 3D virtual environment, the initial attraction and attention led to higher enjoyment, greater effort invested in the tasks, a higher sense of immersion and a higher sense of autonomy. These measures tended to decline when students became relatively experienced and familiar with the environment, consistent with previous findings on the novelty effect [8, 12]. Sequential pattern mining and comparison of overall behavior prevalence using student action log data indicated that novice students engaged in more exploratory behaviors -- they collected more data in the environment and had higher frequency of long sequences comprised of exploratory actions such as talking with NPCs, manipulating objects, and collecting data, as compared to more experienced students. This, again, might be attributed to the novelty effect [cf. 14]. That is, the higher attention of novice students resulted in higher interest and efforts in exploring the new learning environment than students who were more experienced with VPA.

However, another possibility is that the experienced students focused more on the goal at hand, than on the environment they were researching this issue on, leading to less exploration and more attention directly to the information most likely to be useful. This itself may reflect the fact that novelty is wearing off, but may be a positive aspect of the disappearance of the novelty effect. Indeed, despite the experienced students' relatively lower motivation and fewer exploratory behaviors, they outperformed the novice students in identifying a correct final claim in both scenarios and in designing causal explanations (in one scenario). Experienced students generally showed more effective problem solving. They not only tended to read research information pages more often immediately after submitting a final claim or reviewing laboratory test results, but also spent more time reading the information each time they accessed a new page. As such, even after just a half hour completing the first assessment, students demonstrated more expert-like science inquiry behaviors -- they made more use of the research information available as resources [cf. 9], in order to either interpret results, or to monitor and reflect on their final claims [cf. 15]. The information from the

pages may also have added to the domain-specific knowledge base of experienced students, which have been found to be crucial for problem solving and expertise development [5]. This corresponds to earlier findings that the transfer of domain-general inquiry strategy has the potential to facilitate the acquisition of domain-specific knowledge [4]. In conclusion, the experienced students successfully consolidated and transferred science inquiry skills they had learned from the first scenario during the approximately 30-minute engagement to the second scenario.

The current study contributes to research on the assessment of the transfer of science inquiry skills by proposing the application of a combination of educational data mining techniques such as sequential pattern mining as supplements to the traditional analysis of success between conditions. One limitation of this study is that the comparison conducted here involved virtual scenarios within the same VPA architecture. The fact that the two scenarios were highly structurally similar might have facilitated transfer. Future work may involve exploring whether far transfer of science inquiry occurs from VPA to assessments outside the system (e.g., other computer-based learning environments with different domain and interaction design).

## 7. ACKNOWLEDGMENTS

The research presented here was supported by the Bill and Melinda Gates Foundation. We also thank Chris Dede for his support and suggestions.

## REFERENCES

- [1] Agrawal, R., & Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the 11th IEEE International Conference on Data Engineering* (Mar. 1995), 3-14.
- [2] Baker, R.S.J.d., Clarke-Midura, J. 2013. Predicting successful inquiry learning in a Virtual Performance Assessment for science. In *Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization*, 203-214.
- [3] Bazaldua, D. A. L., Baker, R. S., San Pedro, M. O. Z. 2014. Combining expert and metric-based assessments of association rule interestingness. In *Proceedings of the 7th International Conference on Educational Data Mining*, 44-51.
- [4] Chen, Z., & Klahr, D. 1999. All other things being equal: acquisition and transfer of the control of variables strategy. *Child Development*, 70(5), 1098-1120.
- [5] Chi, M. T. H., Glaser, R., & Rees, E. 1982. Expertise in problem solving. In *Advances in the Psychology of Human Intelligence*, R. Sternberg, Ed. Vol. 1, Erlbaum, Hillsdale, NJ, 7-76.
- [6] Clark, R. E. 1983. Reconsidering research on learning from media. *Review of educational research*, 53(4), 445-459.
- [7] Clarke-Midura, J., & Dede, C. 2010. Assessment, technology, and change. *Journal of Research, Education and Technology*, 42(3), 309-328.
- [8] Cuban, L. 1986. *Teachers and Machines: The Classroom Use of Technology since 1920*. Teachers College Press, New York, NY.
- [9] Gilhooly, K. J., McGeorge, P., Hunter, J., Rawles, J. M., Kirby, I. K., Green, C., & Wynn, V. 1997. Biomedical knowledge in diagnostic thinking: the case of

- electrocardiogram (ECG) interpretation. *European Journal of Cognitive Psychology*, 9(2), 199-223.
- [10] Gutierrez-Santos, S., Mavrikis, M., & Magoulas, G. 2010. Sequence detection for adaptive feedback generation in an exploratory environment for mathematical generalisation. In *Artificial Intelligence: Methodology, Systems, and Applications*. Springer Berlin Heidelberg, 181-190.
- [11] Hahsler, M., Gruen, B., & Hornik, K. 2005. Arules - a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15).
- [12] Keller, J. M. 1999. Using the ARCS motivational process in computer-based instruction and distance education. *New Directions for Teaching and Learning*, 1999(78), 37-47.
- [13] Kinnebrew, J. S., Loretz, K. M., & Biswas, G. 2013. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1), 190-219.
- [14] Kubota, C. A., & Olstad, R. G. 1991. Effects of novelty-reducing preparation on exploratory behavior and cognitive learning in a science museum setting. *Journal of research in Science Teaching*, 28(3), 225-234.
- [15] Kuhn, D., & Pease, M. 2008. What needs to develop in the development of inquiry skills? *Cognition and Instruction*, 26(4), 512-559.
- [16] Kuhn, D., Schauble, L., & Garcia-Mila, M. 1992. Cross-domain development of scientific reasoning. *Cognition and Instruction*, 9, 285-327.
- [17] Merceron, A., & Yacef, K. 2008. Interestingness measures for association rules in educational data. *Educational Data Mining*, 8, 57-66.
- [18] National Research Council. 2011. *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. The National Academies Press, Washington, DC.
- [19] Ryan, R., Rigby, C., & Przybylski, A. 2006. The motivational pull of video games: A self-determination theory approach. *Motivation & Emotion*, 30(4), 344-360.
- [20] Sabourin, J., Mott, B., & Lester, J. 2013. Discovering behavior patterns of self-regulated learners in an inquiry-based learning environment. In *Artificial Intelligence in Education* (Jan. 2013), Springer Berlin, Heidelberg, 209-218.
- [21] Sao Pedro, M., Baker, R., Gobert, J., Montalvo, O., & Nakama, A. 2013. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, 23, 1-39.
- [22] Sao Pedro, M., Jiang, Y., Paquette, L., Baker, R.S., Gobert, J. 2014. Identifying transfer of inquiry skills across physical science simulations using educational data mining. *Proceedings of the 11th International Conference of the Learning Sciences*, 222-229.
- [23] Scheuer, O., & McLaren, B. M. 2012. Educational data mining. In *Encyclopedia of the Sciences of Learning*. Springer US, 1075-1079.
- [24] Schofield, J. W. 1995. *Computers and Classroom Culture*. Cambridge University Press, New York, NY.
- [25] Storey J. 2002. A direct approach to false discovery rates. *J Roy. Stat. Soc.*, 64, 479-498.
- [26] Unity Technologies. 2010. *Unity Game Engine*.
- [27] University of Rochester. 2015. *Intrinsic Motivation Inventory*. Retrieved January 25, 2015, from <http://www.selfdeterminationtheory.org/intrinsic-motivation-inventory/>

# The Impact of Incorporating Student Confidence Items into an Intelligent Tutor: A Randomized Controlled Trial

Charles Lang  
Harvard Graduate  
School of Education  
13 Appian Way  
Cambridge, MA  
+1-617-495-7945  
charles\_lang@mail.  
harvard.edu

Neil Heffernan  
Computer Science  
Department, Worcester  
Polytechnic Institute  
100 Institute Road  
Worcester, MA  
+1-508-831-5569  
nth@wpi.edu

Korinn Ostrow  
Learning Sciences &  
Technologies, Worcester  
Polytechnic Institute  
100 Institute Road  
Worcester, MA  
+1-508-831-5569  
ksostrow@wpi.edu

Yutao Wang  
Computer Science  
Department, Worcester  
Polytechnic Institute  
100 Institute Road  
Worcester, MA  
+1-508-831-5569  
yutaowang@wpi.edu

## ABSTRACT

For at least the last century researchers have advocated the use of student confidence as a form of educational assessment and the growth of online and mobile educational software has made the implementation of this measurement far easier. The following short paper discusses our first study of the dynamics of student confidence in an online math tutor. We used a randomized controlled trial to test whether asking students about their confidence while using an Intelligent Tutor altered their performance. We observe that (1) Asking students about their confidence has no statistically significant impact on any of several performance measures (2) Student confidence is more easily reduced by negative feedback (being incorrect) than increased by positive feedback (being correct) and (3) confidence accuracy may be a useful predictor of student behavior. This paper demonstrates how psychological ideas can be imported into Educational Data Mining and our findings point to the possibility of using student confidence to better predict performance and differentiate between students based on the way they approach items.

## Categories and Subject Descriptors

J.4 [Social and Behavioral Sciences]: Psychology

K.3.1 [Computers and Education]: Computer-Assisted Instruction (CAI)

## General Terms

Experimentation, Human Factors

## Keywords

Confidence, certainty, self-efficacy, cognitive tutor, confidence-based assessment, ASSISTments

## 1. INTRODUCTION

Interest in student confidence arose out of investigations into the mathematical formalization of subjective probability at the end of the 19th century [5]. At least since 1913 researchers sought to apply these theories of judgment to educational assessments [19]. The initial motivation from the educationalists' perspective was to determine if querying student confidence could provide useful additional information about student performance [4]. Over the last century the utility of confidence testing has been demonstrated in terms of test reliability [3, 11, 15], identifying

guessing [18], separating students based on their level of understanding [7], increasing student understanding [4, 6, 14] and explaining answer changing [17]. Interest in student confidence has been further extended through work on self-efficacy – “students’ judgments of their capability to accomplish specific tasks” [1]. Self-efficacy studies have made extensive use of Likert-style questions about student confidence [12].

Despite the utility of student confidence it has not gained widespread use within educational assessment. This may be because experimental psychology largely views confidence as an unreliable measure, suggesting that humans generally tend to suffer from overconfidence bias [10]. Overconfidence bias implies that much of the variation in student confidence can be explained by an inclination for students to report that they are better at solving problems than they in fact are rather than explanatory variables that might improve learning [7].

Another reason for the failure of student confidence to become a widespread measure may be that the cost and logistical difficulty in collecting, scoring and storing confidence data was historically high. The comparatively low cost and large scale of online assessment may be diminishing this issue substantially though. In a world of yearly or bi-yearly paper tests it is not feasible to collect and score confidence data, but in an online environment these burdens are lifted.

Yet, there remain some lingering misgivings about the use of self-reported confidence. Overconfidence bias may be an artifact of larger issues with the way that confidence data are collected. Indeed, the concern remains whether simply asking students about their confidence may in fact alter their performance [13]. If requiring students to report their confidence reduces their overall performance then any utility in the measure will be undermined, it is therefore important to study the impact of student confidence measurement within a real-life setting.

The dynamics of student confidence are what concern this short paper. We were concerned primarily with the impact of asking Likert-style confidence questions on other aspects of student performance, and how students’ confidence changed as they navigated tasks within the ASSISTments Intelligent Tutoring System. We are in the beginning stages of mapping out how student confidence changes as students move through online math assessment. Our aim is to identify how student confidence might relate to student behavior with the goal of leveraging this information to increase student learning.

## 2. METHOD

### 2.1 Data

The present study was conducted as a simple randomized controlled trial within ASSISTments, an adaptive mathematics tutor that serves as a free assistance and assessment tool to over 50,000 users around the world [9]. Two problem sets were designed around the multiplication and division of fractions and mixed numbers, using a mastery learning based structure called a Skill Builder. Skill Builder problem sets are unique in that students are randomly dealt questions from a skill bank until they are able to answer three consecutive questions accurately, thus ‘mastering’ the assignment.

Both problem sets were designed with two conditions: an experimental condition in which students were asked to self-assess their confidence in solving similar problems, and a control condition in which students were asked filler questions to control for the effect of spaced assessment. Random assignment was performed by the ASSISTments tutor at the student level. Throughout the course of each assignment, students were asked up to three self-assessment or survey questions. At the start of each assignment, students who were randomly assigned to the experimental condition were introduced to the skill of self-assessment, shown a set of problems isomorphic to those in the problem set, and asked to gauge their confidence in solving the problems using a Likert scale ranging from ‘I cannot solve these problems (0%)’ to ‘I can definitely solve these problems (100%)’. Students who were randomly assigned to the control condition were polled on their current browser in an attempt to ‘improve the ASSISTments tutor.’ Examples of the initial questions posed to each condition are presented in Figure 1 below.

Following these initial questions, students were given three mathematics questions. If students solved each of these three questions accurately, the assignment was considered complete. However, if students answered at least one of the problems incorrectly, they would reach another self-assessment or survey question before being given another set of three math questions to try to master the problem set. This pattern happened a third time for students who were struggling with the content, until finally removing the self-assessment or survey element and simply providing back to back math questions until the student could solve three consecutive problems. Based on this design, high performing students were asked to gauge their confidence only a single time, while students struggling with the topic were asked to reassess their confidence up to two more times throughout the problem set. The confidence question was always formatted using the same Likert scale, while the ‘ASSISTments’ improvement surveys changed slightly, polling students on various elements of accessibility.

These Skill Builders were marked as ASSISTments Certified material and made publicly available to all users. The sets were promoted as new content and received high usage over the course of approximately three months. The tutor logged all student actions throughout the course of the experiment, and a dataset was obtained from the ASSISTments database for analysis. The experiment is still actively running within ASSISTments, gaining sample size for additional analysis to be conducted at a later time.

Problem ID: PRAUWNR [Comment on this problem](#)

Estimating your skill before you solve a problem is a good habit. How confident are you that you could solve problems such as the ones below without an error? Please be honest, as all answers are equally correct:

$$3\frac{5}{18} \times \frac{9}{11} = ?$$
$$\frac{1}{13} \times 2\frac{3}{7} = ?$$
$$7\frac{4}{9} \times \frac{7}{12} = ?$$

Select one:

- I cannot solve these problems (0%)
- I am not confident (25%)
- I feel somewhat confident (50%)
- I feel very confident (75%)
- I can definitely solve these problems (100%)

Submit Answer

---

Problem ID: PRAUWND [Comment on this problem](#)

On this problem set you will be asked a few survey questions to help us make ASSISTments better. Once you answer the survey question you can move forward with your math learning.

Which browser are you using? There is no correct or incorrect answer.

Select one:

- Internet Explorer
- Chrome
- Safari
- I don't know

Submit Answer Show answer

Figure 1. Initial Questions for Students in Experimental (Above) and Control (Below) Conditions

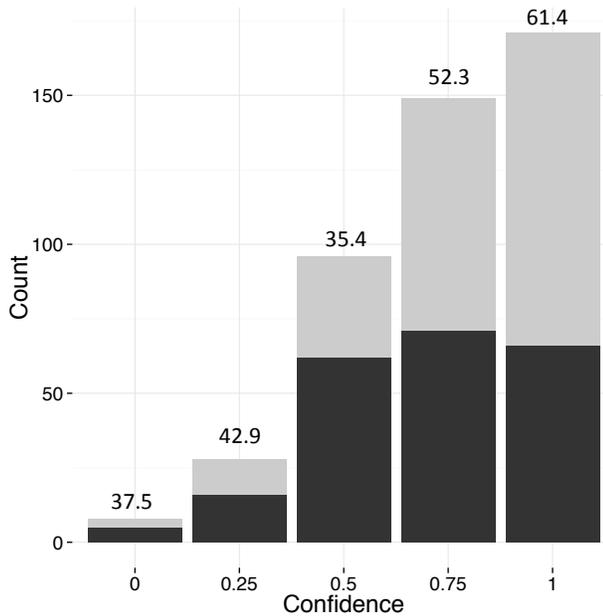
The data set used for the present analysis consisted of 950 12-14 year old students in the eighth grade, from a group of school districts in the North East the United States. Data included 10,770 problem level records including rich details pertaining to student performance. After working with the ASSISTments team to design and run this study, the lead author was provided the data set for primary analysis with all information that could lead to the identification of individual students removed, as set in the

protocol of an IRB exemption granted by the CUHS of Harvard University.

### 3. RESULTS

#### 3.1 Student Confidence

##### 3.1.1 Description of Confidence



**Figure 2. Histogram defining distribution of initial student confidence with the proportion of each group that was correct on the first item above the bar and shaded (gray:correct, black:incorrect). Most students have mid- to high-confidence.**

The initial distribution of student confidence was left skewed, with the majority of students reporting their initial confidence in the problems as being between 0.5 and 1.0 ( $M = 0.75$ ; Figure 2). On subsequent confidence questions the distribution remains left skewed though the mean confidence shifts toward the center as highly confident students exit the system after mastery ( $M = 0.56$ ).

The overall trend in students' estimation of their own skill is that more of the confident students tend to be correct. However, the students at either extreme (not confident at all and 100% confident) do not meet their own expectations. Three of the eight students who estimated that they "cannot solve these problems" were able to solve the first problem and 66 out of the 105 students who estimated that they "can definitely solve these problems" were incorrect on the first problem.

##### 3.1.2 Learning Gains

Overall learning gains were comparable between the experimental and control groups (Table 1). Though differences among different levels of confidence persisted. Highly confident students tended to be more accurate than the control group and continue to improve, while moderately to very unconfident students tended to be far less accurate than the control group, though they tended to improve, with the exception of the students with zero confidence. As occurred in the first question, those students who were "not

confident" outperformed students who were "somewhat confident" on the second and third questions.

**Table 1. Learning paths for students in the experimental and control groups showing percentage of students who were correct on questions 1, 2 and 3.**

	Confidence					Treat	Control
	0.0	0.25	0.5	0.75	1.0		
<b>Q1 Correct (%)</b>	37.5	42.9	35.4	52.3	61.4	51.3	45.4
<b>Q2 Correct (%)</b>	62.5	60.7	55.2	68.5	76.6	68.1	70.7
<b>Q3 Correct (%)</b>	37.5	64.3	59.4	73.2	78.4	71.0	70.7
<b><i>n</i></b>	8	28	96	149	171	452	498

#### 3.2 The Impact of Measuring Confidence on Performance

Since there is some evidence that question format can impact student performance we looked at whether there was a difference between students who were asked confidence style questions and those who were asked "dummy" survey questions. In all but one respect there seems to be no statistically significant effect of asking students what their confidence is within the ASSISTments system.

There was no statistically significant difference with respect to accuracy between students who were asked confidence questions and those who were not (Control = 53% correct, Experimental = 52% correct,  $\chi^2 = 5.7, p = 0.68$ ). Students who were asked confidence questions did not use more or less hints (Control = 0.89 hints/student, Experimental = 0.89 hints/student,  $\chi^2 = 37.1, p = 0.09$ ) nor did they make more or fewer attempts (Control = 1.7 attempts/student, Experimental = 1.6 attempts/student,  $\chi^2 = 46.4, p = 0.41$ ). There was also no difference between students who were asked about their confidence and those who were not with respect to the number of questions they answered (Control = 5.1 questions/student, Experimental = 5.2 questions/student,  $\chi^2 = 169.7, p = 0.10$ ). Nor did asking confidence questions impact the way that students behaved after being incorrect; there is no statistically significant tendency for students who were given confidence questions to ask for hints on the next question after being incorrect on the first question (Control = 8%, Experimental = 10%,  $\chi^2 = 0.11, p = 0.74$ ).

There is one case in which there is a statistically significant difference between the control and experimental groups though: of the students who were incorrect on the first question, more students in the experimental group were incorrect on the second question ( $\chi^2 = 4.63, p = 0.03$ ; Table 2). This suggests that the act of asking confidence questions impairs students' performance in some way. This effect disappears by the third question though ( $\chi^2 = 0.61, p = 0.43$ ).

**Table 2. Students who were correct on Question 2 after being incorrect on Question 1 for control and experimental groups. Fewer students in the experimental group were correct on Question 2.**

	Control	Experimental
<b>Correct (%)</b>	171 (34.3)	125* (27.7)
<b>Incorrect (%)</b>	327 (65.7)	327 (72.3)

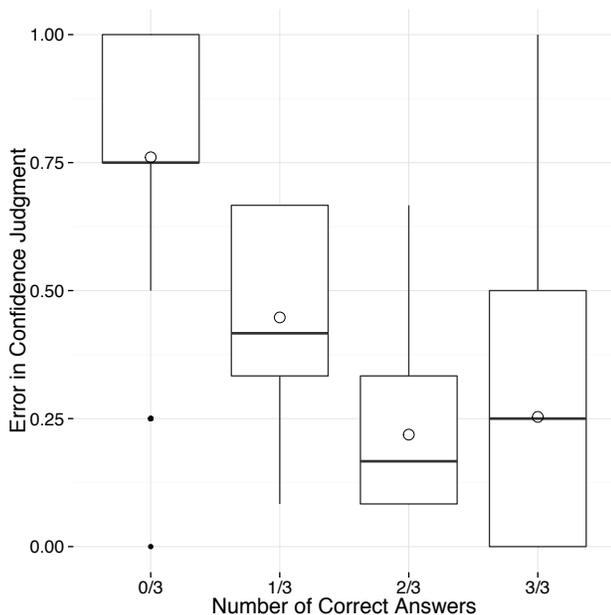
\* Denotes a significant difference between control and experimental  $p < 0.05$ .

### 3.3 The Importance of Confidence

#### 3.3.1 Confidence as a Prediction of Future Performance

If we consider confidence to be a student's prediction of their future performance we can calculate an error measure of this prediction. For example, if a student has a confidence of 0.75 we would assume that they expected to get 75% of the next three questions correct. If they in fact got 100% of the answers correct then their error rate would 0.25 (confidence – percent correct).

Error rates appear to correlate with several factors, including accuracy. Students who are better at predicting their score on the next three questions tend to be those who are more accurate at answering those three questions ( $r(452) = -0.54, p < .001$ ; Figure 3). They also tend to utilize more hints ( $r(452) = 0.42, p < .001$ ) and make more attempts ( $r(452) = 0.31, p < .001$ ).



**Figure 3. Boxplot representing the error associated with student confidence judgment (confidence – percent correct) vs. percent correct for first three questions. Students who are more accurate at judging their ability tend to get more answers correct. Line equals median, circle equals mean.**

#### 3.3.2 Predicting Accuracy Based on Confidence

We can also attempt to predict the outcome of a single question based on student confidence. We built a logistic regression model that predicted whether or not a student was correct on their third item using 1) student confidence, 2) whether the student was correct on previous items, 3) their percentage correct over all problem sets attempted, 4) how many problems they had attempted within the ASSISTments system, and 5) which problem set they were attempting. Of these predictors, the only significant variables were accuracy on previous questions and student confidence, which make up the most parsimonious model (Model IV; Table 3).

There is a more substantial relationship between accuracy on the third item and student confidence than with accuracy on the previous two items. A change in student confidence from zero to 100 is associated with the odds of being correct on the third question increasing by a factor of 3, whereas the odds of being correct on item 3 are increased by a factor of 2.3 with respect to being correct on the first item, and only 1.8 for being correct on the second item.

**Table 3. Taxonomy of logistic regression models that display the fitted relationship between the log odds of being correct on the third item and student confidence, being correct on the first item, being correct on the second item, the prior percent correct, number of prior problems attempted and the problem set (n=452). Model IV is the most parsimonious.**

	Model I	Model II	Model III	Model IV
<b>Intercept</b>	0.5688	-0.4254	-0.5359	-0.6684*
<b>Confidence</b>	0.9974*	1.2248**	1.3163**	1.1234**
<b>Q1 Correct</b>	0.7896***	0.9348***		0.8294***
<b>Q2 Correct</b>	0.6132**		0.7314***	0.5662***
<b>Prior percent correct</b>	-0.0001			
<b>Prior problem count</b>	0.3542			
<b>Problem set</b>	-0.0545			
<b>AIC</b>	517.75	518.3	526	514.15

#### 3.3.3 Changes in Confidence after Incorrect Answers

The impact of incorrect answers on student confidence is clear from a breakdown of how confidence changes before and after completing questions (Figure 4). Students were asked for their confidence before the first and after the third problem. The decision tree below represents the 258 students who did not exit the system before they were asked this second round of

confidence questions. The tree is read top to bottom, in the first tier students are sorted based on how many of the three problems they got correct. In the second tier students are sorted based on how they changed their confidence, did they become less confident, more confident or stay the same.

There are a few trends that can be drawn out from this map. The majority of students (85%) who get three questions incorrect in a row lose confidence, while only 47% of students who get three correct in a row increase their confidence or are already at the maximum confidence. Indeed, 28% of students revise their confidence down after getting three correct answers in a row! Only one student decided to increase their confidence despite getting three incorrect answers in a row.

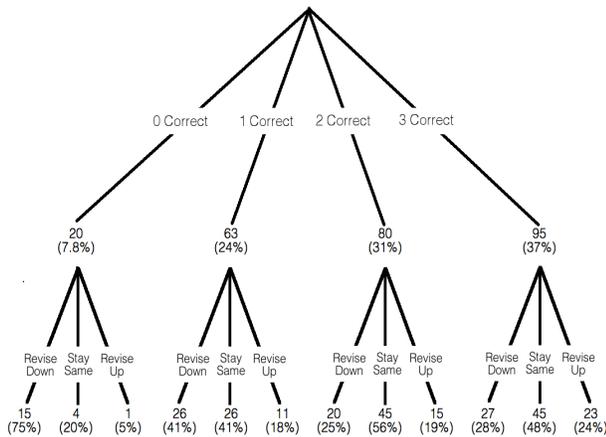


Figure 4. Changes in student confidence with respect to confidence levels at Question 1 and Question 5.

## 4. DISCUSSION

Overall the current study illustrates the trade off between using a different question format and the impact of this format on student behavior. Confidence style questions may provide substantial new utility in predicting and understanding student behavior but this utility may also come at a cost. We want to ensure that we have weighed this cost against the benefits of confidence style questions before further pursuing the benefits they provide. Overall, it appears from the present study that the benefits indeed do outweigh the costs.

### 4.1 Cost vs. Benefit

Beyond the time-cost of adding confidence questions to the problem set we wanted to know if there was any detrimental or beneficial impact on students performance of answering this kind of question and whether the question generates useful information.

The addition of Likert-style confidence questions appears not to impact many relevant behaviors within the ASSISTments system. This is somewhat surprising given methodological research on the impact of phrasing questions [16] and the substantial literatures on the impact of self-efficacy [12] and self-reflection [2] on student performance. However, in this study it seems to have had little discernable impact. The small impact that was detected however is of substantial concern. It appears that students who were given confidence style questions and who were incorrect on their first answer were slightly less likely to be correct on the second question they answered. We might imagine that asking students

their confidence could have myriad effects on the way they answered, perhaps it made them more hesitant or more anxious resulting in poorer performance. In either case this is problematic as the aim of the system is to improve performance and learning.

This is not a definitive finding however, as the effect was small and disappeared by the next question. There are also alternative interpretations. The dip in performance may not necessarily connote a failure to learn. Perhaps it denotes a student wrestling more substantially with the concepts in the problem set, which may result in longer lasting, more robust learning going forward. This hypothesis needs to be tested by looking at future student performance. We also need to test whether any impact diminishes with exposure to the format.

Another reason why we may not want to use confidence style questions is that the information they generate is not useful because it is a poor estimate of student ability. We have substantial evidence of this conclusion. Students appear to be poor estimators of their own skill. For example, although unconfident students answer questions incorrectly more often than confident students, students at the extremes tend to exaggerate their predictions. Students with very low confidence tended to underestimate their ability and students with very high confidence tended to overestimate their ability. This trend may reflect how students approach confidence, although we have presented it as a continuous scale some students may be seeing it more as a binary; they are either confident or not. This would explain why very confident students get wrong answers and very unconfident students get correct answers and is in keeping with the psychological theory of extremeness [8]. In this theory people are thought to concentrate on the extremeness of options above all else. Therefore, students who maybe somewhat confident are drawn to concluding that they are either 0% or 100% confident. To conclude that there is no useful information in confidence because of this tendency would be a mistake though. There are two substantially useful characteristics that are worth pursuing within the ASSISTments system: error rate of student confidence and how confidence changes as students answer questions correctly or incorrectly.

Although students are, on average, poor judges of their own accuracy those who are better at predicting their accuracy tend to be more correct. There seems to be a benefit in being a good predictor of your own performance. This suggests the skill to predict your own performance may be a worthwhile cultivating and therefore measuring. This prediction skill is also correlated with higher levels of engagement with the system when a student is incorrect; asking for more hints and making more attempts. This may indicate that students who are better predictors of their own performance are also more interested in learning. This may help in signaling those students who are not interested in learning for differentiated interventions.

It is also worth thinking about how prediction accuracy is developed. The dynamics of confidence behavior can shed more light on this idea. Confidence seems to be very sensitive to accuracy in an interesting way. The vast majority of students who get incorrect answers tend to reduce their confidence, while a minority of students who get all answers correct seem to increase their confidence. Confidence, it would seem, is easier to lose than to gain. This may be related to another psychological principle, asymmetry. The asymmetry principle states that humans have a tendency to attribute greater weight to negative, rather than positive events. If this effect is cumulative it may explain why

students underestimate their ability at the low end of the confidence scale. Yet it doesn't explain why students overestimate their ability at the other end. Clearly there is more to understand about how students revise their confidence and the rate at which they do it. If being accurate in the prediction of your own performance is important, perhaps we should be more sensitive in how we impact that through the delivery of incorrect/correct answers. Perhaps pushing students away from extreme values is a worthwhile pursuit.

It would appear though that the benefits of studying confidence within this Intelligent Tutor far outweigh the possible cost of diminishing performance on one question. The ability to detect, and possibly increase, student engagement would be a highly useful addition.

## 4.2 Conclusion

The aim of this work is to develop understanding that can improve learning outcomes. It is useful information to know that student confidence is easier to reduce than to build and that accuracy in predicting ones performance is related to engagement in the system and increased performance. This can inform the way that difficulty is used to drive instruction, possibly balancing the difficulty and timing of problems with respect to student tolerances. In future research we hope to draw on the conclusions we have outlined here and to utilize associations with student confidence. In particular, we wish to investigate whether it is possible to improve students' estimates of their confidence and whether this translates into impact on their actions within the online tutor. We wish to know whether it is possible to increase persistence and increase the appropriate use of hints by targeting students' ability to estimate their confidence.

## 6. ACKNOWLEDGMENTS

Our thanks to the ASSISTments team for making this study possible. We acknowledge funding from multiple NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736 & DRL-1031398), the U.S. Department of Education (IES R305A120125 & R305C100024 and GAANN), the ONR, and the Gates Foundation.

## 7. REFERENCES

[1] Bandura, A. 1977. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*. 84, 2 (1977), 191–215.

[2] Van den Boom, G., Paas, F., van Merriënboer, J.J.G. and van Gog, T. 2004. Reflection prompts and tutor feedback in a web-based learning environment: effects on students'

self-regulated learning competence. *Computers in Human Behavior*. 20, 4 (Jul. 2004), 551–567.

[3] Ebel, R.L. 1965. *Measuring educational achievement*. Prentice-Hall.

[4] Echternacht, G. 1972. The use of confidence testing in objective tests. *Review of Educational Research*. 42, 2 (1972), 217–236.

[5] Estes, W.K. 1976. The cognitive side of probability learning. *Psychological Review*. 83, 1 (Jan. 1976), 37–64.

[6] Gardner-Medwin, A. and Gahan, M. 2003. Formative and summative confidence-based assessment. *Proceedings of the 2008 International Computer Assisted Assessment (CAA) Conference* (London, 2003), 147–155.

[7] Gardner-Medwin, A.R. 1995. Confidence assessment in the teaching of basic science. *ALT-J*. 3, 1 (Jan. 1995), 80–85.

[8] Griffin, D. and Tversky, A. 1992. The weighing of evidence and the determinants of confidence. *Cognitive Psychology*. 24, 3 (Jul. 1992), 411–435.

[9] Heffernan, N.T. and Heffernan, C.L. 2014. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education*. 24, 4 (Sep. 2014), 470–497.

[10] Langer, E.J. 1975. The illusion of control. *Journal of Personality and Social Psychology*. 32, 2 (Aug. 1975), 311–328.

[11] Michael, J.J. 1968. The reliability of a multiple-choice examination under various test-taking instructions. *Journal of Educational Measurement*. 5, 4 (Dec. 1968), 307–314.

[12] Pajares, F. and David, M. 1994. Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of Educational Psychology*. 86, 2 (1994), 193–203.

[13] Pajares, F. and Urdan, T.C. 2006. *Self-efficacy Beliefs of Adolescents*. IAP.

[14] Ramsey, P.H., Ramsey, P.P. and Barnes, M.J. 1987. Effects of student confidence and item difficulty on test score gains due to answer changing. *Teaching of Psychology*. 14, 4 (1987), 206–210.

[15] Rippey, R. 1968. Probabilistic Testing. *Journal of Educational Measurement*. 5, 3 (Oct. 1968), 211–215.

[16] Schwarz, N. 1999. Self-reports: How the questions shape the answers. *American Psychologist*. 54, 2 (1999), 93–105.

[17] Skinner, N.F. 1983. Switching answers on multiple-choice questions: Shrewdness or shibboleth? *Teaching of Psychology*. 10, 4 (1983), 220–222.

[18] Taylor, C. and Gardner, P.L. 1999. An alternative method of answering and scoring multiple choice tests. *Research in Science Education*. 29, 3 (Sep. 1999), 353–363.

[19] Woodworth, R.S. 1915. *Archives of Psychology*.

# Analyzing Early At-Risk Factors in Higher Education e-Learning Courses

Ryan S. Baker<sup>1</sup>, David Lindrum<sup>2</sup>, Mary Jane Lindrum<sup>2</sup>, David Perkowski<sup>2</sup>

<sup>1</sup>Teachers College Columbia University, 525 W 120<sup>th</sup> St. New York, NY 10027

<sup>2</sup>Soomo Learning, 9 SW Pack Square, Suite 301, Asheville, NC 28801

baker2@exchange.tc.columbia.edu, david.lindrum@soomolearning.com,  
maryjane.lindrum@soomolearning.com, david.perkowski@soomolearning.com

## ABSTRACT

College students enrolled in online courses lack many of the supports available to students in traditional face-to-face classes on a campus such as meeting the instructor, having a set class time, discussing topics in-person during class, meeting peers and having the option to speak with them outside of class, being able to visit faculty during office hours, and so on. Instructors also lack these interactions, which typically provide meaningful indications of how students are doing individually and as a cohort. Further, online instructors typically carry a heavier teaching load, making it even more important for them to find quick, reliable, and easily understandable indicators of student progress, so that they can prioritize their interventions based on which students are most in need. In this paper, we study very early predictors of student success and failure. Our data is based on student activity, and is drawn from courses offered online by a large private university. Our data source is the Soomo Learning Environment, which hosts the course content as well as extensive formative assessment. We find that students who access the resources early, continue accessing the resources throughout the early weeks of the course, and perform well on formative activities are more likely to succeed. Through use of these indicators in early weeks, it is possible to derive actionable, understandable, and reasonably reliable predictions of student success and failure.

## Keywords

At-Risk Prediction, Prediction Modeling, Predictive Analytics, Activity Analytics, Online Course, Webtexts

## 1. INTRODUCTION

Students enrolled in online courses lack many of the supports available to students in traditional face-to-face classes on campus [13]. Drop rates are typically higher for online courses than traditional courses (see review in [8]), and procrastination is often a major problem in online courses [10]. Part of the reason for the lower success seen in online courses comes from the fact that faculty have less direct contact with students [5, 19] and as a result have fewer indicators of how students are doing, outside of formal assessment. This makes intervention for at-risk students more difficult than in campus-based learning settings.

As a result, many universities and providers of online courseware have moved to models that can automatically identify when students are at risk. These models identify indicators of potential student failure (or lower success). A comprehensive review of work in this area can be found in [10]. In one example of the creation and study of such a model, Barber and Sharkey [4] predicted course failure using a mixture of data from student finances, student performance in previous classes, student forum posting, and assignment performance. In a second example, Whitmer [17] predicted final course grade from student LMS

usage activity, including the number of times a student accessed any content, the number of times a student read or posted to the forum, and the number of times a student accessed or submitted an assignment. In a third example, Romero and colleagues [15] predicted final course grade from activity and performance on assignments, including time taken by the student; this work was followed up by additional work, where the same group studied a more extensive set of interaction variables within the Moodle platform [14]. In a fourth example, Andergassen and colleagues [1] predicted final exam score from completion of online learning activities, including when in the semester students engaged those activities, and the total span of time between a student's first and last activities in the online resource.

An area of particular importance is early prediction, as recommended by Dekker and colleagues [7]. Being able to make predictions early in the semester, using the data available from initial student participation in the course, allows for timely intervention. There have been projects that have been successful in identifying at-risk students early in the semester. For example, Ming and Ming [12] developed models that could predict student course success from the first week of course participation, based on the topics students posted on the online discussion forum. In another example, Jiang and colleagues [11] predicted MOOC course completion from grades and discussion forum social network centrality, at the conclusion of the first course week.

Models that can predict student success early in a course, from course participation data, may be more or less useful depending on the features the models are based upon. If models are based on indicators which are interpretable and meaningful to course staff, these models can then provide instructors with data on which students are at-risk along with information on why those specific students are at risk. Systems of this nature have been successfully embedded within intervention practices and had positive impacts on student outcomes. For example, the Course Signals project at Purdue University provides predictions to instructors along with suggested interventions for specific students, in the form of recommended emails to send the students [2]. In one evaluation, Course Signals was associated with better student grades and better retention [3]. Another project, the Open Academic Support Environment, was associated with better student grades [10].

The attributes of a desirable predictive model are tightly connected to the potential uses of that model. For example, highly complex "black box" indicators are hard for instructors to use in interventions, even if they might be perfectly suitable for automated interventions. Beyond this, demographic variables (such as race and financial need) can be predictive [17, 18], but are less immediately useful for instructors wishing to intervene.

In this paper, we study early predictors of student success based on student activity, with the goal of giving faculty immediately

useful, easy-to-interpret data.

We analyze these predictors within the context of the Soomo Learning Environment, a system used by over 100 universities to deliver course content and extensive formative assessment to over 70,000 undergraduates a year. Specifically, in this paper we study the learning and eventual success of over four thousand students taking an online course on introductory history at a large 4-year private university.

We find that students who access the resources early, continue accessing the resources throughout the early weeks of the course, and perform well on formative activities are more likely to succeed in the course overall. Through use of these indicators in early weeks, it is possible to derive actionable, understandable, and reasonably reliable, predictions of student success, enabling faculty to identify those students most in need of intervention, and suggesting the kind of guidance each student needs.

## 2. DATA

We investigate these issues within the context of data from an introductory history course, offered as an online course by a large 4-year private university, using an interactive web-based learning resource from Soomo. The Soomo Learning Environment (SLE) is a web-based content management system built for hosting instructional content and formative assessment. Typically students click a link in their learning management system to open their webtext, hosted in the SLE, in a new tab. All course content, customized for the specific instructor and institution, is presented within this environment. Courses are typically built with a mix of original, permissioned, and open content, combining text, images, audio, video, hosted and linked artifacts, and tools for study. Webtexts are developed by instructional designers at Soomo Learning in conversation with faculty advisors and subject matter experts. Webtexts are then peer reviewed and finally tailored to the needs of a specific institution and/or faculty member.

Webtexts are not just digital copies of traditional paper textbooks; they are distinguished by hundreds of opportunities for students to respond to the content through the course. Within Soomo's webtexts, "Study Questions" help students assess their own comprehension of what they just read or watched. "Investigations" present opportunities for application, analysis, synthesis, and evaluation, thereby supporting learners in developing richer understanding.

Final student grades in the US History course were based on performance on a range of assignments. The grade weighting was identical across sections in a specific term, but varied term-to-term as the university and Soomo Learning worked together to tune the course. The final course grade was based on a combination of a final paper and milestones to that final paper, work in the Soomo Learning Environment, and participation in class discussion boards. We obtained data on student course performance and webtext activity, for 4,002 students enrolled across 140 sections of this course, taught over six terms in 2013 and 2014. These students performed a total of 2,053,452 actions in the webtext, including opening pages and answering questions.

Student grades below 60% were considered failing grades; however, the target of our at-risk predictions was to predict whether students would fall below 73%, the minimum grade required to get a C. 990 of the 4,002 students (24.7%) obtained a grade below 73%.

## 3. ANALYZING INDIVIDUAL PREDICTORS

One of the major goals of predictive analytics is making predictions early in the semester, before the student has fallen behind on the course's material to an extent that is difficult to repair. It is at this stage where instructor intervention can have the greatest impact. In this paper, therefore, we focus on student performance and usage in the first 4 weeks of a 10-week term.

The Soomo webtexts include formative assessment throughout the course, starting on the first pages of the resource. This gives faculty measures of student engagement and performance from the very first week of the course. The predictors analyzed in this paper are not inherent to the Soomo Learning Environment – they could be applied to other online courses that have online readings and assignments. They rely primarily on having measures of student engagement and understanding on a regular basis, from the start of the course.

### 3.1 Did the student access the webtext at all?

The first feature we analyze is whether students accessed the webtext at all in the early stages of the course. This course was organized into a set of one-week units. Therefore, it might be plausible to analyze whether a student accessed the webtext during the first week of the course; by the end of the first week, the students were expected to have completed the first week's materials. However, many students procrastinate [16], and students are not penalized within this course for completing materials late, so it is possible that many students do not access course materials within this window. We analyze variants of this feature, looking at whether students have failed to access the webtext and activities within the first N days of the course. The canonical value of N is 7; other values are also examined. (We omit data from one course term for this analysis in specific, due to a logging error).



Figure 1. The introductory US History webtext (above) and embedded study questions relevant to that text (below)

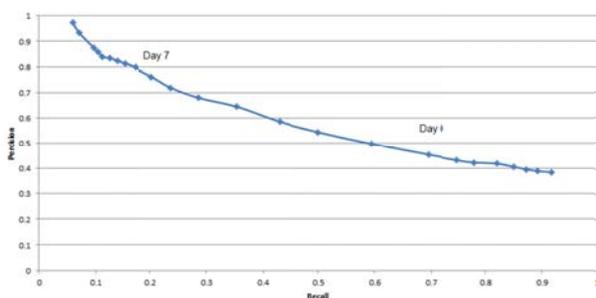
As such, we predict whether a student got a course grade under 73% (a.k.a. eventually failed or got a D), from whether the student had accessed the book yet by day N. A precision-recall curve for this relationship is shown in Figure 2. A precision-recall curve [6] shows the tradeoff between precision and recall for different thresholds of a model. Precision represents the proportion of cases identified as at-risk that are genuinely at-risk; recall represents the proportion of genuinely at-risk cases that are identified as at-risk. They are computed:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Typically, precision-recall curves are used for different confidence thresholds between a positive and negative prediction; in this case, we display the tradeoff between precision and recall for different thresholds of how many days into a course a student can be before we become concerned that they have not accessed the webtext yet. As will be seen in the paper, studying these curves allows us to study the relative trade-off between precision and recall for different model thresholds and different feature variants. Some instructors may want models with higher recall, so that they can contact a larger proportion of at-risk students; other instructors may want more models with higher precision, to avoid contacting too many total students. While some researchers argue for optimizing a single metric, different instructors (or university administrators) may prefer different models.

As Figure 2 shows, there is a clear trade-off between precision and recall for how many days have passed at the start of the course without the student accessing the webtext. On the far left, almost all students who have not yet accessed the webtext by the 14<sup>th</sup> day of the class fail. On the far right, almost all students who eventually fail are captured by a model that looks at whether the student has not yet accessed the webtext seven days before class, but precision is only 40%. On the first day of class (day 0), precision is barely higher but recall is much lower. Seven days later (day 7), precision approaches 80% but recall is just below 20%. As such, this indicator changes its meaning considerably with each day that passes during the first 7 days of the class. On day 0, the Cohen's Kappa for this feature (representing the degree to which the model is better than chance) is 0.207. On day 7, Kappa is 0.200. On day 3, it reaches a maximum of 0.277; any value of N higher or lower than 3 has a lower Kappa.



**Figure 2. Precision-Recall Curve for how well a final grade below 73% is predicted by whether a student has accessed the webtext by day N.**

### 3.2 Has the student accessed the webtext recently?

Accessing the webtext is an important first step, but it is reasonable to believe that students are most successful if they continue to access the course materials weekly. As such, the second feature we analyze is how long it has been since the student accessed the webtext. This feature has two parameters: the current day N, and the number of days D since the student last accessed the webtext.

As such, we are predicting whether a student got a course grade under 73% (a.k.a. eventually failed or got a D), from whether the student had accessed the book in the last D days, at the time of day N. For tractability, we select four possible values for D: the last 3 days, the last 5 days, the last 7 days, and the last 10 days. We also select values between 1 and 28 for N; the model does not go beyond the fourth week of this course, because after this point, it is relatively late for “early” intervention. Note that students can open the book before the first day of the course (so it is meaningful to compare between values of D, even for N=1).

A set of precision-recall curves is given for these model variants in Figure 3. As Figure 3 shows, the models start out very similar, regardless of value of D, at the beginning of the course, with precisions around 44%-46% and recalls around 65%-70%.

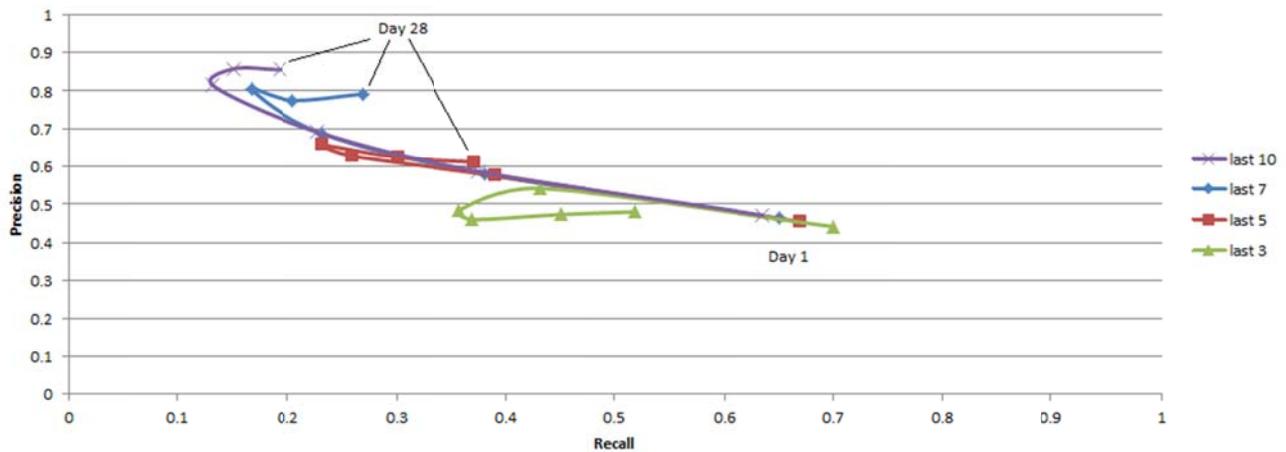
As the value of N goes up, recall drops and precision goes up, until the changes become unstable around the third week of the course. (At that point, however, the changes are relatively minimal). The higher the value of D, the higher the eventual precision and the lower the eventual recall, at the end of the fourth week of the course. For instance, for D = 7, the precision reaches 80.4% by day 14, though the recall is at a relatively low 16.7%. To put this another way, on day 14, a student who has not accessed the textbook in the last 7 days has a 80.4% probability of performing poorly in the course, and 16.7% of students who perform poorly in the course had not accessed the textbook in the last seven days on day 14.

This shift effect is relatively weaker for lower values of D; for instance, for D = 3, the precision goes up relatively little, reaching only 54.2% on day 4, while the recall drops rapidly, reaching 35.8% by day 7. These results, in aggregate, show that this feature manifests different behavior depending on choice of threshold.

Kappa values were relatively unstable across predictors, though the differences in Kappa were generally small, indicating that most of the differences between models reflected a precision-recall tradeoff. The best Kappa, 0.27, was obtained for D=7 and N=28. The second best kappa, 0.247, was obtained for D=7 and N=4. However, the third best kappa, 0.241, was obtained for D=3 and N=4. Kappa values were generally higher for higher values of D, but the differences were extremely small; the average Kappa for each value of D only varied by 0.03.

### 3.3 Is the student doing poorly on exercises in the webtext?

Another indicator that the student is struggling is if the student is performing poorly on the formative exercises in the webtext. These exercises comprise only a third of the student’s eventual grade, but are an indicator that the student does not understand the content. As discussed above, there are two types of assignments within the webtext, Study Questions and Investigations.



**Figure 3: Precision-Recall Curve for how well a final grade below 73% is predicted by whether a student has accessed the webtext in the last D days (indicated by color), by day N.**

We can look at student performance on these two types of assignments, first filtering out students who have not completed any assignments, and then looking for students who by the end of the first or second week of content (day  $N = 7$  or  $14$ ) have an average below a cut-off  $S$  for Study Questions, and a cut-off  $I$  for Investigations. As such, we are predicting whether a student got a course grade under 73% (a.k.a. eventually failed or got a D), from whether the student averaged below  $S$  on Study Questions and  $I$  on investigate assignments, at the time of day  $N$ .

Optimizing based on Cohen’s Kappa, and setting  $N = \text{day } 7$ , we find that the value of  $S$  has almost no impact (and are therefore not shown on Figure 3). For example, if the  $I$  cutoff = 70%, any value of  $S$  from 50% to 95% results in a Cohen’s Kappa between 0.18 and 0.20. If the  $I$  cutoff = 85%, any value of  $S$  from 50% to 95% results in a Cohen’s Kappa between 0.08 and 0.10.

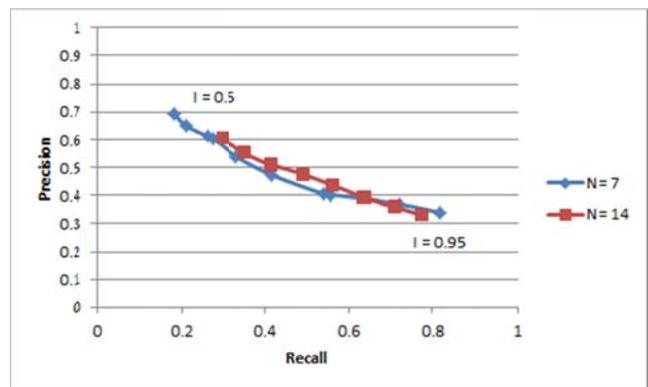
By contrast, the value of  $I$  has substantial impact on model goodness. If the  $I$  cutoff = 65% (and  $S = I$ ), Kappa is 0.20. If the  $I$  cutoff = 95% (and  $S=I$ ), Kappa is -0.05.

The reason for this difference in predictive power between Study Questions and Investigations is likely that Study Questions can be reset. That is, when a student answers a set of Study Questions, the attempt is immediately graded. Students are given feedback and an opportunity to reset the questions and answer them again. Students are encouraged to do this in order to understand the correct answer before they move on. Investigations are more complex, and are also not resettable. In general, then, scores on Study Questions indicate effort and scores on Investigations indicate understanding.

Setting  $S = I$ , we can compute the precision-recall curve for different values of  $I$ , shown in Figure 4.

As Figure 4 shows, as the required grade to not be considered at-risk goes up, the recall goes up but the precision goes down, leading to very different models for different thresholds. It does not appear to make a big difference, however, whether we look at the first week of content, or the first two weeks of content.

To break this down, students who got below 95% on the first week of Soomo Learning Environment content had a 34.0% probability of performing poorly, and 81.8% of students who performed poorly in the course obtained below 95% on the first week of Soomo Learning Environment content. Students who got below 50% on



**Figure 4. Precision-Recall Curve for how well a final grade below 73% is predicted by average grade on assignments ( $I$ ), by day ( $N$ ) 7 and 14.**

the first week of Soomo Learning Environment content had a 69.5% probability of performing poorly, and 18.1% of students who performed poorly in the course obtained below 50% on the first week of Soomo Learning Environment content. As Figure 4 shows, the trade-off between precision and recall is roughly even for values of  $S$  and  $I$  between 50% and 95%.

#### 4. INTEGRATED PREDICTIVE MODEL

Having computed these three indicators, it becomes feasible to look at the three in concert, to see how well we can do overall at predicting whether a student is at risk of obtaining a low grade.

The most straightforward way to do so would simply be to combine the single best version of the three operators described above, with an “or” function. Taking the students who obtained below 95% on the first week of Soomo Learning Environment content, the students who had not yet opened the book on day 2, and the students who had not accessed the book in the last 7 days on day 28, and combining them using an “or” function ends up with the prediction that 98.6% of students are at-risk, a model that is not very usable for intervention (the instructor intervenes for all students).

Alternatively, we can use higher-precision, lower-recall versions of these metrics. Taking the students who obtained below 50% on the first week of Soomo Learning Environment content, the students who had not yet opened the book on day 7, and the students who had not accessed the book in the last 3 days on day 7, and

combining them using an “or” function ends up with the prediction that 84.7% of students are at-risk, still too many interventions.

If, by contrast, we use “and” across the three operators, trying to find students who are definitely not at-risk (e.g. students who demonstrate none of the three behaviors that are indicative of an at-risk student), the higher-precision, lower-recall version of the metrics identifies exactly four students out of 4002 as being at risk. The lower-precision, higher-recall version of the metrics identifies 14.1% of the students as being at-risk, a more workable number for intervention. However, the model achieves a precision of 25.8% and a recall of 10.2%, much worse numbers than single-feature models.

An alternate approach, which we use in this section, is to use a machine-learned model to combine the features in a more complex way. In these analyses, we conduct cross-validation as a check on over-fitting, to determine how reliable these models will be for new students in future sections of the course. Given the focus on predicting performance for future course sections, we conduct the cross-validation at the grain-size of course sections.

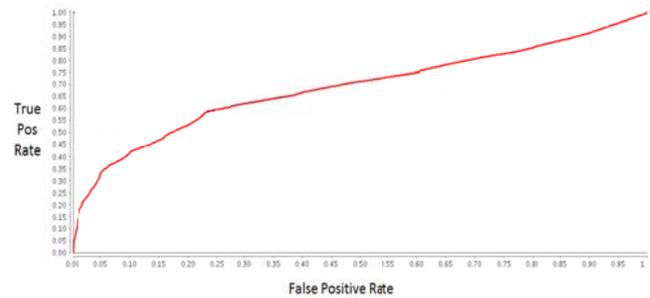
We input to the models the best variants of each feature (in terms of Kappa) seen in the previous sections. We also input extreme threshold variants of the features (high precision-low recall and low precision-high recall) when they achieve comparable Kappa to the best variants. In specific, we include whether the student opened the book on the first N days after the course start (0 days, 2 days, 7 days), whether the student accessed the book recently (D=7, N=28; D=7, N=4; D=3, N=4), and performance on assignments (wk. 1 only, S=I=0.65).

We applied several classification algorithms to these features, and evaluated the resultant models using Kappa, precision, recall, and A', shown in Table 1. A' is the probability that the model can distinguish whether a student is in the at-risk category or not. A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly [9]. A' is used rather than the theoretically equivalent AUC ROC implementation, due to bugs in existing implementations of AUC ROC.

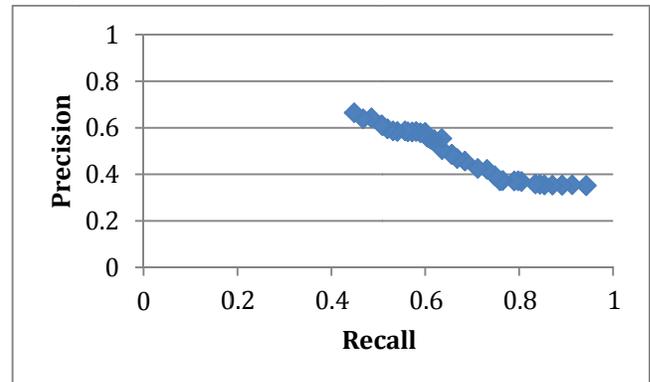
As is often the case, there is not a single best model across all metrics. The best A' is obtained by W-KStar; but this algorithm's Kappa is much lower than other algorithms with very similar A'. Arguably, Logistic Regression, with A' only 0.015 lower than W-KStar, but Kappa 0.111 better, should be preferred. Logistic Regression also achieves the best Recall among the algorithms, while obtaining a middling Precision. Of course, it should be remembered that Recall and Precision can always be traded-off by selecting an alternate threshold based on a Receiver-Operating Characteristic curve, or a Precision-Recall curve (as used throughout this paper), shown in Figures 5 and 6. These curves

**Table 1. Performance of Integrated Predictive Models.**

Algorithm	Kappa	Precision	Recall	A'
W-J48	0.315	0.636	0.435	0.655
W-JRip	0.265	0.570	0.468	0.578
Naïve Bayes	0.231	0.532	0.483	0.666
W-KStar	0.233	0.670	0.288	0.677
Step Regression	0.305	0.697	0.353	0.658
Logistic Regression	0.344	0.568	0.595	0.662



**Figure 5. Receiver-Operating Characteristic Curve for (Cross-Validated) Logistic Regression Version of Integrated Predictive Model.**



**Figure 6. Precision-Recall Curve for (Cross-Validated) Logistic Regression Version of Integrated Predictive Model.**

indicate that recall can be increased to 94.3%, while maintaining precision of 35.1%.

## 5. DISCUSSION AND CONCLUSIONS

In this paper, we have investigated the degree to which student participation in webtext activities within the Soomo Learning Environment, early in the semester, are predictive of eventual student success in a course. We find that it is indeed possible to achieve a reasonable degree of predictive power, and to identify a substantial proportion of the at-risk students, with reasonable precision. Some of these measures have predictive value from the first day of the course, allowing very early intervention.

In aggregate, we find that a combination of these measures leads to A' values in the 0.65-0.7 range, sufficient for intervention, though not quite up to the level of medical diagnostics. The logistic regression version of the combined model can identify 59.5% of students who will perform poorly, achieving precision of 56.8%, 34.4% better than chance. Of course, with any of the approaches used here, confidence thresholds for intervention can be adjusted, leading to more or fewer interventions. If high recall is the goal – attempting to provide intervention to most at-risk students even if some interventions are mis-applied – then the threshold of the logistic regression model can be adjusted, resulting in a model that can identify 94.3% of the students who will perform poorly, but where only 35.1% of the students it identifies performs poorly. This model does better than a single-feature model; even the high recall model from section 3-3 (performance under 95% on early assignments within the webtext) obtained a recall of 81.8% -- lower than the logistic regression model – while achieving comparable precision (34.0%).

However, if the goal is to provide high-cost interventions to the students who are very likely to perform poorly, the logistic regression model is not an optimal choice. The logistic regression model cannot achieve very high precision, even through adjusting thresholds, as shown in Figure 6. However, an alternate approach can be adopted, through using a different predictor algorithm, step regression. This algorithm obtains more precise prediction than logistic regression, with precision of 69.7% and recall of 35.3% for standard thresholds.

Importantly, these measures are based upon interpretable features. They are based upon features that instructors identified as meaningful and having the potential for intervention. The combination of individual-feature models and a comprehensive model enables us to identify which students are at risk, and then to provide instructors with information about which students are at risk, and why. We can specifically identify that a student is at risk because he/she has failed to access the resources, or because he/she has failed to complete the assignments on time, or because he/she has scored poorly on the assignments. With this information, automatically distilled and placed in a user interface within the Soomo platform, faculty will have a means of finding students who most need support and a basis for encouraging them to access the text, do the assigned work, and take the time to do it well.

The first area of future work planned is to enhance the analytics already offered to instructors by Soomo, based on the findings presented here. The success of these interventions, both in terms of improved student grades and improved student retention, will be evaluated in an experiment or quasi-experiment (the final study design will depend upon negotiation with the university which partnered on the analyses discussed in this paper).

However, beyond testing interventions based on the model presented here, there is considerable future work to extend, improve, and study the generalizability of these models. For example, it will be valuable to study what characterizes the students for whom this model functions less effectively. Can additional features, like how much time students spend on assignments, improve overall prediction? And how well will the features identified here apply for different courses, and for different universities, an issue explored by Jayaprakash et al. [10], among others. By answering these questions, we can improve the models, verify their broad applicability, and move to using the models within intervention strategies that can achieve broad positive impact on learners.

## 6. ACKNOWLEDGMENTS

This research was made possible by the active cooperation of our partner university.

## 7. REFERENCES

- [1] Andergassen, M., Modrtischer, F., Neumann, G. (2014) Practice and Repetition during Exam Preparation in Blended Learning Courses: Correlations with Learning Results. *Journal of Learning Analytics*, 1 (1), 48-74.
- [2] Arnold, K. (2010) Signals: Applying Academic Analytics. *Educause Quarterly*. March 2010.
- [3] Arnold, K., Pistilli, M. (2012) Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *Proc. of the 2<sup>nd</sup> International Conference on Learning Analytics*.
- [4] Barber, R., Sharkey, M. (2012) Course Correction: Using Analytics to Predict Course Success. *Proceedings of the 2<sup>nd</sup> International Conference on Learning Analytics*, 259-262.
- [5] Beard, L.A., Harper, C. (2002) Student Perceptions of Online versus on Campus Instruction. *Education*, 122 (4).
- [6] Davis, J., Goadrich, M. (2006) The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*.
- [7] Dekker, G., Pechenizkiy, M., Vleeshouwers, J.M. (2009) Predicting Students Drop Out: A Case Study. *Proc. of the 2<sup>nd</sup> Int'l. Conference on Educational Data Mining*, 41-50.
- [8] Diaz, D.P. (2002) Online Drop Rates Revisited. *The Technology Source*, May/June 2002.
- [9] Hanley, J.A. and McNeil, B.J. (1982) The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143, 29-36.
- [10] Jayaprakash, S.M., Moody, E.W., Lauria, E.J.M., Regan, J.R., Baron, J.D. Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *Journal of Learning Analytics*, 1 (1), 6-47.
- [11] Jiang, S., Williams, A.E., Schenke, K., Warschauer, M., O'Dowd, D. (2014) Predicting MOOC Performance with Week 1 Behavior. *Proceedings of the 7<sup>th</sup> International Conference on Educational Data Mining*, 273-275.
- [12] Ming, N.C., Ming, V.L. (2012) Automated Predictive Assessment from Unstructured Student Writing. *Proceedings of the 1<sup>st</sup> international Conference on Data Analytics*.
- [13] Muilenburg, L.Y., Berge, J.L. (2005) Student Barriers to Online Learning: a factor analytic study. *Distance Education*, 26 (1), 29-48.
- [14] Romero, C., Olmo, J.L., Ventura, S. (2013) A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets. *Proc. of the 6<sup>th</sup> Int'l. Conference on Educational Data Mining*, 268-271.
- [15] Romero, C., Ventura, S., Garcia, E. (2007) Data mining in course management systems: Moodle case study and tutorial. *Computers and Education*, 51 (1), 368-384.
- [16] Thille, C., Schneider, E., Kizilcec, R.F., Piech, C., Halawa, S.A., Greene, D.K. (2014) The Future of Data-Enriched Assessment. *Research and Practice in Assessment*, 9 (4), 5-16.
- [17] Whitmer, J. (2012) *Logging on to improve achievement: Evaluating the relationship between use of the learning management system, student characteristics, and academic achievement in a hybrid large enrollment undergraduate course*. Unpublished Doctoral Dissertation, UC Davis.
- [18] Wolff, A., Zdrahal, Z., Nikolov, A., & Pantucek, M. (2013). Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment. *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 145-149.
- [19] Zhang, J., & Walls, R. (2006). Instructors' self-perceived pedagogical principle implementation in the online environment. *The Quarterly Review of Distance Education*, 7(4), 413-426.

# Do Country Stereotypes Exist in PISA? A Clustering Approach for Large, Sparse, and Weighted Data.

Mirka Saarela  
Department of Mathematical Information  
Technology  
University of Jyväskylä  
Jyväskylä, Finland  
mirka.saarela@gmail.com

Tommi Kärkkäinen  
Department of Mathematical Information  
Technology  
University of Jyväskylä  
Jyväskylä, Finland  
tommi.karkkainen@jyu.fi

## ABSTRACT

Certain stereotypes can be associated with people from different countries. For example, the Italians are expected to be emotional, the Germans functional, and the Chinese hard-working. In this study, we cluster all 15-year-old students representing the 68 different nations and territories that participated in the latest Programme for International Student Assessment (PISA 2012). The hypothesis is that the students will start to form their own country groups when clustered according to the scale indices that summarize many of the students' characteristics. In order to meet PISA data analysis requirements, we use a novel combination of our previously published algorithmic components to realize a weighted sparse data clustering approach. This enables us to work with around half a million observations with large number of missing values, which represent the population of more than 24 million students globally. Three internal cluster indices suitable for sparse data are used to determine the number of clusters and the whole procedure is repeated recursively to end up with a set of clusters on three different refinement levels. The results show that our final clusters can indeed be explained by the actual student performance but only to a marginal degree by the country.

## Keywords

Weighted Clustering, PISA, Sparse Cluster Indices, Country Stereotype

## 1. INTRODUCTION

Certain stereotypes seem to be associated with people from different countries. The French and Italians, for example, are expected to be emotional, while Germany has mainly a functional country stereotype [4], and the Chinese are commonly perceived as hard-working [3]. According to the *Hofstede Model* [6], national cultures can be characterized along six dimensions: power distance, individualism, masculinity, uncertainty avoidance, pragmatism, and indulgence. The

hypothesis in this study is that also the population of 15-year-old students worldwide will start to form their own national groups, i.e., show similar characteristics to their country peers, when clustered according to their attributes and attitudes towards education.

PISA (Programme for International Student Assessment) is a worldwide triannual survey conducted by the Organisation for Economic Co-operation and Development (OECD), assessing the proficiency of 15-year-old students from different countries and economies in three domains: reading, mathematics, and science. Besides evaluating student performances, PISA is also one of the largest public databases<sup>1</sup> of students' demographic and contextual data, such as their attitudes and behaviours towards various aspects of education.

In order to test our hypothesis, we utilize the 15 PISA scale indices (explicitly detailed in [14]), a set of derived variables that readily summarize the background of the students including their characteristics and attitudes. In particular, the *escs* index measures the students' economic, social and cultural status and is known to account for most variance in performance [9]. Additionally, 5 scale indices (*belong*, *atschl*, *atlnact*, *persev*, *openps*) are generally associated with performance on a student-level, while 9 further ones (*failmat*, *intmat*, *instmot*, *matheff*, *anzmat*, *scmat*, *mathbeh*, *matintfc*, *subnorm*) are directly related to attitudes towards mathematics, the main assessment area in the most recent survey (PISA 2012). However, since the assessment material exceeds the time that is allocated for the test, each student is administered solely a fraction of the whole set of cognitive items and only one of the three background questionnaires. Because of this rotated design, 33.24% of the PISA scale indices values are missing.

Moreover, PISA data are an important example of large data sets that include weights. Only some students from each country are sampled for the study, but multiplied with their respective weights they should represent the whole 15-year-old student population. The sample data of the latest PISA assessment, i.e., the data we are working with, consists of 485490 students which, taking the weights into account, represent more than 24 million 15-year-old students in the 68 different territories that participated in PISA 2012.

<sup>1</sup>See <http://www.oecd.org/pisa/pisaproducts/>.

The content of this paper is as follows. First, we describe the clustering algorithm that allows us to work with the large, sparse and weighted data (Sec. 2). Second, we present the clustering results (Sec. 3) and their relevance to our hypothesis, i.e., how the clusters on the different levels can be characterized and to what extent they form their own country groups. Finally, in Sec. 4, we conclude our study and discuss directions for further research.

## 2. THE CLUSTERING APPROACH

Sparsity of PISA data must be taken into account when selecting or developing a data mining technique. With missing values one faces difficulties in justifying assumptions on data or error normality [14, 15], which underlie the classical second-order statistics. Hence, the data mining techniques here are based on the so-called nonparametric, robust statistics [5]. A robust, weighted clustering approach suitable for data sets with a large portion of missing values, non-normal error distribution, and given alignment between a sample and the population through weights, was introduced and tested in [16]. Here, we apply a similar method with slight modifications, along the lines of [7] for sampled initialization and [17] for hierarchical application. All computations were implemented and realized in Matlab R2014a.

### 2.1 Basic method

Denote by  $N$  the number of observations and by  $n$  the dimension of an observation of the data matrix  $\mathbf{X}$ ; and let  $\{w_i\}, i = 1, \dots, N$  be the positive sample-population-alignment weights. Further, let  $\{\mathbf{p}_i\}, i = 1, \dots, N$ , be the projection vectors that define the pattern of the available values [10, 1, 14, 15]. The weighted spatial median  $\mathbf{s}$  with the so-called available data strategy can be obtained as the solution of the projected Weber problem

$$\min_{\mathbf{v} \in \mathbf{R}^n} \mathcal{J}(\mathbf{v}), \quad \mathcal{J}(\mathbf{v}) = \sum_{i=1}^N w_i \|\text{Diag}\{\mathbf{p}_i\}(\mathbf{x}_i - \mathbf{v})\|, \quad (1)$$

where  $\text{Diag}\{\mathbf{p}_i\}$  denotes the diagonal matrix corresponding to the given vector  $\mathbf{p}_i$ . As described in [8], this optimization problem is nonsmooth, i.e., it is not classically differentiable. However, an accurate approximation for the solution of the nonsmooth problem can be obtained by solving the regularized equation (see [1])  $\sum_{i=1}^N \frac{w_i \text{Diag}\{\mathbf{p}_i\}(\mathbf{s} - \mathbf{x}_i)}{\max\{\|\text{Diag}\{\mathbf{p}_i\}(\mathbf{s} - \mathbf{x}_i)\|, \delta\}} = \mathbf{0}$  for  $\delta > 0$ . This is solved using the SOR (Sequential Overrelaxation) algorithm [1] with the overrelaxation parameter  $\omega = 1.5$ . We choose  $\delta = \sqrt{\varepsilon}$  for  $\varepsilon$  representing the machine precision.

In case of clustering with  $K$  prototypes, i.e., the centroids that represent the  $K$  clusters, one determines these by solving the nonsmooth problem  $\min_{\{\mathbf{c}_k\}_{k=1}^K} \mathcal{J}(\{\mathbf{c}_k\})$ , where all  $\mathbf{c}_k \in \mathbf{R}^n$  and

$$\mathcal{J}(\{\mathbf{c}_k\}) = \sum_{k=1}^K \sum_{i \in I_k} w_i \|\text{Diag}\{\mathbf{p}_i\}(\mathbf{x}_i - \mathbf{c}_k)\|. \quad (2)$$

Hereby,  $I_k$  determines the subset of data being closest to the  $k$ th prototype  $\mathbf{c}_k$ . The main body of the so-called iterative relocation algorithm for minimizing (2), which is referred as *weighted k-spatialmedians*, consists of successive application of the two main steps: i) find the closest prototype for each observation, and ii) recompute all prototypes  $\mathbf{c}_k$  using the

attached subset of data. For the latter part, we compute the weighted spatial median as described above. Note that the first step of finding the closest prototype of the  $i$ th observation,  $\min_k \|\text{Diag}\{\mathbf{p}_i\}(\mathbf{x}_i - \mathbf{c}_k)\|$ , does not need to take the positive weight  $w_i$  in (2) into account.

The next issues for the proposed method are the determination of the number of clusters  $K$  and the initialization of the clustering algorithm for a given  $k$ . Basically, the quality of a cluster can be defined by minimal within-cluster distances and maximal between-cluster distances. Therefore, for the first purpose, we use the approach suggested in [16] and apply three internal cluster indices, namely *Ray-Turi (RT)* [13], *Davies-Bouldin (DB)* [2], and *Davies-Bouldin\** ( $DB^*$ ) [11]. All these indices take both aspects of clustering quality into account: In essence, the clustering error (2), i.e., the sum of the within-cluster distances, to be as small as possible, is divided with the distance between the prototypes (minimum distance for RT and different variants of average distance for DB and  $DB^*$ ), to be as large as possible. When testing a number of possible numbers of prototypes from  $k = 2$  into  $K_{\max}$ , we stop this enlargement when all three cluster indices start to increase.

Concerning the initialization, again partly similarly as in [16], we use a weighted k-means++ algorithm in the initialization of the spatial median based clustering with the weights  $\sqrt{w_i}$ . A rigorous argument for such an alignment was given in [9] where the relation between variance (weighted k-means) and standard deviation (weighted *k-spatialmedians*) was established. Because of local character, the initialization and the search are repeated  $N_s = 10$  times and the solution corresponding to the smallest clustering error in (2) is selected. Furthermore, the weighted k-means++ is applied in the ten initializations with ten different, disjoint data samples (10% of the whole data) that were created using the so-called *Distribution Optimally Balanced, Stratified Folding* as proposed in [12], with the modified implementation given in [7]. Such sampling, by placing a random observation from class  $j$  and its  $N_s - 1$  nearest class neighbors into different folds, is able to approximate both classwise densities and class frequencies in all the created data samples. Here, we use the 68 country labels as class indicators in stratification.

### 2.2 Hierarchical application

Because a prototype-based clustering algorithm always works with distances for the whole data, the detection of clusters of different size, especially hierarchically on different scales or levels of abstraction, can be challenging. This is illustrated with the whole PISA data set in Fig. 1, which shows the values of the three cluster indices for  $k = 1, \dots, 68$ . For illustration purposes, also the clustering error as defined in (2), denoted as ‘Elbow’, is provided. All indices have their minimum at  $k = 2$  which suggest the division of the PISA data to only two clusters. Note that the geometrical density and low separability of the PISA scale indices might be related to their standardization to have zero mean and unit variance over the OECD countries.

Hierarchical application of the *k-spatialmedians* algorithm was suggested in [17]. The idea is simple: Similarly to the divisive clustering methods, apply the algorithm recursively

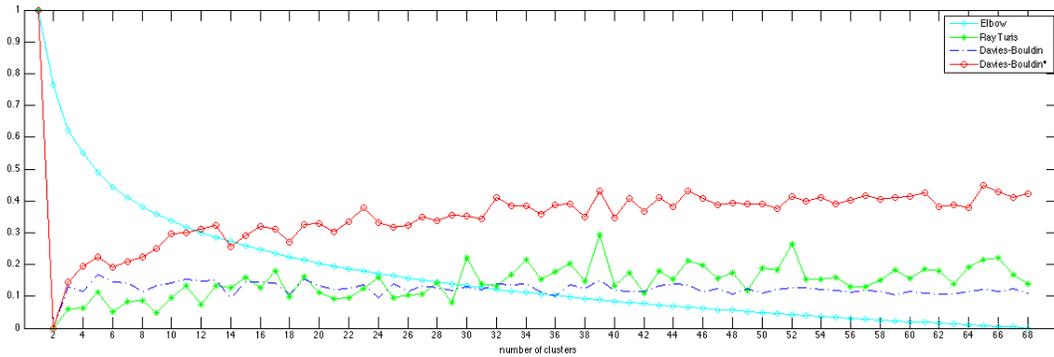


Figure 1: Cluster indices and error slope for the whole sparse PISA data scaled into range  $[0, 1]$ .

to the cluster data sets that have been determined using the basic approach. For the PISA data here, we realized a recursive search of the *weighted k-spatialmedians* with the depth of three levels, ending up altogether with 2 (level 1), 4 + 4 (level 2), and 6 + 12 + 10 + 6 & 2 + 8 + 3 + 6 clusters (level 3). The wall-clock time for each individual clustering problem was several hours.

### 3. RESULTS

As discussed in Sec. 1, we use the 15 PISA scale indices that readily summarize most of the students' background as data input for our clustering algorithm. By following the mixture of the partitional/hierarchical clustering approach as described above, we first of all, provide the results of the weighted sparse data clustering algorithm when applied to the whole PISA data (first level). Then, recursively, the results of the algorithm for the newly obtained clusters at the second and third level of refinement are given. For all the clusters at each level, we compute the relative share of students from each country, i.e., the weighted number of students in the cluster in relation to the whole number of 15-year-old students in the country. Moreover, in order to reveal the deviating characteristics of the appearing clusters, we visualize and interpret (i.e., characterize) the cluster prototypes in comparison to the overall behavior of the entire 15-year-old student population in the 68 countries by always subtracting the weighted spatial median of the whole data from the obtained prototypes.

#### 3.1 First Level

Since, as pointed out in Sec. 2.2, all the sparse cluster indices suggest two, we first run our weighted sparse clustering algorithm for  $K = 2$ . The clustering result on the first level is shown in Fig. 2. The division of these clusters is unambiguous: All scale indices that are associated with high performance in mathematics have a positive value for Cluster 2 and a negative value for Cluster 1. Likewise, those two scale indices that are associated with low performance in mathematics, i.e., the self-responsibility for failing in mathematics (*failmat*) and the anxiety towards mathematics (*anaxmat*), show a positive value for Cluster 1 and a negative value for Cluster 2. As can be expected by these profiles, the mean mathematics performance of Cluster 1 is much lower than the mean math performance of Cluster 2 (see Table 1).

When we consider the relative number of students from dif-

Table 1: Characteristics of global/first level clusters

Cluster	population size ( $\varphi$ in %)	math score		
		$\emptyset$	$\varphi$	$\sigma$
1	13399687 (52%)	445	442	449
2	11321033 (48%)	468	461	475
all	24720720 (50%)	456	451	461

ferent countries, we see that every country has students in both clusters. In fact, the distribution of the 15-year-old student population between the two clusters is quite equal in each country. For Cluster 1, the mean percentage of students from a country is 55% while for Cluster 2, the mean is 45%, and both have the standard deviation of 10. In all of the in PISA participating countries and territories, there are higher and lower performing students and it seems that they share the same characteristics. Additionally, the distribution between girls and boys is quite equal, although somewhat in favor of boys: Only 48% of the students in the cluster with the scale indices that are associated with high performance in mathematics are girls. Moreover, the average math score of the boys is in both clusters higher than the average math score of the girls (see Table 1).

#### 3.2 Second Level

Following the approach as described above, we run the clustering algorithm again, but this time for each of the two global clusters obtained in the first level separately. According to the same rule given in Sec. 2.1, i.e., stop enlarging  $k$  during the search when all the cluster indices are increasing, we get for both of the global clusters  $K = 4$  as a number for their subclusters.

##### 3.2.1 Subclusters of Cluster 1

Table 2: Characteristics of subclusters of Cluster 1

Cluster	population size ( $\varphi$ in %)	math score		
		$\emptyset$	$\varphi$	$\sigma$
1-1	2792046 (56%)	439	438	440
1-2	3873035 (52%)	391	388	394
1-3	3072064 (58%)	466	464	468
1-4	3662542 (45%)	491	489	492

The subclusters of the global Cluster 1 are visualized in Fig. 3 and characterized in Table 2. If we set the threshold

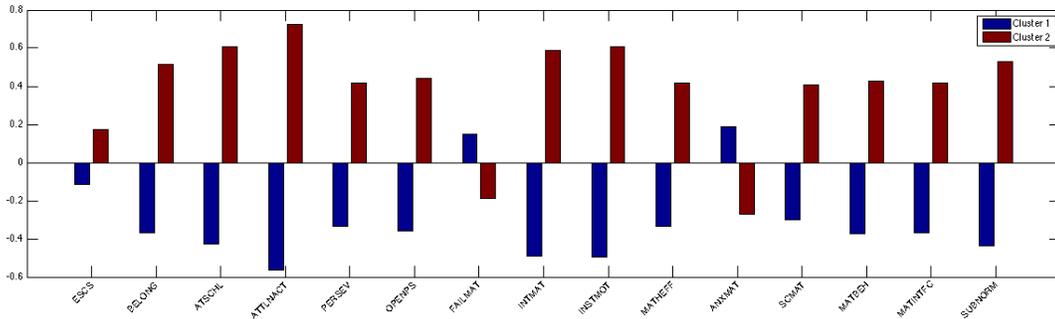


Figure 2: Characterization of the two global clusters.

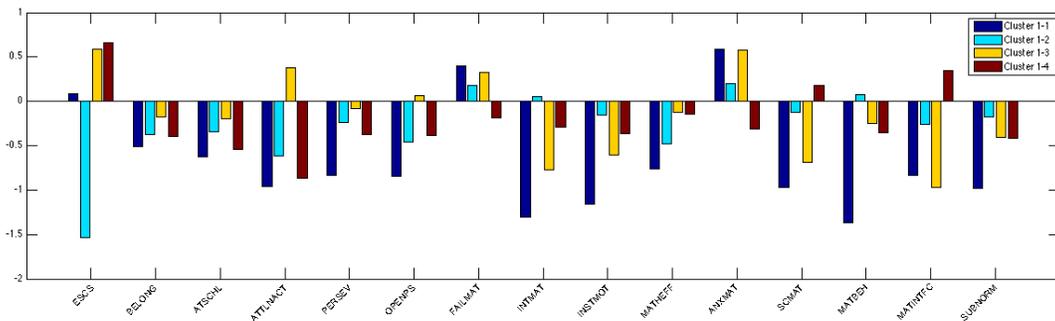


Figure 3: Characterization of the four subclusters of Cluster 1.

of how many students should at least be from one country to 21%, we obtain the following countries for the subclusters: Cluster 1-1 (i.e., subcluster 1 of Cluster 1) contains at most students from East Asia with the exception of China: More than 30% of Japan’s 15-year-old student population belongs to this cluster, 26% of Korea’s and 25% of Taiwan’s. The remaining students represent a mixture from many different countries which, however, are only represented by less than 21% of their 15-year-old student population.

Cluster 1-2 contains almost entirely students from developing countries. Hereby, students from Vietnam form with 49% the majority. Moreover, Indonesia, Thailand (both > 30%) and Brazil, Colombia, Peru, Tunisia, and Turkey (all > 25%) are represented by this cluster. The cluster is, as can be seen from Fig. 3, most notably characterized by a very low economic, social and cultural status (*escs*). That means that the students in this cluster - as a subset of the global Cluster 1 which already represented the more disadvantaged students (see Fig. 2) - are the most disadvantaged.

Cluster 1-3 consists in the majority of students from Eastern Europe: Serbia, Montenegro, Hungary, Slovak Republic (all > 23%) and Romania (almost 22%) constitute the majority. As we can see from Fig. 3, this cluster is the only one in the group of subclusters of the global Cluster 1, that generally was characterized by negative attitudes and perceptions (see Fig. 2), which actually can be distinguished by positive attitudes towards school (*atlnact*). Moreover, it is the cluster with mainly girls in it.

Cluster 1-4 accommodates mainly students from Western

and Central Europe. Most of the 15-year-old student population from the Netherlands (39%) are in this cluster, followed by Belgium with 29%, and the Czech Republic with 27%. This cluster is characterized by the highest *escs* among the students of the global Cluster 1. Furthermore, although they have negative values in most of the scale indices, they have a higher self-concept in math, and also much higher intentions to use mathematics later in life in comparison with their peers.

### 3.2.2 Subclusters of global Cluster 2

Table 3: Characteristics of subclusters of Cluster 2

Cluster	population size (♀ in %)	math score		
		∅	♀	♂
2-1	3127958 (43%)	526	523	528
2-2	2739481 (54%)	457	457	458
2-3	3521092 (50%)	400	397	403
2-4	1932502 (44%)	515	506	523

The subclusters of the global Cluster 2 are characterized in Fig. 4 and summarized in Table 3. Again, we search for clusters that mostly deviate from the others. Cluster 2-1 is such a cluster: The students in this cluster have the highest average math score (see Table 3), the highest intentions to pursue a mathematics related career but a sense of belonging to school (*belong*) and subjective norms in mathematics (*subnorm*) that are only about the same as the average of the whole 15-year-old student population (see Fig. 4). The subjective norms in mathematics measure how people important to the students, such as their friends and parents, view mathematics. In the global Cluster 2, those students

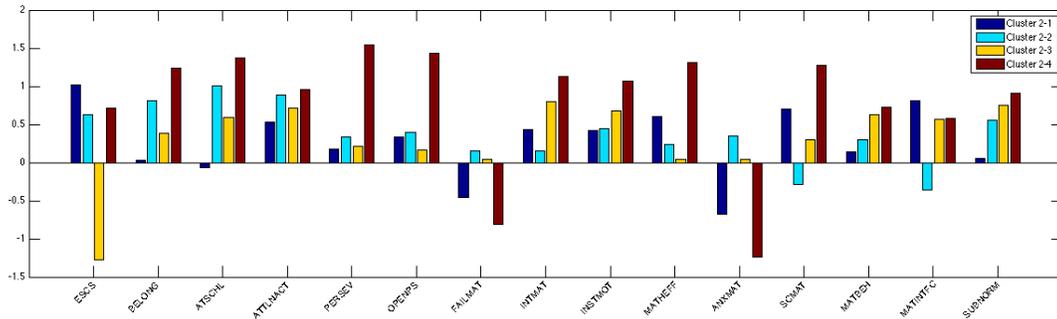


Figure 4: Characterization of subclusters of Cluster 2.

who had high positive values in the other scale indices associated with high performance in mathematics, also thought that their friends and family view mathematics as important (their *subnorm* value is very high, see Fig. 2). Students in this cluster, however, seem not to be influenced or affected by what people close to them think. It appears to be a rather strong cluster that also has the highest percentage of boys in it. For this cluster, we again compute the relative number of students from each country. And indeed, it shows a very clear country-profile. The highest percentage of students come from the English-speaking and Nordic countries: Denmark (more than 30%), Iceland and Sweden (both > 26%) have the highest percentages of their 15-year-old student population in this cluster. Followed by the two highest performing districts in the USA, namely Connecticut and Massachusetts, with both more than 25%. Besides these countries and territories, the cluster has also a high share of students from Norway, Finland, Great Britain, Australia, and Canada (almost 22% or more). Additionally, the USA has with more than 21% still a relatively high share of students in this cluster. According to the Hofstede Model (see Sec. 1), all of these countries are characterized by high individualism.

Also Cluster 2-3 shows an explicit country profile: 36% of the 15-year-old student population from India are in this cluster. Moreover, the cluster consists of students from Peru and Thailand (both 30%), Turkey (27%) and Vietnam (26%). Altogether, we find here the most disadvantaged students (indicated by the very negative *escs*) among the subgroups of the global Cluster 2 and the largest share of students come from the developing countries. However, these students have very positive attitudes towards education and show relatively high values in all scale indices that are associated with high performance in mathematics.

To this end, Cluster 2-2 and Cluster 2-4 have less obvious country affiliations. Cluster 2-2 can at best be described as containing mostly countries with Islamic culture. Most of the students are from the United Arab Emirates and Albania (both 21%), Kazakhstan and Jordan (both 19%). According to the Hofstede Model, these countries are similar in that way that they all show very high power distance. Cluster 2-4 has with 25% the highest share of students also from Kazakhstan, but the remaining countries in this cluster (all have less than 17% of their 15-year-old students population in it) are widely mixed.

Altogether, among the clusters at the second level, Cluster 2-1 appears to be the most interesting one, i.e., the most distinct group with the clearest country profiles.

### 3.3 Third Level

Recursively, we repeat the same approach on the next level, i.e., for the subclusters of the eight clusters identified in Sec. 3.2. For all the new subclusters, the best number of clusters as determined by the cluster indices are as follows: 6, 12, 10, and 6 for the four subclusters of the first global cluster, and 2, 8, 3, and 6 for the four subclusters of the second global cluster. This means that we have 53 different clusters on this level - almost as many as different countries/territories in the whole PISA 2012 data. If our hypothesis is true, we should be able to find clusters that clearly contain more students from certain countries. Exactly as in Sec. 3.2, we first of all compute the basic facts of each cluster. Note, however, that the deeper we go in the hierarchy the more clusters we encounter and the more difficult it becomes to define clear rules and thresholds to distinguish significant characterizations of clusters.

#### 3.3.1 Subclusters of Cluster 1-3

Table 4: Characteristics of subclusters of Cluster 1-3

Cluster	population size (♀ in %)	math score		
		∅	♀	♂
1-3-1	335240 (61%)	493	492	495
1-3-2	262779 (48%)	539	540	538
1-3-3	368591 (51%)	461	460	462
1-3-4	273629 (66%)	492	491	492
1-3-5	359721 (56%)	427	428	426
1-3-6	275513 (63%)	437	436	438
1-3-7	264017 (63%)	443	441	447
1-3-8	318607 (63%)	460	457	464
1-3-9	216704 (60%)	421	418	424
1-3-10	397263 (56%)	481	482	480

The first interesting cluster appears in the 1-3 group. Cluster 1-3-8 accommodates mainly students from South West Europe: Austria, Liechtenstein, Spain, France, and Italy. According to the Hofstede Model, all of these countries are depicted by high avoidance of uncertainty.

#### 3.3.2 Subclusters of Cluster 1-4

The characterization of the subclusters in the 1-4 group are provided in Fig. 6, and summarized in Table 5. Also here,

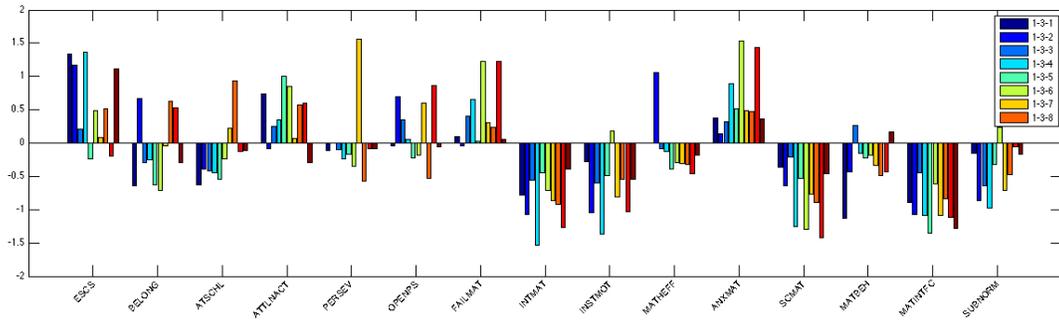


Figure 5: Characterization of subclusters of Cluster 1-3.

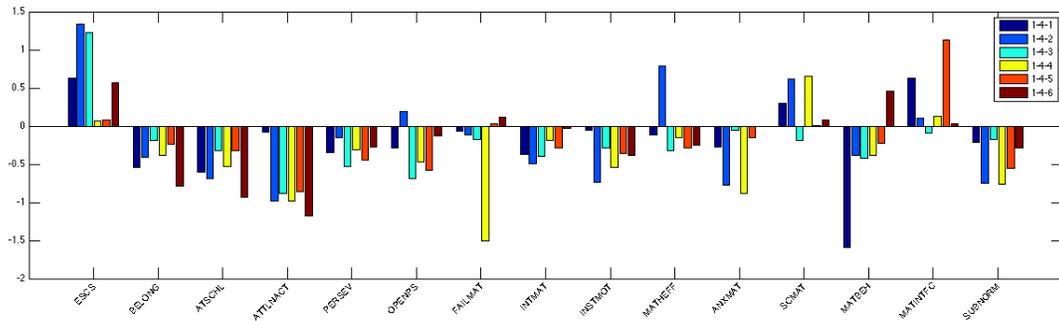


Figure 6: Characterization of the subclusters of Cluster 1-4.

Table 5: Characteristics of subclusters of Cluster 1-4

Cluster	population size ( $\varphi$ in %)	math score		
		$\emptyset$	$\varphi$	$\sigma$
1-4-1	485599 (48%)	481	480	482
1-4-2	520763 (38%)	556	558	555
1-4-3	771799 (53%)	494	494	495
1-4-4	489528 (43%)	497	491	501
1-4-5	754515 (48%)	470	467	473
1-4-6	640338 (38%)	461	465	458'

we are searching for explicit country clusters. This search is realized by looking at the histograms and identifying those clusters that for some countries have a considerably higher share of their 15-year-old student population in it than for the remaining countries. The histogram in Fig. 7 shows one example of this for Cluster 1-4-2: In this cluster, the portion of students in it deviates significantly from the others for exactly one country with 10% of its 15-year-old student population. This country is the Netherlands. For all other countries, the share of their 15-year-old student population in this cluster is less than 6% (see Fig. 7). As can be seen from Fig. 6, this ‘Netherlands Cluster’ is characterized by having the highest math self-efficacy amongst its group.

Cluster 1-4-1 is again a mixture of Nordic and English-speaking countries. The highest share of students in this cluster come from the United Kingdom, Ireland, Norway, New Zealand, and Sweden. As these two country profiles were already detected to be in the same cluster on the higher cluster level (see Sec. 3.2.1), it really seems that students from these countries share many similar characteristics.

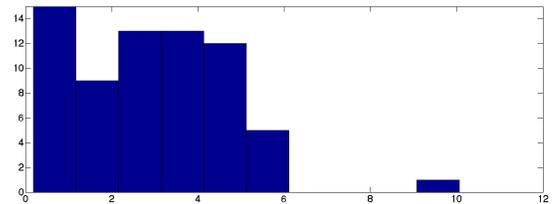


Figure 7: Histogram of the distribution of countries from the students in Cluster 1-4-2.

Cluster 1-4-4 has the highest share of East Asian countries including two of the three districts of China that participated in PISA 2012. Most of the students in this cluster come from Japan, followed by Taiwan, Macao-China and Hong Kong-China. One of the most distinct feature of this cluster is, as can be seen from Fig. 6, the high self-concept in mathematics (*scmat*). According to the Hofstede Model (see Sec. 1), all of these countries show high pragmatism.

### 3.3.3 Subclusters of Cluster 2-1

Table 6: Characteristics of subclusters of Cluster 2-1

Cluster	population size ( $\varphi$ in %)	math score		
		$\emptyset$	$\varphi$	$\sigma$
2-1-1	1346930 (40%)	562	557	566
2-1-2	1781028 (45%)	498	500	497

From Sec. 3.2, we concluded that Cluster 2-1 was the most interesting one. Moreover, Cluster 2-1 was the cluster that had the highest share of two country profiles in it: On the one hand, the English-speaking countries, and, on the other

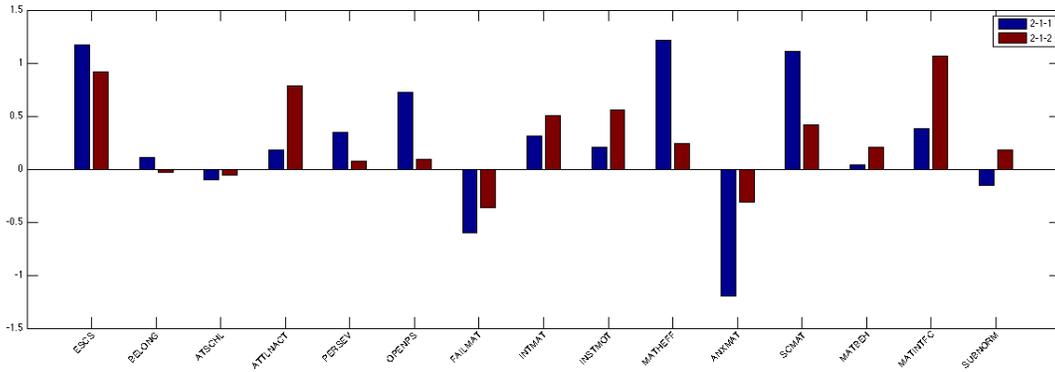


Figure 8: Characterization of subclusters of Cluster 2-1.

hand, the Nordic countries. Interestingly, the cluster indices also suggest to divide this cluster into two further countries. However, when we look again at those countries that have the highest percentages of their 15-year-old students, the two clusters still contain mostly students from both country profiles. For example, 15% of the Danish 15-year-old student population are in Cluster 2-1-1, and 14% are in Cluster 2-1-2. Similarly, 14% of the 15-year-old student population from Connecticut are in Cluster 2-1-1, and 11% in Cluster 2-1-2. Apparently, this cluster does not divide any further between Nordic and English-speaking countries. It only divides the high-performing students from these countries into two types: On the one hand, the type that has a very high self-efficacy (*matheff*) as well as self-concept (*scmat*) in math, i.e., the students that have a very high belief in their own ability, and, on the other hand, the type that has very high intentions to pursue a math related career (*matintfc*).

However, also a new clear group of countries appears. Cluster 2-1-1 has a very high share of German-speaking countries in it: More than 12% of Germany’s and Switzerland’s 15-year-old student population, and 10% of Austria’s can be found in this cluster. None of these countries appear in the sibling Cluster 2-1-2 when the threshold is set to 9%. It seems that high-performing German-speaking students feel very confident in solving mathematical tasks but only show a moderate positive value in the intentions to use mathematics later in life, a characteristic that one would associate the most with the traditional functional German stereotype (see Sec. 1) that is expected to attach great importance to utilitarianism [4]. According to the Hofstede Model, all of these three German-speaking countries are considered to be highly masculine.

### 3.3.4 Subclusters of Cluster 2-4

Table 7: Characteristics of subclusters of Cluster 2-4

Cluster	population size (♀ in %)	math score		
		∅	♀	♂
2-4-1	186107 (37%)	533	528	536
2-4-2	430729 (40%)	582	575	588
2-4-3	261838 (45%)	440	436	443
2-4-4	378120 (50%)	477	468	486
2-4-5	430105 (47%)	520	519	521
2-4-6	245603 (40%)	516	500	526

The subclusters of Cluster 2-4 are summarized in Table 7 and characterized in Fig. 9. The clearest country profile among this group is 2-4-6: It consists to the highest share of students from high-performing Asian countries: Shanghai-China and Singapore. As we can see from Fig. 9, similarly to Cluster 1-4-4 (see Sec. 3.3.2) that also contained a high share of East Asian students, this cluster is characterized as well by a high self-concept in mathematics (*scmat*). The students in this cluster believe that mathematics is one of their best subjects, and that they understand even the most difficult work. Furthermore, as already found for Cluster 1-4-4, also for this cluster the main countries show high pragmatism according to the Hofstede Model.

## 4. CONCLUSIONS

In this article, we have introduced a clustering approach that has both partitional and hierarchical components in it. Moreover, the algorithm takes weights, aligning a sample with its population into account and is suitable for large data sets in which many missing values are present.

The hypothesis in our study was that the different clusters determined by the algorithm, when all students with their attitudes and behaviors towards education are given as input, could be explained by the country of the students in particular clusters. Our overall results on the first level showed that in each cluster students from all countries exist and that the actual test performance (as well as a simple division in positive and negative attitudes towards education) explain the clusters much better than the country from which the students in the particular cluster come from.

However, on the next two levels many clusters were detected that obviously had a much higher share of students from certain countries. For example, an Eastern Europe, a German-speaking, an East Asia, and a developing countries cluster were identified. On the second level, also a very clear cluster that consisted to a high portion of Nordic and English-speaking countries appeared. This cluster did not split further on the next level to fully separate these two distinct country profiles. Instead, the cluster was divided into two student types, of which both the Nordic as well as the English-speaking countries seem to have an almost equal share of their students from.

Summing up, we conclude that groups of similar countries,

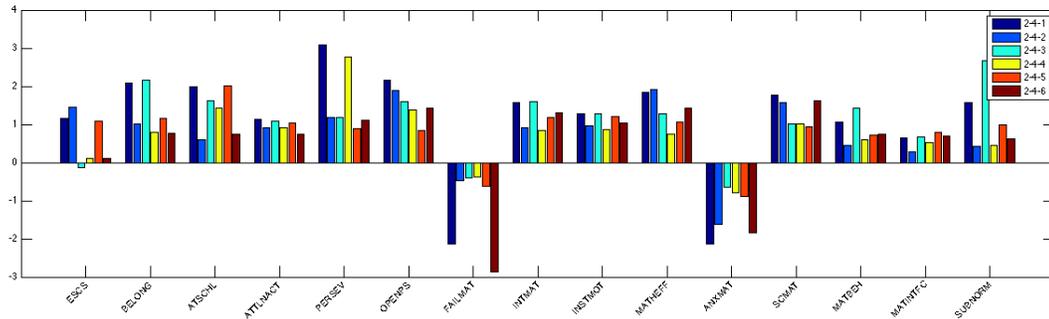


Figure 9: Characterization of subclusters of Cluster 2-4.

e.g., by means of geographical location, culture, stage of development, and dimensions according to the Hofstede Model, can be found by clustering PISA scale indices but the actual country stereotypes exist only to a very marginal extent. However, in a further work the rules how to find relevant clusters could be improved and more variables than the 15 scale indices utilized here could be included to the algorithm. The PISA scale indices are linked to math performance and in every country there are higher and lower performing students who share similar overall characteristics. Nevertheless, we think that the overall results presented here show a very promising behavior already, and we expect that the resulting clusters of our algorithm could be explained even clearer by the country of the students if additional information such as the students' temperament would be available for the clustering algorithm.

## References

- [1] S. Äyrämö. *Knowledge Mining Using Robust Clustering*, volume 63 of *Jyväskylä Studies in Computing*. University of Jyväskylä, 2006.
- [2] D. L. Davies and D. W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.
- [3] S. Harrell. Why do the Chinese work so hard? Reflections on an entrepreneurial ethic. *Modern China*, pages 203–226, 1985.
- [4] M. F. Herz and A. Diamantopoulos. Activation of country stereotypes: automaticity, consonance, and impact. *Journal of the Academy of Marketing Science*, 41(4):400–417, 2013.
- [5] T. P. Hettmansperger and J. W. McKean. *Robust non-parametric statistical methods*. Edward Arnold, London, 1998.
- [6] G. Hofstede. Dimensionalizing cultures: The Hofstede model in context. *Online readings in psychology and culture*, 2(1):8, 2011.
- [7] T. Kärkkäinen. On cross-validation for MLP model evaluation. In *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science (8621), pages 291–300. Springer-Verlag, 2014.
- [8] T. Kärkkäinen and E. Heikkola. Robust formulations for training multilayer perceptrons. *Neural Computation*, 16:837–862, 2004.
- [9] T. Kärkkäinen and M. Saarela. Robust principal component analysis of data with missing values. *To appear in the Proceedings of the 11th International Conference on Machine Learning and Data Mining MLDM*, 2015.
- [10] T. Kärkkäinen and J. Toivanen. Building blocks for odd-even multigrid with applications to reduced systems. *Journal of Computational and Applied Mathematics*, 131:15–33, 2001.
- [11] M. Kim and R. Ramakrishna. New indices for cluster validity assessment. *Pattern Recognition Letters*, 26(15):2353–2363, 2005.
- [12] J. Moreno-Torres, J. Sáez, and F. Herrera. Study on the impact of partition-induced dataset shift on  $k$ -fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1304–1312, 2012.
- [13] S. Ray and R. H. Turi. Determination of number of clusters in  $k$ -means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, pages 137–143, 1999.
- [14] M. Saarela and T. Kärkkäinen. Discovering Gender-Specific Knowledge from Finnish Basic Education using PISA Scale Indices. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 60–68, 2014.
- [15] M. Saarela and T. Kärkkäinen. Analysing Student Performance using Sparse Data of Core Bachelor Courses. *JEDM-Journal of Educational Data Mining*, 7(1):3–32, 2015.
- [16] M. Saarela and T. Kärkkäinen. Weighted clustering of sparse educational data. *To appear in 23rd Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.
- [17] P. Warttinen and T. Kärkkäinen. Hierarchical, prototype-based clustering of multiple time series with missing values. *To appear in 23rd Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.

# Student Privacy and Educational Data Mining: Perspectives from Industry

Jennifer Sabourin   Lucy Kosturko   Clare FitzGerald   Scott McQuiggan  
Curriculum Pathways  
SAS Institute Inc.  
100 SAS Campus Drive  
Cary, NC, USA  
1.919.677.8000  
{Jennifer.Sabourin, Lucy.Kosturko, Clare.FitzGerald, Scott.McQuiggan}@sas.com

## ABSTRACT

While the field of educational data mining (EDM) has generated many innovations for improving educational software and student learning, the mining of student data has recently come under a great deal of scrutiny. Many stakeholder groups, including public officials, media outlets, and parents, have voiced concern over the privacy of student data and their efforts have garnered national attention. The momentum behind and scrutiny of student privacy has made it increasingly difficult for EDM applications to transition from academia to industry. Based on experience as academic researchers transitioning into industry, we present three primary areas of concern related to student privacy in practice: policy, corporate social responsibility, and public opinion. Our discussion will describe the key challenges faced within these categories, strategies for overcoming them, and ways in which the academic EDM community can support the adoption of innovative technologies in large-scale production.

## Keywords

Student privacy, student data, policy

## 1. INTRODUCTION

Educational data mining (EDM) is chiefly defined by the application of sophisticated data mining techniques to solving problems in education [1]. A powerful tool, EDM has been successfully incorporated into applications that optimize student learning in both research and commercial products. EDM's proven effectiveness has led many—from the U.S. government to individual teachers—to recognize the ability of student data in guiding education and to support the development and use of these technologies in schools. Consequently, applications utilizing EDM technologies have become more prevalent in school systems [2], [3].

However, the increase in EDM usage has raised public awareness of how much data is being collected about students. The applications and companies that collect and use student data are coming under scrutiny, as parents, advocates, and public officials grow concerned over student privacy. A recent cascade of events has focused attention on privacy concerns [4]. For example, there has been a rise in high-profile attacks on consumer data from online retailers and financial institutions. Large, well-trusted institutions have been targeted for using student data in undesirable ways [5]. Promising companies driven by student data have been brought down by public opinion with no evidence of wrong-doing. Calls for stricter policy from privacy advocates have led to more than 100 bills being introduced in U.S. state legislatures to address issues of student privacy in 2014 [4]. In response, the White House has

announced plans for federal legislation modeled after state policies [6].

Negative media attention and increased legislation threaten to stifle EDM, particularly in commercial settings. Public opinion may make organizations wary to invest in and use EDM techniques while legislation could make it more difficult to collect and use student data in effective ways. We believe it is an incredibly important time for the EDM community to be aware of the challenges being faced in industry. The rise of concern over student privacy has strong implications for how new EDM approaches can be integrated into wide-reaching applications as well as the amount of funding available to public and private entities wishing to innovate in this space.

These issues are receiving rapidly increasing attention and driving action at the national level. It is critical that the discussions around these issues include experts from the EDM community. This paper discusses the issues and implications faced by commercial applications of educational data mining because of recent focus on student privacy. In this paper, we discuss the role of policy, corporate social responsibility, and public opinion in framing the work of and challenges to industry. We discuss strategies for overcoming these challenges and present opportunities for the EDM community to address rising concerns.

## 2. EDM AND INDUSTRY

The profile of the EDM community has risen in the past decade—in research, commercial products, public attention—bolstered by three related shifts. First, educational technology has been more widely adopted. School systems are investing in laptops, mobile devices and other technologies in favor of static textbooks. These technologies offer opportunities for data collection that did not exist before. Student records are also increasingly digitized including test scores, attendance records, and bus schedules. These digitized records have generated a wealth of longitudinal data that was previously difficult and expensive to collect [7].

Second, there has been a dramatic rise in computational power and storage capacities. This storage allows for the collection and housing of large amounts of data, even data that is not presently known to be useful. The increased computational power has generated sophisticated algorithms that can mine large corpora of data to identify connections that would previously be impossible [8] and has even created the possibility for robust decision engines to operate in real time learning systems.

Finally, public officials and industry experts are starting to recognize the power of educational data mining [9]. Government funding opportunities for data-driven education solutions are on the

rise, and reports estimate that educational data mining has the potential to provide meaningful economic impact worldwide [10].

There are many areas of EDM research, each with unique applications to industry. At the individual level, data on student behavior, from mouse clicks to eye tracking, provide insight on how students interact with educational technology. For example, EDM has produced models of help abuse [11], attention to hints [12], and conversational dynamics in online forums [13]. These insights and techniques can help commercial educational technology providers design better applications that support positive interactions with students while being user-friendly.

Another key area of research at the individual level is assessment. EDM applications have been used to identify student mastery as well as knowledge gaps. Frequently, these models are based on student performance on relevant tasks but can go beyond measuring what students did correctly and incorrectly by modeling underlying knowledge [14]. Some assessments are cleverly hidden, called “stealth assessment,” in games or other non-threatening applications [15]. These systems develop robust models of student knowledge while avoiding the negative effects associated with test performance; in fact, students may not even know they are being tested. These techniques have important implications for educational technologies, ranging from the design of new systems that can revolutionize the way assessment is done in formal learning environments, to technologies that can identify gaps in student knowledge and recommend resources to help fill them.

EDM technologies have also driven personalized learning beyond tailoring instruction to what students know, but also to how they learn based on needs and preferences. Systems can identify commonly used strategies by students and select which are most effective, for particular individuals, under specific circumstances [16]. EDM techniques have also supported technologies that guide students towards learning how to regulate their own learning, by helping them to recognize and overcome weaknesses in their current approaches [17]. These techniques are critical in creating applications that use the most effective techniques and support personalized learning.

Finally, EDM research has examined mining data at higher levels, including schools and districts, for a variety of purposes such as exploring college readiness [18], identifying the best teachers [19], or driving district spending [7]. Commercial products are commonly used to house this level of data and communicate findings to necessary stakeholders. Data mining on this organizational or even regional level has allowed for the development of early warning systems to predict student drop-out before it happens as well as identify holes in district-level education [7].

In essence, “educational data mining and learning analytics have the potential to make visible data that have heretofore gone unseen, unnoticed, and, therefore, unactionable” [9]. The approaches outlined in this section offer significant promise in helping to improve education delivery and outcomes, but their success is contingent on the collection, storage, and use of large amounts of quality student data. Companies who wish to collect and use student data must operate under increased public and governmental scrutiny, which can, and has, created barriers to the use of EDM in industry.

### 3. STUDENT PRIVACY

Privacy is chiefly a question of access. Unlike anonymity or confidentiality, peoples’ interest in privacy is about controlling the

access of others to themselves [20]. How to safeguard a child’s privacy is a particularly complex question because of their vulnerability. Children are incapable of “protecting their own interests through negotiation for informed consent” because they are likely to misunderstand risks or be coerced into participating [20].

This need to protect has led to the formation of student privacy advocacy groups and driven the adoption of legislation. The restrictions required to comply with this legislation and maintain good public opinion have a significant impact on the adoption of data-based solutions in education.

### 3.1 Policy

In the U.S., we have established privacy protections for children by asking for consent from parents or guardians and implementing policies which hold organizations, both public and private, accountable for obtaining consent when collecting, storing or disclosing data, and ensuring proper usage. There are two federal acts that address children’s privacy directly: the Federal Education Rights and Privacy Act (FERPA), and the Children’s Online Privacy Protection Act (COPPA).

#### 3.1.1 Federal Education Rights and Privacy Act

Before the enactment of the Federal Education Rights and Privacy Act (FERPA) in 1974, parents and students had little access to education records. Meanwhile, that same information was widely available to outside authorities without requiring the consent of parents or students [21]. FERPA applies to any school receiving federal funds and levies financial penalties for not following it. While complying with FERPA is a local responsibility [22], the way it defines education records and regulates third party access to them matters to private companies.

According to FERPA, education records contain information on student background, academic performance, grades, standardized test results, psychological evaluations, disability reports, and anecdotal remarks from teachers or school authorities regarding academic performance or student behavior (FERPA, 1974, 20 U.S.C § 1232g(a)(1)(D)(3)). Generally, schools looking to disclose information contained in these records must have written permission from a parent or eligible student, an individual who is 18 or attending post-secondary school. Education record information is only shared with a third party on the assurance that that third party will not allow further outside access to requested information without additional written parental consent (FERPA, 1974, 20 U.S.C § 1232g(b)(4)(B)). Some activities, however, do not require written consent. Under FERPA, third parties, including private companies, may use information within education records for official or contracted evaluation, audit, and compliance activities without parental or student consent but are barred from using that data for marketing [23].

FERPA is not without controversy. Some have argued that schools improperly apply FERPA in order to protect information that does not fall under its definition of an education record and that such denials of disclosure are in violation of state open record laws [24]. Others voice concern over contracted service providers’ use of data not covered by FERPA citing that the content of emails housed in cloud services, data from identification cards, or data collected by schools to outsource a service could, depending on the contract, be used or sold for marketing purposes [23].

#### 3.1.2 Children’s Online Privacy Protection Act

While FERPA affects private interests, the Children’s Online Privacy Protection Act (COPPA) speaks more directly to

operations, particularly to online service providers that have direct or actual knowledge of users under 13 and collect information online. Made effective in 2000, COPPA “requires web hosts and content providers to seek parental consent to store data about children under age 13” [25]. To be fully compliant, parents must be given the opportunity to review terms of service and privacy policies of each commercial website where their child’s information may be stored. Parental consent is required before any information can be collected, and parents can retract this permission and request all data be deleted at any time. Technology providers are required to disclose what data is being collected about children and what it is being used for. They are also expected to provide reasonable measures of security and discard of data once it is no longer needed. [25], [26]. Overall, COPPA seeks to encourage responsible business practice and reduce “imprudent disclosures of personal information by children” [27].

COPPA, too, has fallen under criticism. It is difficult to enforce and there many ways in which companies can comply with the “letter of the law” without truly protecting student privacy. COPPA has also been criticized for not reflecting the changes in online technologies accessed by children. In an effort to stay current with technological advancement, COPPA underwent revisions in 2013 to “address changes in the way children use and access the Internet, including the increased use of mobile devices and social networking” [28] by widening the definition of what constitutes children’s personal information to include cookies, geolocation, photos, videos, and audio recordings [28]. These updates bolstering safeguards for student data appear further scaffolded by actions from the White House.

### 3.1.3 Student Digital Privacy

Driven by concerns over the efficacy of national policies, state legislators have seen the introduction of a large number of policies aimed at protecting student data [4], [29]. New national legislation may also be on the horizon for protecting student privacy [30]. The proposed Student Digital Privacy Act, modeled after a California statute, prohibits companies from selling student data to third parties except for educational purposes [6]. While it is unclear when, or if, this legislation will be enacted, it has already drawn criticism. Parents and privacy advocates fear it is too lenient while industry experts warn that increased legislation may limit development of important educational solutions [31].

These industry experts point to the voluntary Student Privacy Pledge (<http://studentprivacypledge.org/>) as a means to achieve better management of student data without federal legislation [32]. At the time of writing, 108 companies have chosen to sign the pledge, vowing that they will not sell student data or use data for targeted advertisement, and will maintain transparency about how data is being collected and used. This pledge is an indication that commercial education technology providers are taking steps towards the corporate social responsibility that will garner respect among users and privacy advocates.

### 3.1.4 Student Privacy: International Perspectives

The United States has relied on a piecemeal approach to regulating privacy where legislation is sector driven and may be enacted at state and/or federal levels [33]. Conversely, the European Union enacted a comprehensive set of regulations in the Data Protection Directive under which student privacy issues are largely subsumed. This set of regulations requires unambiguous consent of individuals before collecting or processing personal data as well as a prohibition on collecting sensitive information with few exceptions [34].

Canadian national privacy legislation is stipulated in the Personal Information Protection and Electronic Documents Act which, like COPPA, is focused on how commercial entities use personal information, as well as the Privacy Act which limits the collection, use, and disclosure of personal information by federal government entities. Meanwhile, similar to United States, Canadian provinces follow their own patchwork of student specific legislation. Ontario, for instance, follows the Education Act, the Municipal Freedom of Information and Protection of Privacy Act as well as the Personal Health Information Protection Act. The Canadian system is less comprehensive than the EU, but is perhaps more effective in safeguarding student interests than the US due to an “all-encompassing and prescriptive nature” [34].

## 3.2 Corporate Social Responsibility

Corporate social responsibility refers to companies taking an active part ensuring they have a positive impact on social welfare. In the case of privacy, this means working to truly protect student data and collect and use it responsibly. Design weaknesses and enforcement shortcomings in student privacy legislation can often allow companies to appear more responsible than they are. Organizations can legally comply, a potentially cumbersome process on its own, but do little to actually ensure best practices are being followed and student interests are protected.

This is a significant issue in markets of educational technologies designed for children under the age of 13, the population protected by COPPA. True compliance with the intents behind COPPA can be “both overwhelming and prohibitive” [35] which privacy scholar, Danah Boyd, believes has led to an apprehension to target users under thirteen. Avoiding the issue is often seen as “easier and more cost effective than attempting to tackle COPPA compliance.” [35]

Currently there are many websites, online services, and mobile apps that are widely used in classroom settings including those classrooms with younger students. For example, Google Apps for Education reportedly serves an estimated 40 million students, teachers, and administrators. Similarly, over 47 million teachers have accounts with Edmodo, the “world’s largest K-12 social learning community”. Education technology is estimated to be an 8 billion dollar industry [30] and technology providers are often trying to find their niche while maintaining competitive advantage. Issues arise when creating a product that will be useful to education, ensuring that student data is collected and managed responsibly, and managing profit and competition are at conflict with one another. This balance of constraints is one of the strongest challenges faced by companies seeking to gather and use educational data responsibly.

### 3.2.1 Supporting Shared-Device Settings

Classroom constraints make the educational market particularly unique. While 1:1 schools (1 device per student) and Bring Your Own Device (BYOD) integrations are on the rise, many schools reflect a shared-device model (e.g., classroom sets, device carts). In order to achieve personalized learning in this setting, individual accounts are often necessary. Yet individual accounts raise several issues.

The first is that secure account authentication can be troublesome. Expecting students, especially younger students, to remember their login credentials is unreasonable in many cases. Keeping up with login information is particularly challenging when classrooms attempt to take advantage of multiple systems each requiring their own unique username and password. In fact, a report by the National School Board Association notes “password reuse due to

lax controls (i.e., password written on a sticky note)” as a particular concern for using online educational services [36]. Some systems utilize password pictures or avatars for younger populations, which could be a viable option depending on the type of data; however, when sensitive data such as images, video, and performance evaluations are often protected behind account logins, it is important to enable users to securely protect their data.

Furthermore, for those companies without any interest in storing student data on servers, shared-device settings can unintentionally force this responsibility. In a 1:1 environment, user data can simply be stored on students’ devices as there is little concern over other individuals gaining access to the data; thus, eliminating the need to device solutions for complying to privacy legislation and avoiding security breaches. Appealing to shared-device environments, on the other hand, necessitates such measures including cloud storage, a solution known to concern parents [37]. Moreover, when schools rely on online educational resources and mobile apps that utilize cloud storage, they often relinquish control of that student data, which is particularly alarming given the fact that FERPA “generally requires districts to have direct control of student information when disclosed to third-party service providers” [23]. A recent report by Fordham Law School on the issue of student privacy and cloud computing found “school district cloud service agreements generally do not provide for data security and even allow vendors to retain student information in perpetuity with alarming frequency” [23]. The report goes on to point out that “fewer than 25% of the agreements [pulled from a national sample and reviewed by the committee] specify the purpose for disclosures of student information, fewer than 7% of the contracts restrict the sale or marketing of student information by vendors, and many agreements allow vendors to change the terms without notice.” In sum, supporting ubiquitous student access through cloud computing necessitates a great deal of legal accommodations.

### 3.2.2 Consent

The process for simply creating an account can be cumbersome and time-consuming for two primary reasons: 1) companies cannot collect personal information from students under thirteen without parental consent, and 2) students under 18 cannot legally agree to the Terms of Service agreement accompanying many registration processes. In some cases, schools obtain a blanket agreement from parents at the beginning of the year allowing instructors to create accounts for students. Although, if teachers do not have legal consent from parents to create accounts on their students’ behalf, having to wait for parental approval can easily derail an entire lesson quickly making the resource obsolete to the instructor.

Unfortunately, many companies find “restricting” users, even audiences for which the product is intended, streamlines the registration process by avoiding parental consent. Susan Fox of the Walt Disney Company articulates this concern by stating “Operators are keenly aware that consumers will quickly move on if websites are slow to load, functionality is delayed, or registration-type processes stand between users and their content.” [38] Furthermore, because virtual age verification is difficult and easily bypassed, compliance can still be met by adding statements such as “we do not knowingly collect data” from persons under thirteen in privacy policies. As a result, sidestepping the intentions of COPPA makes it difficult for other companies to remain competitive and “discourage[s] startups from innovating for the under-thirteen market” [38].

### 3.2.3 Disclosure

Parental consent and disclosure are two of the major tenants of COPPA compliance. Responsible adherence suggests that companies are forthcoming with information and present details clearly to parents when asking consent. However, this can be troublesome and may serve to harm parental opinions of an application rather than help. For example, there is concern that anything requiring parental permission (e.g., PG-13 or R-rated movies) is somehow objectionable. This misconception stems from the fact that “parents and youth believe that age requirements are designed to protect their safety, rather than their privacy.” [39] As a result, companies attempting to be compliant may be inadvertently penalized because of public opinion.

Privacy policies are another form of disclosure that may be open to misinterpretation. Regulated by the FTC, privacy policies require companies to be upfront about the collection and use of user data. There is, however, much debate about their effectiveness. In a recent survey, over half of interviewed online Americans agreed with the statement, “When a company posts a privacy policy, it ensures that the company keeps confidential all the information it collects on users” and even fewer users read—or, in the case of these younger populations, can read and comprehend—them [40]. Others have proposed alternative solutions that more clearly convey the purposes of data collection [41] yet truly articulating the intricacies of EDM and personalized learning environments will take proofs of concept and time.

## 3.3 Public Opinion

One of the largest drivers behind the focus on privacy of student data is the vocal concern of parents and stakeholders in the media. The issue has been gaining a great deal of attention and has already had serious impacts on the landscape of educational technology providers.

Perhaps one of the best examples of the power of backlash from parents and media is the demise of a well-funded nonprofit company based entirely on the promise of educational data mining [5]. Though it was widely supported by districts, industry experts, and funding agencies, its efforts were undermined by parental protests and media frenzy. The company did not respond to rising concerns and failed to staunch fears over data misuse and protection. Though there was no evidence of any wrong-doing on the part of the company, parents and privacy advocates protested that the risk was too great. As the protest grew larger and more vocal districts began withdrawing participation in early 2014.

While anecdotal, this example demonstrates the need for industries relying on student data to get ahead of the rising panic by demonstrating value (i.e. driving innovation and/or supporting student learning). While EDM has its proponents [2], [9], their beliefs do not propagate to the general public. Parents and privacy advocates do not believe the benefits to be gained by educational technologies driven by student data outweigh the risks. The top concerns for these individuals are varied, as are their levels of awareness with various issues. Commonly discussed areas of concern with regards to student data include marketing, security, decision-making, and the “unknown”.

### 3.3.1 Marketing

A primary purpose behind existing and proposed legislation is to limit the use of children’s data to drive targeted advertisements [42]. It is, therefore, unsurprising that this is one of the top concerns of parents and school officials. However, much of this legislation and parental concern stems from children’s interactions with non-educational sites and technologies. In this case, it makes sense to

limit targeted advertising of toys, food items, and other commercial goods, especially when considering findings that children are mostly unable to distinguish advertisement from regular content [43].

However, it is not clear that this protection is warranted in educational contexts. Much of the “advertisement” promoted by the EDM community centers around identifying gaps in a student’s understanding and surfacing the most effective and engaging ways to fill those gaps. These advertisements have strong potential to benefit students, but some parents and other privacy advocates are only able to see that their children are being exploited for profit [2][3].

### 3.3.2 Decision-Making

Several EDM technologies provide a promise to support data-driven decisions about how best to help students learn. This is seen regularly in tools that select problem sets, feedback, or lesson plans based on students’ prior interactions [44]. Data may also be presented to educators or administrators making decisions about whether a student needs additional attention or if they are college-ready [18]. These types of decisions start drawing parental concern. While parents understand (though they may not agree with) data from high stakes examinations being used to drive decisions about their children’s education, data from private learning technologies is more unclear. Parents fear that undisclosed “stealth assessments” could negatively impact their children’s future – from academics through the work force [42].

### 3.3.3 Security.

In addition to concerns over what companies may do with the data they collect, many parents are also fearful over what may happen if that data enters the wrong hands. The news is rife with incidents of data breaches with individual financial and other personal data being accessed by malicious parties. Parents concerns over student data security is certainly valid, though experts think it unlikely that this type of data would draw attack as it is less obviously lucrative when compared with financial and other personal records [2].

Existing legislation does put restrictions on the collection and storage of personally identifiable information (PII) of minors and responsible companies do strive to ensure anonymization of data. However, the rapid increase in the quantity of data collected and the sophistication of data mining procedures increase the likelihood that data that does not seem like PII on the surface could be combined to identify individuals [8].

### 3.3.4 The “Unknown”.

Finally, many fears from parents and the media cannot be vocalized. There is something unsettling about the quantity of data being collected, stored, and mined about children, even if there is

no real threat to safety or happiness. Much of this fear stems from the lack of transparency that surrounds the issues. Companies want to keep practices secret to avoid giving competitors an advantage. Privacy policies are often vague and uninformative to reduce the risk of drawing criticism or lawsuits. This is especially a concern as media tensions and attacks rise. Parents know that large quantities of data are being collected about their children, and it is unclear why it is being collected, how it is being used, and what it could be used for in the future. Rising distrust between parents, stakeholders and technology providers shuts down constructive conversation and only serves to exacerbate the issue.

## 4. ROLE OF THE EDM COMMUNITY

The barriers to industry applications of educational data mining techniques stem from several sources. Existing and proposed policy put restrictions on how data can be collected, stored and used. Companies can technically comply with legislation without much impact on their product or processes. However, strictly adhering to policies and offering real privacy protection often makes accessing and using educational tools more difficult, giving less socially responsible companies a competitive advantage. Public opinion can lead to the destruction of companies with no unethical practices and can drive money away from investment in data-based educational technologies. The EDM community has an important role to play in keeping these challenges in check and allowing innovation to thrive (Table 1).

### 4.1 Transparency

A lack of clarity, rampant misunderstanding, and a high degree of uncertainty fuel sentiment against the collection and use of student data. The main concerns of many parents and privacy advocates are largely not reflective of actual practice.

Consequently, the EDM community is uniquely positioned to advance public understanding for what student data is really being used. EDM professionals can better describe how data is being used, what innovations it supports, explain the focus of current research, and portray likely research foci of the field. Parental concerns may be allayed knowing that people are not actively contributing to the outcomes they most fear.

The community can also disseminate details about the effectiveness of these approaches beyond the research community. Showing the strengths of these techniques may help concerned individuals see the benefits that individual children and the education system as a whole stand to gain.

As new approaches are developed, consider creating public-facing talking points that can be used to communicate with concerned parties. These points should describe what data is being used and

**Table 1.** The role of the EDM community on the issue of student privacy.

Point of Concern	Proposed Solution	Action Item
Policy	<ul style="list-style-type: none"> <li>Policy Activism</li> </ul>	<ul style="list-style-type: none"> <li>Remain abreast of proposed or approved policy changes.</li> <li>Actively voice expert opinions to policy makers.</li> </ul>
Corporate Social Responsibility	<ul style="list-style-type: none"> <li>Awareness of classroom constraints</li> </ul>	<ul style="list-style-type: none"> <li>Develop algorithms that minimize the amount of data needed to produce effective results where possible.</li> <li>Avoid requirements for individual accounts when possible.</li> </ul>
Public Opinion	<ul style="list-style-type: none"> <li>Understanding public opinion</li> <li>Transparency</li> </ul>	<ul style="list-style-type: none"> <li>Actively work to correct misconceptions about student data and privacy concerns.</li> <li>Set research agendas aimed at better understanding public understanding of privacy issues.</li> </ul>

how it can benefit students. They should be written in a way that is clear and easy for non-experts to understand.

## 4.2 Research Agendas

The EDM community can also drive research towards areas that may help compliance with legislation and improve public opinion. Algorithms that minimize the amount of data needed to produce effective results would be beneficial to companies wishing to keep privacy concerns at bay. Researchers should consider the tradeoffs when developing new “big data” approaches. More data may lead to more effective techniques but it also may represent an increased violation of privacy. Finding a balance can support widespread dissemination in commercial technologies

It is important that researchers understand the classroom constraints of commercial educational technologies, especially when it comes to privacy. For a variety of reasons it is often less feasible to guarantee that data comes from a specific individual. Approaches that are robust enough to take this into account will allow educational technologies to be successful in more environments.

An additional area of research that could benefit from the involvement of the EDM community is research on the public understanding of privacy issues. The EDM community could be involved in cross-disciplinary research to ensure that communication surrounding EDM techniques is accurate and clear, and organizational privacy policies are widely understood.

## 4.3 Policy Activism

Finally, we encourage members of the EDM community to become active as policy debates grow. It is important to stay up to date on proposed policy changes and to consider how these changes may impact research agendas and the commercial applicability of those findings. Policy changes may increase constraints in commercial applications that could drive shifts in funding made available to EDM research. The policy changes affect both communities.

The discussion also needs more contributions from EDM experts. Consider voicing concerns to local officials and provide guidance on how policy should be directed. Too much of the current dialogue is based on a fear and misunderstanding. These voices are currently overpowering the experts who support the use of data in education.

## 5. CONCLUSION

Educational data mining offers significant promise in improving student learning and education systems as a whole. However, these systems are often driven by the collection of large amounts of student data, which is a growing concern to many. Shifts in public opinion and policy have led to barriers to the adoption of EDM technologies in commercial applications and threaten to stifle future innovation. Several fundamental issues are driving this trend.

The first is the role of trust, fear, and misunderstanding. It is difficult to combat the fear associated with the unknown. Companies and experts in the field must work hard to both gain the trust of the public and communicate what is actually being done with student data. Trust must extend the other way as well. Companies need to trust that by being open about their practices they will not be attacked by concerned external stakeholders. Fear from companies about the reactions of privacy advocates encourages silence on their parts and serves to reduce overall transparency. Both parties must build trust to move towards an open and productive dialogue.

Another recurring theme centers on legislation that has not yet had the desired effect. Privacy advocates view current legislation as too

lenient and many companies are able to comply without actually protecting student data. In fact, the legislation may actually harm companies that do the most to protect student privacy. Voluntary pledges offer one solution, though they are not without problems; conflicts of interest often erode even the best self-policing strategies. Many, if not most, companies may support the spirit of such pledges but be unable to sign due to any number of various technicalities. Active involvement from all invested parties will be crucial to designing new legislation that will strike a balance between allowing data to be used for the good of education, while protecting the privacy of individual students.

Finally, differing views on the appropriateness of private institutions delivering public goods underscore many of the issues discussed. If commercial vendors are going to be the major providers of educational technologies to school systems there needs to be a shift in how the public perceives these companies. Stifling the success of these companies only serves to keep innovative learning technologies out of the classroom. Still, deference to privacy concerns is an important component of occupying a role in part characterized by public stewardship. Discussions about the ethical limits of financially profiting off of student data need to be addressed directly by corporate, research, and public interests with adequate emphasis on risk and potential system improvements.

Overall, there are a variety of issues contributing to concerns over student privacy and how these concerns impact industry applications of educational data mining. These issues are extremely prominent and are not expected to lose momentum soon. The EDM community stands to play an important role in how discussions and legislation around student privacy evolve in the coming years. The landscape of educational data and privacy will continue to shift, and we hope with increased involvement this shift will be positive for researchers and industries interested in using educational data mining to support student learning.

## 6. REFERENCES

- [1] G. Siemens and R. S. J. Baker, “Learning Analytics and Educational Data Mining: Towards Communication and Collaboration,” pp. 252–254, 2012.
- [2] S. Simon, “Data Mining Your Children,” *Politico*, 15-May-2014.
- [3] N. Singer, “With Tech Taking Over in Schools, Worries Rise,” *The New York Times*, 14-Sep-2014.
- [4] S. Trainor, “Student data privacy is cloudy today, clearer tomorrow,” *Phi Delta Kappan*, vol. 96, no. 5, pp. 13–18, 2015.
- [5] B. Herold, “inBloom to Shut Down Amid Growing Data-Privacy Concerns,” *Education Week*, 04-Feb-2014.
- [6] “FACT SHEET: Safeguarding American Consumers & Families,” *The White House*, 2015. [Online]. Available: <http://www.whitehouse.gov/the-press-office/2015/01/12/fact-sheet-safeguarding-american-consumers-families>.
- [7] J. McQuiggan and A. W. Sapp, *Implement, Improve and Expand Your Statewide Longitudinal Data System: Creating a Culture of Data in Education*. 2014.
- [8] Mayer-Schonberger and K. Cukier, *Big Data*. New York, New York: Houghton Mifflin Harcourt Publishing Company, 2013.
- [9] U. S. D. of Education, “Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: An Issue Brief,” 2012.

- [10] J. Manyika, M. Chui, D. Farrel, S. Van Kuiken, P. Groves, and E. Almasi, "Open data: Unlocking Innovation and Performance with Liquid Information," 2013.
- [11] V. Aleven and K. Koedinger, "Limitations of Student Control: Do Students Know When They Need Help?," in *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, 2000, pp. 292–303.
- [12] C. Conati, N. Jaques, and M. Muir, "Understanding attention to adaptive hints in educational games: an eye-tracking study," *Int. J. Artif. Intell. Educ.*, vol. 23, pp. 136–161, 2013.
- [13] M. Wen, D. Yang, and C. Rose, "Sentiment Analysis in MOOC Discussion Forums: What does it tell us?," in *Proceedings of the 7th International Conference on Educational Data Mining*, 2014, pp. 257–260.
- [14] R. S. J. Baker, A. T. Corbett, and V. Aleven, "More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing," *Knowl. Creat. Diffus. Util.*, pp. 406–415, 2008.
- [15] V. Shute, "Stealth Assessment in Computer-Based Games to Support Learning," in *Computer Games and Instruction*, 2011, pp. 503–523.
- [16] J. P. Rowe, L. R. Shores, B. W. Mott, and J. C. Lester, "Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments," *Int. J. Artificial Intell. Educ.*, vol. 21, no. 1–2, pp. 115–133, 2011.
- [17] J. Sabourin, L. R. Shores, B. W. Mott, and J. C. Lester, "Predicting Student Self-Regulation Strategies in Game-Based Learning Environments," in *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, 2012.
- [18] H. Chen, "Identifying Early Indicators for College Readiness," 2007.
- [19] L. Pappano, "Using Research to Predict Great Teachers," *Harvard Education Letter*, 2011.
- [20] M. Sieber, J. Tolich. "Planning ethically responsible research" Sage Publications, 2012.
- [21] S. Carey, "Students, Parents and the School Record Prison A Legal Strategy for Preventing Abuse.pdf," *J. Law Educ.*, vol. 3, p. 365, 1974.
- [22] T. L. Elliott, D. Fatemi, and S. Wasan, "Student Privacy Rights — History , Owasso , and FERPA," *J. High. Educ. Theory Pract.*, vol. 14, no. 4, 2014.
- [23] J. R. Reidenberg, N. C. Russell, J. Kovnot, T. B. Norton, and R. Cloutier, "Privacy and Cloud Computing in Public Schools," 2013.
- [24] R. Silverblatt, "Hiding behind ivory towers: Penalizing schools that improperly invoke student privacy," *Georgetown Law J.*, vol. 101, pp. 493–517, 2013.
- [25] B. Smith and J. Mader, "Protecting Students' Privacy - By Law," *Sci. Teach.*, vol. 81, no. December, 2014.
- [26] Children's Online Privacy Protection Act of 1998, 5 U.S.C. 6501-6505.
- [27] A. Allen, "Minor Distractions: Children, Privacy and E-commerce," *Houston Law Review*, 2001.
- [28] J. Mayfield, "Revised Children's Online Privacy Protection Rule Goes Into Effect Today Federal Trade Commission," *Federal Trade Commission*, 01-Jul-2013.
- [29] Data Quality Campaign, "2014 Student Data Privacy Bills," 2014.
- [30] E. Brown, "Obama to propose new student privacy legislation," *The Washington Post*, Washington D.C., 19-Jan-2015.
- [31] S. Simon, "Barack Obama to seek limits on student data mining," *Politico*, 11-Jan-2015.
- [32] H. Tsukayama, "More than 70 companies just signed a pledge to protect student data privacy - with some notable exceptions" *The Washington Post*, 12-Jan-2015.
- [33] D. Banisar, "Privacy and data protection around the world," in 21st International Conference on Privacy and Personal Data Protection, 1999.
- [34] G. Yee, "Security and Privacy in Distance Education," in *Information Security and Ethics: Concepts, Methodologies, Tools, and Applications*, 1st ed., H. Namati, Ed. 2007, p. 4110.
- [35] D. Boyd, "Response to COPPA Rule Review, 16 CFR part 312, Project No. P-104503," Washington D.C., 2011.
- [36] N. S. B. Association, "Data in the Cloud: A Legal and Policy Guide for School Boards on Student Data Privacy in the Cloud Computing Era," Alexandria, VA, 2014.
- [37] C. S. Media, "Student Privacy Survey," 2014.
- [38] S. Fox, "In the Matter of COPPA Rule Review, 16 CFR Part 312, Project No. P-104503," Washington D.C., 2011.
- [39] J. Palfrey, D. Boyd, and U. Gasser, "How the COPPA, as Implemented, Is Misinterpreted by the Public: A Research Perspective," 2010.
- [40] Pew Research Center, "What Internet Users Know about Technology and the Web," 2014.
- [41] C. DeLorme, "Response to COPPA Rule: Comments to be placed on the public record," Washington D.C., 2012.
- [42] M. Madden, S. Cortesi, U. Gasser, A. Lenhart, and M. Duggan, "Parents, Teens, and Online Privacy," 2012.
- [43] B. L. Wilcox, D. Kunkel, J. Cantor, P. Dowrick, S. Linn, and E. Palmer, "Report of the APA Task Force on Advertising and Children," 2004.
- [44] K. Vanlehn, "The Behavior of Tutoring Systems," *Int. J. Artif. Intell. Educ.*, vol. 16, no. 3, pp. 227–265, 2006.

# Beyond Prediction: First Steps Toward Automatic Intervention in MOOC Student Stopout

Jacob Whitehill  
Harvard University  
jacob\_whitehill@harvard.edu

Joseph Williams  
Harvard University  
joseph\_jay\_williams@harvard.edu

Glenn Lopez  
Harvard University  
glenn\_lopez@harvard.edu

Cody Coleman  
MIT  
colemanc@mit.edu

Justin Reich  
Harvard University  
justin\_reich@harvard.edu

## ABSTRACT

High attrition rates in massive open online courses (MOOCs) have motivated growing interest in the automatic detection of student “stopout”. Stopout classifiers can be used to orchestrate an intervention before students quit, and to survey students dynamically about why they ceased participation. In this paper we expand on existing stop-out detection research by (1) exploring important elements of classifier design such as generalizability to new courses; (2) developing a novel framework inspired by control theory for how to use a classifier’s outputs to make intelligent decisions; and (3) presenting results from a “dynamic survey intervention” conducted on 2 HarvardX MOOCs, containing over 40000 students, in early 2015. Our results suggest that surveying students based on an automatic stopout classifier achieves higher response rates compared to traditional post-course surveys, and may boost students’ propensity to “come back” into the course.

## 1. INTRODUCTION

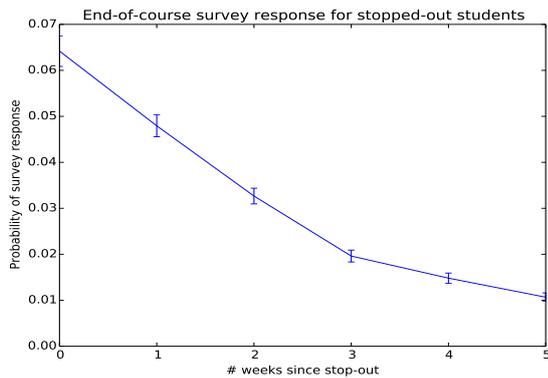
Massive open online courses (MOOCs) enable students around the world to learn from high-quality educational content at low cost. One of the most prominent characteristics of MOOCs is that, partly due to the low cost of enrollment, many students may casually enroll in a course, browse a few videos or discussion forums, and then cease participation [12, 6, 10]. Some MOOCs offer the ability to receive a “certificate” by completing a minimum number of assignments or earning enough points, and for the most part the number of students who certify in MOOCs is far lower than the number of students who register. This is not necessarily a problem – students may enroll for different reasons, not everyone cares about formal certification, and if students learn anything from a MOOC, that is arguably an important gain.

On the other hand, the fact that most students who enroll in a MOOC do not complete the course still warrants further

investigation. For example, there may be some students who genuinely intended to complete a course when they enrolled but, upon encountering the lecture materials, quiz problems, or even other students, felt discouraged, frustrated, or bored, and then stopped participating in the course. Indeed, Reich [11] found that, of students who completed HarvardX pre-course surveys and expressed the *intent* to complete the course, only 22% of such students actually did so. A deeper understanding of the reasons why students stop out of a course could help course developers improve course content.

HarvardX, Harvard’s strategic initiative for online education, is interested in understanding students’ learning experiences in order to improve both online and residential education. Some of the questions we are currently tackling include *who* is enrolling in HarvardX courses, *why* are they enrolling, and *how* can we improve their educational experiences. In particular, we would like to know whether students stop out of HarvardX courses for reasons exogenous to their course experience – e.g., increased stress at work – or whether they quit because they disliked something about the course, especially things that course developers might be able to improve. One step towards answering this question, which we instituted starting in 2014, was to request of every student who enrolled in a HarvardX course to answer a *post-course survey*, which asks whether they liked the course and how it could be improved. Unfortunately, this effort was largely unsuccessful: response rates to these surveys were very low (around 2% of all course registrants, and less than 1% of students who had stopped out) and heavily biased toward students who had already persisted through weeks of voluntary challenges and were likely very satisfied with the course. It seems that the traditional approach to course evaluation – asking all students to evaluate a course at its end – is unlikely to work in a MOOC context.

One possible reason for the low response rate from students who stopped out is that such students quickly disengage after leaving the course, so that the likelihood of responding to a survey weeks or even months after they quit is small. Indeed, we found (see Fig. 1) that the probability of responding to (i.e., starting, but not necessarily completing) the post-course surveys decays rapidly as the time since stopout increases. It is possible that higher response rates could be achieved if students could be contacted, through some automatic mechanism, in a more timely fashion. This could potentially increase the amount of information that



**Figure 1: Mean probability ( $\pm$  std.err.) of responding to the post-course survey versus time-since-stopout, over 6 HarvardX MOOCs.**

HarvardX, and other MOOC providers, can glean from students who choose not to complete their courses.

In January-April 2015, we pursued this idea of a *dynamic survey mechanism* designed specifically to target students who recently “stopped out”. In particular, we developed an automatic *classifier* of whether a student  $s$  has “stopped out” of a course by time  $t$ . Our **definition of stopout** derives from the kinds of students we wish to survey: we say a student  $s$  has *stopped out* by time  $t$  if and only if:  $s$  does not subsequently earn a certificate *and*  $s$  takes no further action between time  $t$  and the course-end date when certificates are issued. The rationale is that students who *either* certify in a course *or* continue to participate in course activities (watch videos, post to discussion forums, etc.) can reasonably be assumed to be satisfied with the course; it is the *rest* of the students whom we would like to query. In addition to developing a stopout classifier, we developed a survey *controller* that decides, based on the classifier’s output, whether or not to query student  $s$  at time  $t$ ; the goal here is to maximize the rate of survey response while maintaining a low spam rate, i.e., the fraction of students who had not stopped out but were incorrectly classified as having done so (false alarms). In our paper we describe our approaches to developing the classifier and controller, as well as our first experiences in querying students and analyzing their feedback. To a modest extent, even just emailing students with “Returning to course?” in the subject line (see Sec. 6) constitutes a small “intervention”; the architecture we develop for deciding which students to contact may be useful for researchers developing automatic mechanisms for preventing student stopout.

**Contributions:** (1) Most prior work on stopout detection focuses on training detectors for a *single* MOOC, without examining generalization to *new* courses. For our purpose of conducting dynamic surveys and interventions, generalization to new MOOCs is critical. We thus focus our machine learning efforts on developing features that predict stopout over a wide variety of MOOCs and conduct analyses to measure cross-MOOC generalization accuracy. (2) While a variety of methods have been investigated for *detecting* stopout,

almost no prior research has explored how to *use* a stopout detector to survey students or conduct an intervention. We present a principled method, based on optimization via simulation, to choose a threshold on the classifier’s output so as to maximize a performance criterion. Finally, (3) we conduct one of the first MOOC “survey interventions” using an automatic stopout classifier (to our knowledge, the only other work is [7]) and report initial findings.

## 2. STRUCTURE OF HARVARDX MOOCs

Most HarvardX MOOCs (all those which are analyzed in this paper) are hosted on servers owned and managed by edX, which is a non-profit multi-university consortium located in Cambridge, Massachusetts. Student enrollment and event data are stored at edX and then transferred periodically (daily and weekly depending on the dataset) from edX to HarvardX. Hence, there is a “time gap” between when students generate events and when these event data are available at HarvardX.

Every HarvardX MOOC has a *start date*, i.e., the first day when participation in the MOOC (e.g., viewing a lecture, posting to the discussion forum) is possible. HarvardX MOOCs also have an *end date* when certificates are issued. At the end date, all students whose grade exceeds a minimum *certification threshold*  $G$  (which may differ for each course) receive a certificate. HarvardX courses allow students to register even after the course-end date, and they may view lectures and read the discussion forums; in most MOOCs these students cannot, however, earn a certificate. For the analyses in this paper we normalize the start date for each course to be 0 and denote the end date as  $T_e$ .

## 3. RELATED WORK

Over the past 3 years, since MOOCs have proliferated and the low proportion of students who complete them has become apparent, researchers from a variety of fields, including computer science, education, and economics, have begun developing quantitative models of when and why student stop out from MOOCs. The motivation for such work varies – some researchers are more interested in estimating the relative weight of different causes of stopout, whereas others (including ourselves) are primarily interested in developing automatic classifiers that could be used for real-time interventions. Work on stopout/dropout detection in MOOCs varies along several dimensions, described below:

**Definition of stopout/dropout:** Some researchers treat a student’s last “event” within a MOOC as the stopout/dropout date, where “event” could be submitting an assignment or quiz solution [14, 13], watching a video [13], posting to a discussion forum [17], or any event whatsoever [8, 1]. Others define stopout as not earning a certificate within a course [5, 2, 4]. Hybrid definitions, such as having watched fewer than 50% of the course’s videos and having executed no action during the last month [3], are also possible. Our own “stopout” definition (see Introduction) is a hybrid of lack of certification and last event.

**Features used for prediction:** The most commonly used features are derived from *clickstream data* [4, 1, 8, 2, 3, 14, 7] (e.g., when students play videos, post to discussion forums, submit answers to quiz problems), *grades* [4, 5, 3, 14, 7]

(e.g., average grade on quizzes), and *social network analysis* [17, 5] (e.g., eigenvector centrality of a node in a discussion forum graph). Biographical information (e.g., job, age) has also been used [5, 13, 17].

**Classification method:** Most existing work uses standard supervised learning methods such as support vector machines [8] and logistic regression [4, 5, 14, 7]; the latter has the advantage of probabilistic semantics and readily interpretable feature coefficients. Another approach is to use a generative model such as a Hidden Markov Model [1]; this could be useful for control-theoretic approaches to *preventing* stopout. Survival analysis techniques such as the Cox proportional hazards model have also been used [17, 13].

**Classification setting:** A critical issue is whether a stopout detector is highly tuned to an existing course that will never be offered again; whether it could generalize to a future offering of the same course; or whether it could generalize to other courses. Detectors that are tuned to perform optimally for only a single course are useful for exploring different classification architectures and features, but their utility for predicting stopout in new students is limited (since typically the entire course has ended before training even begins). Most existing work focuses on a single MOOC (which may or may not be offered again); to our knowledge, only [7, 3] explore stopout detection across multiple courses.

To our knowledge, the only prior work that explores how to use a stopout detector to conduct dynamic surveys is [7]. In contrast to their work, we take a more formal optimization approach to deciding how to use the classifier’s output to make intelligent survey decisions (see Sec. 5).

## 4. STOPOUT DETECTOR

The first step toward developing our dynamic survey system is to train a classifier of student stopout. In particular, we wish to estimate the probability that a student  $s$  has stopped out by time  $t$ , given the event history up to time  $t$ . We focus on *time invariant* classifiers, i.e., classifiers whose input/output relationship is the same for all  $t$ . (An alternative approach, which we discuss in Sec. 4.3, is to train a separate classifier for each week, as was done in [14].) In correspondence with the interventions that we conduct (see Sec. 6), we vary  $t$  over  $\mathcal{T} = \{10, 17, 24, \dots, T_e\}$  days; these days correspond to the timing of the survey interventions that we conduct. In our classification paradigm, if a student  $s$  stops out at time  $t = 16$ , then the label for  $s$  at  $t = 10$  would be negative (since he/she had not yet stopped out), and the labels for times 17, 24,  $\dots$ ,  $T_e$  would all be positive. Note that, since students may enroll at different times during the course (between 0 and  $T_e$ ), not all values of  $t$  are represented for all students.

For classification we use multinomial logistic regression (MLR) with an  $L_2$  ridge term ( $10^{-4}$ ) on every feature except the “bias” term (which has no regularization). Prior to classifier training, features are normalized to have mean 0 and variance 1; the same normalization parameters (mean, standard deviation) are also applied to the testing set. For each course, we assign each student to either the training (50%) or testing (50%) group based on a hash of his/her username; hence, students who belong to the testing set for one course

will belong to the testing set for *all* courses. For all experiments, we include all students who enrolled in the MOOC prior to the course-end date when certificates are issued.

As accuracy metric we use Area Under the Receiver Operating Characteristics Curve (AUC) statistic, which measures the probability that a classifier can discriminate correctly between two data points – one positive, and one negative – in a two-alternative forced-choice task [15]. An AUC of 1 indicates perfect discrimination whereas 0.5 corresponds to a classifier that guesses randomly. The AUC is *threshold independent* because it averages over all possible thresholds of the classifier’s output. For a *control* task in which we use the classifier to make decisions, we face an additional hurdle of how to select the threshold (see Sec. 5).

## 4.1 Features

Our focus is on finding features that are predictive of stopout for a wide variety of MOOCs, rather than creating specialized features (via intensive feature engineering [14]) that are tailored to a particular course. We extract these features from two tables generated by edX: the “tracking\_log” table (containing event data), and the “courseware student module” table (containing grades). The features we extract and the motivation for them are listed below:

1. The absolute time (in days, since course start)  $t$ , as well as the relative time through the course ( $t/T_e$ ) – it is possible that students who persist through most of the course are unlikely to stop out.
2. The elapsed time between the last recorded event and time  $t$  – recent activity is likely negatively correlated with stopping out.
3. The total number of events of different types that were triggered by the student up to time  $t$ , where event types includes forum posts, video plays, etc.
4. 1-D temporally-local band-pass (Gabor [9]) filters (6 frequencies, 3 bandwidths) of all event times before  $t$ . Temporal Gabor filters capture sinusoidal patterns (with frequency  $F = 2^f$ ,  $f \in \{-10, -9, \dots, -5\}$  days) in the *recent* history of events by attenuating with a Gaussian envelope (with bandwidth  $\sigma \in \{14, 28, 56\}$  days); see Fig. 5 for examples. Gabor filters have been used previously for automatic event detection (e.g., [16]), and it is possible that “regularity” in event logs is predictive of whether a student stops out.
5. The student’s grade at time  $t$  relative to the certification threshold ( $g_t/G$ ), as well as a binary feature encoding whether the student already has enough points to certify ( $\mathbb{I}[g_t \geq G]$ ). If the latter feature equals 1, then by definition the student has not stopped out.

See Appendix for more details. Including a “bias” feature (constant 1), this amounts to 37 features.

## 4.2 Experiments

We investigated the following questions:

Course ID	Year	Subject	# students	# certifiers	# events	# data	# + data
AT1x	2014	Anatomy	971	60	384747	7588	5895
CB22x	2013	Greek Heroes	34615	1407	11017890	671894	555581
CB22.1x	2013	Greek Heroes	17465	731	5195716	250205	201836
ER22x	2013	Justice	71513	5430	16256478	1209515	926067
GSE2x	2014	Education	37382	3936	13474171	209097	159639
HDS1544.1x	2013	Religion	22638	1546	6837110	144233	108848
PH525x	2014	Public Health	18812	652	5567125	124592	96836
SW12x	2013	Chinese History	18016	3068	7638660	78821	50431
SW12.2x	2014	Chinese History	9265	2137	3544666	25885	15741
USW30x	2014	History	14357	1089	2171359	107789	86043

**Table 1: MOOCs for which we trained stopout classifiers, along with # students who enrolled up till the course-end date, # students who earned a certificate, # events generated by students up till the course-end date, # data points (summed over all students and all times  $t$  when classification was performed) for training and testing, and # positively labeled data points (time-points after the student had stopped out).**

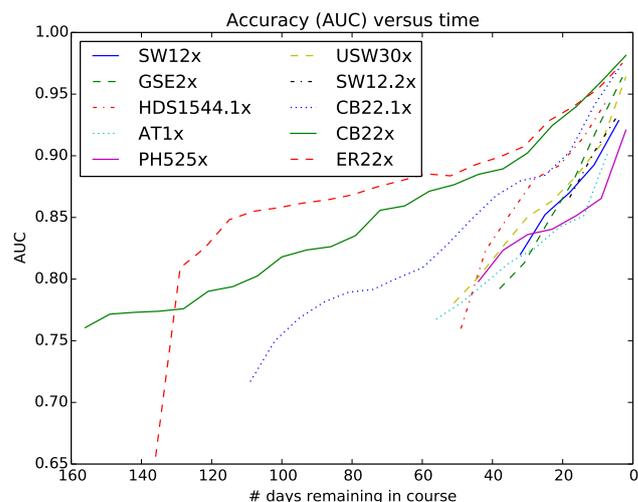
1. **Accuracy within-course:** How much variation in accuracy is there from course to course? How does this accuracy vary over  $t \in [0, T_e]$  within each course?
2. **Accuracy between-courses:** How well does a classifier trained on the largest course in Table 1 (ER22x) perform on the other courses?
3. **Training set size & over/under-fitting:** Does accuracy improve if more data are collected? Is there evidence of over/under-fitting?
4. **Feature selection:** Which features are most predictive of stopout? How much accuracy is gained by adding more features?
5. **Confidence:** Does the classifier become more confident as the time-since-stopout increases?

### 4.3 Accuracy within-course

For this experiment we trained a separate classifier for each of 10 HarvardX MOOCs (see Table 1) using only training data and then evaluated on testing data. Accuracy for each course as a function of time-to-course-end ( $T_e - t$ ) is shown in Fig. 2. In this graph we observe substantially lower accuracy during the beginning of each course (left side of the graph) than at the end, suggesting that longer event histories (larger  $t$ ) yield more accurate classifications. In addition, accuracy varies considerably from course to course, especially at the beginning of each course.

Table 2 (middle column) shows accuracy for each course aggregated over all  $t \in \mathcal{T}$ . Comparing classification architectures across different courses is approximate at best; however, we do observe a large performance gap between our numbers and the accuracy reported in [1] (AUC=0.71), who also use “last event” as their definition of stopout. One possible explanation is the lack of a “time since last event” feature (see Sec. 4.6) in their feature set. [8] use a similar definition of stopout but only report percent-correct, not AUC.

Based on Fig. 2, it is conceivable that students’ behavior (or the set of students) is qualitatively different during the first week of a course compared to later weeks, and that training a specialized classifier to predict stopout only during the first week might perform better than a classifier trained on



**Figure 2: Accuracy (area under the receiver operating characteristics curve (AUC)) of the various stopout classifiers as a function of time, expressed as number of days until the course-end date.**

all weeks’ data. We explored this hypothesis in a follow-up study (ER22x only) and found minor evidence to support it: train on week 1, test on week 1 gives an AUC of 0.69; train on all weeks, test on week 1 gives an AUC of 0.66.

### 4.4 Accuracy between-courses

Here, we consider only the classifier for course ER22x, containing the largest number of students and the most training data. We assessed how well the ER22x stopout classifier generalized to other courses compared to training a custom classifier for each course. We assess accuracy over all students and all  $t \in \mathcal{T}$  to obtain an overall AUC score for each course. Results are shown in Table 2. The middle column shows testing accuracy when training on each course, whereas the right column shows testing accuracy when trained on ER22x. Interestingly, though a small consistent performance gain can be eked by training a classifier for each MOOC, the gap is quite small, typically  $< 0.02$ . This suggests that the features described in Sec. 4.1 are quite

Course	Within-course	Cross-train (ER22x)
AT1x	0.850	0.832
CB22x	0.879	0.876
CB22.1x	0.868	0.866
ER22x	0.895	0.895
GSE2x	0.892	0.881
HDS1544.1x	0.897	0.887
PH525x	0.860	0.847
SW12x	0.890	0.880
SW12.2x	0.907	0.896
USW30x	0.884	0.875

**Table 2:** Accuracy (AUC, measured over all students in the test set and all times  $t$ ) of stopout classification for each course, along with accuracy when cross-training from course ER22x.

general; on the other hand, it also points to the possibility of underfitting (see Sec. 4.5).

#### 4.5 Training set size & over/under-fitting

We examined how testing accuracy (AUC) increases as the number of training data increases. For ER22x, we found that, even if the number of training students is drastically reduced to 1000 (down from around 36000), the testing accuracy is virtually identical at 0.894. Moreover, the *training* accuracy for a training set of 1000 students is only 0.91 (and slightly lower when using the full training set) and does not improve by reducing the ridge term. These numbers suggest that (a) the feature space may be too impoverished (under-fitted) to classify all data correctly; and/or (b) there is a large amount of inherent uncertainty in a student’s future action given only his/her event logs and grades.

#### 4.6 Feature selection

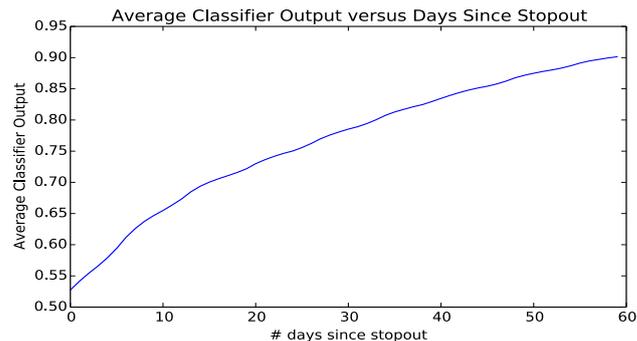
While some insight into feature salience can be gleaned by examining the regression coefficients, in practice it is difficult to interpret these coefficients because the  $L_2$  regularizer distributes weight across multiple correlated features. We thus used the following greedy feature selection procedure: Initialize a feature set  $\mathcal{F}$  to contain only the “bias” feature; find the feature (not already in  $\mathcal{F}$ ) that maximally increases the AUC on training data (for ER22x); add this feature to  $\mathcal{F}$  and record the associated AUC score; repeat  $N - 1$  times.

We executed this procedure for  $N = 5$  rounds and obtained the results in Table 3. The most predictive feature was time-since-last-action (which corroborates a similar result in [7]); using this feature alone (along with the “bias” feature), the AUC was already 0.867. The student’s normalized grade ( $g_t/G$ ) was the second most predictive feature; this is intuitive since our definition of stopout includes certification as one of the criteria. Next, time into the course ( $t$ ) was selected, suggesting there are certain times in the course when students are more likely to stop out. The fourth feature selected was a Gabor feature; rather than capturing periodicity in a student’s events, the high bandwidth ( $\sigma = 56$  days) and low frequency ( $F = 2^{-10}$  days) of the feature can more aptly be described as a weighted sum of event counts favoring the recent past more than the distant past (see Fig. 5).

**Top 5 Most Predictive Features**

#	Feature	Cumulative AUC (training)
1	Time since last event	0.867
2	Normalized grade ( $g_t/G$ )	0.880
3	Time into course ( $t$ )	0.886
4	Gabor ( $\sigma = 56, F = 2^{-10}$ days)	0.889
5	Total # events	0.890

**Table 3:** The top 5 most predictive features and associated cumulative AUC on *training* data, for ER22x. Feature  $i$  is chosen so as to maximize the training AUC given the previously selected features  $1, \dots, i - 1$ .



**Figure 3:** The average output of the ER22x stopout classifier, as a function of time-since-stopout, on students who had stopped out of the course.

In retrospect, it is clear that “time since last event” would be salient – the longer it has been since a student has done anything, the less likely he/she is to do anything in the future. It may be useful, in future stopout detection research, to compare with this single feature as a baseline.

#### 4.7 Confidence

When building a real-time system that uses the probability estimates given by a classifier to make decisions, it can be useful to “wait” before acting until the classifier becomes more confident (so as to avoid false alarms). For course ER22x, we found that the expected classifier output at time  $t$ , averaged over every student who stopped out at time  $t' < t$ , increases with time-since-stopout ( $t - t'$ ). The Pearson correlation of the classifier output  $y$  with  $t - t'$  was 0.73, and the Spearman rank correlation was even higher (0.93). A graph displaying the expected classifier output versus time-since-stopout is shown in Fig. 3.

### 5. CONTROLLER

Given a trained classifier of student stopout, how can we use it to decide which students to contact and when to contact them? At each week  $t$ , the classifier estimates for each student  $s$  the probability  $y_{st}$  that the student has stopped out. How high must  $y_{st}$  be in order to justify querying that student at that time? In this decision problem, we are faced with the following **trade-off**:

**Factor 1:** The sooner we contact a student after he/she has stopped out, the higher the probability that he/she will respond (see Fig. 1); this suggests using a lower threshold.

**Factor 2:** On the other hand, the longer we wait after he/she has stopped out, the more accurate our classifier becomes (see Fig. 3); this suggests using a higher threshold.

Depending on how the “response fall-off curve” (factor 1) and the “confidence increase” curve (factor 2) are shaped, it is possible that a more efficient (higher response rate, lower spam rate) system can be constructed if the threshold  $\theta$  on the classifier’s output is chosen carefully. Factor 2 was estimated in Sec. 4.7. Factor 1 can be roughly estimated using response rate data collected from the *post-course* surveys (see Introduction) and back-dating when students who responded to the survey had stopped out.

In collaboration with the HarvardX course creation teams, we also decided on additional constraints: (1) each student can be contacted during the course at most once (so as to avoid irking students with multiple email messages), and (2) the fraction of students whom we query but who had not actually stopped out (false alarms) should not exceed  $\alpha = 20\%$ . Note that this false alarm rate, which is computed over students’ entire trajectories through the MOOC, is different from the false alarm rate of *classification* described in Sec. 4, which is computed at multiple timepoints within each trajectory. Subject to these constraints, we wish to choose a threshold  $\theta$  (a scalar) on the classifier’s output  $y_{st}$  so as to *maximize* the rate of survey response from students who had stopped out. Our approach to tackling this problem is based on *optimization via simulation*.

**Optimization via simulation:** We built a simulator of how students generate events, what grades they earn, and when they stop out, based on historical data from prior HarvardX MOOCs. We can also simulate whether a student who stopped out at time  $t'$  responds to a survey given at time  $t$  using the “response fall-off curve” described above. Then, for any given value of  $\theta$ , we can estimate how many query responses and how many false alarms it generates by averaging over many runs (we chose  $N = 50000$ ) of the simulator: for each run, we randomly choose a student  $s$  from our training set, and at each time point  $t$  (every 7 days until  $T_e$ ), we extract a feature vector  $x_t$  based on  $s$ ’s event log and grade up to time  $t$ . We then classify  $x_t$  using a trained classifier (from Sec. 4) and threshold the result  $y_{st}$  using  $\theta$ . If  $y_{st} > \theta$  and if we had not previously queried  $s$  during the current simulation run, then we query the student. If the student had indeed stopped out before  $t$ , then we sample the student’s response (reply, not reply) from the response fall-off curve. During all simulation runs we maintain counts of both false alarms and hits (stopped-out student replies to query). Since  $\theta$  is a scalar, we can use simple grid-search to find  $\theta^*$  that maximizes the hit rate subject to a false alarm rate below  $\alpha$ . Note that more sophisticated controllers with multidimensional parameter vectors  $\theta$  are also possible (e.g., a different threshold for every week of the MOOC) using policy gradient optimization methods.

## 6. SURVEY INTERVENTION

Using the classifier and controller described above, we conducted a “dynamic survey intervention” on two live Har-

vardX courses: HLS2x (“ContractsX”) and PH525x (“Statistics and R for the Life Sciences”), which started on Jan. 8 and Jan. 19, 2015, respectively. The goals were to (1) collect feedback about why stopped-out students left the course and (2) explore how sending a simple survey solicitation email affects students’ behavior.

We trained separate stopout classifiers, using previous HarvardX courses for which stopout data were already available, for HLS2x and PH525x. For PH525x, there was a 2014 version of the course on which we could train. For HLS2x, we trained on a 2014 course (“AT1x”) whose lecture structure (e.g., the frequency with which lecture videos were posted) was similar. Then, using each trained classifier and the response fall-off curve estimated from post-course survey data (see Sec. 5), we optimized the classifier threshold  $\theta$  for each MOOC ( $\theta = 0.79$  for HLS2x,  $\theta = 0.75$  for PH525x).

We emailed students in batches once per week. Each week, we ran the stopout classifier on all students who had registered and were active in the course (i.e., had not de-registered). Each student was assigned a condition (50% experimental, 50% control) based on a hash of his/her username. To every student  $s$  in the experimental group whose  $y_{st}$  at time  $t$  exceeded  $\theta$ , we sent an email (see Fig. 4) asking whether he/she intended to complete the course and why/why not. After clicking on a link, the user is given the opportunity to enter free-response feedback in a textbox. We used Qualtrics to manage the surveys, send the emails, and track the results. Students in the control group were not emailed; instead, we used them to measure the accuracy of our stopout classifier and to compare the “comeback rates” across conditions.

We delivered 3 batches (Jan. 21, Jan. 26, Feb. 2) of survey emails to 5073 students in HLS2x and 1 batch (Feb. 2) to 3764 students in PH525x. These dates were chosen to occur shortly after the data transfers from edX to HarvardX (see Sec. 2). Except in Sec. 6.2, we exclude students (138 (2.7%) from HLS2x, 201 (5.4%) from PH525x) from our analyses whom we *would not have emailed* if we had had real-time access to students’ event data. Hence, the results below estimate the response rates, accuracy, and comeback rates if we could run our intervention directly on edX’s servers (with 0 time-gap).

### 6.1 Response rate from stopped-out students

We investigated whether the dynamic survey intervention induced more stopped-out students to respond compared to the conventional post-course survey mechanism. Because the HarvardX post-course surveys are much longer than our stop-out survey, we compared the rates with which stopped-out students *started* the surveys (without necessarily completing them) to enable a fairer comparison. We analyzed response rates for HLS2x only (PH525x is still ongoing).

To measure response rates, we computed the number of students  $D$  whom we emailed *and* who had actually stopped out (which we now know since the course has ended) before the email was sent. Then, of these  $D$  students, we compute the number  $N$  of students who responded to (started, but not necessarily completed) the survey, and then calculated the response rate  $N/D$ . Since the last intervention for HLS2x was on Feb. 2, which was 32 days before the course-end date

Dear Jake,

We hope you have enjoyed the opportunity to explore ContractsX. It has been a while since you logged into the course, so we are eager to learn about your experience. Would you please take this short survey, so we can improve the course for future students? Each of the links below connects to a short survey. Please click on the link that best describes you.

- I plan on continuing with the course
- I am not continuing the course because it was not what I expected when I signed up.
- I am not continuing the course because the course takes too much time.
- I am not continuing the course because I am not happy with the quality of the course.
- I am not continuing the course because I have learned all that I wanted to learn.
- I am not continuing the course now, but I may at a future time.

Your feedback is very important to us. Thank you for registering for ContractsX.

Figure 4: A sample email delivered as part of our dynamic survey intervention for HLS2x.

(Mar. 6), we also calculated the corresponding fraction of students in previous HarvardX courses who responded to the post-course surveys who had stopped out at least 32 days before the course-end date (c.f. Fig. 1).

**Result:** The response rate from stopped-out students for the dynamic survey intervention was 3.7% compared to 1.0% for the post-course survey mechanism; the difference was statistically significant ( $\chi^2(1) = 183, p < 10^{-15}$ , 2-tailed). In other words, the dynamic survey mechanism achieved over 3x higher response rate.

## 6.2 Survey responses

For this analysis we included *all* students whom we emailed (even those whom we would not have emailed with real-time data; see above). From HLS2x, 336 students (6.6%) responded to (i.e., started but not necessarily finished) the survey. From PH525x, 353 students (9.4%) responded to the survey. Note that, in contrast to [7], who reported a 12.5% response rate for a computer science MOOC, we did not condition on students having watched at least one video.

Of students who started the survey *and* answered whether or not they planned to continue (329 for HLS2x, 328 for PH525x), most replied that they planned to continue the course (242 for HLS2x, 203 for PH525x). Of those who replied they did *not* wish to continue (87 for HLS2x, 125 for PH525x), the reasons are broken down as follows:

Reason	Freq.
“It was not what I expected when I signed up”	8.4%
“The course takes too much time”	5.0%
“I am not happy with the quality of the course”	0.5%
“I have learned all that I wanted to learn”	5.5%
“I may at a future time”	80.7%

In other words, many respondents who confirmed they had stopped out indicated that they also might resume the course in the future. Notably, very few respondents reported that the courses were of poor quality. However, we emphasize that the full population of registrants who stop out could potentially be very different from the sample who responded to the survey; hence, the numbers above should be interpreted with caution. Our stopout detector may disproportionately identify students who stop out because they are too busy, or students who stop out because they are too busy may disproportionately respond to our survey and students unhappy with the course may choose not to respond.

## 6.3 Accuracy

As a further assessment of the stopout detector described in Sec. 4, we computed the accuracy of the classifier on students in the control group of our HLS2x intervention.

**Results:** The accuracy (AUC) for HLS2x was 0.74 for week 1, 0.78 for week 2, and 0.80 for week 3. These numbers are consistent with the results in Sec. 4.3.

## 6.4 Effect on student “comeback”

One survey respondent wrote: “I was not allocating time for edX, but receiving your survey e-mail recaptured my attention.” This raises the question of whether the mere act of notifying students that we believed they had lost interest might cause them to “come back”. To test this hypothesis, we compared the fraction of students in the experimental group who “came back” – i.e., took at least one action (other than de-registering and/or responding to the survey) in the course after we sent the emails – to the corresponding fraction of students in the control group. We assessed comeback rates at two different timepoints – Feb. 12 (before we submitted the paper for review) and Apr. 20 (before we submitted the paper for final publication) – using all event data available by those dates.

**Results:** For all 4 interventions (3 weeks of HLS2x, and 1 week of PH525x), the comeback rates were higher at both timepoints for the experimental group (who received an email) than for the control group (who did not receive an email). Aggregated over all weeks of both courses, the comeback rate by Feb. 12 was 12.4% for the experimental group versus 11.2% for the control group; the difference was statistically significant ( $\chi^2(1) = 5.63, p = 0.018$ , 2-tailed). By Apr. 20, however, the difference was smaller – 22.1% for the experimental group versus 21.4% for the control group – and not statistically significant ( $\chi^2(1) = 1.25, p = 0.26$ , 2-tailed).

Together, these results suggest that the intervention induced students to come back *sooner* into the course, even if the overall comeback rates are similar. To confirm this hypothesis, we compared the mean “comeback time” (time between last action before intervention, and first action after intervention, among students who came back) between the two groups and across all 4 interventions. We found that students in the experimental group came back significantly sooner: 51.68 days for the experimental group versus 55.02 days for the control group (Mann-Whitney  $U = 1458393, n_1 = 1725, n_2 = 1831, p < 10^{-4}$ , 2-tailed). These

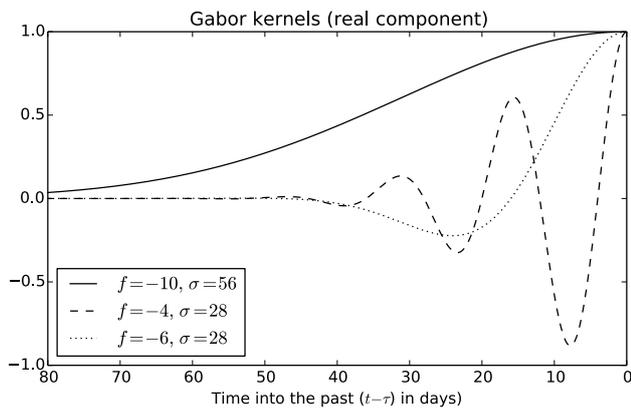


Figure 5: Sample Gabor kernels.

results provide evidence that an “intervention” consisting of an email indicating that a student has been flagged as having potentially stopped out, can affect students’ behavior.

## 7. CONCLUSIONS

We developed an automatic classifier of MOOC student “stop-out” and showed that it generalizes to new MOOCs with high accuracy. We also presented a novel end-to-end architecture for conducting a “dynamic survey intervention” on MOOC students who recently stopped out to ask them why they quit. Compared to post-course surveys, the dynamic survey mechanism attained a significantly higher response rate. Moreover, the mere act of asking students why they had left the course induced students to “come back” into the course more quickly. Preliminary analysis of the surveys suggest students quit due to exogenous factors (not enough time) rather than poor quality of the MOOCs.

**Limitations:** The subset of stopped-out students who responded to the survey may not be a representative sample; thus, results in Sec. 6.2 should be interpreted with caution.

**Future work:** In future work we will explore whether more sophisticated, time-variant classifiers such as recurrent neural networks can yield better performance. With more accurate classifiers we can conduct more efficient surveys and more effective interventions to reduce stopout.

## APPENDIX

**Event count features:** We counted events of the following types (using the “event\_type” field in the edX “tracking\_log” table): “showanswer”, “seek\_video”, “play\_video”, “pause\_video”, “stop\_video”, “show\_transcript”, “page\_close”, “problem\_save”, “problem\_check”, and “problem\_show”. We also measured activity in discussion forums by counting events whose “event\_type” field contained “threads” or “forum”.

**Gabor features:** A Gabor filter kernel (see Fig. 5) is the product of a Gaussian envelope and a complex sinusoid. At time  $t-\tau$  (i.e.,  $\tau$  days before  $t$ ), the real and imaginary components are given by  $K_r(\tau) = \exp(-\pi\tau^2/(2\sigma^2)) \cos(2\pi F\tau)$  and  $K_i(\tau) = \exp(-\pi\tau^2/(2\sigma^2)) \sin(2\pi F\tau)$  (respectively), where  $\sigma$  is the bandwidth of the Gaussian envelope and  $F$  is the frequency of the sinusoid. When extracting Gabor features at

time  $t$ , we convolve this complex kernel with a  $t$ -dimensional “history vector”  $h$  whose  $\tau$ th component contains the total number of events generated by that student on day  $t-\tau$ . We then compute the magnitude of the complex filter response, i.e.,  $|\sum_{\tau=1}^t (K_r(\tau)h_\tau + jK_i(\tau)h_\tau)|$ , where  $j = \sqrt{-1}$ .

## 8. REFERENCES

- [1] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. Technical report, UC Berkeley, 2013.
- [2] C. Coleman, D. Seaton, and I. Chuang. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Learning at Scale*, 2015.
- [3] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in MOOCs using learner activity features. In *European MOOC Summit*, 2014.
- [4] J. He, J. Bailey, Benjamin, I. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *AAAI*, 2015.
- [5] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O’Dowd. Predicting MOOC performance with week 1 behavior. In *Educational Data Mining*, 2014.
- [6] H. Khalil and M. Ebner. MOOCs completion rates and possible methods to improve retention - a literature review. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2014.
- [7] R. Kizilcec and S. Halawa. Attrition and achievement gaps in online learning. In *Learning at Scale*, 2015.
- [8] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting MOOC dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65, 2014.
- [9] J. Movellan. Tutorial on Gabor filters. Technical report, UCSD Machine Perception Laboratory, 2002.
- [10] D. Onah, J. Sinclair, and R. Boyatt. Dropout rates of massive open online courses: behavioural patterns. In *Conf. on Education and New Learning Tech.*, 2014.
- [11] J. Reich. MOOC completion and retention in the context of student intent. *EDUCAUSE Review*, 2014.
- [12] R. Rivard. Measuring the MOOC dropout rate. *Insider Higher Ed*, 2013.
- [13] R. Stein and G. Allione. Mass attrition: An analysis of drop out from a principles of microeconomics MOOC. *PIER Working Paper*, 14(031), 2014.
- [14] C. Taylor, K. Veeramachaneni, and U.-M. O’Reilly. Likely to stop? Predicting stopout in massive open online courses. *arXiv*, 2014. <http://arxiv.org/abs/1408.3382>.
- [15] C. Tyler and C.-C. Chen. Signal detection theory in the 2AFC paradigm: attention, channel uncertainty and probability summation. *Vision Research*, 40(22):3121–3144.
- [16] E. Vural, M. Bartlett, G. Littlewort, M. Cetin, A. Ercil, and J. Movellan. Discrimination of moderate and acute drowsiness based on spontaneous facial expressions. In *ICPR*, 2010.
- [17] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. “Turn on, tune in, drop out”: Anticipating student dropouts in massive open online courses. In *NIPS Workshop on Data-Driven Education*, 2014.

# From Predictive Models to Instructional Policies

Joseph Rollinson  
Computer Science Department  
Carnegie Mellon University  
jrollinson@gmail.com

Emma Brunskill  
Computer Science Department  
Carnegie Mellon University  
ebrun@cs.cmu.edu

## ABSTRACT

At their core, Intelligent Tutoring Systems consist of a student model and a policy. The student model captures the state of the student and the policy uses the student model to individualize instruction. Policies require different properties from the student model. For example, a mastery threshold policy requires the student model to have a way to quantify whether the student has mastered a skill. A large amount of work has been done on building student models that can predict student performance on the next question. In this paper, we leverage this prior work with a new when-to-stop policy that is compatible with any such predictive student model. Our results suggest that, when employed as part of our new predictive similarity policy, student models with similar predictive accuracies can suggest that substantially different amounts of practice are necessary. This suggests that predictive accuracy may not be a sufficient metric by itself when choosing which student model to use in intelligent tutoring systems.

## 1. INTRODUCTION

Intelligent tutoring systems offer the promise of highly effective, personalized, scalable education. Within the ITS research community, there has been substantial work on constructing student models that can accurately predict student performance (e.g. [6, 3, 15, 5, 10, 9, 14, 7]). Another key issue is how to improve student performance through the use of instructional policy design. There has been significant interest in cognitive models used for within activity design (often referred to as the inner-loop) and even authoring tools developed to make designing effective activities easier (e.g. CTAT [1]). However, there has been much less attention to outer-loop (what problem to select or when to stop) instructional policies (though exceptions include [5, 12, 17]).

In this paper we focus on a common outer-loop ITS challenge, adaptively deciding when to stop teaching a certain skill to a student given correct/incorrect responses. Somewhat surprisingly, there are no standard policy rules or al-

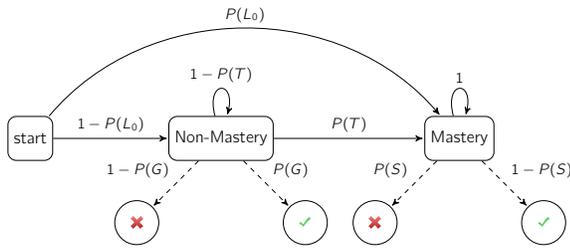
gorithms for deciding when to stop teaching for many of the student models introduced over the last decade. Bayesian Knowledge Tracing [6] naturally lends itself to mastery teaching, since one can halt when the student has mastered a skill with probability above a certain threshold. Such a mastery threshold has been used as part of widely used tutoring systems, but typically in conjunction with additional rules since a student may never reach a sufficient mastery threshold given the available activities.

We seek to be able to directly use a wide range of student models to create instructional policies that halt both when a student has learned a skill and when the student seems unlikely to make any further progress given the available tutoring activities. To do so we introduce an instructional policy rule based on change in predicted student performance.

Our specific contributions are as follows:

- We provide a functional interface to student models that captures their predictive powers without knowledge of their internal mechanics (Section 3).
- We introduce the *predictive similarity policy*: a new when-to-stop policy that can take as input any predictive student-model (Section 4) and can halt both if students have successfully acquired a skill or do not seem able to do so given the available activities.
- We analyze the performance of this policy compared to a mastery threshold policy on the KDD dataset and find our policy tends to suggest similar or a smaller number of problems than a mastery threshold policy (Section 5).
- We also show that our new policy can be used to analyze a range of student models with similar predictive performance (on the KDD dataset) and find that they can sometimes suggest very different numbers of instructional problems. (Section 5).

Our results suggest that predictive accuracy alone can mask some of the substantial differences among student models. Policies based on models with similar predictive accuracy can make widely different decisions. One direction for future work is to measure which models produce the best learning policies. This will require new experiments and datasets.



**Figure 1: BKT as a Markov process. *Mastery* and *Non-Mastery* are hidden states. Arrow values represent the probability of the transition or observation.**

## 2. BACKGROUND: STUDENT MODELS

Student models are responsible for modeling the learning process of students. The majority of student models are *predictive models* that provide probabilistic predictions of whether a student will get a subsequent item correct. In this section we describe two popular predictive student models, *Bayesian knowledge tracing* and *latent factor models*. Note that other predictive models, such as Predictive State Representations (PSRs), can also be used to calculate the probability of a correct response [7].

### 2.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [6] tracks the state of the student’s knowledge as they respond to practice questions. BKT treats students as being in one of two possible hidden states: *Mastery* and *Non-Mastery*. It is assumed that a student never forgets what they have mastered and if not yet mastered, a new question always has a fixed static probability of helping the student master the skill. These assumptions mean that BKT requires only four trained parameters:

- $P(L_0)$  Initial probability of mastery.
- $P(T)$  Probability of *transitioning* to mastery over a single learning opportunity.
- $P(G)$  Probability of *guessing* the correct answer when the student is not in the mastered state.
- $P(S)$  Probability of *slipping* (making a mistake) when the student is in the mastered state.

After every response, the probability of mastery,  $P(L_t)$ , is updated with Bayesian inference. The probability that a student responds correctly is

$$P_{\text{BKT}}(C_t) = (1 - P(S))P(L_t) + P(G)(1 - P(L_t)). \quad (1)$$

Prior work suggests that students can get stuck on a particular activity. Unfortunately, BKT as described above assumes that students will inevitably master a skill if given enough questions. As this is not always the case, in industry BKT is often used together with additional rules to make instructional decisions.

### 2.2 Latent Factor Models

Unlike BKT models, Latent Factor Models (LFM) do not directly model learning as a process [3]. Instead, LFMs assume that there are latent parameters of both the student and skill that can be used to predict student performance. These parameters are learned from a dataset of students answering

questions on multiple skills. The probability that the student responds correctly to the next question is calculated by applying the sigmoid function to the linear combination of parameters  $p$  and features  $f$ .

$$P_{\text{LFM}}(C) = \frac{1}{1 + e^{-f \cdot p}} \quad (2)$$

Additive Factor Models (AFM) [3] are based on the assumption that student performance increases with more questions. A student is represented by an aptitude parameter ( $\alpha_i$ ) and a skill is represented by a difficulty parameter ( $\beta_k$ ) and learning rate ( $\gamma_k$ ). AFM is sensitive to the number of questions the student has seen, but ignores the correctness of student responses. The probability that student  $i$  will respond correctly after  $n$  responses on skill  $k$  is

$$P_{\text{AFM}}(C) = \frac{1}{1 + e^{-(\alpha_i + \beta_k + \gamma_k n)}}. \quad (3)$$

Performance Factor Models (PFM) [15] are an extension of AFMs that are sensitive to the correctness of student responses. PFMs separate the skill learning rate into success and failure parameters,  $\mu_k$  and  $\rho_k$  respectively. The probability that student  $i$  will respond correctly after  $s$  correct responses and  $f$  incorrect responses on skill  $k$  is

$$P_{\text{PFM}}(C) = \frac{1}{1 + e^{-(\alpha_i + \beta_k + \mu_k s + \rho_k f)}}. \quad (4)$$

LFMs can easily be extended to capture other features. For example, the instructional factors model [5] extends PFMs with a parameter for the number of tells (interactions that do not generate observations) given to the student. To our knowledge there is almost no work on using LFMs to capture temporal information about the order of observations. Unlike BKT, LFMs are not frequently used in instructional policies.

Though structurally different, BKT models, AFMs and PFMs tend to have similar predictive accuracy [9, 15]. This raises the interesting question of whether instructional policies that use these models are similar.

## 3. WHEN-TO-STOP POLICIES

We assume a simple intelligent tutoring system that teaches students one skill at a time. All questions are treated the same, so the system only has to decide when to stop providing the student questions. In this section, we provide a general framework for the when-to-stop problem. In particular, we describe an interface that abstracts out the student model from instructional policies, which we will use to define the MASTERY THRESHOLD policy and use in the next section as the foundations of a model-agnostic instructional policy.

### 3.1 Accessing Models

Policies require a mechanism for getting values from student models to make decisions. We describe this mechanism as a state type and a set of functions. A student model consists of two types of values: immutable parameters that are learned on training data and mutable state that changes over time. For example, the parameters for BKT are  $(P(L_0), P(T), P(G), P(S))$  and the model state is the probability of mastery  $(P(L_t))$ . Policies treat the state

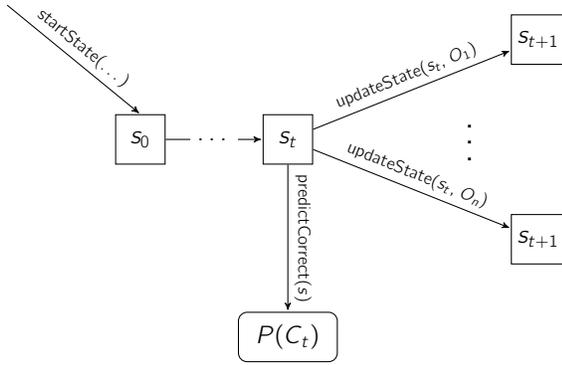


Figure 2: Model process with functional interface

as a black box, which they pass to functions. All predictive student models must provide the following functions. **startState**(...) returns the model state given that the student has not seen any questions. **updateState**(state, obs) returns an updated state given the observation. For this paper, observations are whether the student got the last question correct or incorrect. Finally, predictive student models must provide **predictCorrect**(state), which returns the probability that the student will get the next question correct. The function interfaces for BKT models and PFM are provided in table 1. Under this abstraction, when-to-stop policies are functions **stop**(state) that output true if the system should stop providing questions for the current skill and false if the system should continue providing the student with questions.

### 3.2 Mastery Threshold Policy

The MASTERY THRESHOLD POLICY halts when the student model is confident that the student has mastered the skill. This implies that we want to halt when the student masters the skill. Note that if the estimate of student mastery is based solely on a BKT<sup>1</sup> then a mastery threshold policy implicitly assumes that every student will master the skill given enough problems. Mathematically, we want to stop at time  $t$  if  $P(L_t) > \Delta$ , where  $\Delta$  is our mastery threshold. The MASTERY THRESHOLD policy function can be written as:

$$\text{stop}_M(\text{state}) = \text{predictMastery}(\text{state}) > \Delta. \quad (5)$$

The MASTERY THRESHOLD policy can only be used with models that include **predictMastery**(state) in their function set. BKT models are compatible, but LFM are not. By itself, the MASTERY THRESHOLD does not stop if the student has no chance of attaining mastery in the skill with the given activities. Students on poorly designed skills could be stuck learning a skill indefinitely.

## 4. FROM PREDICTION TO POLICY

In educational data mining, a large emphasis is put on building models that can accurately predict student observations. Our goal is to build a new when-to-stop policy that will work with any predictive student model.

<sup>1</sup>In practice, industry systems that use mastery thresholds and BKTs often use additional rules as well.

Our new instructional policy is based on a set of assumptions. First, students working on a skill will eventually end in one of two hidden end-states. Either, they will master the skill, or they will be unable to master the skill given the activities available. Second, once students enter either end-state, the probability that they respond correctly to a question stays the same. Third, if the probability that a student will respond correctly is not changing, then the student is in an end-state. Finally, we should stop if the student is in an end-state.

From these assumptions it follows that if the probability that the student will respond correctly to the next question is not changing, then we should stop. In other words, we should stop if it is highly likely that showing the student another question will not change the probability that the student will get the next question correct by a significant amount. We propose to stop if

$$(P(|P(C_t) - P(C_{t+1})| < \epsilon)) > \delta \quad (6)$$

where  $P(C_t)$  is the probability that the student will get the next question right. This can be thought of as a threshold on the sum of the probabilities of each observation that will lead to an insignificant change in the probability that a student will get the next question correct, which can be written as

$$\sum_{o \in \mathcal{O}} P(O_t = o) \mathbb{1}(|P(C_t) - P(C_{t+1}|O_t = o)| < \epsilon) > \delta \quad (7)$$

where  $P(C_{t+1}|O_t = o)$  is the probability that the student will respond correctly after observation  $o$ ,  $O_t$  is the observation at time  $t$ , and  $\mathbb{1}$  is an indicator variable. In our case  $\mathcal{O} = \{C, -C\}$ . This expression is true in the following cases:

1.  $P(C_t) > \delta$  and  $|P(C_t) - P(C_{t+1}|C_t)| < \epsilon$
2.  $P(-C_t) > \delta$  and  $|P(C_t) - P(C_{t+1}|-C_t)| < \epsilon$
3.  $|P(C_t) - P(C_{t+1}|C_t)| < \epsilon$  and  $|P(C_t) - P(C_{t+1}|-C_t)| < \epsilon$

First, if a student is highly likely to respond correctly to the next question and the change in prediction is small if the student responds correctly, then we should stop. Second, if a student is highly unlikely to respond correctly to the next question and the change in prediction is small if the student responds incorrectly, then we should stop. Third, if the change in prediction is small no matter how the student responds, then we should stop. All terms in these expressions can be calculated from the predictive student model interface as shown in equations 8 and 9. We call the instructional policy that stops according to these three cases the PREDICTIVE SIMILARITY policy. The function for the PREDICTIVE SIMILARITY policy is provided in algorithm 1

$$P(C_t) = \text{predictCorrect}(s) \quad (8)$$

$$P(C_{t+1}|O_t) = \text{predictCorrect}(\text{updateState}(s, O_t)) \quad (9)$$

## 5. EXPERIMENTS & RESULTS

We now compare the PREDICTIVE SIMILARITY policy to the MASTERY THRESHOLD policy and see if using different student models as input to the predictive SIMILARITY POLICY yields quantitatively different policies.

**Table 1: Functional interfaces for BKT and PFM**

	BKT	PFM
<b>startState</b> (...)	$P(L_0)$	$(\alpha_i + \beta_k, \mu_k, \rho_k, 0, 0)$
<b>updateState</b> ( $s, o$ )	$P(L_{t+1} P(L_t), O_{t+1} = o)$	$\begin{cases} (w, \mu, \rho, s + 1, f) & \text{if } o = C \\ (w, \mu, \rho, s, f + 1) & \text{if } o = \neg C \end{cases}$
<b>predictCorrect</b> ( $s$ )	$P(\neg S)P(L_t) + P(G)(1 - P(L_t))$	$(1 + e^{-(w+s\mu+f\rho)})^{-1}$
<b>predictMastery</b> ( $s$ )	$P(L_t)$	—

**Algorithm 1** PREDICTIVE SIMILARITY policy stop function

```

1: function STOP(state)
2:    $P(C_t) \leftarrow \text{predictCorrect}(\text{state})$ 
3:   total  $\leftarrow 0$ 
4:   if  $P(C_t) > 0$  then
5:     state'  $\leftarrow \text{updateState}(\text{state}, \text{correct})$ 
6:      $P(C_{t+1}|C_t) \leftarrow \text{predictCorrect}(\text{state}')$ 
7:     if  $|P(C_t) - P(C_{t+1}|C_t)| < \epsilon$  then
8:       total  $\leftarrow \text{total} + P(C_t)$ 
9:   if  $P(C_t) < 1$  then
10:    state'  $\leftarrow \text{updateState}(\text{state}, \text{incorrect})$ 
11:     $P(C_{t+1}|\neg C_t) \leftarrow \text{predictCorrect}(\text{state}')$ 
12:    if  $|P(C_t) - P(C_{t+1}|\neg C_t)| < \epsilon$  then
13:      total  $\leftarrow \text{total} + (1 - P(C_t))$ 
14:   return total  $> \delta$ 

```

## 5.1 ExpOps

In order to better understand the differences between two instructional policies we will measure the expected number of problems to be given to students by a policy using the ExpOps algorithm. The ExpOps algorithm allows us to summarize an instructional policy into a single number by approximately calculating the expected number of questions an instructional policy would provide to a student. A naive algorithm takes in the state of the student model and returns 0 if the instructional policy stops at the current state or recursively calls itself with an updated state given each possible observation as shown in equation 10. It builds a synthetic tree of possible observations and their probability using the model state. The tree grows until the policy decides to stop teaching the student. This approach does not require any student data nor does it generate any observation sequences. However, this algorithm may never stop, so ExpOps approximates it by also stopping if we reach a maximum length or if the probability of the sequence of observations thus far drops below a path threshold as shown in algorithm 2. In this paper, we use a path threshold of  $10^{-7}$  and a maximum length of 100.

$$E[Ops] = \begin{cases} 0 & \text{if } \text{stop}(s) \\ 1 + \sum_{o \in O} P(O_t = o)E[Ops|o] & \text{otherwise} \end{cases} \quad (10)$$

Lee and Brunskill first introduced this metric to show that individualized models lead to significantly different policies than the general models [12].

## 5.2 Data

**Algorithm 2** Expected Number of Learning Opportunities

```

1: function EXPOPS(startState)
2:   function EXPOPS'(state, P(path), len)
3:     if  $P(\text{path}) < \text{pathThreshold}$  then
4:       return 0
5:     if  $\text{len} \geq \text{maxLen}$  then
6:       return 0
7:     if stop(state) then
8:       return 0
9:      $P(C) \leftarrow \text{predictCorrect}(\text{state})$ 
10:     $P(W) \leftarrow 1 - P(C)$ 
11:    expOpsSoFar  $\leftarrow 0$ 
12:    if  $P(C) > 0$  then
13:       $P(\text{path} + c) \leftarrow P(\text{path}) * P(C)$ 
14:      state'  $\leftarrow \text{updateState}(\text{state}, C)$ 
15:      ops  $\leftarrow \text{EXPOPS}'(\text{state}', P(\text{path} + c), \text{len} + 1)$ 
16:      expOpsSoFar  $\leftarrow \text{expOpsSoFar} + (\text{ops} * P(C))$ 
17:    if  $P(W) > 0$  then
18:       $P(\text{path} + w) \leftarrow P(\text{path}) * P(W)$ 
19:      state'  $\leftarrow \text{updateState}(\text{state}, \text{incorrect})$ 
20:      ops  $\leftarrow \text{EXPOPS}'(\text{state}', P(\text{path} + w), \text{len} + 1)$ 
21:      expOpsSoFar  $\leftarrow \text{expOpsSoFar} + (\text{ops} * P(W))$ 
22:    return 1 + expOpsSoFar
23:   return EXPOPS'((startState, 1, 0))

```

For our experiments we used the Algebra I 2008–2009 dataset from the KDD Cup 2010 Educational Data Mining Challenge [18]. This dataset was collected from students learning algebra I using Carnegie Learning Inc.’s intelligent tutoring systems. The dataset consists of 8,918,054 rows where each row corresponds to a single step inside a problem. These steps are tagged according to three different knowledge component models. For this paper, we used the SubSkills knowledge component model. We removed all rows with missing data. We combined the rows into observation sequences per student and per skill. Steps attached to multiple skills were added to the observation sequences of all attached skills. We removed all skills that had less than 50 observation sequences. Our final dataset included 3292 students, 505 skills, and 421,991 observation sequences.

We performed 5-fold cross-validation on the datasets to see how well AFM, PFM, and BKT models predict student performance. We randomly separated the dataset into five folds with an equal number of observation sequences per skill in each fold. We trained AFM, PFM, and BKT models on four of the five folds and then predicted student performance on

**Table 2: Root Mean Squared Error on 5 Folds**

Fold	BKT	PFM	AFM
0	0.353	0.364	0.368
1	0.359	0.367	0.371
2	0.358	0.368	0.371
3	0.366	0.369	0.374
4	0.353	0.365	0.368

the leftover fold. We calculated the root mean squared error found in Table 2. Our results show that the three models had similar predictive accuracy, agreeing with prior work.

### 5.3 Model Implementation

We implemented BKT models as hidden Markov models using a python package we developed. We used the Baum-Welch algorithm to train the models, stopping when the change in log-likelihood between iterations fell below  $10^{-5}$ . For each skill, 10 models with random starting parameters were trained, and the one with the highest likelihood was picked. Both AFM and PFM were implemented using scikit-learn’s logistic regression classifier [16]. We used L1 normalization and included a fit intercept. The tolerance was  $10^{-4}$ . We treated an observation connected to multiple skills as multiple observations, one per skill. It is also popular to treat them as a single observation with multiple skill parameters. In the interest of reproducibility, we have published the models used as a python package.<sup>2</sup>

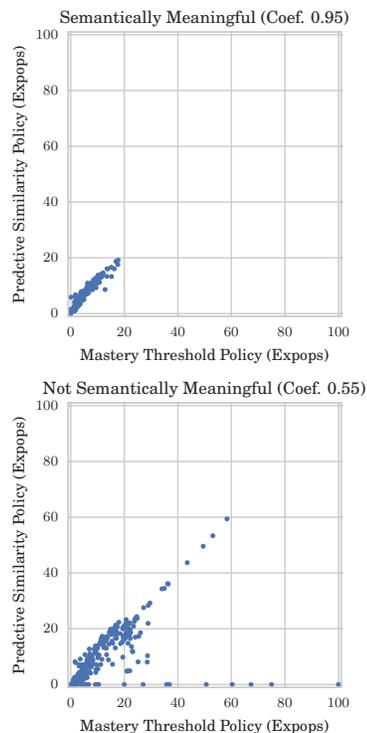
### 5.4 Experiment 1: Comparing policies

The MASTERY THRESHOLD policy is frequently used as a key part of deciding when to stop showing students questions. However without additional rules, it does not stop if students cannot learn the skill from the current activities. In this experiment we compare the PREDICTIVE SIMILARITY policy to the MASTERY THRESHOLD policy to see if the PREDICTIVE SIMILARITY policy acts like the MASTERY THRESHOLD policy when students learn and stops sooner when students are unable to learn with the given tutoring. We based both policies on BKT models.

We ran ExpOps on each skill for both policies. For the MASTERY THRESHOLD policy, we used the community standard threshold of  $\Delta = 0.95$ . For the PREDICTIVE SIMILARITY policy, we decided that the smallest meaningful change in predictions is 0.01 and that our confidence should be 0.95, so we set  $\epsilon = 0.01$  and  $\delta = 0.95$ . We then split the skills into those where the BKT model trained on them had semantically meaningful parameters and the rest. A BKT model was said to have semantically meaningful parameters if  $P(G) \leq 0.5$  and  $P(S) \leq 0.5$ . 218 skills had semantically meaningful parameters and 283 did not.<sup>3</sup>

<sup>2</sup>The packages are available at <http://www.jrollinson.com/research/2015/edm/from-predictive-models-to-instructional-policies.html>.

<sup>3</sup>We found similar results for both experiments using BKT models trained through brute force iteration on semantically meaningful values. These results can be found at <http://www.jrollinson.com/research/2015/edm/from-predictive-models-to-instructional-policies.html>



**Figure 3: ExpOps using the mastery threshold policy and the predictive similarity policy on skills with and without semantically meaningful parameters.**

The Pearson correlation coefficient between ExpOps values calculated using the two policies on skills with semantically meaningful parameters was 0.95. This suggests that the two policies make very similar decisions when based on BKT models with semantically meaningful parameters. However, the Pearson correlation coefficient between ExpOps values calculated using the two policies on skills that do not have semantically meaningful parameters was only 0.55. To uncover why the correlation coefficient was so much lower on skills that do not have semantically meaningful parameters, we plotted the ExpOps values calculated with the MASTERY THRESHOLD policy on the X-axis and the ExpOps values calculated with the PREDICTIVE SIMILARITY policy on the Y-axis for each skill as shown in figure 3. This plot shows that the PREDICTIVE SIMILARITY policy tends to either agree with the MASTERY THRESHOLD policy or have a lower ExpOps value on skills with parameters that are not semantically meaningful. This suggests that the PREDICTIVE SIMILARITY policy is stopping sooner on skills that students are unlikely to learn. The mastery policy does not give up on these skills, and instead teaches them for a long time.

### 5.5 Experiment 2: Comparing models with the predictive similarity policy

The previous experiment suggests that the PREDICTIVE SIMILARITY policy can effectively mimic the good aspects MASTERY THRESHOLD policy when based on a BKT model. We now wish to see how using models with similar predictive accuracy, but different internal structure will affect it. LFM

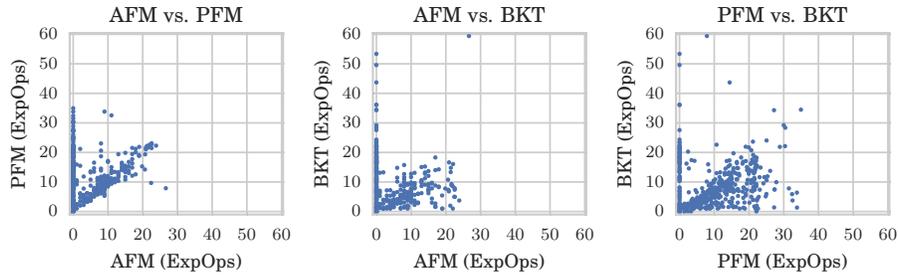


Figure 4: ExpOps plots for the predictive similarity policy using BKT, AFM, and PFM.

Table 3: Correlation coefficients on ExpOps values from policies using BKT, AFM, and PFM.

Models	Coefficient with all skills	Coefficient with skills not stopped immediately
AFM vs. PFM	0.32	0.72
AFM vs. BKT	-0.06	0.44
PFM vs. BKT	0.16	0.46

and BKT models have vastly different structure making them good models for this task. Our earlier results also found that AFM, PFM, and BKT models have similar predictive accuracy. We ran ExpOps on each skill with the PREDICTIVE SIMILARITY policy based both on AFM and PFM. AFM and PFM require a student parameter, which we set to the mean of their trained student parameters. This is commonly done when modeling a student that has not been seen before. We compared the ExpOps values for these two models with the values for the BKT-based PREDICTIVE SIMILARITY policy calculated in the previous experiment.

We first looked at how many skills the different policies immediately stopped on. We found that the BKT-based policy stopped immediately on 31 (6%) of the skills, whilst PFM stopped immediately on 130 (26%) and AFM stopped immediately on 295 (59%).

We calculated the correlation coefficient between each pair of policies on all skills as well as just on skills in which both policies did not stop immediately as shown in table 3. We found that AFM and PFM had the highest correlation coefficient. For each pair of policies, we found that removing the immediately stopped skills had a large positive impact on correlation coefficient. The BKT-based policy had a correlation coefficient of 0.44 with the AFM-based policy and 0.46 with the PFM-based policy on skills that were not immediately stopped on. This suggests that there is a weak correlation between LFM-based and BKT-based policies.

We plotted the ExpOps values for each pair of policies, shown in figure 4. The AFM vs. PFM plot reiterates that the AFM-based and PFM-based policies have similar ExpOps values on skills where AFM does not stop immediately. The BKT vs. PFM plot shows that the PFM-based policy either immediately stops or has a higher ExpOps value than

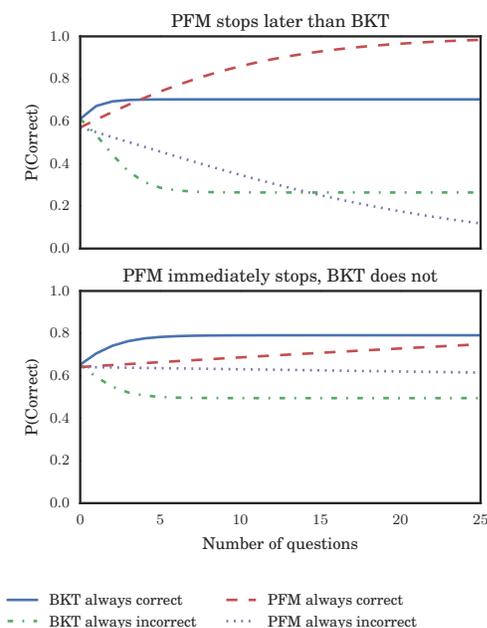
the BKT-based policy on most skills.

To understand why the PFM-based policy tends to either stop immediately or go on for longer than the BKT-based policy, we studied two skills. The first skill is ‘Plot point on minor tick mark — integer major fractional minor’ on which the BKT-based policy has an ExpOps value of 7.0 and the PFM-based policy has an ExpOps value of 20.7. The second skill is ‘Identify solution type of compound inequality using and’ on which the BKT-based policy has an ExpOps value of 11.4 and the PFM-based policy immediately stops. We calculated the predictions of both models on two artificial students, one who gets every question correct and one who gets every question incorrect. In figure 5, we plot the prediction trajectories to see how the predictions of the two models compare. In both plots, the PFM-based policy asymptotes slower than the BKT-based policy. Since LFM’s calculate predictions with a logistic function, PFM predictions asymptote to 0 when given only incorrect responses and 1 when given only correct responses, whereas the BKT model’s predictions asymptote to  $P(G)$  and  $1 - P(S)$  respectively. In the first plot, the PFM-based policy learns at a slower rate than the BKT-based policy, but the predictions do begin to asymptote by the 20<sup>th</sup> question. In the second plot, the PFM-based policy learns much more slowly. After 25 correct questions, the PFM-based policy’s prediction changes by just over 0.1, and after 25 incorrect questions, the PFM-based policy’s predictions changes by less than 0.03. In contrast, the BKT-based policy asymptotes over 10 questions to  $1 - P(S) = 0.79$  when given correct responses and  $P(G) = 0.47$  when given incorrect responses.

This figure also shows how the parameters of a BKT model affect decision making.  $P(L_0)$  is responsible for the initial probability of a correct response.  $P(S)$  and  $P(G)$  respectively provide the upper and lower asymptotes for the probability of a correct response.  $P(T)$  is responsible for the speed of reaching the asymptotes. For the PREDICTIVE SIMILARITY policy, the distance between the initial probability of a correct response and the asymptotes along with the speed of reaching the asymptotes is responsible for the number of questions suggested.

## 6. DISCUSSION

Our results from experiment 1 show that the PREDICTIVE SIMILARITY policy performs similarly to the MASTERY THRESHOLD policy on BKT models with semantically meaningful parameters and suggests the same or fewer problems



**Figure 5: Predictions of BKT models and PFMs if given all correct responses or all incorrect responses on two skills.**

on BKT models without semantically meaningful parameters. Thus, this experiment suggests that the two instructional policies treat students successfully learning skills similarly. The lower ExpOps values for the PREDICTIVE SIMILARITY policy provide evidence that the PREDICTIVE SIMILARITY policy does not waste as much student time as the MASTERY THRESHOLD policy on its own. Fundamentally, the MASTERY THRESHOLD policy fails to recognize that some students may not be ready to learn a skill. The PREDICTIVE SIMILARITY policy does not make the same error. Instead, the policy stops either when the system succeeds in teaching the student or when the skill is unteachable by the system. In practice MASTERY THRESHOLD policies are often used in conjunction with other rules such as a maximum amount of practice before stopping. A comparison of such hybrid policies to the PREDICTIVE SIMILARITY policy is an interesting direction for future work. However, it is important to note that such hybrid policies would still require the underlying model to have a notion of mastery, unlike our predictive similarity policy.

The PREDICTIVE SIMILARITY policy can be used to uncover differences in predictive models. Experiment 2 shows that policies based on models with the same predictive power can have widely different actions. AFMs had a very similar RMSE to both PFMs and the BKT models, but immediately stopped on a majority of the skills. An AFM must provide the same predictions to students who get many questions correct and students who get many questions incorrect. To account for this, its predictions do not change much over time. One may argue that this suggests that AFM models are poor predictive models, because their predictions hardly change with large differences in state. Both AFMs and PFMs have inaccurate asymptotes because it is

likely that students who have mastered the skill will not get every question correct and that students who have not mastered the skill will not get every question incorrect. This means that these models will attempt to stay away from their asymptotes with lower learning rates. One possible solution would be to build LFMs that limit the history length. Such a model could learn asymptotes that are not 0 and 1.

## 7. RELATED WORK

Predictive student models are a key area of interest in the intelligent tutoring systems and educational data mining community. One recent model incorporates both BKT and LFM into a single model with better predictive accuracy than both [10]. It assumes that there are many problems associated with a single skill, and each problem has an item parameter. If we were to use such a model in a when-to-stop policy context, the simplest approach would be to find the problem with the highest learning parameter for that skill, and repeatedly apply it. However, this reduces Khajah et al.'s model to a simple BKT model, which is why we did not explicitly compare to their approach.

Less work has been done on the effects of student models on policies. Fancsali et al. [8] showed that when using the MASTERY THRESHOLD policy with BKT one can view the mastery threshold as a parameter controlling the frequency of false negatives and false positives. This work focused on simulated data from BKT models. Since BKT assumes that students eventually learn, this work did not consider wheel-spinning. Rafferty et al. [17] showed that different models of student learning of a cognitive matching task lead to significantly different partially observable Markov decision process policies. Unlike our work which focuses on deciding when-to-stop teaching a single activity type, that work focused on how to sequence different types of activities and did not use a standard education domain (unlike our use of KDD cup). Mandel et al. [13] did a large comparison of different student models in terms of their predicted influence on the best instructional policy and expected performance of that policy in the context of an educational game; however, like Rafferty et al. their focus was on considering how to sequence different types of activities, and instead of learning outcomes they focused on enhancing engagement. Chi et al. [5] performed feature selection to create models of student learning designed to be part of policies that that would enhance learning gains on a physics tutor; however, the focus again was on selecting among different types of activities rather than a when-to-stop policy. Note that neither BKT nor LFMs in their original form can be used to select among different types of problems, though extensions to both can enable such functionality. An interesting direction of future work would be to see how to extend our policy to take into account different types of activities.

Work on when-to-stop policies is also quite limited. Lee and Brunskill [12] showed that individualizing student BKT models has a significant impact on the expected number of practice opportunities (as measured through ExpOps) for a significant fraction of students. Koedinger et al. [11] showed that splitting one skill into multiple skills could significantly improve learning performance; this process was done by human experts and leveraged BKT models for the policy design. Cen et al. [4] improved the efficiency of student learn-

ing by noticing that AFM models suggested that some skills were significantly over or under practiced. They created new BKT parameters for such skills and the result was a new tutor that helped students learn significantly faster. However, the authors did not directly use AFM to induce policies, but rather used an expert based approach to transform the models back to BKT models, which could be used with existing mastery approaches. In contrast, our approach can be directly used with AFM and other such models.

Our policy assumes that learning is a gradual process. If you were to instead subscribe to an all-at-once method of learning, you could possibly use the moment of learning as your stopping point. Baker et al. provide a method of detecting the moment at which learning occurs [2]. However, this work does not attempt to build instructional policies.

## 8. CONCLUSION & FUTURE WORK

The main contribution of this paper is a when-to-stop policy with two attractive properties: it can be used with any predictive student model and it will provide finite practice both to students that succeed in learning a given skill and to those unable to do so given the presented activities.

This policy allowed us for the first time to compare common predictive models (LFMs and BKT models) in terms of their predicted practice required. In doing so we found that models with similar predictive error rates can lead to very different policies. This suggests that if they are to be used for instructional decision making, student models should not be judged by predictive error rates alone. One limitation of the current work is that only one dataset was used in the experiments. To confirm these results it would be useful to compare to other datasets.

One key issue raised by this work is how to evaluate instructional policy accuracy. One possible solution is to run trials with students stopping after different numbers of questions. The student would take both a pre and post-test, which could be compared to see if the student improved. However, such a trial would require many students and could be detrimental to their learning.

There is a lot of room for extending this instructional policy. First, we would like to incorporate other types of interactions, such as dictated information (“tells”) or worked examples, into the PREDICTIVE SIMILARITY policy. This would give student models more information and hopefully lead to better predictions. Second, the PREDICTIVE SIMILARITY policy is myopic, and we are interested in the effects of expanding to longer horizons. Third, we are excited about extending this instructional policy to choosing between skills. Instead of stopping when there is a high probability of predictions not changing, the instructional policy could return either the skill that had the highest chance of a significant change in prediction, or the skill with the highest expected change in prediction.

## 9. REFERENCES

- [1] V. Alevan, B. M. McLaren, J. Sewall, and K. R. Koedinger. The cognitive tutor authoring tools (ctat): preliminary evaluation of efficiency gains. In *ITS*. Springer, 2006.
- [2] R. S. Baker, A. B. Goldstein, and N. T. Heffernan. Detecting learning moment-by-moment. *IJAIED*, 21(1), 2011.
- [3] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis—a general method for cognitive model evaluation and improvement. In *ITS*. Springer, 2006.
- [4] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary?—improving learning efficiency with the cognitive tutor through educational data mining. *FAIA*, 158, 2007.
- [5] M. Chi, K. R. Koedinger, G. J. Gordon, P. W. Jordan, and K. VanLehn. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *EDM*, 2011.
- [6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMAP*, 4(4), 1994.
- [7] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *EDM 2013*, 2010.
- [8] S. E. Fancsali, T. Nixon, and S. Ritter. Optimal and worst-case performance of mastery learning assessment with bayesian knowledge tracing. In *EDM*, 2013.
- [9] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *ITS*, 2010.
- [10] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. *EDM*, 2014.
- [11] K. R. Koedinger, J. C. Stamper, E. A. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *AIED*. Springer, 2013.
- [12] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *EDM*, 2012.
- [13] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. *AAMAS*, 2014.
- [14] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *UMAP*. Springer, 2010.
- [15] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis—a new alternative to knowledge tracing. 2009.
- [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12, 2011.
- [17] A. Rafferty, E. Brunskill, T. Griffiths, and P. Shafto. Faster teaching by POMDP planning. In *AIED*, 2011.
- [18] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra 1 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge. find it at <http://pslccdatashop.web.cmu.edu/kddcup/downloads.jsp>.

# *Your model is predictive— but is it useful?*

## Theoretical and Empirical Considerations of a New Paradigm for Adaptive Tutoring Evaluation

José P. González-Brenes  
Digital Data, Analytics and Adaptive Learning  
Pearson School Research  
Philadelphia, PA, USA  
jose.gonzalez-brenes@pearson.com

Yun Huang  
Intelligent Systems Program  
University of Pittsburgh  
Pittsburgh, PA, USA  
yuh43@pitt.edu

### ABSTRACT

Classification evaluation metrics are often used to evaluate adaptive tutoring systems— programs that teach and adapt to humans. Unfortunately, it is not clear how intuitive these metrics are for practitioners with little machine learning background. Moreover, our experiments suggest that existing convention for evaluating tutoring systems may lead to suboptimal decisions. We propose the Learner Effort-Outcomes Paradigm (Leopard), a new framework to evaluate adaptive tutoring. We introduce Teal and White, novel automatic metrics that apply Leopard and quantify the amount of effort required to achieve a learning outcome. Our experiments suggest that our metrics are a better alternative for evaluating adaptive tutoring.

### Keywords

evaluation, efficacy, classification evaluation metrics

## 1. INTRODUCTION

A fundamental part of the scientific and engineering process is *testability*— the property of evaluating whether a hypothesis or method can be supported or falsified by data of actual experience. For example, in educational data mining, we formulate testable hypotheses that claim that the methods we engineer improve the outcomes of learners. In this manuscript, we study how to verify learner outcome hypotheses.

We focus on evaluating a popular type of educational method called *adaptive intelligent tutoring system*. Adaptive systems teach and adapt to humans; their promise is to improve education by optimizing the subset of *items* presented to students, according to their historical performance [5], and on features extracted from their activities [10]. In this context, items are questions, problems, or tasks that can be graded individually.

Evaluation metrics are important because they quantify the extent of whether an educational system helps learners. For example, a practitioner may use an evaluation method to choose which of the alternative adaptive tutoring systems to deploy in a classroom, or school district. On the other hand, a researcher may be interested in quantifying the improvements of her system compared to previous technology.

Our main contributions are proposing a novel evaluation paradigm for assessing adaptive tutoring and examples of when traditional evaluation techniques are misleading. This paper is organized as follows: § 2 reviews related methods for evaluating adaptive systems; § 3 describes the paradigm we propose for automatic evaluation of tutoring systems; § 4 provides a meta-evaluation of our novel evaluation techniques; and, § 5 provides some concluding remarks.

## 2. BACKGROUND

Adaptive tutoring is often implemented as a complex system with many components, such as a student model, content pool, and a cognitive model. Adaptive tutoring may be evaluated with randomized control trials. For example, in a seminal study [5] that focused on earlier adaptive tutors, a controlled trial measured the time students spent on tutoring and their performance on post-tests. The study reported that the tutoring system enabled significantly faster teaching, while students maintained the same or better performance on post-tests

Unfortunately, controlled trials can become extremely expensive and time consuming to conduct: they require institutional review board approvals, experimental design by an expert, recruiting (and often payment!) of enough participants to achieve statistical power, and data analysis. Automatic evaluation metrics improve the engineering process because they enable less expensive and faster comparisons between alternative systems. Fields that have agreed on automatic evaluation have seen an accelerated pace of technological progress. For example, the widespread adoption of the Bleu metric [15] in the machine translation community has lowered the cost of development and evaluation of translation systems. At the same time, it has enabled machine translation competitions that result in great advances of translation quality. Similarly, the Rouge metric [13] has helped the automatic summarization community transition

from expensive user studies of human judgments that may take thousands of hours to conduct, to an automatic metric that can be computed very quickly.

The adaptive tutoring community has tacitly adopted conventions for evaluating tutoring systems [6, 16, 18]. Researchers often evaluate their models with classification evaluation metrics that assess the *student model* component of the tutoring system— student models are the subsystems that forecast whether a learner will answer the next item correctly. Popular classification evaluation metrics include accuracy, log-likelihood, Area Under the Curve (AUC) of the Receiver Operating Characteristic curve, and, strangely for classifiers, the Root Mean Square Error. However, automatic evaluation metrics are intended to measure an outcome of the end user. For example, the PARADISE [22] metric used in spoken dialogue systems correlates to user satisfaction scores. Not only is there no evidence that supports that classification metrics correlate with learning outcomes; but, prior work [2] has identified serious problems with them. For example, classification metrics ignore that an adaptive system may not help learners— which could happen with a student model with a flat or decreasing learning curve [1, 20]. A decreasing learning curve implies that student performance decreases with practice; this curve is usually interpreted as a modeling problem, because it operationalizes that learners are better off with no teaching. Therefore, an adaptive tutor with a student model with a decreasing learning curve does not teach students.

Surprisingly, in spite of all of the evidence against using classification evaluation metrics, their use is still very widespread in the adaptive literature [6, 16, 18]. Moreover, there is very little research on alternative evaluation techniques. A noticeable exception is recent work on individualizing student models [12]. The authors evaluated their approach using a method called *ExpOppNeed*, which calculates the expected number of practice opportunities that learners require to master the content of the tutoring curriculum. Though their evaluation methodology is extremely interesting and promising, it was not intended to be generalizable. In the next section we extend on prior work and present a novel general paradigm for evaluating adaptive systems.

### 3. LEOPARD EVALUATION

Adaptive tutoring implies making a trade-off between minimizing the amount of student *effort*, by carefully personalizing the curriculum, and maximizing student *outcomes* [4]. For example, repeated practice on a skill may improve student proficiency, at the cost of a missed opportunity for teaching new material. Adequate values for student effort and outcomes respond to external expectations from the social context. For example, it is not acceptable for a tutor to minimize effort by not teaching any content at all, or to maximize outcomes by taking twenty years to teach a simple concept. The right trade off is defined by subject matter experts.

We propose the novel Learner Effort-Outcomes Paradigm (Leopard) for automatic evaluation of adaptive tutoring. At its core, Leopard quantifies the effort and outcomes of students in adaptive tutoring. Even though measuring effort and outcomes is not novel by itself, our contribution is mea-

suring both without a randomized control trial.

- **Effort:** Quantifies how much practice the adaptive tutor gives to students. In this paper we focus on counting the number of items assigned to students but, alternatively, amount of time could be considered.
- **Outcome:** Quantifies the performance of students after adaptive tutoring. For simplicity, we operationalize performance as the percentage of items that students are able to solve after tutoring. We assume that the performance on solving items is aligned to the long-term interest of learners.

We argue that Leopard is more intuitive than classification metrics because the effort and outcome resonate to educational principles. We now describe two novel metrics that apply the Leopard philosophy. In § 3.1, we describe Teal, a metric that calculates the theoretical expected behavior of students when interacting with a family of student models; and in § 3.2, we describe White<sup>1</sup> a metric that uses empirical data that may have not been collected on a control trial.

### 3.1 Theoretical Evaluation of Adaptive Learning Systems (Teal)

We formulate Theoretical Evaluation of Adaptive Learning Systems (Teal) to evaluate adaptive tutoring from the expected behavior of their student model. Teal focuses on models of the *Knowledge Tracing Family*— a very popular set of student models [10].

To use Teal on data collected from students, we first train a model using an algorithm from the Knowledge Tracing family (§ 3.1.1), then we use the learned parameters to calculate the effort (§ 3.1.2) and outcome (§ 3.1.3) for each skill. We discuss how to use Teal on models that use features (§ 3.1.4) and our design decisions (§ 3.1.5).

#### 3.1.1 Knowledge Tracing Family

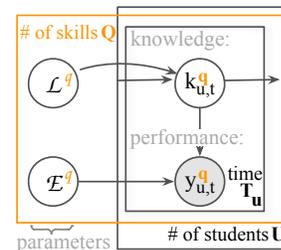


Figure 1: Knowledge Tracing plate diagram. The color of the circles represent whether the variable is latent (white), or observed in training (light), and plates represent repetition.

Figure 1 describes the Knowledge Tracing [5] model, the most simple member of the family. Knowledge Tracing requires a mapping of items to skills, often built by subject matter experts, although automatic approaches exist [8]. These skill mappings are also called cognitive models, or Q-matrices. Knowledge Tracing uses a Hidden Markov Model (HMM) per skill to model the student’s knowledge as latent variables. The binary observation variable  $y_{u,t}^q$  represents

<sup>1</sup>Tradition names metrics like colors! E.g., Rouge, Bleu.

whether the student  $u$  applies the  $t^{\text{th}}$  practice opportunity of skill  $q$  correctly. The latent variable  $k_{u,t}^q$  models the latent student proficiency, which is often modeled with a binary variable to indicated mastery of the skill. To declutter notation, we may not explicitly write the indices  $q$  and  $u$ . There are two conventions for naming the skill-specific parameters of Knowledge Tracing. In the HMM tradition, the parameters are simply named transition or learning ( $\mathcal{L}$ ), and emission ( $\mathcal{E}$ ). In the educational tradition when using two latent states the parameters are called initial knowledge ( $l_0$ ), learning ( $l$ ), forgetting ( $f$ ), guess ( $g$ ) and slip ( $s$ ). The Knowledge Tracing family includes models that parameterize the emission probabilities, transition probabilities, or both. For example, in Knowledge Tracing, the emission probability of emitting an answer  $\mathbf{y}$  when the student has knowledge  $\mathbf{k}$  is:

$$\mathcal{E}_{\mathbf{y},\mathbf{k}} = p(\mathbf{y}|\mathbf{k}) \quad (1)$$

Which is simply a binomial probability. To allow features in the emissions, we replace the binomial with a logistic regression [10]:

$$\mathcal{E}_{\mathbf{y},\mathbf{k}}(\boldsymbol{\beta}, \mathbf{X}_t) = p(\mathbf{y}|\mathbf{k}; \boldsymbol{\beta}, \mathbf{X}_t) \quad (2)$$

$$= \frac{1}{1 + \exp(-\boldsymbol{\beta} \cdot \mathbf{X}_t)} \quad (3)$$

Here  $\mathbf{X}_t$  is the feature vector extracted at time  $t$ , and  $\boldsymbol{\beta}$  is the regression coefficient vector. The feature may indicate, for example, if the student requested a hint.

### 3.1.2 Effort

Teal calculates the expected number of practice that an adaptive tutor gives to students. We assume a policy that the tutor stops teaching a skill once the student is very likely to answer the next item correctly according to a model from the Knowledge Tracing Family. For notational convenience, we define the probability of answering the next item correctly as:

$$c_{t+1}(\mathbf{y}_1, \dots, \mathbf{y}_T) \equiv p(y_{t+1} = \text{correct} | \mathbf{y}_1, \dots, \mathbf{y}_t; \mathcal{L}, \mathcal{E}) \quad (4)$$

Here  $\mathcal{L}$  and  $\mathcal{E}$  are the parameters of the Knowledge Tracing Family model. We can estimate  $c_{t+1}$  using conventional inference techniques for HMMs [19], such as the Forward-Backward algorithm.

The adaptive tutor teaches an additional item if two conditions hold: (i) it is likely that the student will get the next item wrong— in other words, the probability of answering correctly the next item is below a threshold  $R$ ; and (ii) the tutor has not decided to stop instruction already. More formally, the tutor keeps teaching if:

$$\text{teach}(\mathbf{y}_1, \dots, \mathbf{y}_t, R) \equiv \begin{cases} 1 & \text{if } \forall_{t' < t} c_{t'+1}(\mathbf{y}_1, \dots, \mathbf{y}_{t'}) < R \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

We now can calculate at which practice opportunity the tutor should stop instruction. For simplicity, we assume all sequences are of length  $T$ . We simply count all of the times the tutor decides to teach a new item:

$$\text{cost}_R(\mathbf{y}_1, \dots, \mathbf{y}_T) \equiv \sum_{t=1}^T \text{teach}(\mathbf{y}_1, \dots, \mathbf{y}_t, R) \quad (6)$$

Note that if the probability of answering correctly the next item has not reached the threshold in  $T$  time steps, the cost is defined as  $T$ . Teal defines effort as the expected value of the number of practice opportunities a tutor gives. This is:

$$\begin{aligned} \text{effort}(R) &\equiv \mathbb{E}(\text{cost}_R(\mathcal{Y}_T)) & (7) \\ &= \sum_{\mathbf{y}_1, \dots, \mathbf{y}_T \in \mathcal{Y}_T} \underbrace{\text{cost}_R(\mathbf{y}_1, \dots, \mathbf{y}_T)}_{\text{amount of practice}} \cdot \underbrace{p(\mathbf{y}_1, \dots, \mathbf{y}_T)}_{\text{sequence likelihood}} & (8) \end{aligned}$$

Here,  $\mathcal{Y}_T$  is the set of all sequences of length  $T$ . When we have binary student outcomes (correct or not), the cardinality of this set is  $2^T$ , which makes Teal only tractable for sequences of a few dozens of observations. In our experience, the sequences of adaptive tutoring systems are often in this range. In a companion paper [9] we give an alternative formulation of Teal that allows approximate calculations. The likelihood of the sequence can be efficiently estimated using the Forward-Backward algorithm.

### 3.1.3 Outcome

We define the outcome of a student as the mean performance after the tutor should stop instruction. For a particular sequence with student cost  $k = \text{cost}_R(\mathbf{y}_1, \dots, \mathbf{y}_T)$ , this is:

$$\text{outcome}(\mathbf{y}_1, \dots, \mathbf{y}_T, k) \equiv \begin{cases} \text{mean}(y_k \dots y_T) & \text{if } k < T \\ \text{impute value} & \text{otherwise} \end{cases} \quad (9)$$

We map the correct and incorrect student responses  $y_t$  into 1 or 0, respectively. If the student sequence does not reach the performance threshold, we impute the value of the outcome. In this paper, we set the imputation value to 0. We define the score as the expected value of the outcome:

$$\begin{aligned} \text{score}(R) &\equiv \mathbb{E}(\text{outcome}(\mathcal{Y}_T, k)) & (10) \\ &= \sum_{\mathbf{y}_1, \dots, \mathbf{y}_T \in \mathcal{Y}_T} \text{outcome}(\mathbf{y}_1, \dots, \mathbf{y}_T, R) \cdot p(\mathbf{y}_1, \dots, \mathbf{y}_T) & (11) \end{aligned}$$

### 3.1.4 Usage on Models With Features

For models that parameterize emission or transitions we first must build a counterfactual feature vector  $\mathbf{X}$ , and use it to calculate model parameters that do not depend on features. For example, consider a model that uses a binary feature vector that encodes students in different conditions. Conditions can be any feature of interest of the tutoring system, such as the ability to display multimedia content. We can use Teal to calculate the effort of students in each of the specific conditions.

For example, consider a feature vector  $\mathbf{X} = (f_1, f_2, \dots, f_n)$ . Feature  $f_1$  is 1 iff the student is using condition 1 (e.g., multimedia content is available), feature  $f_2$  is 1, iff the student is using condition 2, etc. The vector is all zeros if the student is in the control condition. If we activate feature  $f_1$ , we can calculate the effort or score of students in the treatment 1. To apply Teal we first estimate counterfactual slip and guess parameters using Equation 3. We can use the counterfactual parameters with Teal.

For some models with features, Teal may require that students are assigned randomly to feature activation conditions, so that the regression coefficients can be interpreted as causal effects. Teal may not be appropriate if – for example – the features have reverse causality, or if there are omitted variables in the model.

### 3.1.5 Design Discussion

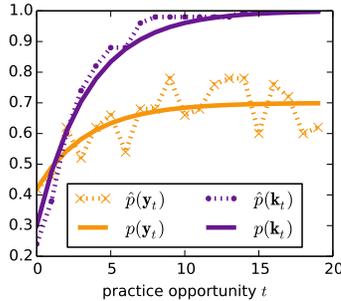


Figure 2: Expected and empirical student performance for a skill ( $l_0 = 0.3$ ,  $l = 0.25$ ,  $g = 0.3$ ,  $s = 0.3$ ,  $f = 0$ ).

Teal extends the ExpOppNeed algorithm discussed on § 2. We compare both approaches to justify our design decisions.

1. **When to stop tutoring.** Teal expects tutoring to stop once the student is very likely to apply the skill correctly. On the other hand, ExpOppNeed relies on stopping tutoring once the posterior probability of the latent variable for knowledge is above a threshold. Figure 2 compares both approaches for some Knowledge Tracing parameters. The solid lines represent the expected values derived theoretically<sup>2</sup> for both strategies. To illustrate what actual student behavior may look like, we plotted dotted lines for 50 synthetic students sampled from a HMM. Although individual students vary, their average behavior is close to theoretical.

In the figure, with 15 practice opportunities the students have close to 100% probability of skill mastery, while they only have 65% probability of applying the skill correctly. This big gap between the probability of mastery and probability of correct (the two solid lines) implies that the model is defining mastery as a state when students have low probability of applying the skill correctly. Low probability of answering correctly in a mastery state can occur due to a number of problems, for example, an incorrect item-to-skill mapping, or confusing tutoring content. We argue that an evaluation metric should penalize such models to be consistent with the Mastery Learning Theory [3].

Moreover, prior work [1] has demonstrated that some ill-defined models have probability of correct decreasing with practice opportunities, at the same time that the probability of mastery increases. ExpOppNeed does not penalize such ill-defined models, but Teal does.

<sup>2</sup>Prior work derived [21]:  $p(y_t = \text{correct}) = 1 - s - A\beta^t$ . Here,  $\beta = (1 - l)$ , and  $A = (1 - s - g) \cdot (1 - l_0)$

---

#### Algorithm 1 Single-Skill White

---

**Require:** performance sequences  $\mathbf{y}_{u,q,t}$ , student model predictions  $\hat{\mathbf{c}}_{u,q,t}$  (the subscripts index students, skills, and practice opportunities), threshold  $R$

- 1: **function** WHITE( $\mathbf{y}_{u,q,t}, \hat{\mathbf{c}}_{u,q,t}, R$ )
- 2:   **for** each student  $u$  **do**
- 3:     **for** each skill  $q$  **do**
- 4:        $\triangleright$  Select data for student  $u$  and skill  $q$  only:
- 5:        $\mathbf{y}', \hat{\mathbf{c}}' \leftarrow \text{filter}(\mathbf{y}, \hat{\mathbf{c}}, u, q)$
- 6:       effort( $q, u$ )  $\leftarrow 0$
- 7:       **for** each practice opportunity  $t$  in  $\mathbf{y}'$  **do**:
- 8:         **if**  $\hat{c}'_{t+1} \geq R$  **then**
- 9:         score( $q, u$ )  $\leftarrow \text{mean}(y_{t+1}, \dots, y_T)$
- 10:         **next** skill  $q$
- 11:         **else if** last( $t$ ) **then**
- 12:         score( $q, u$ )  $\leftarrow$  impute
- 13:         effort( $q, u$ )  $\leftarrow$  effort( $q, u$ ) + 1
- 14:     **return** effort, score

---

2. **What to measure.** ExpOppNeed does not calculate expected outcome of students. Teal considers both student outcome and effort because it is trivial to optimize one of the metrics if the other one is ignored.
3. **Precision of the results** Both ExpOppNeed and Teal have exponential computational complexity. However, ExpOppNeed uses a heuristic to prune sequences with low probability. Unfortunately, if the effort is very high (or infinite), the likelihood of the individual sequences becomes very low, and ExpOppNeed prunes the sequences too soon and therefore it may underestimate the effort. Teal improves on ExpOppNeed by defining effort on fixed-length sequences and not doing pruning.

We now summarize some limitations of our approach. Teal assumes that the model parameters are correct, and does not take into account potential modeling problems—such as misspecification, or over-fitting. By design, Teal only is able to evaluate models in the Knowledge Tracing Family. We now present a novel evaluation method that addresses these limitations.

## 3.2 Whole Intelligent Tutoring System Empirical Evaluation (White)

We propose Whole Intelligent Tutoring System Evaluation (White), a novel automatic method that evaluates the recommendations of an adaptive system using data. White does not assume the student data is generated by a Knowledge Tracing model; instead, it relies on counterfactual simulations. White reproduces the decisions that the tutoring system *would* have made given the input data on the test set, by counting how many items the adaptive tutor would ask students to solve, and what is the mean student performance after tutoring.

Algorithm 1 describes White for a tutoring system that assumes an item is assigned to exactly one skill. We leave more complex tutors for future work. The input of White is the student performance sequences  $\mathbf{y}$ , the predictions of answering correctly  $\hat{\mathbf{c}}$ , and a threshold  $R$  that defines what is the

		predicted performance			
		actual performance			
	t	student u	skill q	$\hat{c}_{u,q,t+1}$	$y_{u,q,t}$
effort=	0	Alice	s1	.6	
	1	Alice	s1	.5	0
	2	Alice	s1	.5	1
	3	Alice	s1	.6	1
	0	Bob	s1	.4	
effort=	1	Bob	s1	.7	1
	2	Bob	s1	.7	1
	3	Bob	s1	.7	1
	4	Bob	s1	.8	0
	4	Bob	s1	.9	1
	6	Bob	s1	.9	1

Figure 3: Example of White calculating counterfactual score and effort using empirical data ( $R = 0.6$ ).

target probability of correct. White assumes that the students are a random sample of the student population. The predictions are calculated by the student model component of the adaptive tutoring. For a data-driven student model, the predictions can be informed with the history preceding the current time step. For instance, to predict on the third time step, the student model may use the data up to the second time step. For example, for Knowledge Tracing:

$$\hat{c}_t = \hat{p}(y_t = \text{correct} | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) \quad (12)$$

Figure 3 shows example data of how White works for a 60% threshold ( $R = 0.6$ ). For each student and skill in the test set, White estimates their counterfactual effort—how many items the student *would* have solved using the tutoring system. In our example, Alice does not get to practice the skill because the student model believes that she is likely to already know it (effort=0), but Bob is given one practice opportunity (effort=1). After Bob answers correctly the item, he is not given any more practice. White also calculates a counterfactual score to represent the student learning. It is the percentage of correct answers after the instruction would have stopped. The score is related to an existing classification evaluation metric called precision. Precision aggregates the entire dataset, while score is computed by students and skills. Although superficially it may sound as a small difference, our strategy allows us to avoid a special case of the Simpson’s Paradox. In § 4.1.1 we discuss the issue more.

In this paper, when we report results with White, we impute the score of students that do not reach the threshold with their average performance. This is deliberately a different imputation strategy that we use with Teal, which assigns a score of zero to students that do not reach the threshold.

## 4. META-EVALUATION

In this section we meta-evaluate Leopard. We experiment with data from students (§ 4.1) and simulations (§ 4.2).

We compare these sets of metrics:

- **Conventional metrics.** We use classification evaluation metrics to evaluate how the student models predict future student performance. For this, we allow student models to use the history preceding the time step we want to predict.
- **Leopard metrics.** We use the score and effort as calculated by White and Teal. For simplicity we report the average scores across skills, and the sum of the mean effort. For  $U$  students and  $Q$  skills, this is:

$$\text{dataset score}(R) = \frac{1}{Q \cdot U} \sum_q \sum_u \text{score}(q, u) \quad (13)$$

$$\text{dataset effort}(R) = \frac{1}{U} \sum_q \sum_u \text{effort}(q, u) \quad (14)$$

## 4.1 Real Student Data

We use data collected from a commercial non-adaptive tutoring system for middle school Math. Our dataset includes only the first part of the entire curriculum, and contains students from the same grade from multiple schools. It contains approximately 1.2 million observations from 25,000 students. We randomly split the dataset into three sets of students. The training and test set have 60% and 20% of the students, respectively. The remainder of the data is reserved for future experiments not described in this paper. The item bank was mapped to skills in three different ways—the *coarse* definition maps the items into 27 skills, the *fine* definition into 90 skills, and the proprietary one is not reported.

### 4.1.1 Are predictive models always useful?

Assessing an evaluation metric with real student data is difficult because we often do not know the ground truth. To get around this, we now describe a strategy to select a subset of the dataset that we know the behavior of. Our main insight is that for adaptive tutoring to be able to optimize when to stop instruction, the student performance should increase with repeated practice (the learning curve should be increasing). Our strategy consists on selecting the subset of the data where student modeling may fail, because student performance remains flat or decreases with practice.

We first train a simplified Performance Factors Analysis [17] (PFA) model. We use a logistic regression for each skill:

$$p(y_{u,t}^q) = \frac{1}{1 + \exp(\beta^q \cdot \mathbf{X}^q)} \quad (15)$$

The dimensions of  $\mathbf{X}^q$  are the count of prior correct responses of the student and an intercept. We learn the parameters of the model  $\beta^q$  using constrained optimization—the regression coefficient for the effect of prior correct responses has to be non-negative.

We only use data from the skills that have zero regression coefficient for the effect of prior correct responses (flat or decreasing learning curve). Such skills are not suitable for an adaptive tutor because the PFA student model believes that practice does not influence student performance. More concretely, this PFA model would give infinite practice to difficult skills, or no practice to easy skills. Table 1 compares the results of using White and two conventional metrics on

the test set of the selected skills. We compare with a majority class model that always predicts students answers as correct. The conventional metrics we report are the AUC, because of its popularity, and the F-metric, because in experiments we report later correlates highly with White. For White we use a threshold of 60%. We cannot report on Teal because PFA is not part of the Knowledge Tracing Family.

Table 1: Evaluation metric comparison.

	White		conventional	
	score	effort	F	AUC
Performance Factors Analysis	.18	10.1	<b>.79</b>	<b>.85</b>
Majority Class	.18	11.2	0	.50

The AUC and F-metric results are arguably very high, indicating that the PFA model is highly predictive—yet by construction, we know that the model is *not useful* for adaptivity. The high prediction power of PFA is explained only by the intercepts of the model. That is, the predictions are based on the skill difficulty, independently of the student performance. We argue that White communicates better the unfavorable nature of the model because it reports a very low score, and only a small improvement of effort when compared to a baseline.

The problem with metrics that aggregate over the entire dataset, like the AUC and the F-metric, can be explained by Simpson’s paradox—a trend that appears in different groups of data that disappears or reverses when the groups are combined. Because adaptive tutors learn a model from each skill independently, it is effectively a group of models. White and Teal evaluate each skill independently and are not susceptible to this problem. Consider the alternatives:

- Reporting as a baseline the *difficulty classifier*—a classifier that only considers the fraction of correct answers of each skill in the training set. For example, in Table 1, the PFA model has an AUC of 0.8, the same as the difficulty classifier. Because PFA did not outperform this baseline, it suggests the student model has a problem. However, simulations [8] provide evidence that useful student models may have predictive performance similar to the difficulty classifier. Therefore, the difficulty classifier baseline may reject some useful student models. Moreover, convention expects classifiers to have an AUC of higher than 0.5 to be useful, and this new baseline would break this interpretation.
- Calculating classification metrics over skills independently. This would only be useful when the skills are known beforehand, and not discovered with data [8]. We now provide evidence that suggests that classification metrics may be misleading, even when they are not affected by the Simpson’s paradox.

#### 4.1.2 Do traditional metrics lead to good decisions?

We now compare Leopard and traditional metrics for choosing an item-to-skill mapping. We train a PFA model using our Math dataset. Table 2 compares the results of White ( $R = 0.6$ ) and AUC.

If we were to choose the best skill mapping by AUC alone, we

Table 2: Comparisons of item-to-skill definitions.

	White		AUC
	score	effort	
coarse	<b>.41</b>	<b>55.7</b>	.69
fine	.36	88.1	<b>.74</b>

would choose the finer item-to-skill mapping, while White selects the coarser one. Why do they disagree? The fine skill mapping has almost three times the number of skills (90 skills) than the coarse mapping (27 skills). This means that for the effort to be the same on both models, the finer model should give a third of the practice of the coarser model. Even though the finer model is slightly more predictive, we argue that the coarser model is better suited for adaptive tutoring.

#### 4.1.3 Case Study

For completeness, Table 3 demonstrates using different student modeling techniques with the coarse item-to-skill mapping. For Knowledge Tracing, we show both the White estimates, and the Teal estimates (in parenthesis). We use the average sequence length for each skill because Teal requires a sequence length as an input. The estimates of Teal and White for effort are very similar, but their scores mismatch—possibly due to the differences in imputation for skills that don’t reach the threshold. The low score metrics are indicative of students not reaching the performance threshold. This suggests that further inspection is necessary, because the learning curves may be decreasing or some skills may have high slip probabilities. One of the advantages of White is that it can be used to evaluate non-probabilistic student models. For example, we use White to evaluate the student model that gives practice of a skill until the student gets three correct answers in the skill.

Table 3: Student model comparison using Leopard

	Leopard		AUC
	score	effort	
Knowledge Tracing	.39 (.18)	<b>49.5</b> (50.9)	<b>.70</b>
Performance Factor Analysis	.41	55.7	.69
Three Correct	.39	59.1	n/a
Majority Class	.41	65.6	.50

## 4.2 Simulations

With real data, we do not know the extent that the parameters are learned correctly, or affected by modeling problems—such as misspecification. We now use synthetic data to evaluate different metrics and compare them to a ground truth. Given that we know the Knowledge Tracing parameters that were used to generate the synthetic datasets, we can use Teal to calculate *exactly* the student effort and outcomes.

We sample 500 different datasets using random Knowledge Tracing parameters. In none of the datasets we allow forgetting, but we do not impose any other constraint (not even that students improve with practice). Each dataset has only a single skill, and has 200 students with 10 practice opportunities. We do not learn parameters from the synthetic

dataset, so we do not cross-validate.

#### 4.2.1 Which metrics correlate best with the truth?

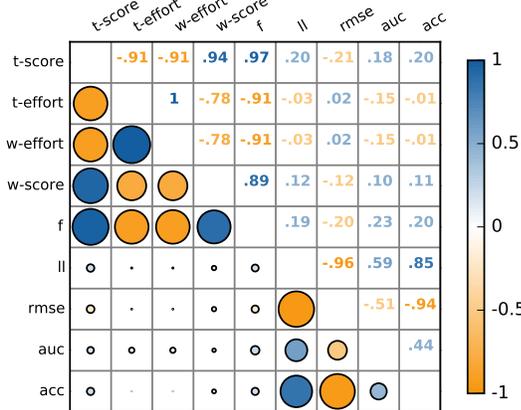


Figure 4: Correlation matrix of Leopard and conventional metrics. The size of the circles indicate the magnitude of the Pearson  $\rho$  correlation coefficient.

Figure 4 shows the pairwise Pearson- $\rho$  correlations across 500 synthetic datasets on Teal (score), Teal (effort), White (effort), White (score), F-metric, Log-likelihood, RMSE, AUC, and Accuracy.

The metrics that correlate the most with the ground truth are White and the F-metric. Interestingly, the ground truth effort and score have low correlation with all the conventional metrics, except the F-metric, but the conventional metrics have relatively high correlation among each other (except the F-metric). In other words, most conventional metrics seem to be exchangeable.

We now investigate the effect of the imputation strategy of White. We are mindful that all of the synthetic students have 10 practice opportunities. Therefore, if White reports an effort of 10 for a dataset, it is likely that the dataset is not suitable for adaptivity, and that White may be imputing missing data to calculate the score. Figure 5 compares the 324 datasets that White reports effort lower than 9.99. Each dot in the scatterplot represents a different dataset. We see that effort computed with White has an almost perfect correlation with the ground truth ( $\rho = 1.00$ ,  $p < 0.05$ ). On the other hand, the score computed with White is affected by our imputation strategy, but still has near perfect correlation ( $\rho = 0.98$ ,  $p < 0.05$ ) with the ground truth. The correlation of the F-metric with the ground truth effort ( $\rho = -0.47$ ) and score ( $\rho = 0.89$ ) is relatively lower than White's. E.g., when the ground truth effort is 0, the F-metric ranges from very bad (0.2) to very good (1.0) predictive power, but White's effort is close to 0. Moreover, we speculate that score and effort may be more relatable to practitioners with little background of machine learning than the F-metric.

#### 4.2.2 Does White Converge to True Values?

We now investigate whether White converges to the true values calculated by Teal. We use the same parameters used to plot Figure 2, and we manipulate the number of synthetic

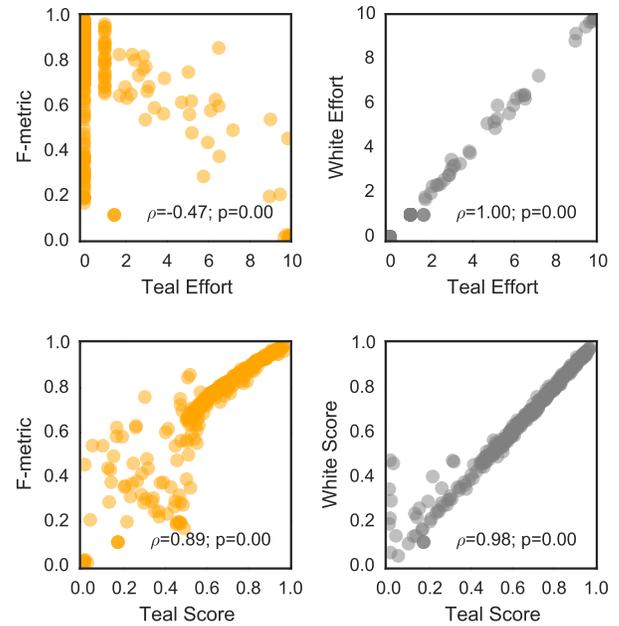


Figure 5: Comparison between F-metric and White to the ground truth.

students, each student with 20 practice opportunities, Figure 6 shows that with little data, White converges to the true value computed by Teal. Future work may provide a formal argument of when and how much data White requires to convergence.

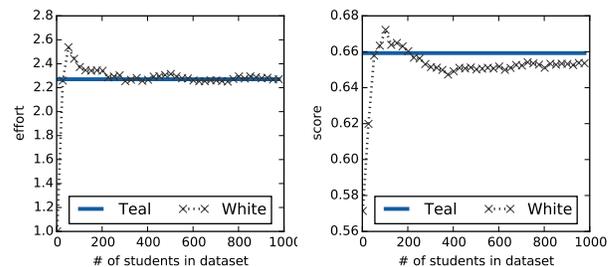


Figure 6: Example of White converging to Teal.

## 5. DISCUSSION

Our main contribution is the Leopard framework that automatically assesses adaptive tutoring systems in dimensions that relate to learner effort and outcomes. These dimensions were previously measured only in randomized control trials. We present Teal and White, two novel metrics that apply Leopard and are useful to evaluate adaptive tutoring systems. Secondary contributions include a novel methodology to assess evaluation metrics, the insight of Simpson's paradox affecting adaptive tutoring evaluation, and the implementation of the techniques we propose in this paper<sup>3</sup>.

Classification evaluation metrics are very widespread in many disciplines, and their use in education is very important.

<sup>3</sup><http://josepablogonzalez.com>

For example, for Computer-Adaptive Testing (CAT), classification metrics provide very useful insights to psychometric models. Leopard is not intended to replace classification metrics, randomized control trials, automatic experimentation [14], or visualization approaches [7, 11]. Leopard is a complementary approach to existing techniques, and we claim that it is specially useful when *in vivo* and online experimentation is not feasible.

We argue against the *de facto* standard of evaluating adaptive tutoring solely on classification metrics. Our experiments on real and synthetic data reveal that it is possible to have student models that are very predictive (as measured by traditional classification metrics), yet provide little to no value to the learner. Moreover, when we compare alternative tutoring systems with classification metrics, we discover that they may favor tutoring systems that require higher student effort with no evidence that students learn more. That is, when comparing two alternative systems, classification metrics may prefer a suboptimal system.

An interesting future direction may be to relax Teal’s assumption that all sequences have fixed-length. Future work may provide more rigorous theoretical analysis on convergence, confidence intervals, validate our metrics with randomized control trials, or derive White for policies with multiple skills per item.

We are excited to see future work in adaptive tutoring systems reporting their contributions in terms of learner effort and outcomes. Besides the technical contributions of our evaluation metrics, we hope that our work contributes to the mission of driving the student modeling community to have a more learner-centric perspective.

## 6. REFERENCES

- [1] R. Baker, A. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In B. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 406–415. Springer Berlin / Heidelberg, 2008.
- [2] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In S. K. D’Mello, R. A. Calvo, and A. Olney, editors, *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013*, pages 4–11. International Educational Data Mining Society, 2013.
- [3] B. S. Bloom. Learning for mastery. *Evaluation Comment*, 1(2):1–12, 1968.
- [4] H. Cen, K. R. Koedinger, and B. Junker. Is Over Practice Necessary?—Improving Learning Efficiency with the Cognitive Tutor Through Educational Data Mining. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 511–518, Amsterdam, The Netherlands, 2007. IOS Press.
- [5] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.
- [6] A. Dhanani, S. Y. Lee, P. Pothilimthana, and Z. Pardos. A comparison of error metrics for learning model parameters in bayesian knowledge tracing. Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley, May 2014.
- [7] I. M. Goldin and A. Galyardt. Viz-r: Using recency to improve student and domain models. In *Proceedings of the 2nd ACM conference on Learning At Scale*, Vancouver, Canada, Mar. 2015.
- [8] J. P. González-Brenes. Modeling Skill Acquisition Over Time with Sequence and Topic Modeling. In G. Lebanon and S. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics AISTATS 2015*, pages 296–305, 2015.
- [9] J. P. González-Brenes and Y. Huang. Using data from real and simulated learners to evaluate adaptive tutoring systems. In *Proceedings of the Workshops at the 18th International Conference on Artificial Intelligence in Education AIED 2015*, Madrid, Spain, 2015.
- [10] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General Features in Knowledge Tracing: Applications to Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. In M. Mavrikis and B. M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining*, London, UK, 2014.
- [11] Y. Huang, J. P. González-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In J. G. Boticario, O. C. Santos, C. Romero, and M. Pechenizkiy, editors, *Proceedings of the 8th International Conference on Educational Data Mining*, Madrid, Spain, 2015.
- [12] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In K. Yacef, O. R. Zaïane, A. Hershkovitz, M. Yudelson, and J. C. Stamper, editors, *Proceedings of the 5th International Conference on Educational Data Mining*, pages 118–125, Chania, Greece, 2012.
- [13] C. Lin and E. Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51. Association for Computational Linguistics Morristown, NJ, USA, 2002.
- [14] Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popović. Towards automatic experimentation of educational knowledge. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3349–3358. ACM, 2014.
- [15] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics Morristown, NJ, USA, 2001.
- [16] Z. A. Pardos and M. V. Yudelson. Towards moment of learning accuracy. In *Simulated Learners Workshop of Artificial Intelligence in Education*, 2013.
- [17] P. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis—A New Alternative to Knowledge Tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 531–538. IOS Press, 2009.
- [18] R. Pelánek. A Brief Overview of Metrics for Evaluation of Student Models. In S. Gutierrez-Santos and O. C. Santos, editors, *Approaching Twenty Years of Knowledge Tracing Workshop of the 7th International Conference on Educational Data Mining*, London, UK, 2014.
- [19] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.
- [20] D. Rai, Y. Gong, and J. E. Beck. Using dirichlet priors to improve model parameter plausibility. In T. Barnes, M. Desmarais, C. Romero, and S. Ventura, editors, *Proceedings of the 2nd International Conference on Educational Data Mining*, Cordoba, Spain, 2009.
- [21] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, 5(2):1–10, 2013.
- [22] M. Walker, C. Kamm, and D. Litman. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3):363–377, 2001.

# Automated Session-Quality Assessment for Human Tutoring Based on Expert Ratings of Tutoring Success

Benjamin D. Nye<sup>\*</sup>  
The University of Memphis  
Memphis, TN 38152  
benjamin.nye@gmail.com

Donald M. Morrison  
The University of Memphis  
Memphis, TN 38152  
dmmrrson@memphis.edu

Borhan Samei  
The University of Memphis  
Memphis, TN 38152  
bsamei@memphis.edu

## ABSTRACT

Archived transcripts from tens of millions of online human tutoring sessions potentially contain important knowledge about how online tutors help, or fail to help, students learn. However, without ways of automatically analyzing these large corpora, any knowledge in this data will remain buried. One way to approach this issue is to train an estimator for the learning effectiveness of an online tutoring interaction. While significant work has been done on automated assessment of student responses and artifacts (e.g., essays), automated assessment has not traditionally automated assessments of human-to-human tutoring sessions. In this work, we trained a model for estimating tutoring session quality based on a corpus of 1438 online tutoring sessions rated by expert tutors. Each session was rated for evidence of learning (outcomes) and educational soundness (process). Session features for this model included dialog act classifications, mode classifications (e.g., Scaffolding), statistically distinctive subsequences of such classifications, dialog initiative (e.g., statements by tutor vs. student), and session length. The model correlated more highly with evidence of learning than educational soundness ratings, in part due to the greater difficulty of classifying tutoring modes. This model was then applied to a corpus of 242k online tutoring sessions, to examine the relationships between automated assessments and other available metadata (e.g., the tutor's self-assessment). On this large corpus, the automated assessments followed similar patterns as the expert rater's assessments, but with lower overall correlation strength. Based on the analyses presented, the assessment model for online tutoring sessions emulates the ratings of expert human tutors for session quality ratings with a reasonable degree of accuracy.

## Keywords

Automated Assessment, Tutoring Dialog, Dialog Acts, Dia-

<sup>\*</sup>Corresponding Author

log Modes, Natural Language Processing, Educational Data Mining

## 1. INTRODUCTION

As online learning has expanded, computer-mediated tutoring and help-seeking has become more prevalent and accessible. This tutoring occurs in a variety of forms, ranging from large commercial platforms employing certified teachers down to ad-hoc peer tutoring in rudimentary learning management systems (LMS). These systems generate a wealth of data about human tutoring interactions that can provide significant insights into the processes of online learning, the space of effective tutoring strategies, and the effectiveness of different platforms and contexts for tutoring. However, to study successful tutoring, tools are needed that can help distinguish between more and less successful sessions.

Quality ratings for tutoring sessions are often only available from self-reports by the tutor and student. However, these ratings have significant problems. Students typically have limited metacognitive skills and need training to assess their own learning [17]. Tutors can be more effective judges of learning, but a tutor's assessments of their students' learning can be biased and hard to compare due to these rating biases. Some of these biases may be individual variation (easy vs. hard raters), while others are systematic, such as less-expert tutors reporting higher average learning from their sessions. Other tutoring session sources have no real quality measure. For example, peer tutoring often lacks any assessment of the quality of the tutoring session, and hand-tagging these sessions for quality measures would be very time-consuming.

A standardized, automated estimator for the effectiveness of online tutoring sessions is arguably essential to the analysis of large-scale transcript corpora. Such a tool can be used to identify especially high-rated sessions, to track the results of improvement efforts, and to identify patterns in associated metadata. Also, differences between the automated estimator and tutors' self-reports could be used to identify new features that indicate effective tutoring strategies (i.e., an active learning approach). As such, the iterative improvement of a session success indicator would provide new insights into the features of effective tutoring and how they relate to other sets of data.

In this work, we have used a two-step supervised learning approach to train an estimator for session effectiveness. This

estimator was trained on a corpus of 1438 human-to-human tutoring sessions, where each session was rated in terms of two quality measures and each statement was annotated with a dialog act tag (e.g., *Confirmation:Positive*) and a dialog mode (e.g., *Scaffolding*). Based on the quality ratings assigned by independent expert tutors, features related to tutoring session success were identified using sequential pattern mining and statistical analysis of high-level session features (e.g., duration). Second, regression models that employed these features were trained to rate the quality of the tutoring sessions. Finally, this model was applied to a large sample of 246k tutoring sessions to examine the consistency of these ratings against metadata associated with each session, such as the original tutor’s rating of student learning and of the student’s knowledge of necessary prerequisites.

## 2. BACKGROUND AND RELATED WORK

Studying strategies and patterns in tutoring transcripts is a longstanding research area with roots in speech act theory [21]. Key techniques from this literature include dialog act classification [8], identifying dialog modes [1], and identifying statistically significant sequence patterns [3]. Our research described here relies on the use of all three levels of analysis to identify significant features that can be used to assess session quality. Dialog act classification involves binning each tutor or student statement into distinct taxonomy categories, which represent the functional purpose of the statement (e.g., an “Assertion” that states a fact). Dialog act taxonomy distinctions vary depending on the research focus, such as question types [8], higher-level dialog acts and feedback [1], and finer-grained pedagogical acts [3]. Our research extended this prior work in several ways, including a highly granular coding scheme, developed in collaboration with professional online tutors, which will be discussed later.

Dialog modes are a more recent area of focus for machine learning, but their theoretical underpinnings for studying learning are equally mature. In our work, modes represent shared understandings regarding hidden, higher-order dialog states with associated roles and expectations concerning the likelihood and appropriateness of particular dialog acts given that state [16]. In tutoring research, theoretically-based modes typically represent pedagogical strategies, such as Modeling, Scaffolding, and Fading. More recent studies of modes have used unsupervised approaches, such as Hidden Markov Models to detect patterns of dialog acts that match such theoretical modes [1]. However, such discovered states are not always guaranteed to be modes as we frame them here: others likely represent intermediate structures, such as adjacency pairs (e.g., a question followed by an answer). As such, in this research, we have relied on human-tagged modes and supervised mode-classifiers based on such modes, so that each mode can be linked more clearly to theoretical descriptions of pedagogy.

Finally, this research relies on features extracted using sequence data mining. A good review of prior work for sequence mining tutoring transcripts is presented by D’Mello and Graesser [3], which outlines conventional approaches (e.g., association rule mining) as well as a novel method based on transition likelihoods. In general, traditional analyses of tutoring sessions focus on identifying frequent or distinctive dialog act transitions and subsequences. However,

where supervised labels exist (e.g., quality tags), alternative sequence analysis techniques can be applied to identify sequences that occur more frequently in certain session types. This type of analysis detects distinctive subsequences, which discriminate between one group of sequences versus another group of sequences [5].

Since online human tutoring is a dyadic interaction, it also has similarities with computer-supported collaborative learning (CSCL). CSCL analysis often considers higher-level constructs related to collaboration, such as reaching consensus and division of tasks [13]. Many of these constructs are less central to a professional tutoring process, which has predefined roles (tutor vs. student) and associated cultural expectations for dialog behavior. However, aspects of these more general interactions were incorporated, such as dialog management (a “Process Negotiation” mode) and interpersonal relationships (a “Rapport Building” mode).

The quality of a tutoring session can be measured in two ways: “objective” assessments, such as tests given to the student [1], or “subjective” assessments, based on expert ratings or tags assigned to the session. However, even objective assessments require subjective decisions about their criteria. Additionally, expert raters can often provide higher granularity for tagging events during the tutoring process. As such, process-focused machine learning often focuses on building classifiers and estimators trained on expert tags and ratings [18]. Our research builds on this approach, so our automated assessments model how expert tutors *perceive* session quality rather than necessarily the resulting learning gains. In future work, we feel that there would be great value in contrasting a session quality assessment trained on tested learning gains against the one developed in this paper. Such an assessment might identify session features that help identify when illusions of mastery and other rating biases occur [6].

## 3. DATA SET

This research analyzes a full data set of 246k online human-to-human tutoring transcripts from a major commercial tutoring service (Tutor.com). Thousands of different tutors, and tens of thousands of different students participated in these sessions, but all focused on Algebra and Physics topics. As an on-demand service, each session was initiated by a student who requested help on a problem or concept (e.g., at an impasse). Of these transcripts, approximately 4k were excluded from analyses on the full data set due to missing data or formatting issues. Each session contained a timestamped line-by-line text transcript of the statements typed by the student, the tutor, and system messages (e.g., file uploads). Every session was also associated with metadata collected before and after the session. This metadata included the tutor’s assessment of evidence of learning during the session (EL1) and the tutor’s assessment of the student’s level of prerequisite knowledge (PREREQ). Metadata was also available for a subset of tutors, which included their “Tutor Level,” an internal performance level that ranged from “Probationary” (0) to “Level III” (Highest). The tutor level was determined by each tutor’s mentor, based on internal reviews of the tutor’s sessions, and is correlated with experience. On average, Level III tutors had five years experience, Level II had two to four years, and Level I had a little over

a year. Probationary tutors averaged 6 months.

Of the total set of transcripts, 1438 sessions were annotated by a panel of 19 subject matter experts (SMEs), selected from a pool of some 2,800 Tutor.com tutors using a rigorous screening process, which included analysis of answers to a set of survey questions designed to gather initial expert opinion about tutoring, and also to assess the respondents' ability to critique session transcripts. The training process and details on inter-rater reliability are described in more detail in related work [15]. As part of the annotation process, the SMEs rated each session on two scales: evidence of learning (EL2) and educational soundness (ES). Annotators were instructed to consider different criteria for each: EL2 targets outcomes (i.e., did the student learn) and ES targets process (i.e., did the tutor use good tutoring strategies). This is important because sometimes good tutoring steps can still fail to produce learning for a given student. EL1, EL2, ES, and PREREQ were all rated on a 0-5 scale, where zero represents a low rating and five represents a top rating.

Each line in the tutoring session was also tagged for a dialog act and was also part of a dialog mode. Given the size of the taxonomies (126 dialog acts and 16 dialog modes), a full review of each tag would be infeasible, so specific tags that showed value as features will be noted as they are discussed. The taxonomy of dialog acts included 126 distinct tags, organized into 15 main categories. At a macro-level, these categories focus on traditional dialog act classes such as Questions, Assertions, Requests, Directives, and Expressives [21]. Within the tutoring context, these categories tend to be used to provide information (Answer, Assertion, Clarification, Confirmation, Correction, Expressive, Explanation, Reminder), asking for information (Hint, Prompt, Question), and managing the tutoring process (Directive, Promise, Request, Suggestion). Within each of the 15 main categories, subtypes capture key differences such as positive versus negative feedback (e.g., *Expressive:Positive* vs. *Expressive:Negative*).

Annotators also tagged student or tutor contributions that signaled the start of a dialog mode, or a switch from one dialog mode into another. The 16 included modes associated with classic tutoring strategies (Fading, Modeling, Scaffolding, Sensemaking, Session Summary, Telling), identifying the problem (Method Identification, Problem Identification) or learner prerequisites (Assessment), interpersonal strategies (Metacognitive Support, Rapport Building), and session process (Process Negotiation, Opening, Closing, Method Road Map, Off Topic). The time spent in each mode was far from uniform. Tutoring strategy modes, particularly Scaffolding, accounted for a majority of most sessions. Session process modes were also significant, such as Process Negotiation (i.e., getting on the same page), Openings, and Closings. Other modes were fairly rare, such as Method Identification.

Based on these annotated tags, complementary research on this data set developed a logistic regression dialog act classifier [20] and a conditional-random fields (CRF [11]) mode classifier [19]. This tagging methodology followed similar principles to Moldovan et al. [14]. These classifiers ap-

**Table 1: Reliability Scores for Tagging**

Tagger	Main Act		Sub-Act		Mode	
	Acc	Kappa	Acc	Kappa	Acc	Kappa
Human	81%	0.77	65%	0.63	56%	0.47
Machine	77%	0.71	53%	0.50	57% (43%)	0.52 (0.21)

proached the level of reliability shown by independent tagging by human experts, as noted in Table 3. The figures in this table show the best performance by both the human taggers (i.e., their final inter-rater reliability tests) and the performance of the classifiers used for automated tagging in this paper. Machine tagging statistics shows cross-validation results. As can be observed, the classification of the main dialog acts (15 categories) and full set of sub-acts (126 categories) approximated human inter-rater tagging fairly closely. Classifying modes was fairly effective also, but lost nearly half of its accuracy the tagger trained on human speech act tags was applied to the machine-labeled dialog acts (29% accuracy). Retraining on machine tags before testing on machine tags improved overall accuracy, but still produced a significantly lower kappa (43% and 0.21, respectively, as shown in parentheses), as compared to training and testing on human tags. As such, mode tags will be less accurate for machine-tagged sessions.

From the standpoint of analysis, the 1,438 human-tagged training set was used for initial feature identification and training of the session quality assessment model. The full set of 242k machine-tagged sessions were then treated as a second data sample for analysis, which included the original training set but tagged using the automated dialog act and mode classifier models. This research builds on the prior research that developed dialog act classifiers [20] and mode classifiers [19], as well as development of a taxonomy for speech acts and modes in human tutoring [15]. The novel contributions reported in this paper include identifying patterns in speech acts and modes (subsequence analysis), identifying features that help estimate tutoring session quality, training machine learning models that estimate tutoring session quality, examining the strength of features in these models, and examining the correlation between estimated session quality against other indicators of session quality (e.g., the original tutor's rating of learning during the session). This work was done to target the research questions described in the following section.

## 4. RESEARCH METHODOLOGY

Based on these data sets, this work approaches five primary research questions:

1. How closely can we model expert judgments about session quality, based on domain-independent dialog acts and modes?
2. What models show the most promise for assessing session quality?
3. What features are the strongest predictors in these models?
4. What features lose predictive power when trained on machine tags rather than human tags?

5. How closely do the results from machine quality tags correlate with metadata on the full corpus (e.g., EL1), as compared to the training corpus?

To examine these questions, a session quality classifier was trained using a two-step process of feature selection followed by supervised learning. First a set of high-level features was selected that correlated with the rater's evidence of learning (EL2) and educational soundness (ES). These features included the duration of the session, the average number of words typed by the student per contribution (verbosity), the number of dialog acts typed by the tutor and by the student, and the number of short and long pauses between dialog acts. Additionally, the counts of each mode tag and of each individual dialog act by a given speaker were used as features (e.g., *Confirmation:Positive [Tutor]*).

Next, to capture more complex features of the tutoring process, sequence pattern mining was applied to tutoring sessions to identify subsequences of dialog acts or dialog modes that help distinguish between excellent and poor tutoring sessions. For this analysis, two subsets of human-annotated tutoring sessions were selected that included the most successful sessions ( $N=261$ , where  $ES = 5$  and  $EL2 = 5$ ) and the least successful sessions ( $N=93$ , where  $ES \leq 2$  and  $EL2 \leq 2$ ). Subsequences of dialog modes consider dialog mode switches, where there was a change from one mode to another. This is important because modes often span multiple dialog acts.

The subsequence analysis used the TraMiner package for sequence analysis [5], which contains an algorithm for detecting discriminant event subsequences between two groups of sequences. At a high level, this algorithm calculates the frequency of all subsequences up to a given length for each group of sequences, then applies a Chi-squared test (Bonferroni-adjusted) to identify subsequences that are statistically more (or less) frequent in each group. In this context, a subsequence must be distinguished from a substring: subsequences are ordered, but do not necessarily have to be contiguous. Three sets of distinctive subsequence analyses were performed: 1) dialog act subsequences, 2) mode subsequences, and 3) dialog acts within each type of mode. Any subsequence which was distinctive at the  $p < 0.4$  level was included as a candidate feature. The  $p < 0.4$  cutoff was selected to allow a large set of candidate features, while still likely performing better than chance. This analysis was performed on the human-annotated tags. Each subsequence was treated as a feature whose incidence would be counted within a session (i.e., a count of the number of times that tags occurred in that order, without reusing any tags).

Four algorithms were trained to estimate the average of ES and EL2 based on the full feature set: linear regression with feature selection, support vector machine (SVM) regression [10], and additive regression based on decision stumps [4]. In general, these algorithms were selected and tuned to try to avoid over-fitting: the final number of active candidate features was 1465, which was comparable to the number of training sessions (1438). Ridge regression reduces the number of parameters by penalizing additional factors. Support Vector Machines are resistant to overfitting because they regularize the space solution space. Additive regression

(also called Stochastic Gradient Boosting) uses smoothing that reduces the impact of each additional factor. Each algorithm was evaluated using 10-fold cross validation, using Weka [9]. After evaluating the effectiveness of each algorithm on the human-annotated data, the best of these algorithms was then tested on the machine-tagged sessions to examine performance. The best algorithm was re-trained using machine-tagged sessions, to test if calibrating to the dialog act and mode classifier outputs would improve performance.

Finally, the full set of 242k tutoring sessions was tagged using the best-fit model for session quality. These quality tags were correlated against session metadata available for the larger corpus of sessions: the original tutor's evidence of learning (EL1), the original tutor's assessment of the student's prerequisite knowledge (PREREQ), and the level of the tutor (Tutor Level). These correlations were compared against the correlations observed between the automated assessments and these same metadata variables for the training set. The goal of this analysis was to examine the consistency of the automated assessment with other ratings of session quality that were available for all tutoring sessions.

## 5. RESULTS AND DISCUSSION

The results from each step are discussed in this section, including sequence mining for session features, training and evaluating the session assessment model, and applying this model to a large corpus of online tutoring session transcripts. For the sake of brevity, dialog acts in this section are displayed using the shorthand form  $\langle \text{Main Dialog Category} \rangle : \langle \text{Sub Act} \rangle [\langle \text{Speaker} \rangle]$ , such that *Expr:Praise [T]* means "expressive praise from the tutor."

### 5.1 Sequence Pattern Mining

Discriminate sequence analysis that compared the most successful and least successful tutoring sessions identified 1151 better-than-chance ( $p < 0.4$ ) distinctive subsequences from 2 to 7 elements long. The majority of these sequences were sequences of dialog acts (1062) and a significant number of these sequences captured variations on similar patterns. Due to the granularity of the taxonomy, distinctions occurred such as *Assertion:Calculation [S]*  $\Rightarrow$  *Expressive:Confirmation:Positive [T]* versus *Assertion:Calculation [S]*  $\Rightarrow$  *Confirmation:Positive [T]*, where the only difference was whether the tutor's feedback took the form of an Expressive. Moreover, such distinctions sometimes showed slightly higher distinctiveness. For example, in the above case, *Expressive:Confirmation:Positive* feedback (e.g., "Great!") was a stronger indicator of session success than *Confirmation:Positive* (e.g., "Right").

A total of 89 distinctive mode subsequences were identified as candidate features that distinguished between session quality. Many of these were variants of eight patterns that were supported by Bonferroni-adjusted Chi-squared tests at the  $p < 0.05$  level. Six of these patterns were indicators of positive sessions. 1) Successful sessions almost always ended with a Closing/WrapUp, suggesting that both the tutor and student are satisfied with the progress. 2) Successful sessions had more Fading. The existence of even one Fading segment was an indicator of success, though Scaffolding preceding Fading was a better indicator; 3) Successful ses-

sions tended to have repeated Scaffolding or Sensemaking segments (the conceptual equivalent of Scaffolding), where Scaffolding was interleaved with other modes. 4) Successful sessions were more likely to have late-session Rapport Building is after Scaffolding or Fading, but preceding the Closing. 5) A Telling mode (i.e., mini-lecture) before Rapport Building was also a positive feature, which likely indicates that a summary is positive. 6) The presence of a single Opening mode was also an indicator of a good session, where less-successful sessions skipped the Opening greetings and moved immediately to Problem Identification.

Two patterns of mode subsequences tended to be associated with less successful tutoring sessions. 1) Unsuccessful sessions tended to have repeated Modeling mode cycles. While a single Modeling mode segment was not indicative of a poor session, two or more in series was associated with worse ratings. 2) Unsuccessful sessions were also indicated by repeated Process Negotiation, particularly if Process Negotiation alternated with Modeling (the tutor solving the problem) or Problem Identification (figuring out what problem the student has). It was also a negative indicator when Process Negotiation started early in a session sequence. Process Negotiation is a mode that is associated with discussing the tutoring process itself, which includes figuring out who should be speaking or addressing technical issues. Process Negotiation itself was not a bad mode, and was also present in many good characteristic sequences. In these good sequences, it tends to occur late in the session (preceding a Closing) rather than early-on. In general, long or early cycles of Process Negotiation likely indicate that the student is unable to contribute meaningfully to the problem due to lack of prerequisites, technical issues, or poor dialog coordination (e.g., student interrupting).

From aligning these distinctive subsequences, an ideal path of modes for a session might be framed as: Opening  $\Rightarrow$  ProblemID  $\Rightarrow$  Scaffolding  $\Rightarrow$  Fading  $\Rightarrow$  ProcessNegotiation  $\Rightarrow$  Telling  $\Rightarrow$  RapportBuilding  $\Rightarrow$  Closing, where some modes (e.g., Scaffolding and Fading) optimally alternate multiple times. This successful mode sequence shows some similarities and differences when compared to Graesser et al.'s 5-step frame for in-person tutoring, which can be described as: [Tutor poses a question]  $\Rightarrow$  [Student attempts to answer]  $\Rightarrow$  [Tutor provides brief feedback]  $\Rightarrow$  [Collaborative interaction]  $\Rightarrow$  [Tutor checks if student understands] [7]. The final two frames align well with Scaffolding  $\Rightarrow$  Fading  $\Rightarrow$  ProcessNegotiation pattern observed in the successful online sessions. The main differences likely stem from the tutoring context. The Graesser tutoring frame assumes a tutor-driven process in which the student is attempting to answer a question, typically conceptual, posed by the tutor. In our data, the student is typically coming to the tutor for help on a specific problem, and the session is in this sense student-driven. As such, Problem Identification occurs first, instead of the tutor posing an initial question.

The insights from the dialog act sequences for successful versus less successful sessions show similar patterns as those based on sequences of modes. However, they are more granular and some of the distinctive sequences tend to be longer or repeating (e.g., repeated answers by a student alternating with *Confirmation:Positive* by the tutor are better).

These patterns match loosely to the learning-relevant affective states noted by D'Mello and Graesser [2], which were: Achievement, Engagement, Disengagement, Confusion / Uncertainty, and Frustration. Evidence of achievement (i.e., answers that received positive feedback, explanations followed by expressions of understanding) corresponded with higher session ratings. Likewise, engagement (student answer attempts and sequences with multiple student statements) were positive.

Disengagement indicators, such as questions followed by *Expressive:LineCheck* (e.g., "Are you there?") and *Expressive:Neutral* statements by the student (e.g., "ok") were associated with lower ratings. Raters likely interpreted neutral responses as indicating that the learner was passively processing the session. By comparison, tutor questions that transitioned to *Confirmation:Understanding:Negative* (e.g., "No, I don't understand") were not strong indicators of an unsuccessful session. Frustration was not significantly observed in the corpus, in part due to a lack of taxonomy tags devoted to detecting it and in part due to a relatively low prevalence of obvious frustration within the training corpus. While taxonomy acts for confusion and uncertainty were available in the taxonomy, these were less common and did not have a clear correlation to successful or unsuccessful sessions. This is somewhat expected, since a limited amount of confusion tends to be productive [2], but a large amount can lead to unproductive frustration. More nuanced techniques might be needed to monitor these cycles in tutoring sessions.

## 5.2 Automated Assessment Models

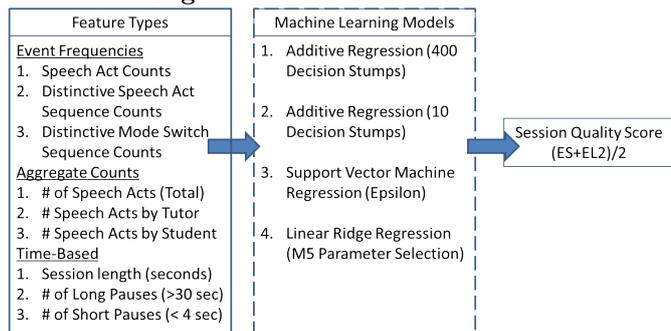
The total feature set was used to train a series of machine-learning models: linear ridge regression with parameter selection (Linear), SVM regression (SVM), and additive regression with decision stumps (Add.). The outcome variable for this training was a unified quality score based on the average of the rater's assessment of educational soundness (ES) and evidence of learning (EL2). The process for training these models is outlined in Figure 1. The results of 10-fold cross-validation for the best-fit models are presented in Table 5.2, in terms of the correlations between the machine-generated tags and the hold-out folds. Additive regression outperformed the other models, even with a fairly small number of decision branches (10). However, it improved significantly when allowed to use additional decisions (400). From examining the decision stumps, these additional stumps allowed it to incorporate additional factors and also form piecewise curves for some of the strongest factors.

**Table 2: Regression Fits for (ES+EL2)/2 (10-fold CV)**

	Linear	SVM	Add. (10)	Add. (400)
Human Tags	0.24	0.55	0.62	0.69
Machine Tags	0.24	0.49	0.52	0.56

The linear model performed very badly, despite parameter selection: it tended to overfit the data and did not seem to model the expert ratings very well. SVM performed slightly better, but was not the best model overall. The Additive model, which was based on decision thresholds, worked best

Figure 1: Model Data Flow



out of the three. This may indicate that the human raters tended to implicitly use heuristics such as “too many Modeling modes,” or “not enough Student contributions.” The nature of features was also a factor, since many features were relatively sparse in each session (e.g., only occurred once or twice within an average session), which lends itself to rules related to the existence of a feature (i.e.,  $N > 0$ ).

Models trained on the machine-generated tags followed a similar pattern, but with slightly worse estimates. Retraining the classifiers on the machine-labeled tags did not significantly improve estimates based on those tags. When applying the model trained on human tags to the training set with machine tags, the model fit is  $R=0.54$ , as compared to  $R=0.56$  for the cross-validated model built on the machine tags. As such, the machine tags appear to lose certain information, rather than simply categorizing it differently.

Since the smallest Additive Regression model worked so effectively, it is worthwhile to examine the features that were included. These models differed slightly when trained on human tags versus machine-labeled tags. The top features for this model on human tags vs. machine-labeled tags are shown in Table 5.2, in order of their importance (note: *Confirmation* is shortened to *Conf*). The presented analysis used non-standardized data, which is reasonable partly because the length of Tutor.com sessions tends to be fairly regular (i.e., a typical session is 15-25 minutes). Normalization would likely be needed to apply this to significantly different corpora. In general, many of the same patterns are important for both the human and machine tagged models. At least some of the judgments are based on a required minimal session length (e.g., # of Tutor Acts). Certain features appear to target evidence of learning (EL2), such as tutor actions that indicate the student has provided correct answers (*Confirmation:Positive*, *Expr:Praise*) and not passive in the tutoring session (*Expr:Neutral*, *Expr:LineCheck*). Other features appear to be associated with educational soundness (ES) for tutoring process (e.g., existence of a Closing, Scaffolding, and no excessive Modeling). Machine tagging appears to lose some of these nuances with respect to modes, probably due to the significantly lower accuracy for classifying modes.

Overall, the model appears to capture evidence of learning (EL2) better than educational soundness (ES). When trained on the full training data set (human tags), the Ad-

Table 3: Top-10 Features in Additive Regression

Trained on Human Tags	Trained on Machine Tags
Closing > 0	# of Tutor Acts > 11
<i>Expr:Conf:Positive [T]</i> ⇒ <i>Expr:Conf: Positive [T] &gt; 0</i>	RapportBuild ⇒ Closing > 0
Scaffolding > 0	<i>Expr:Conf:Positive [T]</i> ⇒ <i>Expr:Conf: Positive [T] &gt; 0</i>
Closing > 0	<i>Assertion:Concept [T] &lt; 18</i>
<i>Expr:Apology [T] = 0</i>	# of Tutor Acts < 12
# of Tutor Acts > 6	# of Tutor Acts > 5
ProcessNegotiation ⇒ Modeling ⇒ Modeling < 4	<i>Request:Conf: Understanding [S] &lt; 3</i>
<i>Expr:Praise [T] &gt; 0</i>	Scaffolding ⇒ Scaffolding ⇒ Closing > 4
<i>Expr:LineCheck [T] = 0</i>	# of Tutor Acts < 12
<i>Expr:Neutral [S] &gt; 15</i>	<i>Expr:Conf:Positive [S] &gt; 1</i>

ditive Regression (400) correlates with the average of ES and EL2 at  $R=0.8$ . By comparison, the correlation to these estimates is  $R=0.76$  for EL2 versus  $R=0.63$  for ES. Clearly, this is not the result of the outcome variable itself, which is a straight average of the two ratings ( $R=0.93$  with EL2 and  $R=0.92$  with ES). Instead, this indicates that the features for evidence of learning are more easily detected using the available taxonomy tags and features. This limitation was amplified when using the machine-generated tags, where the fit to  $(ES+EL2)/2$  was  $R=0.54$  but the correlation with the components was  $R=0.55$  for ES2 and  $R=0.38$  for ES. As such, improving the automated tagging of dialog modes would improve the automated assessments significantly.

### 5.3 Tagging Large Tutoring Data Set

To examine the consistency of this assessment model on out of sample data, it was applied to a corpus of 242k machine-tagged sessions. The features for each tutoring session were extracted from parsing the transcript. Metadata about the session and the tutor were collected and aligned to the automated session assessments for analysis. The correlations between the Automated Estimates (Estimates), EL1, and PREREQ were available for almost the full corpus of 242k sessions. Other metadata was not always complete (e.g., not all tutor level data was available), so each pairwise correlation may have a slightly different N. However, all comparisons involve thousands of values and are statistically significant at the  $p < 0.01$  level.

Table 4: Correlations of Quality Scores with Session Metadata

	Estimate	(ES+EL)/2	EL1	PREREQ
(ES+EL)/2	0.54	-	-	-
EL1	0.45	0.56	-	-
PREREQ	0.39	0.49	0.87	-
Tutor Level	0.05	0.11	-0.02	-0.04

Table 5.3 shows the correlations between the automated estimate of session quality (Estimate), the average quality score for human raters  $(ES+EL2)/2$  (available for the training set only), the original tutor’s ratings for evidence of learning (EL1) and the learner’s prerequisite knowledge (PREREQ), and the Tutor Level. The first two columns of this ta-

ble show that the estimate maintains similar correlations to those for the ratings that it was based on, across the larger data set, but slightly weaker overall. For example, the session tutor's rating of learning for the student correlates at  $R=0.56$  ( $N=1438$ ) for the training tags, but only  $R=0.45$  ( $N=242k$ ) for the automated tags across the full session data. With that said, the automated session rating maintains a similar pattern as the supervised tags across the full corpus. This indicates that the automated assessment captures significant information from the original expert raters, but with additional noise due to the machine-tagging process (particularly for modes).

This table also indicates why an external rating source can be important for evaluating the quality of tutoring sessions, even for well-trained professional tutors. Despite being rated independently by tutors with no knowledge of the original tutor, a higher Tutor Level correlated with significantly higher external quality ratings ( $R=0.11$ ,  $N=1328$ ). However, these more-expert tutors rated both the learning ( $R=-0.02$ ) and the prerequisite knowledge ( $R=-0.04$ ) lower than lower-level tutors. Or, put another way, less-expert tutors probably over-estimate both the learning and initial understanding of their students.

Moreover, it may be difficult for session tutors to provide ratings for the session that capture distinct features. For example, the original tutors expressed an  $R=0.87$  ( $N=242k$ ) correlation between learning (EL1) and prerequisite knowledge (PREREQ). While one would expect these factors to be related, that level of correlation is nearly identical. By comparison, the external quality ratings correlated with the PREREQ assessments much more loosely ( $R=0.49$ ,  $N=1438$ ) and the automated assessments shadow this pattern ( $R=0.39$ ,  $N=242k$ ). So then, this automated rater provides a unique source of information modeled after the judgments of the external raters, which can be complementary to other sources of information about tutoring session quality.

## 6. CONCLUSIONS AND FUTURE WORK

This research has offered some insights into the five primary research questions posed earlier in Section 4. First, this work demonstrates the feasibility of an automated assessment model that models human expert judgments about the learning that took place during an online human-to-human tutoring session, at a level of  $R=0.54$ . While room for improvement exists, this model is already functionally useful. At least in this work, non-linear meta-models based on decision stumps (e.g., Additive Regression) outperformed more linear approaches such as Linear Regression and SVM Regression. This finding indicates that Random Forests [12] and similar algorithms are probably also promising for this type of problem. The strongest predictors of session quality in these models tended to be features where the tutor confirmed the accuracy of the student's responses, the session process indicated that progress was occurring (e.g., Scaffolding, Fading), or a consensus about successful learning was reached (i.e., a mutually-agreed Closing). Of these features, modes were fragile when machine tags were used: the level of noise in the mode classification appears to wash out information that is needed to evaluate the tutoring process. Finally, the resulting model was shown to follow similar patterns to the original training ratings, even over a much larger data

set. This indicates that the automated assessments offer a reasonable proxy for expert human assessment when needed.

Notably, these ratings are calculated without a domain model that can directly assess the quality of students' answers. Instead, the model captures more general features of the tutoring interaction that relate to engagement and consensus between the tutor and student about learning accomplishment. As such, this model should be effective across a variety of tutoring domains beyond those analyzed in this work (Algebra and Physics). These session features are, in principle, domain-independent: they are based on classifications of tutoring dialog acts and modes.

However, this is also a limitation. Since the automated assessment system lacks the ability to assess the correctness of student input, it relies significantly on the session tutor's domain knowledge and basic capabilities to provide correctness feedback. As such, the session assessments can detect aspects of the pedagogy and student progress, but are unlikely to work appropriately if the tutors are entirely unqualified. This is, in part, because the training corpus includes only professional tutors who are rated and evaluated for quality. As such, additional quality-rated corpora might be needed to transition this estimator to other tutoring contexts where session quality assessments are important (e.g., peer-tutoring).

Additionally, significant drops in performance were observed when using machine-annotated sessions instead of human-annotated sessions. These drops were particularly severe for mode classifications, which had a direct impact on the ability of the session quality estimates to model the educational soundness of a session. This functionality would be helpful, as it allows credit for "good process" even when strong learning outcomes are not observed. Improving the accuracy of dialog mode classification would significantly strengthen the assessment of tutoring sessions, and is an important area for further research. One way to approach this problem would be to use active learning where machine-annotated transcripts are corrected by human taggers.

Finally, an important next direction for this research would be to train a similar tutoring session assessment model based on pre-test and post-test assessments, such as the approach taken by Boyer et al. [1]. This step would enable a comparison between the features underlying our expert ratings of session quality against the features associated with measured learning gains. This work may show notable qualitative differences related to not only the key features, but also the algorithms involved (e.g., discontinuous algorithms such as Additive Regression might not be as dominant). Features associated with learning gains that are not associated with human ratings might also help detect illusions of mastery or expert blind spots. Likewise, integrating both approaches for analysis of tutoring sessions would offer the potential to identify authentic "Eureka moments" where the learner's sense of sudden understanding can be shown to correlate with subsequent performance on a similar problem. In the long term, the process of maintaining and improving this model should provide insights into new features of successful tutoring that may even be more valuable than the automated assessments calculated by the model.

## 7. REFERENCES

- [1] K. E. Boyer, R. Phillips, A. Ingram, E. Y. Ha, M. Wallis, M. Vouk, and J. Leste. Investigating the relationship between dialogue structure and tutoring effectiveness: a hidden markov modeling approach. *International Journal of AI in Education*, 21(1-2):65–81, Jan. 2011.
- [2] S. D’Mello and A. Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157, 2012.
- [3] S. D’Mello, A. Olney, and N. Person. Mining collaborative patterns in tutorial dialogues, Dec. 2010.
- [4] J. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378, 2002.
- [5] A. Gabadinho, G. Ritschard, N. S. Muller, and M. Studer. Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software*, 40(4):1–37, 2011.
- [6] A. Graesser, S. D’Mello, and W. Cade. Instruction based on tutoring. In *Handbook of Research on Learning*, pages 408–426. 2011.
- [7] A. Graesser and N. K. Person. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9(6):495—522, 1995.
- [8] A. C. Graesser and N. K. Person. Question asking during tutoring. *American Educational Research Journal*, 31(1):104–137, Jan. 1994.
- [9] M. Hall, E. Frank, and G. Holmes. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [10] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18—28, 1998.
- [11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001.
- [12] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.
- [13] A. Meier, H. Spada, and N. Rummel. A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning*, 2(1):63–86, Feb. 2007.
- [14] C. Moldovan, V. Rus, and A. Graesser. Automated speech act classification for online chat. In *Midwest Artificial Intelligence and Cognitive Science Conference*, pages 23–29, 2011.
- [15] D. Morrison, B. D. Nye, and V. Rus. Tutorial dialogue modes in a large corpus of online tutoring transcripts. In *Artificial Intelligence in Education (AIED) 2015*, Under review.
- [16] D. Morrison and V. Rus. Defining the nature of human pedagogical interaction. In R. A. Sottolare, X. Hu, H. Holden, and K. Brawner, editors, *Generalized Intelligent Framework for Tutoring Systems, Volume 2: Pedagogical Strategies*, pages 217–224. 2014.
- [17] I. Roll, V. Aleven, B. M. McLaren, and K. R. Koedinger. Metacognitive practice makes perfect: Improving students’ self-assessment skills with an intelligent tutoring system. In A. Biswas, G and Bull, S and Kay, J and Mitrovic, editor, *AIED 2011*, volume 6738 of *LNAI*, pages 288–295, 2011.
- [18] C. Rosé, Y.-C. Wang, Y. Cui, J. Arguello, K. Stegmann, A. Weinberger, and F. Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning*, 3(3):237–271, Jan. 2008.
- [19] V. Rus and N. Niraula. Automated labeling of dialogue modes in tutorial dialogues,. Technical report, The Language and Information Processing Lab, The University of Memphis, 2014.
- [20] B. Samei, V. Rus, B. D. Nye, and D. M. Morrison. Hierarchical dialogue act classification in online tutoring sessions. In *Educational Data Mining (EDM) 2015*, In Press.
- [21] J. Searle, F. Kiefer, and M. Bierwisch. *Speech act theory and pragmatics*. 1980.

# A Framework for Multifaceted Evaluation of Student Models

Yun Huang  
Intelligent Systems Program  
University of Pittsburgh  
Pittsburgh, PA, USA  
yuh43@pitt.edu

José P. González-Brenes  
Pearson Research &  
Innovation Network  
Philadelphia, PA, USA  
jose.gonzalez-  
brenes@pearson.com

Rohit Kumar  
Speech, Language and  
Multimedia  
Raytheon BBN Technologies  
Cambridge, MA, USA  
rkumar@bbn.com

Peter Brusilovsky  
Intelligent Systems Program  
University of Pittsburgh  
Pittsburgh, PA, USA  
peterb@pitt.edu

## ABSTRACT

Latent variable models, such as the popular Knowledge Tracing method, are often used to enable adaptive tutoring systems to personalize education. However, finding optimal model parameters is usually a difficult non-convex optimization problem when considering latent variable models. Prior work has reported that latent variable models obtained from educational data vary in their predictive performance, plausibility, and consistency. Unfortunately, there are still no unified quantitative measurements of these properties. This paper suggests a general unified framework (that we call Polygon) for multifaceted evaluation of student models. The framework takes all three dimensions mentioned above into consideration and offers novel metrics for the quantitative comparison of different student models. These properties affect the effectiveness of the tutoring experience in a way that traditional predictive performance metrics fall short. The present work demonstrates our methodology of comparing Knowledge Tracing with a recent model called Feature-Aware Student Knowledge Tracing (FAST) on datasets from different tutoring systems. Our analysis suggests that FAST generally improves on Knowledge Tracing along all dimensions studied.

## Keywords

Student Modeling, Knowledge Tracing, parameter estimation, Identifiability, Model Degeneracy

## 1. INTRODUCTION

Adaptive tutoring systems often rely on student models to trace the progress of student knowledge to personalize instruction. Such student models are usually latent variable models with the state of student knowledge as the latent variable. However, finding optimal model parameters is usually a difficult non-convex optimization problem for latent variable models. Moreover, in the context of tutoring systems, even global optimum model parameters may not be interpretable (or plausible). Knowledge Tracing [4] is one such latent variable model that has been widely used, and different properties of its estimated parameters have been presented in many previous studies: predictive performance [6], plausibility [1, 6, 19], and consistency [2, 6, 16, 19, 9]. Unfortunately, there are still no unified quantitative measurements of these properties. If prediction of student performance is our only goal, this need is less urgent, since we can simply pick a model according to classification metrics. However, parameters with varying properties might have different inferences about knowledge, which may result in different tutoring decisions that can have a large impact on students. To illustrate, we show examples where two models that both belong to Knowledge Tracing are fitted from the same data, and where predictive performance is not sufficient to pick a good model:

- One model with higher predictive performance asserts that student knowledge decreases with correct practices, while the other model asserts the opposite. In such cases, the former model will suggest continuing practicing even if students get a lot of correct answers in a row, while the latter will suggest moving to other skills in a shorter amount of time.
- Two models have the same predictive performance, yet one asserts that about 20 practices are required to reach mastery of a skill, while the other asserts that only about 3 practices are enough. In such cases, a student needs to practice a lot under the former model, but under the latter model, students can move to learning other skills more quickly.

In the first example, the more predictive model lacks plausibility; in the second example, two models lack consistency, even though they have the same predictive performance. As a result, we advocate that a student model should be examined from dimensions besides predictive performance. We propose a unified quantitative framework, called Polygon, for the multifaceted evaluation and comparison of student models. The framework suggests novel metrics to quantify the properties of a student model along multiple dimensions, including predictive performance, plausibility, and consistency. Polygon is designed for general latent variable models that model latent student knowledge and is domain-independent. In the present work, we demonstrate how we apply Polygon to evaluate and compare classic Knowledge Tracing with a recent generalized model called Feature-Aware Student Knowledge Tracing (FAST) [8] in four different domains. Section 2 reviews some latent variable student models and prior work examining their properties; Section 3 describes our Polygon framework and metrics; Section 4 studies the relationship among these metrics and compares Knowledge Tracing with FAST; Section 5 concludes the work.

## 2. BACKGROUND

### 2.1 Latent Variable Student Models

We now review two effective latent variable models for predicting student performance and inferring student knowledge: Knowledge Tracing [4] and Feature-Aware Student Knowledge Tracing (FAST) [8]. Knowledge Tracing uses Hidden Markov Models to model student knowledge as binary latent variables (either learned or unlearned), given the observed practice performance (correct or incorrect) and using four parameters: Init (initial knowledge level), Learn (learning rate), Guess, and Slip. We learn the parameters of Knowledge Tracing using the Expectation Maximization algorithm. A recent model FAST incorporates features into Knowledge Tracing by replacing the binomial distributions by logistic regression distributions. It encodes contextual information as features for the original Knowledge Tracing parameters. It allows flexible features to affect student performance or knowledge directly. For simplicity, we use features in all four parameters in the study. FAST trains feature coefficients jointly with other parameters using the Expectation Maximization with Features algorithm [3]. This algorithm keeps the original E-step and replaces the M-step by training a weighted regularized logistic regression using a gradient-based search algorithm (LBFGS). While FAST has been shown to outperform Knowledge Tracing in many prediction tasks, we are interested in comparing it with Knowledge Tracing in other dimensions.

### 2.2 Prior Work Examining Properties of Knowledge Tracing

Prior work has examined Knowledge Tracing models from predictive performance, plausibility, and consistency. We now review previous studies in each dimension.

*Predictive Performance.* Measurements of predictive performance have been broadly applied to evaluate student models. Prior studies have shown several problems with parameter estimation for Knowledge Tracing, which predictive performance metrics often fail to detect [2, 16, 7]. We

examine this traditional dimension in more depth for both Knowledge Tracing and FAST, and complement it in other dimensions, including plausibility and consistency.

*Plausibility.* Interpretability of a model is a desired property because it allows for better scientific claims and practical applications. Prior studies have used external measurements for validating the plausibility of fitted parameters, such as pre-test scores [6], exercise scores [4], or some domain-specific measurements [2]. However, such external resources are not always available. Many studies also examined plausibility by internal validity. Learning curves plotted using fitted parameters are inspected [2], and extremely low learning rates are considered implausible. However, very difficult skills can have very low learning rates, and it is not clear what is the suitable threshold for defining low learning rates. Implausibility has been formally defined using model degeneracy [1], which refers to situations where parameter values violate the model’s conceptual meaning. They defined strong empirical constraints to detect theoretical degeneracy, and designed two specific metrics involving empirical parameters to detect empirical degeneracy: (i) the model’s estimated probability that a student knows a skill is not higher than before the student’s first  $N$  actions, or (ii) the model doesn’t assess that the student has mastered the skill, even though the student has made a large number  $M$  of correct responses in a row. Under these two cases, the model is judged to be empirically degenerate. They arbitrarily chose  $N=3$  and  $M=10$  for the study. A later theoretical fixed point analysis [19] has precisely identified the conditions where models will be empirically degenerate. We are interested in generally quantifying the plausibility property based on such a theoretical conclusion, avoiding imposing empirical parameters during evaluation.

*Consistency.* Prior work has focused on two aspects of this dimension. First, the optimization algorithm (namely, the Expectation Maximization algorithm) can converge to the local optima of the log likelihood space yielding different properties of parameters that depend on the initial values [5, 16]. Although there are studies on setting good initial values to tackle this problem [5], practically, the strategy of setting randomly distributed initial values is usually taken. Yet there is still no principled way to measure the models’ difference in the variation of convergence, and as a result, it is difficult to get a quantitative view of such a property. Second, multiple global optima of Knowledge Tracing exist [16, 2] where observed student performance corresponds to different sets of parameter estimates that make different assertions about student knowledge, yet have identical (under finite precision) performance predictions [2]. This problem is referred to as the identifiability problem [2]. Later studies have presented different (and even contradictory) views of this problem [19, 9]. These two aspects all relate to the consistency of the parameter space, and in order to determine their practical implications, we offer a unified view of them.

## 3. POLYGON EVALUATION FRAMEWORK

Polygon is a novel framework proposed for evaluating general latent variable student models from multiple dimensions with multiple metrics, besides simply predictive performance. It considers three dimensions, predictive performance, plausibility, and consistency, along with novel met-

rics that instantiate each dimension. Polygon can evaluate a single model which contains only one set of parameters fitted from the data, because in practice we usually deploy a single model into a tutoring system after model selection. Polygon’s predictive performance and plausibility metrics can be used to evaluate single models. However, latent variable models can converge to different points with different initial parameter values due to the non-convexity of the negative log-likelihood. A better model should be more likely to converge to points with higher predictive performance and plausibility, and also give more stable predictions and inferences. So we also use Polygon to evaluate a student model fitted from a large number of random initializations. This provides an examination on the parameter space that is useful for single model selection or construction. In our study, we call these final fitted models random restarts. We mainly focus on evaluating the parameter space from random restarts, but also include evaluating a single model. Each Polygon metric evaluates the trained model(s) of a skill. To get an overall evaluation across skills, we aggregate by averaging each skill’s individual metric. All metrics range from 0 to 1, with a higher positive value indicating higher quality. We focus on the evaluation on Knowledge Tracing and FAST in this study. We now introduce Polygon in detail.

### 3.1 Predictive Performance

Predictive performance has been the previous standard of evaluating student models. It provides useful validation for the inference of knowledge, since accurate knowledge estimation should imply accurate prediction of student performance. We apply a widely used classification metric for this.

*AUC and P-RAUC.* We use Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to evaluate each single model on test set, which gives an overall summary of diagnostic accuracy. AUC equals 0.5 for a random classifier and 1.0 for perfect accuracy. For assessing multiple random restarts, we compute the average of AUC values from single models and define it as P-RAUC, where P- stands for prediction performance, R stands for random restart, and r indicates the  $r^{th}$  random restart:

$$P\text{-RAUC} = \frac{1}{R} \sum_{r=1}^R \text{AUC}^r \quad (1)$$

### 3.2 Plausibility

The conceptual idea behind using Knowledge Tracing to model student knowledge is that knowing a skill generally leads to correct performance, and conversely, that correct performance implies that a student knows the relevant skill [1]. We define plausibility metrics based on this idea.

*Guess+Slip<1 (GS) and P-RGS.* Several prior studies have empirically addressed the issue of plausibility, as mentioned in Section 2. A recent study [19] has provided a theoretical ground that we think can be used to formally define plausibility. This study used theoretical fixed point analysis to prove that when  $\text{Guess} + \text{Slip} > 1$ , the probability that a student has learned a skill just after a practice, given the student’s previous performance, decreases for correct practices and increases for incorrect practices. In this case, the model is empirically degenerate [1]. This is different from theoretically degenerate [1] constraining  $\text{Guess} \leq 0.5$  and  $\text{Slip} \leq 0.5$

to be plausible estimations, which we think is somewhat too strong. For example, it is possible that a student may answer a problem correctly after receiving strong scaffolding (help), even though the skill has not yet been learned. As a result, we propose a metric constructed using the  $\text{Guess} + \text{Slip} < 1$  condition. We use an indicator for  $\text{Guess} + \text{Slip} < 1$  for a single model and refer to it as GS (Equation 2). For assessing random restarts, we compute the average of the GS values from single models and define it as P-RGS, where P- stands for plausibility and R stands for random restart (Equation 3):

$$GS^r = \mathbb{1}(\text{Guess}^r + \text{Slip}^r < 1) \quad (2)$$

$$P\text{-RGS} = \frac{1}{R} \sum_{r=1}^R GS^r \quad (3)$$

Here,  $\mathbb{1}$  is an indicator function and  $\text{Guess}^r$  and  $\text{Slip}^r$  are the  $r^{th}$  random restart’s fitted probabilities. For FAST, with the change of feature values, Guess and Slip can change. We focus on capturing the average behavior of guessing and slipping across contexts, so we compute Guess and Slip with only the intercepts in the logistic regression component (note that other features are activated according to context during training). The interpretation of our computation depends on the construction of features. For example, when using item indicator features, the computation captures the average values of Guess and Slip of a skill.

### *Non-decreasing Predicted Probability of Learned (NPL) and P-RNPL.*

In addition to the above metric grounded in a theoretical analysis [19] for Knowledge Tracing, we construct another empirical metric to capture the behavior of a general latent variable model, since it is not always easy or feasible to conduct theoretical analysis of complex models. Our proposed metric captures how likely a model gives a non-decreasing estimation of knowledge levels with an increase in practice opportunities. This idea is consistent with constraining the learning rate to be non-negative, as in [17, 6]. We think that a decreasing predicted probability of learned is not plausible, based on the interpretation that such a decrease implies practices that hurt learning. We are aware that a decreasing knowledge estimate can also be interpreted as a decrease in the model’s belief that a student might reach a high knowledge level, where the model adjusts itself when observing a lot of incorrect practices. However, we focus on the first interpretation, because in real world tutoring systems where students are aware of their knowledge level as provided by the systems, decreasing knowledge estimates with more practices might discourage students from trying more.

To construct this new metric, we first obtain the estimation of a student reaching leaned state at each  $t^{th}$  practice opportunity given prior  $1^{th}$  to  $(t - 1)^{th}$  performance  $O_1$  to  $O_{t-1}$  on the test set. We denote this probability as  $P(L_t = \text{Learned} | \mathbf{O}_{1:t-1})$ , and also refer to it as  $P(\tilde{L}_t | \mathbf{O})$  for simplicity. Then we count the total number of consecutive pairs with non-decreasing  $P(\tilde{L}_t | \mathbf{O})$  across each skill-student sequence, and then divide it by the the total number of observations of the current skill. We define this as NPL as an indicator of its plausibility for assessing a single model (Equation 4). For assessing random restarts, we compute the average of the NPL values obtained from single models, and define it as P-RNPL, where P- stands for plausibility

and R stands for random restart (Equation 5):

$$\text{NPL}^r = \frac{1}{D} \sum_{s=1}^S \sum_{t=1}^{T_s-1} \mathbb{1}[\text{P}(\tilde{L}_{t+1}^{rs} | \mathbf{O}^{rs}) \geq \text{P}(\tilde{L}_t^{rs} | \mathbf{O}^{rs})] \quad (4)$$

$$\text{P-RNPL} = \frac{1}{R} \sum_{r=1}^R \text{NPL}^r \quad (5)$$

where  $\mathbb{1}$  is an indicator function,  $r, s, t$  indicates random restarts, students, and practice opportunities, respectively.  $T_s$  is the total number of practices of student  $s$ , and  $D$  is the total number of practices of all students of current skill.

### 3.3 Consistency

Depending on different initial values of parameters, Knowledge Tracing and FAST can converge to points with different properties (such as plausibility or prediction of mastery). We favor a consistent model that has a low variance in properties across random restarts. Here, we extend the problem of Identifiability, where only global optimal log likelihood points are involved, into a more general problem of consistency, where all converged points are examined. The measurement of all converged points might be more operational in practice since it can be hard to judge whether the algorithm reaches a local or global optimum. For example, it is not clear how many random restarts are needed. Also, it is not sure whether converged points with log likelihood very close to the identified highest one can be treated as global optima or not.

*Consistency of AUC, GS, NPL (C-RAUC, C-RGS, C-RNPL).* Based on the explained importance of the performance metric AUC and the plausibility metrics GS and NPL, we think that a good model should also present low variance in these metrics across random restarts. As a result, we define consistency metrics C-RAUC, C-RGS, C-RNPL correspondingly by computing the standard deviation<sup>1</sup> of each single model's metrics across multiple random restart runs ( $r$ ) on the test set with some transformation to map them into  $[0, 1]$  interval. Here, C- stands for consistency and R stands for random restarts. For example, for computing C-RAUC, we use the following formula:

$$\text{C-RAUC} = 1 - \sqrt{\frac{1}{R} \sum_{r=1}^R (\text{AUC}^r - \overline{\text{AUC}})^2} \quad (6)$$

*Consistency of the Predicted Probability of Mastery (C-RPM).* Student models are usually used to assess whether and when students reach mastery, based on which tutoring systems give adaptive instruction. A model lacking consistency in mastery prediction will lead to varying decision in instruction, which can have a significant impact on students. So we also construct a metric to quantify this consistency, inspired by previous studies [2, 15, 7]. We use the conventional definition of Mastery as the probability of Learned reaching 0.95 [4]. We compute  $\text{P}(L_t = \text{Learned} | \mathbf{O}_{1:t})$ , the posterior knowledge estimation of being in the Learned state at  $t^{\text{th}}$  practice updated by  $1^{\text{st}}$  to  $t^{\text{th}}$  practice observations  $\mathbf{O}_{1:t}$ .

<sup>1</sup>We use uncorrected sample standard deviations to map the metric to  $[0, 1]$ . With a large enough sample size (100 in our study), the bias of this estimator is small. For a smaller sample size, the corrected version might be considered.

We also refer to it as  $\text{P}(\tilde{L}_t | \mathbf{O})$  for simplicity. We then compute the probability of reaching Mastery as the percentage of students predicted to ever have  $\text{P}(\tilde{L}_t | \mathbf{O}) \geq 0.95$ , which means achieving a 0.95 posterior knowledge estimation in a practice sequence for the current skill. We refer to this probability as  $\text{P}(\text{Mastery})$  or PM (Equation 7). We then compute the standard deviation of  $\text{P}(\text{Mastery})$  across different runs, transform it to map to  $[0, 1]$  interval, and refer to it as C-RPM where C- stands for consistency, R stands for random restarts (Equation 8):

$$\text{PM}^r = \frac{1}{S} \sum_{s=1}^S \mathbb{1}\{\text{P}(\tilde{L}_t^{rs} | \mathbf{O}^{rs}) \geq 0.95, \exists t \in [1, T_s]\} \quad (7)$$

$$\text{C-RPM} = 1 - \sqrt{\frac{1}{R} \sum_{r=1}^R (\text{PM}^r - \overline{\text{PM}})^2} \quad (8)$$

where  $r, s, t$  indicates random restarts, students, and practice opportunities respectively.  $T_s$  is the total number of practices of student  $s$  of current skill.

*Cohesion of the parameter vector space (C-RPV).* Fixed point analysis has been used to show that we need all four parameters to define the overall behavior of Knowledge Tracing [19] during the prediction phase, when knowledge estimation is updated by prior observations. We use this conclusion to construct another consistency metric. To capture all four parameters, we construct a Euclidian vector based on the four fitted parameters Init, Learn, Guess, and Slip for each single model. For FAST, we compute the four parameters with only the intercepts in the logistic regression components after fitting with features during training. We then compute the Euclidian distance of each vector to the mean of the parameter vectors (similar to the cluster cohesion measurement), and then perform a transformation to map this value to  $[0, 1]$  interval. We define it as C-RPV where C- stands for consistency, R stands for random restarts, and PV stands for parameter vector:

$$\text{C-RPV} = 1 - \frac{1}{2R} \sum_{r=1}^R \|\mathbf{V}^r - \overline{\mathbf{V}}\| \quad (9)$$

where  $\mathbf{V}^r$  is the parameter vector of the  $r^{\text{th}}$  random restart.  $\mathbf{V}^r = (\text{Init}^r, \text{Learn}^r, \text{Guess}^r, \text{Slip}^r)$ .  $\overline{\mathbf{V}}$  is the mean of the parameter vectors across the random restarts.

### 3.4 Metric Selection

Our proposed Polygon framework consists of three dimensions: prediction, plausibility, and consistency, and allows flexibly designed metrics for each dimension. The metrics we introduced before are the potential ones to be considered. We propose a principled way to select metrics to instantiate the framework: selected metrics should cover all three dimensions while having the smallest pairwise correlation. To achieve this, we examine the scatterplot and correlation of each pair of the metrics and conduct a significance test. Finally, we report our selected metrics in Section 4.3.1.

## 4. STUDIES AND RESULTS

### 4.1 Datasets and Features

We conducted experiments on datasets from different tutoring systems: Geometry Cognitive Tutor [12], OLI Engineering Statics [18], Java programming tutor [10], and the

Physics tutoring instance of the BBN learning platform [14]. Table 1 shows descriptive statistics (#observations indicates the smallest assessable practice units of students).

**Geometry, Statics.** We obtained these datasets from PSLC Datashop [13]. The Geometry dataset has data from the area unit of the Geometry course, which was conducted during the 1996-1997 school year. The Statics dataset has data from multiple schools during Fall 2011. We defined a problem (item) by concatenating the problem hierarchy, problem name, and step name. We defined a skill by concatenating the problem hierarchy and original skills, and treated the combination of skills as one unique skill if multiple skills are associated with a problem. For the Statics dataset, we randomly selected 20 skills (from the total of 156) to avoid bias towards this dataset when we aggregate across datasets. We further removed 3 skills where there are fewer than 10 observations in total, resulting in 17 skills. For FAST models, we constructed binary item indicator features for each problem with fitted coefficients represent item difficulties. Such models have been known for their high predictive performance [11, 8], and we plan to examine other dimensions as well.

**Java.** The Java dataset was collected from an online Java programming tutoring system [10] from Fall 2010 to Fall 2014. For each problem, students are asked to give the value of a variable or the printed output of a Java program after they have executed the code in their mind, and the system assesses correctness. The Java programs are instantiated randomly from templates on every attempt. Students can make multiple attempts until they think they have mastered the skill, or just give up. Problems are grouped by Java topics (each problem is mapped to a single topic), and we considered each topic as a skill. We consider each problem template as a single item. For FAST models, we also constructed binary item indicator features, adding to the exploration of the effect of item difficulties.

**Physics.** The Physics dataset was collected from the BBN Learning Platform [14], a domain-independent, problem-solving-based online learning platform. Students can solve problems without any help, or request a decomposition of the problem into steps. The steps lead students through a carefully crafted directed path to help solve the problem. We used logs collected from 40 users solving 10 problems from the Electric Circuits units. Each of these problems and steps are annotated with electric circuits skills (in total 10). In addition to capturing student actions at the items, the platform logs requests for help, feedback received, and problem navigation actions. We derived 105 numeric features from these logs, performed feature selection, and finally used the top ranked feature for FAST. This allows us to inspect the effect of help in the Knowledge Tracing framework.

## 4.2 Experimental Setup

We used Expectation Maximization (EM) for training Knowledge Tracing, and Expectation Maximization with features for FAST [8]. We uniformly initialized each parameter within (0, 1) at each run for Knowledge Tracing, and we uniformly initialized each feature coefficient within (-10, 10) for FAST, which resulted in original parameters approximately covering (0, 1). We drew 100 different initial values for each parameter. We set 500 as the maximum EM iteration, 50 as the maximum LBFGS iteration and the log likelihood’s rela-

Table 1: Dataset descriptive statistics.

Dataset	#observations	#skills	#students	%correct
Geometry	5,055	18	59	75%
Statics	23,390	17	326	77%
Java	43,696	20	328	67%
Physics	10,063	10	40	62%

Table 2: Scatterplot and Kendall rank correlation among metrics of all skills (65) from Knowledge Tracing. Metrics selected into Polygon are shown in blue. Values shown in blue indicate a low correlation, and values shown in YellowOrange with asterisks indicate statistical significance ( $\alpha=0.05$ ).

	1	2	3	4	5	6	7	8
1.P-RAUC		.13	-.01	-.16	.07	-.00	.16	.14
2.P-RGS			.09	-.09	.25*	-.02	.05	.11
3.P-RNPL				-.06	.29*	-.07	-.07	.00
4.C-RAUC					.13	.31*	.11	.14
5.C-RGS						.22*	.26*	.49*
6.C-RNPL							.39*	.36*
7.C-RPM								.57*
8.C-RPV								

tive change within  $10^{-6}$  as convergence criteria. We trained each skill independently and used a user-stratified data split: 80% of the students were randomly selected into the training set, and the remaining students were assigned to the test set. In this way, models can be generalized to unseen students.

## 4.3 Results

### 4.3.1 Metric Selection

In order to obtain a compact instantiation of the Polygon evaluation framework, we analyze the pairwise correlation among the proposed metrics on Knowledge Tracing models. For each skill we compute eight metrics based on 100 random restarts and analyze the relationship across skills. Table 2 shows that C-RGS, C-RNPL and C-RPV all include significant correlations with other metrics. Particularly, the scatterplot of P-RGS and C-RGS shows a U-shape; we think this finding is because the mean and standard deviation of Bernoulli-distributed variables (GS) have this property. Finally, we instantiate the **Polygon** framework with five metrics in our study: **P-RAUC**, **P-RGS**, **P-RNPL**, **C-RAUC** and **C-RPM**, where they cover three dimensions and have low, non-significant pairwise correlations.

### 4.3.2 Evaluation on Multiple Random Restarts

We now present how we use Polygon to evaluate multiple random restart models and single models on Knowledge Tracing and FAST. Figure 1 shows Polygon evaluation per dataset aggregated across skills. Overall, FAST mostly have Polygon areas covering that of Knowledge Tracing. Considering the variance across skills, FAST has significantly higher values in all five metrics ( $\alpha=0.05$ ,  $p < 0.0001$  by Wilcoxon signed-rank test), suggesting that it might promise not only higher predictive performance, but also higher plausibility and consistency. One possibility is that the constructed features indirectly constrain the optimization algorithm to

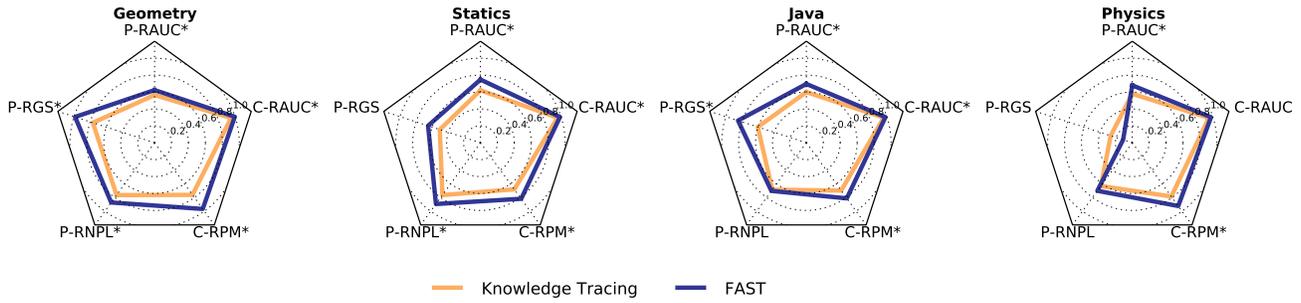


Figure 1: Polygon metrics per dataset comparing Knowledge Tracing and FAST. An asterisk (\*) indicates statistical significance under Wilcoxon signed-rank test ( $\alpha=0.05$ ). FAST's Polygon area mostly covers that of Knowledge Tracing.

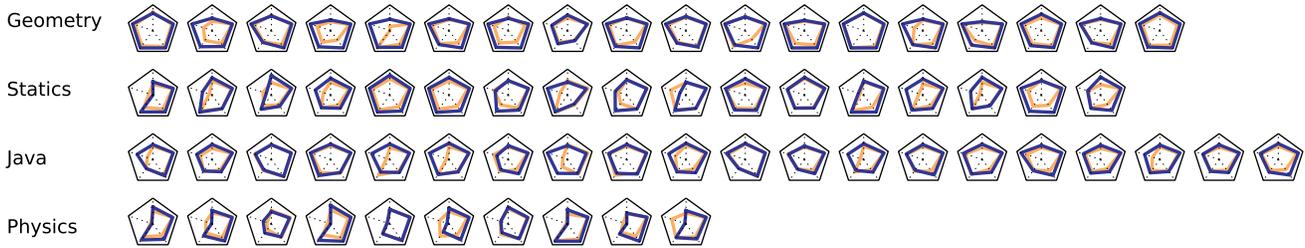


Figure 2: Polygon metrics per skill comparing Knowledge Tracing and FAST. FAST's Polygon area mostly covers that of Knowledge Tracing.

search within regions with both high fitness and plausibility. However, FAST's plausibility seems to be less stable, as compared to other properties, since its improvement varies across datasets.

We further examine Geometry, Statics and Java datasets where we use FAST with item difficulty features. As shown in Figure 1, FAST significantly outperforms Knowledge Tracing in all metrics, except for P-RGS on Statics and P-RNPL on Java, where FAST still presents positive tendencies. Generally speaking, using item difficulty features in Knowledge Tracing not only increases the model's predictive performance, but also its plausibility and consistency. However, the relative improvement in plausibility varies across datasets.

In the Physics dataset, FAST using problem decomposition requested features has a higher P-RAUC (significant), P-RNPL, C-RPM (significant), and C-RAUC, yet it also has a lower P-RGS, compared with Knowledge Tracing (not significant). Noticing that both methods have very low P-RGS, we suspect that skill definitions may be too coarse-grained, meaning that latter practices may involve potential new skills, where students fail more often than in the beginning. Thus, student models fitted from such data might be prone to estimating high Guess and Slip. FAST may be more vulnerable to bad skill definitions, since it might seek to fit the data as the primary goal, given that it has significantly higher predictive performance. In order to find out more about these potentially ill-defined skills, we further examine Polygon for each skill, as shown in Figure 2. This analysis shows that more than half of the skills in the Physics dataset have very low P-RGS, and particularly, there are two skills where FAST and Knowledge Tracing have an obvious gap on P-RGS (6<sup>th</sup> and the last one), which should cause Knowledge Tracing to obtain a higher average value over FAST. We plan to examine whether refinement of the skill definitions will increase plausibility of both methods and FAST's relative quality for P-RGS in next steps.

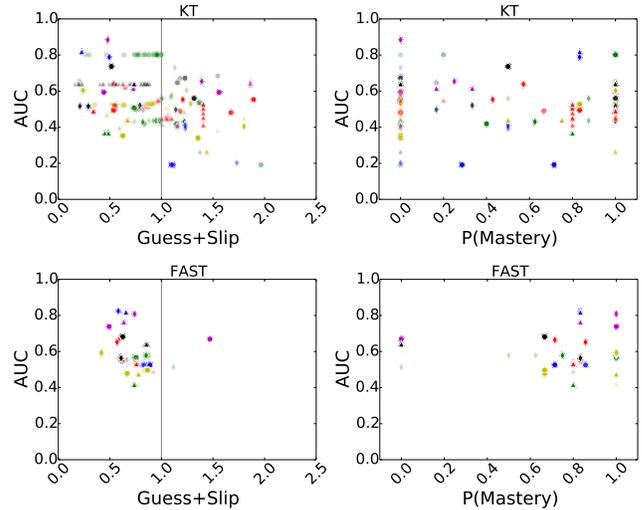


Figure 3: Evaluation on each skill's each random restart on Geometry dataset. Each color-shape corresponds to one skill. Each point corresponds to one random restart convergence point. Comparing with Knowledge Tracing, FAST generates more consistent, plausible models.

### 4.3.3 Drill-down Evaluation of Single Models

Polygon not only evaluates a method from multiple random restarts, but also contains components that can evaluate a single model. We use AUC, GS (Guess+Slip<1), and NPL to analyze each single model's predictive performance and plausibility, and also use the component PM (P(Mastery)) to get an intuitional understanding of a single model's effect on tutoring. Figure 3 visualizes AUC, Guess+Slip, and P(Mastery) of each random restart of each skill for Knowledge Tracing and FAST on Geometry dataset. Each color-shape corresponds to one skill, while each point corresponds to one random restart convergence point. We can easily determine different behaviors between Knowledge Tracing and FAST. FAST generates more consistent solutions than

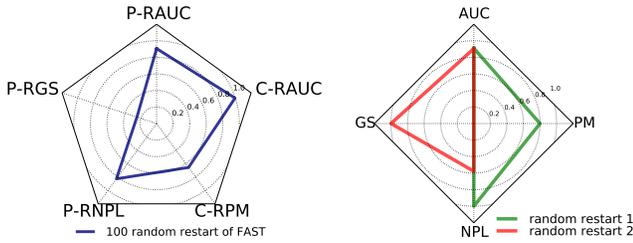


Figure 4: Polygon evaluation on a skill (id=154) on Statics dataset. The multi-model pentagon reveals this skill has high AUC consistency but low P(Mastery) consistency. The single-model quadrangle further reveals the contradictory properties of two random restart single models even they have very similar AUC.

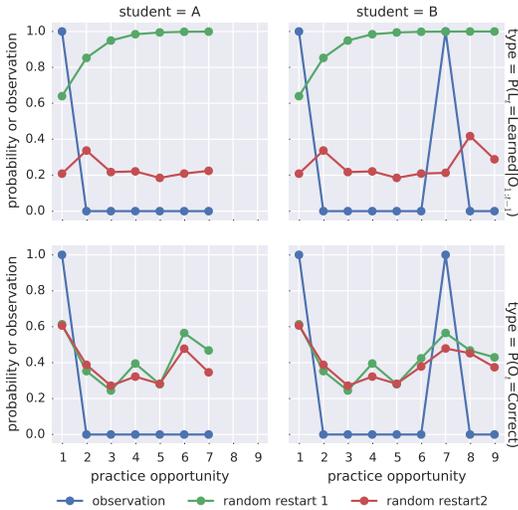


Figure 5: Comparison of two random restart single FAST models of a skill (id=154) from Statics dataset on two students. Both models have similar curves of predicted  $P(O_t=Correct)$  but have substantially different curves of predicted  $P(L_t=Learned | O_{1:t-1})$ .

Knowledge Tracing, since there is less spread both horizontally and vertically of the random restart points within the same skill for all three metrics. FAST also generates more plausible models than Knowledge Tracing, since most of the points fall into  $Guess+Slip < 1$  region. Note that FAST asserts that students are more likely to reach mastery, since the converged points mostly lie in the higher-value region.

However, does FAST perform well on every skill? If not, can we use Polygon to effectively identify such skills and better understand the behavior? Based on previous skill-specific polygon evaluations (Figure 2), we identify one skill ( $3^{rd}$  polygon on the  $2^{nd}$  row) on the Statics dataset, where Knowledge Tracing has better P-RGS than FAST. In Figure 4 the left-hand figure shows that this skill has a very high consistency of predictive performance (C-RAUC), yet a very low consistency of PM (C-RPM) across 100 random restarts. We further pick two of the random restarts and compute the polygon metrics for single models, as shown in Figure 4 right-hand single-model quadrangle. The quadrangle reveals that these two random restarts have almost identical AUC, yet have contradictory assertions about learning and mastery. In order to better understand the behavior, we

Table 3: Kendall rank correlation among single model AUC, GS, NPL and log likelihood (LL) on training set for the same skill across 100 random restarts on Knowledge Tracing. We report the number of skills and in the bracket the average of the correlation values across skills under each positive (+) or negative (-) correlation relation (zero correlation ignored) among all skills (65).

	AUC		GS		NPL	
	+	-	+	-	+	-
AUC			41(0.6)	23(-0.6)	35(0.7)	30(-0.5)
LL	46(0.5)	19(-0.4)	34(0.5)	30(-0.5)	30(0.4)	35(-0.5)

pick two students from each one of these random restarts, and plot the predicted correctness curve and knowledge level curve (conditioned on prior observations). Figure 5 shows a severe problem in comparing these two random restarts: they have very similar predicted correctness, yet present fundamentally different predicted knowledge levels. We think that this problem extends the identifiability problem, in the sense that similar predicted correctness curves though not identical can be problematic if the predicted knowledge level curves differ greatly. Also, we observe the empirical degeneracy of random restart 1: with more incorrect practices, the predicted probability of Learned increases. This analysis showcases the deficiency of using only predictive performance to evaluate student models, and the effectiveness of Polygon metrics in identifying hidden problems.

#### 4.3.4 Implications for Single Model Selection

We further examine the deficiency of using prediction performance or fitness metrics to select single models. We compute the Kendall rank correlation between AUC and the plausibility metrics for each run of each skill of Knowledge Tracing. Table 3 shows the deficiency of using only AUC to select the best random restart. There are more than one-third of skills that show a negative correlation between predictive performance and plausibility across different runs, and the magnitude of the negative correlation on average is not small. What about choosing the model with the maximum likelihood (LL) on the training set? Table 3 also shows the correlation between LL, AUC, and the plausibility metrics across different random restarts. Overall, about 71% (46/65) of the time, choosing the maximum LL on the training set can lead to a higher predictive performance in the test set, yet we have no more than 46% (30/65) of the time to get a more plausible model. These findings show that LL fails to offer a better choice than AUC. We think that a practical generalizable way to obtain a latent variable student model with both high predictive performance and plausibility remains to be explored, and Polygon provides important insights.

## 5. CONCLUSIONS

In this paper, we propose a general unified evaluation framework (that we call Polygon) to evaluate student models with latent knowledge estimates. Prior studies have presented different properties of the estimated parameters of Knowledge Tracing, yet there are no unified, quantitative evaluations for general student models. Our primary contribution lies in the quantitative unification of three aspects for general latent variable student models: predictive performance, plausibility, and consistency. We propose novel metrics and present a principled way to select proper metrics. Our defined dimensions extend the definitions of previously defined Identifi-

bility and Model Degeneracy, which allows us to understand such problems more practically and more generally. A secondary contribution is that we show that a recent model with proper features, known as FAST, generally provides higher predictive models with higher plausibility and consistency than Knowledge Tracing. This suggests that proper features might help the optimization algorithm to constrain the search towards more plausible, more predictive regions.

There are several areas in which we can further extend our study. First, a single metric or perspective considering the multiple facets introduced in our analysis can further improve the accessibility of the evaluation. Also, each single metric can be further improved. For example, we can investigate the proper number of random restarts. However, Polygon’s current individual metrics already provide insights for training student models. For example, incorporating the plausibility metric as a penalty into the optimization objective function can guide the algorithm to search within the high plausibility region. Second, external measurements applied in prior studies [4, 2, 6] may help to validate our framework. However, Polygon primarily serves as domain-independent internal validity, which is useful when external resources are not available. Third, the plausibility measurement can be a mixture of both student model and skill model evaluations. Will each model’s relative quality be different when we examine well-defined vs. ill-defined skills? Can we utilize plausibility metrics to inspect skill model qualities? These are questions that remain unanswered. Fourth, we need to further understand and improve FAST. Since there are still cases where FAST generates models with low plausibility or low consistency, is there a principled way to construct features that maximize all three dimensions? Also, as we have only studied cases where a single feature (besides the intercept) is activated for each observation, will increasing the number of features change FAST’s behavior?

Our study is still exploratory and serves as a first step towards a more theoretical, deeper understanding of the parameter estimation of complex latent variable student models. We hope that our work can open the door to more studies in the community on building student models that can yield not only better predictions of student performance but also more reliable, effective tutoring systems.

## 6. ACKNOWLEDGMENTS

This research is supported the Advanced Distributed Learning Initiative<sup>2</sup>, Pearson<sup>3</sup> and the US Office of Naval Research (ONR) contract N00014-12-C-0535.

## 7. REFERENCES

[1] R. Baker, A. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems 2008*, pages 406–415. Springer.

[2] J. E. Beck and K.-m. Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146.

[3] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with

features. In *HLT: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590.

[4] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[5] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *6th International Conference on Educational Data Mining*, pages 28–35, 2013.

[6] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, pages 35–44. Springer, 2010.

[7] J. P. González-Brenes and Y. Huang. Your model is predictive— but is it useful? theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In *Proceedings of the 8th Intl. Conf. on Educational Data Mining*, 2015.

[8] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proceedings of the 7th Intl. Conf. on Educational Data Mining*, 2014.

[9] G. Gweon, H.-S. Lee, C. Dorsey, R. Tinker, W. Finzer, and D. Damelin. Tracking student progress in a game-like learning environment with a monte carlo bayesian knowledge tracing model. In *Learning Analytics and Knowledge Conference 2015*.

[10] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning*, 2010.

[11] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, 2014.

[12] K. R. Koedinger. Geometry area (1996-97), February 2014. In *URL* <https://pslccdatashop.web.cmu.edu/DatasetInfo?datasetId=76>.

[13] K. R. Koedinger, R. S. J. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*, pages 43–55, Boca Raton, FL, 2010. CRC Press.

[14] R. Kumar, G. Chung, A. Madni, and B. Roberts. First evaluation of the physics instantiation of a problem-solving based online learning platform. In *Intl. Conf. on Artificial Intelligence in Education 2015*.

[15] J. I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th Intl. Conf. on Educational Data Mining*, pages 118–125, 2012.

[16] Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. *EDM*, 2010:161–170, 2010.

[17] P. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis—A New Alternative to Knowledge Tracing. In *Proceeding of the 2009 conference on Artificial Intelligence in Education*, pages 531–538.

[18] P. Steif and N. Bier. Oli engineering statics - fall 2011, February 2014. . In *URL* <https://pslccdatashop.web.cmu.edu/DatasetInfo?datasetId=507>.

[19] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, 5(2):1–10, 2013.

<sup>2</sup><http://www.adlnet.gov/>

<sup>3</sup><http://researchnetwork.pearson.com/>

# Predicting Student Performance In a Collaborative Learning Environment

Jennifer K. Olsen

Human Computer Interaction Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
jkolsen@cs.cmu.edu

Vincent Aleven

Human Computer Interaction Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
aleven@cs.cmu.edu

Nikol Rummel

Institute of Educational Research  
Ruhr-Universität Bochum  
Bochum, Germany  
nikol.rummel@rub.de

## ABSTRACT

Student models for adaptive systems may not model collaborative learning optimally. Past research has either focused on modeling individual learning or for collaboration, has focused on group dynamics or group processes without predicting learning. In the current paper, we adjust the Additive Factors Model (AFM), a standard logistic regression model for modeling individual learning, often used in conjunction with knowledge component models and tutor log data. The extended model predicts performance of students solving problems collaboratively with an ITS. Specifically, we address the open questions: Does adding collaborative features to a standard AFM provide a better fit than the standard AFM? Also, does the impact of these features change based on the nature of the knowledge (conceptual v. procedural) that is being acquired? In our extended AFM models, we include a variable indicating if students are working individually or in pairs. Also, for students working collaboratively, we model both the influence on learning of being helped by a partner and helping a partner. For each model, we analyzed conceptual and procedural datasets separately. We found that both collaborative features (being helped and helping) improve the model fit. In addition, the impact of these features differs between the collaborative and procedural datasets, suggesting collaboration may affect procedural and collaborative learning differently. By adding collaborative learning features into an existing regression model for individual learning over a series of skill opportunities, we gain a better understanding of the impact that working with a partner has on student learning, when working with a step-based collaborative ITS. This work also provides an improved model to better predict when students have reached mastery while collaborating.

## Keywords

knowledge tracing, collaborative learning, educational data mining, Additive Factors Model

## 1. INTRODUCTION

The modeling of student knowledge has been shown to be an important aspect of Intelligent Tutoring Systems (ITSs) technology. A variety of modeling approaches have been used to model student knowledge and have often been used to support

individualized learning [2, 3, 15, 25]. Models can provide an accurate prediction of learning and also provide insights into how people learn. However, these types of models typically account for students who work individually with an ITS; they do not account for situations in which students learn collaboratively in dyads or small groups, supported by ITS technology. Yet collaboration cannot be ignored since it has been shown to be beneficial for student learning [6, 19] and there may be relative strengths for collaborative and individual learning [11]. Students who work collaboratively may have different learning rates than when working individually; this effect may be caused from being helped by a partner or helping a partner. A key question is, therefore: How can modeling techniques used for individual learning be adapted so they help provide predictions and insights into collaborative learning, in addition to individual learning? Specifically, how can these models be adapted to account for the fact that the collaborating partners may influence each other's learning? What insight can models provide regarding this influence? In our ITS, students work either collaboratively or individually on the problem sets. We extend the Additive Factors Model (AFM) [2, 15] by including features that are unique to collaboration, in an attempt to better model both individual and collaborative learning.

Much of the research on learning prediction has focused on modeling individual learning such as through Bayesian Knowledge Tracing [3], AFM [2, 15], and Knowledge Decomposition Model [25]. These models accurately predict student performance and can advance our understanding of how students learn. Previous research has adapted these types of models to better predict and understand individual learning, such as by treating correct and incorrect attempts differently [15] or by including the transfer that may happen between similar skills [25]. For our work, we are using a version of the AFM. The AFM has frequently been used to assess and predict individual student performance. The AFM is a generalized logistic mixed model [1]. It is widely used to fit learning curves and to analyze and improve student learning [1]. To adapt the AFM to account for aspects of collaborative learning, we can apply the same types of principles that have been applied to increase our understanding of individual learning and apply them to collaborative learning. For example, individual models can account for the transfer of learning from previous similar opportunities [25]; the same method can be applied to collaborating students having an opportunity to learn from watching their partner solve steps.

Prior research within collaborative learning has focused on analyzing collaborative processes to better understand learning and social influence [5, 20]. Within this area, there are multiple approaches for better understanding the collaborative processes.

### Equivalent Fractions

**A Let's find equivalent fractions.**

The purple circle shows the fraction:  $\frac{1}{3}$

Select twice as many pieces but have the same total pieces as the purple fraction.  Do  Ask

Make the pieces half as big but the same selected pieces as the purple fraction.  Do  Ask

Make the pieces half as big and select twice as many pieces as the purple fraction.  Do  Ask

Name the fraction

What do you multiply the numerator and denominator by to get the new fraction?

How has the amount changed compared to the purple fraction?

Which fraction is equivalent to the purple fraction?

**B Let's define equivalence.**

1 For a fraction to be equivalent with another fraction: (Answer individually and then as a group)

- The numerators must be the same
- The denominators must be the same
- The numerator and denominator must be multiplied by the same number to get the second fraction
- The amounts need to be different

Done

**Figure 1. An example of a conceptual problem showing the different steps assigned to the partners in the collaborative condition based on the “Do” and “Ask” icons.**

Some research aims to detect and classify collaboration skills, such as social deliberation skills and collaborative networks [21, 24]. Other research looks at the change in communication and processes that happen over time [10, 18]. Research has also focused on group dynamics and how we can recognize and intervene with groups that are not collaborating well [8, 9, 16]. Another aspect of collaboration that has been studied is asynchronous work that occurs on discussion boards and how this can influence learning and retention [22, 23]. Although this research is broad in the types of research questions that are addressed and covers many aspects of collaboration, much of the work does not attempt to predict student performance as students collaboratively solve problems. Such predictions could support student learning, for example by informing problem choices for dyads to help students where they are struggling. There has been previous work that has studied predicting performance by predicting posttest scores based on pair actions and found student interactions are predictive of the posttest score [17]; however, this work focuses on environments where the actions of collaborating students within a dyad or group cannot be distinguished (i.e., it is not known who took the action). In collaborative environments, in which the actions of the students within a collaborating group can be distinguished (e.g., a collaborative ITS), including collaborative features in models that have typically been used to predict individual performance may support a better understanding of the collaborative learning process and the ability to predict performance when students are collaborating. Previous work has attempted to address this issue by predicting performance of students based on their speech with an intelligent agent and found semantic match scores as a key predictor of later test performance [12]. Our work adds to this body of literature by investigating the prediction based on student actions within a

system and how students will later do on similar items. The analysis of the student actions may provide different insights into the collaborative processes.

Extending the AFM with collaborative features enables us to study how collaboration might influence learning. Prior research with collaborative learning has shown that within mathematics, collaborative learning may better help students acquire conceptual knowledge, whereas individual learning activities may be more conducive to learning procedural knowledge [11]. Since our data set, obtained with a fractions tutor that supports collaborative learning, described below, includes both conceptual and procedural activities [13], we can study whether and how collaboration affects learning differently for these types of activities. By separately fitting models capturing collaborative and individual learning to data from procedurally versus conceptually oriented problems, we may be able to add to the understanding of how the different aspects of collaborative learning may have different strengths for different types of knowledge.

In this paper, we extend the AFM to (a) distinguish the learning that may occur when working individually versus collaboratively and (b) to capture learning that may occur from observing a partner’s answers to steps. We also explore (c) whether the effect of these features is different in activities designed to support learning of concepts, compared to activities designed to support learning of procedures. By modeling student knowledge when working collaboratively, we aim to develop a better understanding of collaborative and individual learning processes. An improved model would also allow us to more accurately predict student performance and has the potential to support learning more effectively within an ITS, for example through improved problem selection for collaborative learning.

Figure 2. An example of a conceptual problem showing the different steps assigned to the partners in the collaborative condition based on the “Do” and “Ask” icons.

## 2. METHODS

In the following sections, we present the collaborative ITS for fractions learning that was used in our study and explain the experimental set-up that was used for data collection.

### 2.1 Individual and Collaborative Fraction Tutors

In the study that produced the data set that we analyze in the current work, students worked with an ITS that targeted equivalent fractions knowledge either working individually or with a partner. We developed two parallel versions of a fractions tutor, one with embedded collaboration scripts and one for individual learning. We created all tutor versions using the Cognitive Tutor Authoring Tools (CTAT), which we extended so it supports the authoring of tutors with embedded (static) collaboration scripts that are tied to the problem state [14]. Both the individual and the collaborative tutor versions had procedural and conceptual problem sets. Figure 1 shows an example of a conceptual problem, which shows the student different relationships between the numerators and denominators and that only the one where the amount stays the same shows an equivalent fraction. On the other hand, Figure 2 shows an example of a procedural problem where the student makes equivalent fractions by multiplying the numerator and denominator by the same number. The individual ITS provides standard ITS support (step-level guidance for problem solving, with correctness feedback, next-step hints, and error-specific feedback messages) while the collaborative ITS also has embedded collaborative scripts. The students working collaboratively did so through a synchronous, networked collaboration. That is, collaborating students sat at their own computer and had a shared (though differentiated) view of the problem state. They could discuss the activity through audio by using Skype.

The collaboration was supported through proven collaboration scripts such as the use of roles, cognitive group awareness, and unique information, embedded in the interactions with the ITS. First, the embedded collaboration scripts defined roles that distribute the activities between the students and provide guidance to the students about what they should be doing to interact with their partner and help to scaffold this interaction. A second collaborative support feature we used in the collaborative problem sets is cognitive group awareness. Cognitive group awareness means that group members have information about other group members' knowledge, information, or opinions and has been shown to be effective for the collaboration process [7]. The last collaborative support feature is the use of unique information to create a sense of individual accountability. Individual accountability means that each group member takes responsibility for the group reaching its goal [19]. All of these collaboration features, as implemented, assigned different problem steps to each student within a collaborating dyad. The “Do” and “Ask” icons shown in Figures 1 and 2 indicate which student was responsible for solving a given step and which student had the role of supporting the other student; on the screen of the collaborating partner, the “Do” and “Ask” icons would be flipped. Therefore, problem steps divide into a student's own steps and that student's partner's steps. This distinction is important because, we will see, our extended AFMs treat these steps differently.

Our ITS is uncommon in that it was developed to support both collaborative and individual learning. This means that our data logs contain both records of individual and collaborative sessions, with a common set of features that is typical of ITS log data. (The data from the collaborative sessions were captured as separate streams from each student, where a partner's actions are not associated with a student's id.) Although the collaborative tutor had three different types of support for collaboration, each

scaffolding the interactions between the students in different ways, each of these support type led to the same pattern of information in the log data. For every step in a tutor problem, one student was responsible for answering the step and the other student's role was to monitor and help; therefore, the steps in the log data can be assigned to one partner or the other. Although not all collaboration environments allow for the distinction between student actions within a group, many environments can record this data and would then have similar log data to what we have, possibly even when student roles are not as clearly defined and supported.

## 2.2 Data Source

Our data is a set of collaborative and individual data that had been collected from a study [13] in which 4<sup>th</sup> and 5<sup>th</sup> grade students engaged in a problem-solving activity with the ITS for fractions learning described above. The experiment was a pull-out design, in which the students left their normal instruction during the school day to participate in the study. The data set comprises 84 students. Each teacher paired the students participating in the study based on students who would work well together and had similar math abilities. These pairs were then randomly assigned to one of four conditions: collaborative conceptual, collaborative procedural, individual conceptual, and individual procedural. Twice as many students were assigned to the collaborative conditions as to the individual conditions, so that the number of dyads in the collaborative conditions equaled the number of individual students in the individual conditions. Each student or dyad worked with the tutor for 45 minutes in a lab setting at their school during the school day.

We analyzed all tutor problems in terms of the underlying knowledge components (KCs) related to fraction equivalence. For the four conditions, the KCs were the same between the individual and collaborative conditions, but there was no overlap in the KCs between the conceptual and procedural items, as conceptual and procedural KCs were modeled separately.

## 3. MODELS

In this section we review the standard AFM and then present the models we made by adding collaborative features to this model.

### 3.1 Additive Factors Model

We first present the standard AFM, because this model is the basis on which all of our other models are built. The AFM [2] shows that the log-odds that a given student correctly solves a given step in a problem are a function of three parameters capturing, respectively, the given student's proficiency, the ease of the given knowledge component (KC, the skill the student is learning), and the learning rate. It assumes that the learning rate differs by KC but, for any given KC, is equal for all students. It further assumes that students differ in their general proficiency but in a way that affects all KCs and KC opportunities equally.

The AFM is a generalized mixed model.  $p_{ij}$  is the probability that student  $i$  gets step  $j$  right,  $\theta_i$  is the random effect representing the proficiency of the student  $i$ . The fixed effect portion of the model includes  $\beta_k$  (the ease of KC  $k$ ),  $\gamma_k$  (the learning rate of this KC), and  $N_{ik}$  (the prior learning opportunities the student had to apply KC  $k$ ). The  $Q_{kj}$  term represents if an item the student encounters (i.e., a step in a tutor problem) uses KC  $k$ .

$$\ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj} (\gamma_k N_{ik}) \quad (1)$$

The standard AFM presented in Formula 1 is based on individual learning parameters of the opportunities that the individual has had with the KC. For the individual learning condition, these are all steps the student encountered in which the given KC applies. When this model is applied to the collaborative learning condition, on the other hand, these are the steps with the given KC that the given student is responsible for solving. This model however does not take into account that the learning rate for students may be different when working in a group compared to individually or that the students may learn from watching their partner solve problems.

### 3.2 Additive Factors Model with Condition

To investigate the difference in learning rates that may occur when students work individually, as compared to working in pairs, we added a feature to the original AFM that changes the slope based on condition (individual v. collaborative). Similar to the assumption that students learn at different rates from correct and incorrect answers in Pavlik, Cen, and Koedinger's Performance Factors Analysis, PFA [15], students may learn different amounts (per opportunity) when they are working individually versus collaboratively. In the collaborative condition, students are talking with their partner (through Skype) while solving steps that have been assigned to them. Having a partner may have an influence on their learning, even on steps that they (and not their partner) are responsible for solving. A student may get more learning out of a step they solved because of fruitful discussion with the partner, but could conceivably also learn less than when solving the step alone, with tutor help only, for example if the partner simply tells them the answer and the student does not reflect on the answer. In Formula 2, we capture the influence that the presence of a partner has on the student's own opportunities. A term  $c$  is added to represent the condition that the student is in at a given step. This allows the learning rate of a KC,  $\gamma_{kc}$ , to vary depending on the condition, so as to capture a difference in the learning that occurs between individual and collaborative work, on the student's own steps

$$\ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj} (\gamma_{kc} N_{ikc}) \quad (2)$$

By adding the condition parameter to the model, we can capture any differences in learning rates that may occur between working individually and within a group.

### 3.3 Additive Factors Model with Partner Opportunities

Within collaborative learning, there is an opportunity for students to learn from their partner's actions. Recall that when students work collaboratively in our tutoring system, the students are assigned to different roles for any given step (either solve it or help the partner solve it). Therefore, steps in tutor problems classify as the student's own steps or the partner's steps. On the partner's steps the student is watching and possibly providing advice, feedback, and explanations, which may create a learning opportunity for that student, even though he or she is not solving this step. Thus, we need to model the learning that occurs not only

**Table 1. Prediction accuracy for the individual and collaborative procedural dataset across all models. The asterisks indicates the model with the best performance for that criterion.**

Procedural Models	Log Likelihood	RMSE	AIC	Parameters
Standard AFM	-2010.34	0.4738	4080.69	30
AFM with Condition	-1983.39	0.4717	4056.77*	45
AFM with Partner Opportunities	-1984.59	0.4712	4059.17	45
AFM with Condition and Partner Opportunities	-1972.97*	0.4674*	4065.94	60

on a student's own opportunities (as modeled in Formulas 1 and 2) but also on their partner's opportunities. Learning on partner opportunities may be analogous to the learning decomposition that happens as students learn reading and their learning of a certain word benefits from seeing words with identical stems [25]. Although the student is not interacting directly with the tutor, there may still be learning. We assume that the learning that occurs when watching and/or helping a partner is possibly different from that which occurs when *doing* steps. We therefore added a new fixed parameter that takes into account the learning that could happen on a partner's opportunities. In the model seen in Formula 3,  $\rho_k N_{ik}$  represents the learning (with its own learning rate) from a partner's opportunities on KC  $k$  ( $N_{ik}$ ).

$$\ln \frac{p_{ij}}{1 - p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj} (\gamma_k N_{ik}) + \sum_k Q_{kj} (\rho_k N_{ik}) \quad (3)$$

By adding the learning from partner's opportunities to the model, we can capture how students learn from their partner's opportunities, when their role is to observe and provide help and advice. This provides insights into the importance of helping a partner's work. The model also may provide better predictions of student performance when working in a collaborative condition where the student's actions can be differentiated.

### 3.4 Additive Factors Model with Condition and Partner Opportunities

The final model combines the collaborative features of the previous two. This model takes into account both the differences in learning rates that may occur for a student's own opportunities between individual and collaborative learning (captured in Formula 2) and also includes the learning that may occur by observing a partner's opportunities while working collaboratively (captured in Formula 3). Please note that the  $c$  (condition term) was not included in the partner's opportunities, because students who work individually do not have any partner opportunities to observe, making the partner opportunities always be 0 for students working individually.

$$\ln \frac{p_{ij}}{1 - p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj} (\gamma_{kc} N_{ikc}) + \sum_k Q_{kj} (\rho_k N_{ik}) \quad (4)$$

This model combines the collaborative features of the previous two models to capture how these two ways of possibly benefitting from collaboration might balance.

## 4. RESULTS

For our analysis of the models, we evaluated the data from the procedurally-oriented tutor problems and the conceptually-oriented tutor problems separately to be able to see if the collaborative features that were added to the model have different effects for these two types of knowledge. Because students were assigned to either work on procedurally or conceptually oriented problems, there was no overlap in the students in the two datasets. Additionally, there was no overlap in the KCs in the datasets since any given KC captured either procedural or conceptual knowledge. With neither an overlap in students nor KCs between the datasets, the datasets can be analyzed separately, so as to analyze how collaboration (versus individual learning) might influence the learning of conceptual and procedural knowledge differently.

We measured the prediction accuracy of all of the models across the two data groups using the log likelihood, the root mean squared error on the training set (RMSE), and the Akaike information criterion (AIC). The log likelihood and RMSE provide a measure of fit not taking into account the complexity of the model. The AIC takes into account the complexity of the model when determining the fit of the model; it imposes a penalty based on the number of parameters. All of the models were run through a LIBLINEAR library in C [4]. Although in a standard AFM, the learning rate is restricted to be greater than or equal to zero, this restriction was not enforced in our models.

### 4.1 Procedurally-Oriented Problems

On the procedural dataset (see Table 1), the more complex models (i.e., the models that capture the influence of working with a partner in the ways discussed above) have a better fit in terms of log likelihood and RMSE, compared to the standard AFM. When comparing the models based on the AIC, all of the models that model aspects of collaborative learning have an improved AIC over the standard AFM. The AFM with Condition has the best AIC fit. Since the parameters are the same for the AFM with Condition and the AFM with Partner Opportunities, yet the former has a lower AIC, the condition the students are working in may be a better predictor of performance than having additional opportunities to observe a partner solving a step. Put differently, on procedural problems, having partner help when solving a step may influence learning more than helping a partner solve a step. It should be noted, however, that the difference in AIC between the two models is very small. The AIC for the model that combines the two collaborative features (AFM with Condition and

**Table 2. Prediction accuracy for the individual and collaborative conceptual dataset across all models. The asterisks indicates the model with the best performance for that criterion.**

Conceptual Models	Log Likelihood	RMSE	AIC	Parameters
Standard AFM	-1383.81	0.4815	2843.61	38
AFM with Condition	-1362.72	0.4804	2839.44	57
AFM with Partner Opportunities	-1359.67	0.4815	2833.33*	57
AFM with Condition and Partner Opportunities	-1344.50*	0.4772*	2841.01	76

Partner Opportunities) is higher, even though the log likelihood and RMSE are lower, indicating that the complexity of the model out-weighs the added gains.

## 4.2 Conceptually-Oriented Problems

For the models that were run on the conceptual dataset (see Table 2), the more complex models (i.e., those modeling how collaboration might influence learning) again have a better fit in terms of log likelihood and RMSE. As with the procedural dataset, these results indicate the importance of both the condition the students are working in (i.e., influence of partner help on the student's own opportunities) and of the partner opportunities (i.e., influence of helping a partner). When comparing the models based on the AIC, all of the models with collaborative features have an improved AIC over the standard AFM, and the AFM with Partner Opportunity has the best fit. Unlike with the procedural dataset, on conceptual problems, being able to observe a partner solving a step has more of an impact on predicted performance than condition.

## 5. DISCUSSION

AFMs are widely used models for predicting student performance. However, these models have mostly been used to predict the performance of students who are working individually. Students who are working collaboratively may go through different learning processes as they interact with other students, which currently are not accounted for in the standard AFM. In this paper, we wanted to see if adding collaborative features to AFM had an impact on the accuracy of the predicted learning performance of students in ITSs. Specifically, we investigated two mechanisms by which collaboration might influence learning. First, students might have different learning gains on steps they are responsible for solving because of the influence of a partner, such as through productive discussion or by being distracted. Second, a student might benefit from collaboration through engaging in discussion with a partner on steps that the partner is solving or by observing a partner as the partner solves the step. These two mechanisms were tested by two different ways of extending the AFM. First, we took into account the condition the student is working in (collaborative v. individual) by allowing the learning slope to vary based on condition. Second, we included the partner opportunities to capture the learning that may occur from observing/discussing a partner's answers to steps. These different learning mechanisms may differ for students who are working to acquire different types of knowledge. To take this into account, we analyzed our datasets for conceptual and procedural knowledge separately.

We first investigated if there is a difference between the learning rate of students working individually and those working collaboratively. To model the effect a partner may have on the steps that a student is responsible for solving, we added condition

as a feature to the learning slope parameter. For both the procedural dataset and the conceptual dataset, the models that included condition outperformed the standard AFM based on AIC and log likelihood. Condition may be a useful predictor to include in a model for performance when students work collaboratively (or even, alternate between working collaboratively or individually) to more accurately predict performance.

To answer the question if observing and working with a partner on the partner's opportunities has an impact on learning (the second mechanism by which collaborative learning might help), we added an additional learning slope for a partner's opportunities to the standard AFM. Again, for both the procedural and conceptual datasets, the models that included the partner's opportunities outperformed the standard AFM based on AIC and log likelihood. This indicates that observing and helping a partner solve problems has an impact on a student's learning when working on either procedurally oriented problems or conceptually oriented problems. A partner's opportunity to practice a KC may be important to include in a learning model where students have the potential to work with another student.

Although the models built on the procedural and conceptual datasets cannot be compared directly, we can observe some differences in the order of the model fits that may indicate differences in the importance of different learning processes when acquiring different types of knowledge. The best model for the procedural dataset was the AFM with Condition, whereas the best fitting model for the conceptual dataset was the AFM with Partner Opportunity. These differences in the best-fitting model may indicate that collaboration might influence learning differently when learning procedural knowledge than when learning conceptual knowledge. When students are acquiring conceptual knowledge, observing a partner or helping a partner solve a step may have more of an impact than when a student is acquiring procedural knowledge.

The work makes a number of contributions to the field of EDM. It is one of the few to address how standard student modeling techniques in EDM can be applied to collaborative learning. Our modified AFM model predicts student performance as students collaboratively solve problems. The model can be applied to learning in collaborative environments in which the actions of the students within a collaborating group can be distinguished. The work extends the AFM so it can be applied to collaborative learning, capturing two different mechanisms by which collaboration might help students learn with a collaborative ITS. By applying these new models to a data set on both collaborative and individual learning, the work demonstrates that these two mechanisms might both be at work in conceptual and procedural learning, although to varying degrees. These findings contribute to enhance the understanding of the relative strengths of collaborative and individual learning.

A limitation of this dataset is that we do not have a comparison between the difficulty of the procedural and conceptual datasets. Any differences between the models for these datasets may not be due to the type of knowledge that is being acquired but may be related to where the students were in the learning process for these different types of data while learning. For future work, we are interested in using these models for student data where the students switch between working individually and collaboratively on the same sets of KCs, both conceptual and procedural. By modeling this data using the new AFMs we have created, we can better understand how the models will generalize to a more natural learning situation in the classroom. In addition, the models can be applied to situations where students come to the collaboration with different skills to see how students learn the skills from their partner. The AFMs with the added parameters provide improved models to better predict when students have reached mastery while collaborating or working individually.

## 6. ACKNOWLEDGMENTS

We thank the CTAT team, Daniel Belenky, Ryan Carlson, and Michael Yudelson for their help. This work was supported by Graduate Training Grant # R305B090023 and by Award # R305A120734 both from the US Department of Education (IES).

## 7. REFERENCES

- [1] Boeck, P. 2008. Random Item IRT Models. *Psychometrika*, 73(4), 533–559.
- [2] Cen, H., Koedinger, K. R., and Junker, B. 2007. Is Over Practice Necessary? - Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. *In Proc. AIED*, pages 511–518.
- [3] Corbett, A. T. and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 4(4):253–278.
- [4] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9(2008), 1871-1874.
- [5] Janssen, J., & Bodemer, D. 2013. Coordinated computer-supported collaborative learning: Awareness and awareness tools. *Educational Psychologist*, 48(1), 40-55.
- [6] Li, Y., Wang, J., Liao, J., Zhao, D., and Huang, R. 2007. Assessing collaborative process in CSDL with an intelligent content analysis toolkit. *In ICALT*.
- [7] Lou, Y., Abrami, P. C., & d'Apollonia, S. 2001. Small group and individual learning with technology: A meta-analysis. *Review of educational research*, 71(3), 449-521.
- [8] Martinez-Maldonado, R., Yacef, K., & Kay, J. 2013. Data Mining in the Classroom: Discovering Groups' Strategies at a Multi-tabletop Environment. *Proc. EDM, 2013*.
- [9] McNely, B. J., Gestwicki, P., Hill, J. H., Parli-Horne, P., & Johnson, E. 2012. Learning analytics for collaborative writing: a prototype and case study. *In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 222-225). ACM.
- [10] Mercer, N. 2008. The seeds of time: Why classroom dialogue needs a temporal analysis. *The Journal of the Learning Sciences*, 17(1), 33-59.
- [11] Mullins, D., Rummel, N., & Spada, H. 2011. Are two heads always better than one? Differential effects of collaboration on students' computer-supported learning in mathematics. *International Journal of Computer-Supported Collaborative Learning*, 6(3), 421-443.
- [12] Nye, B. D., Hajeer, M., Forsyth, C. M., Samei, B., Millis, K., & Hu, X. 2014. Exploring real-time student models based on natural-language tutoring sessions.
- [13] Olsen, J. K., Belenky, D. M., Aleven, V., & Rummel, N. 2014. *Using an intelligent tutoring system to support collaborative as well as individual learning*. *In 12<sup>th</sup> International Conference on Intelligent Tutoring Systems*, 134-143. Springer International Publishing.
- [14] Olsen, J. K., Belenky, D. M., Aleven, A., Rummel, N., Sewall, J., & Ringenberg, M. 2014. Authoring Tools for Collaborative Intelligent Tutoring System Environments. *In 12th Int'l Conference on Intelligent Tutoring Systems*.
- [15] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. 2009. Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Online Submission*.
- [16] Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. 2009. Clustering and sequential pattern mining of online collaborative learning data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(6), 759-772.
- [17] Rafferty, A. N., Davenport, J., & Brunskill, E. 2013. Estimating Student Knowledge from Paired Interaction Data. *Proc. EDM*.
- [18] Reimann, P.: Time is precious 2009 Variable-and event-centred approaches to process analysis in CSDL research. *International Journal of Computer-Supported Collaborative Learning*, 4(3), 239-257.
- [19] Slavin, R. E. 1996. Research on cooperative learning and achievement: What we know, what we need to know. *Contemporary educational psychology*, 21(1), 43-69.
- [20] Stahl, G., Koschmann, T., and Suthers, D.. 2006. Computer supported collaborative learning: An historical perspective. *In R. K. Sawyer, editor, Cambridge Handbook of the Learning Sciences*. Cambridge University Press.
- [21] Suthers, D., & Chu, K. H. 2012. Multi-mediated community structure in a socio-technical network. *In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 43-53). ACM.
- [22] Wen, M., Yang, D., & Rosé, C. P. 2014. Sentiment Analysis in MOOC Discussion Forums: What does it tell us. *Proceedings of Educational Data Mining*.
- [23] Wise, A. F., & Chiu, M. M. 2011. Analyzing temporal patterns of knowledge construction in a role-based online discussion. *International Journal of Computer-Supported Collaborative Learning*, 6(3), 445-470.
- [24] Xu, X., Murray, T., Woolf, B. P., & Smith, D. 2013. Mining Social Deliberation in Online Communication--If You Were Me and I Were You. *In Educational Data Mining*.
- [25] Zhang, X., Mostow, J., & Beck, J. E. 2007. All in the (word) family: Using learning decomposition to estimate transfer between skills in a Reading Tutor that listens. *In AIED2007 Educational Data Mining Workshop* (pp. 80-87).

# Learning Instructor Intervention from MOOC Forums: Early Results and Issues

Muthu Kumar  
Chandrasekaran<sup>1</sup>

Min-Yen Kan<sup>1</sup>

Bernard C.Y. Tan<sup>2</sup>

Kiruthika Ragupathi<sup>3\*</sup>

<sup>1</sup> Web IR / NLP Group (WING)

<sup>2</sup> Department of Information Systems

<sup>3</sup> Centre for Development of Teaching and Learning  
National University of Singapore

{muthu.chandra, kanmy}@comp.nus.edu.sg, {pvotcy, kiruthika}@nus.edu.sg

## ABSTRACT

With large student enrollment, MOOC instructors face the unique challenge in deciding when to intervene in forum discussions with their limited bandwidth. We study this problem of *instructor intervention*. Using a large sample of forum data culled from 61 courses, we design a binary classifier to predict whether an instructor should intervene in a discussion thread or not. By incorporating novel information about a forum's type into the classification process, we improve significantly over the previous state-of-the-art.

We show how difficult this decision problem is in the real world by validating against indicative human judgment, and empirically show the problem's sensitivity to instructors' intervention preferences. We conclude this paper with our take on the future research issues in intervention.

## Keywords

MOOC; Massive Open Online Course; Instructor Intervention; Discussion Forum; Thread Recommendation

## Categories and Subject Descriptors

H.3.3. [Information Search and Retrieval]: Information filtering;  
K.3.1. [Computers and Education]: Computer Uses in Education

## 1. INTRODUCTION

MOOCs scale up their class size by eliminating synchronous teaching and the need for students and instructors to be co-located. Yet, the very characteristics that enable scalability of massive open online courses (MOOCs) also bring significant challenge to its teach-

\*This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

ing, development and management [7]. In particular, scaling makes it difficult for instructors to interact with the many students — the lack of interaction and feelings of isolation have been attributed as reasons for why enrolled students drop from MOOCs [9].

MOOC discussion forums are the most prominent, visible artifact that students use to achieve this interactivity. Due to scale of contributions, these forums teem with requests, clarifications and social chatting that can be overwhelming to both instructors and students alike. In particular, we focus on how to best utilize instructor bandwidth: with a limited amount of time, which threads in a course's discussion forum merit instructor intervention? When utilized effectively, such intervention can clarify lecture and assignment content for a maximal number of students, promoting the enhancing the learning outcomes for course students.

To this end, we build upon previous work and train a binary classifier to predict whether a forum discussion thread merits instructor intervention or not. A key contribution of our work is to demonstrate that prior knowledge about forum type enhances this prediction task. Knowledge of the enclosing forum type (i.e., discussion on *lecture*, *examination*, *homework*, etc.) improves performance by 2.43%; and when coupled with other known features disclosed in prior work, results in an overall, statistically significant 9.21% prediction improvement. Additionally, we show that it is difficult for humans to predict the actual interventions (the gold standard) through an indicative manual annotation study.

We believe that optimizing instructor intervention is an important issue to tackle in scaling up MOOCs. A second contribution of our work is to describe several issues pertinent for furthering research on this topic that emerge from a detailed analysis of our results. In particular, we describe how our work at scale details how personalized and individualized instructor intervention is — and how a framework for research on this topic may address this complicating factor through the consideration of normalization, instructor roles, and temporal analysis.

## 2. RELATED WORK

While the question of necessity of instructor's intervention in online learning and MOOCs is being investigated [12, 20], technologies to enable timely and appropriate intervention are also required.

The pedagogy community has recognized the importance of instructor intervention in online learning prior to the MOOC era (e.g., [10]). Taking into consideration the pedagogical rationale for effective intervention, they also proposed strategic instructor postings: to guide discussions, to wrap-up the discussion by responding to unanswered questions, with “Socrates-style” follow-up questions to stimulate further discussions, or with a mixture of questions and answers [13]. However, these strategies must be revisited when being applied to the scale of typical MOOC class sizes.

Among works on forum information retrieval, we focus on those that focus on forum moderation as their purpose is similar to the instructor’s role in a course forum. While early work focused on automated spam filtering, recent works shifted focus towards curating large volumes of posts on social media platforms [4] to distil the relevant few. Specifically, those that strive to identify thread solvedness [21, 8] and completeness [3] are similar to our problem.

Yet all these work on general forums (e.g., troubleshooting, or threaded social media posts) are different from MOOC forums. This is due to important differences in the objectives of MOOC forums. A typical thread on a troubleshooting forum such as Stack Overflow is centered on questions and answers to a particular problem reported by a user; likewise, a social media thread disseminates information mainly to attract attention. In contrast, MOOC forums are primarily oriented towards learning, and also aim to foster learning communities among students who may or may not be connected offline.

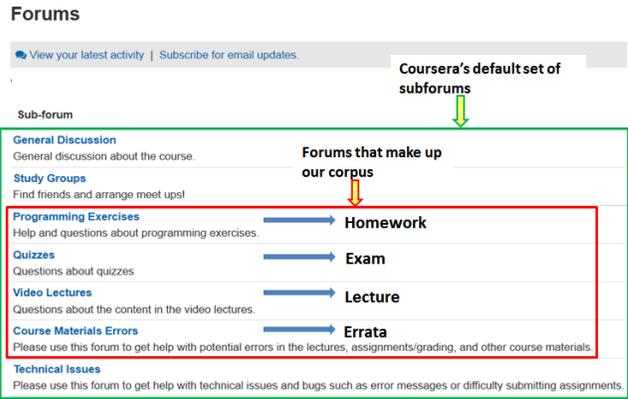
Further, strategies for thread recommendation for students such as [23] may not apply in recommending for instructors. This difference is partially due to scale: while the number of students and threads are large, there are few instructors per course. In this case, reliance on collaborative filtering using a user–item matrix is not effective. Learning from previous human moderation decisions [2], therefore, becomes the most feasible approach. Prior work on MOOC forums propose categorisation of posts [16, 5, 19] to help instructors identify threads to intervene. Chaturvedi *et al.* [5], the closest related work to ours, show each of the four states of their sequence models to predict instructor intervention to be distributed over four post categories they infer. In this paper, we use their results for comparison.

Different from previous works, we propose thread–level categories rather than post–level categories, since an instructor needs to first decide on a thread of interest. Then they need to read its content, at least in part, before deciding whether to intervene or not. We make the key observation that show thread–level categories identified as by the forum type, help to predict intervention.

Previous work has evaluated only with a limited number of MOOC instances. One important open question is whether those reported results represent the diverse population of MOOCs being taught. In this paper, we address this by testing on a large and diverse cross-section of Coursera MOOC instances.

### 3. METHODS

We seek to train a binary classifier to predict whether a MOOC forum thread requires instructor intervention. Given a dataset where instructor participation is labeled, we wish to learn a model of thread intervention based on qualities (i.e., features) drawn from the dataset. We describe our dataset, the features distilled used for our classifier, how we obtain class labels, and our procedure for instance weighting in the following.



**Figure 1: Typical top-level forum structure of a Coursera MOOC, with several forums. The number of forums and their labels can vary per course.**

Forum type	All		Intervened	
	# threads	# posts	# threads	# posts
<b>D61 Corpus</b>				
Homework	14,875	127,827	3993	18,637
Lecture	9,135	64,906	2,688	10,051
Errata	1,811	6,817	654	1,370
Exam	822	6,285	405	1,721
<b>Total</b>	<b>26,643</b>	<b>205,835</b>	<b>7,740</b>	<b>31,779</b>
<b>D14 Corpus</b>				
Homework	3,868	31,255	1,385	6,120
Lecture	2,392	13,185	1,008	3,514
Errata	326	1,045	134	206
Exam	822	6,285	405	1,721
<b>Total</b>	<b>7,408</b>	<b>51,770</b>	<b>2,932</b>	<b>11,561</b>

**Table 1: Thread statistics from our 61 MOOC Coursera dataset and the subset of 14 MOOCs, used in the majority of our experiments.**

### 3.1 Dataset

For our work, we collected a large-scale, multi-purpose dataset of discussion forums from MOOCs. An important desideratum was to collect a wide variety of different types of courses, spanning the full breadth of disciplines: sciences, humanities and engineering. We collected the forum threads<sup>1</sup> from 61 completed courses from the Coursera platform<sup>2</sup>, from April to August 2014, amounting to roughly 8% of the full complement of courses that Coursera offers<sup>3</sup>.

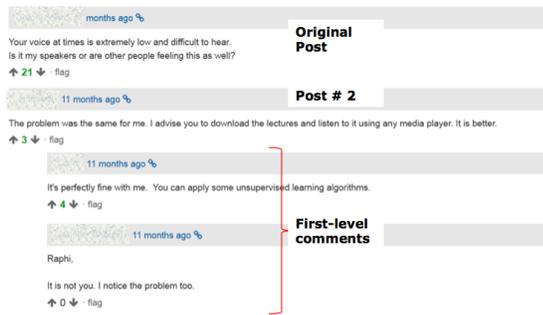
For each course, we first assigned each forum<sup>4</sup> to one of several types based on the forum’s title. For this study we focus on threads that originated from four prevalent types: (i) errata or course material errors, (ii) video lectures, (iii) homework, assignments or prob-

<sup>1</sup>We collected all threads and their component posts from four sub-forum categories as in Section 3.1. We did so, as we hypothesize that they would necessitate different levels of instructor intervention and that such interventions may be signaled by different features.

<sup>2</sup>The full list of courses is omitted here due to lack of space.

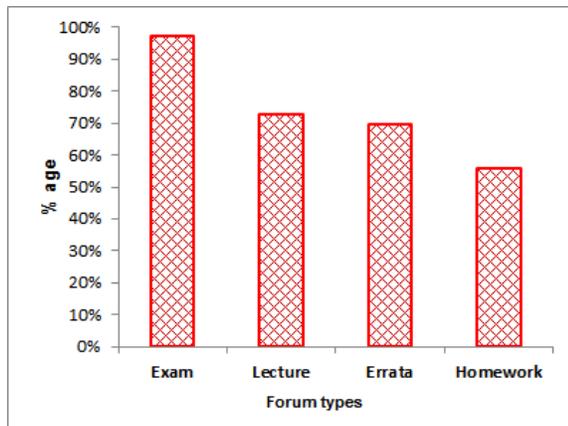
<sup>3</sup>As of December 2014, Coursera, a commercial MOOC platform: <https://www.coursera.org>, hosted 761 courses in English spanning 25 different subject areas.

<sup>4</sup>“Subforum” in Coursera terminology.



**Figure 2: Coursera’s forums allow threads with posts and a single level of comments.**

lem sets, and (iv) exams or quizzes (see Figure 1)<sup>5</sup>. All 61 courses had forums for reporting errata and discussing homework and lectures. For more focused experimentation, we selected the 14 largest courses within the 61 that exhibited all four forum types (denoted “D14” hereafter, distinguished from the full “D61” dataset). Table 1 provides demographics of both D61 and D14 datasets. In our corpus, there were a total of 205,835 posts including posts and comments to posts. The Coursera platform only allows for a single level of commenting on posts (Figure 2). We note that this limits the structural information available from the forum discourse without content or lexical analysis. We observed that posts and comments have similar topics and length, perhaps the reason why previous work [18] ignored this distinction. We have retained the distinction as it helps to distinguish threads that warrant interven-



**Figure 3: Thread distribution over errata, homework, lecture and exam forums in D14 by their *intervention ratio*.**

### 3.2 System Design

From the dataset, we extract the text from the posts and comments, preserving the author information, thread structure and posting timestamps, allowing us to recreate the state of the forum at any timestamp. This is important, as we first preprocessed the dataset to remove inadmissible information. For example, since we collected the dataset after all courses were completed, instructors’ posts as

<sup>5</sup>Some courses had forums for projects, labs, peer assessment, discussion assignments. We omit from the collection these and other miscellaneous forums, such as those for general discussion, study groups and technical issues.

well as any subsequent posts in a thread need to be removed. We also do not use the number of votes or views in a thread as these are summary counts that are influenced by intervention<sup>6</sup>.

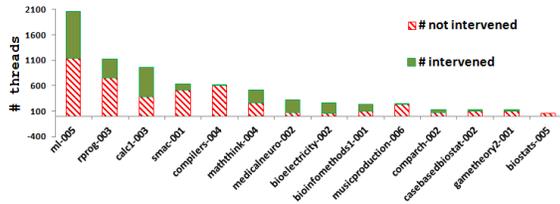
We used regular expressions to further filter and canonicalize certain language features in the forum content. We replaced all mathematical equations by <MATH>, URLs by <URLREF> and references to time points in lecture videos by <TIMEREF>. We removed stopwords, case-folded all tokens to lowercase, and then indexed the remaining terms and computed the product of term frequency and inverse thread frequencies ( $tf \times itf$ ) for term importance. The weighted terms form a term vector that we further normalized using the L2-norm. Other real-valued features were max-min normalized. Categorical features such as the forum type were encoded as bit vectors.

Each thread is represented as bag of features consisting of terms and specific thread metadata as disclosed below. We indicate each new feature that our study introduces with an asterisk.

1. Terms (unigrams);
- 2\*. Forum type to which the thread belongs: Figure 3 shows a clear difference in *intervention ratio*, the ratio of number of threads intervened to those that weren’t, across different forum types. Forum type thus emerges as a feature to use to discriminate threads worthy of intervention. The forum type encapsulating the thread could be one of homework, lecture, exam or errata.
- 3\*. Number of references to course materials and other sources external to the course: includes explicit references by students to course materials within and outside the course e.g., *slide 4, from wikipedia, lecture video 7*.
- 4\*. Affirmations by fellow students; Count of agreements made by fellow students in response to a post. Mostly, first posts in a thread receive affirmations.
5. Thread properties (Number of posts, comments, and both posts / comments, Average number of comments per post): expresses a thread’s length and structural properties in terms of number of posts and comments posted.
6. Number of sentences in the thread: This feature intends to capture long focussed discussions that may be intervened more often than the rest.
- 7\*. Number of non-lexical references to course materials: (number of URLs, references to time points in lecture videos). This feature is similar to course material references but includes only non-lexical references (Item #1) such as URLs and time points in lecture videos.

Importantly, as part of the author information, Coursera also marks instructor-intervened posts / comments. This supplies us with automatically labeled gold standard data for both training and evaluating our supervised classifier. We use threads with instructor posts / comments as positive instances (intervened threads). However, we note that the class imbalance is significant: as the instructor-student ratio is very low, typical MOOC forums have fewer positives (interventions) than negative ones. To counter skewness, we weigh

<sup>6</sup>Previous work such as [5] utilize this as they have access to time-stamped versions of these statistics, since they use privately-held data supplied by Coursera for MOOCs held at their university.



**Figure 4: Thread distribution over the errata, homework, lecture and exam forums in D14. Corresponds to numeric data in Table 2.**

majority class (generally positive) instances higher than minority class (generally negative) instances. These weights are important parameters of the model, and are learned by optimizing for maximum  $F_1$  over the training / validation set.

## 4. EVALUATION

We performed detailed experimentation over the smaller D14 dataset to validate performance, before scaling to the D61 dataset. We describe these set of experiments in turn. As our task is binary classification, we adopted L1-regularized logistic regression as our supervised classifier in all of our experimentation.

We first investigated each of the 14 courses in D14 as 14 separate experiments. We randomly used 80% of the course’s threads for training and validation (to determine the class weight parameter,  $W$ ), and use the remaining 20% for testing. Our experimental design for this first part closely follows the previous work [5] for direct comparison with their work. We summarise these results in Table 2, in the columns marked “(II) Individual”, averaging performance over ten-fold cross validation for each course.

The results show a wide range in prediction performance. This casts doubt on the portability of the previously published work [5]. They report a baseline performance of  $F_1 \approx 25$  on both their courses each having an intervention ratio  $\approx 0.13^7$ . In contrast, our results show the instability of the prediction task, even when using individualized trained models. Nevertheless, on average our set of features performs better on  $F_1$  by at least 10.15%.<sup>8</sup>

We observe the true intervention ratio correlates to performance, when comparing Columns I.2 and II.3 ( $\rho = 0.93$ ). We also see that intervention ratio varies widely in our D14 dataset (Figure 4). This happens to also hold for the larger D61 dataset. In some courses, instructors intervene often (76% for medicalneuro-002) and in some other courses, there is no intervention at all (0% for biostat-005).

To see whether the variability can be mitigated by including more data, we next perform a leave-one-course-out cross validation over the 14 courses, shown in “Columns (III) LOO-course C.V.”. *I.e.*, we train a model using 13 courses’ data and apply the trained model to the remaining unseen test course. While not strictly comparable with (II), we feel this setting is more appropriate, as it: allows training to scale; is closer to the real scenario discussed in Section 6, Item 4.

Separately, we studied the effectiveness of our proposed set of fea-

<sup>7</sup>Based on test data figures [5] had disclosed in their work

<sup>8</sup>Due to non-availability of experimental data, we can only claim a 10.15% improvement over the highest  $F_1$  they reported, 35.29.

Feature	Precision	Recall	$F_1$
1. Unigrams	41.98	61.39	45.58
2. (1) + Forum Type	41.36	69.13	48.01
3. (2) + Course_Ref	41.09	66.57	47.22
4. (3) + Affirmation	41.20	68.94	47.68
5. (4) + T Properties	42.99	70.54	48.86
6. (5) + Num Sents	43.08	69.88	49.77
7. (6) + Non-Lex Ref	42.37	74.11	50.56
8. (7) – Forum Type	41.33	83.35	51.16
9. (7) – Course Ref	45.96	79.12	54.79
10. (7) – Affirmation	42.59	71.76	50.34
11. (7) – T Properties	40.62	84.80	51.35
12. (7) – Num Sents	42.37	73.05	49.32
13. (7) – Non-Lex Ref	43.08	69.88	49.77

**Table 3: Feature study. The top half shows performance as additional features are added to the classifier. Ablation tests where a single feature is removed from the full set (Row 7) are shown on the bottom half. Performance given as weighted macro-average over 14 courses from a leave-one-out cross course validation over D14.**

tures over the D14 dataset. Table 3 reports performance averaged over all 14 courses weighted by its proportion in the corpus. In the top half of the table, we build Systems 1–7 by cumulatively adding in features from the proposed list from Section 3.2. Although the overall result in Row 7 performs  $\sim 5\%$  better than the unigram baseline, we see that the classifier worsens when the count of course references are used as a feature (Row 2). Other rows all show an additive improvement in  $F_1$ , especially the forum type and non-lexical reference features, which boost recall significantly.

The performance drop when adding in the number of course references prompts us to investigate whether removing some features from the full set would increase prediction quality. In the bottom half of Table 3, we ablate a single feature from the full set.

Results show that removing forum type, number of course references and thread length in a thread all can improve performance. Since the different rows of Table 3 are tested with weights  $W$  learnt from its own training set the changes in performance observed are due to the features and the learnt weight. When we tested the same sequence with an arbitrary constant weight we observed all features but Course\_Ref improved performance although not every improvement was significant.

Using the best performing feature set as determined on the D14 experiments, we scaled our work to the larger D61 dataset. Since a leave-one-out validation of all 61 courses is time consuming we only test on the each of the 14 courses in D14 dataset while training on the remaining 60 courses from D61. We report a **weighted averaged  $F_1 = 50.96$  ( $P = 42.80$ ;  $R = 76.29$ )** which is less than row 9 of Table 3. We infer that scaling the dataset by itself doesn’t improve performance since  $W$  learnt from the larger training data no longer counters the class imbalance leaving the testset with a much different class distribution than the training set.

### 4.1 Upper bound

The prediction results show that forum type and some of our newly-proposed features lead to significant improvements. However, we suspect the intervention decision is not entirely objective; the choice to intervene may be subjective. In particular, our work is based on

Course	(I) Demographics		(II) Individual				(III) LOO-course C.V.			
	1. # of Threads	2. I. Ratio	1. Prec.	2. Rec.	3. $F_1$	4. $W$	1. Prec.	2. Rec.	3. $F_1$	4. $W$
ml-005	2058	0.45	51.08	89.19	64.96	2.06	48.10	68.63	56.56	2.46
rprog-003	1123	0.32	50.77	48.53	49.62	2.41	35.88	75.77	48.70	2.45
calc1-003	965	0.60	60.98	44.25	51.29	0.65	65.42	72.79	68.91	2.45
smac-001	632	0.17	21.05	30.77	25.00	5.29	22.02	67.93	33.26	2.00
compilers-004	624	0.02	8.33	50.00	14.28	37.23	2.53	80.00	4.91	2.33
maththink-004	512	0.49	46.59	100.00	63.56	2.13	50.24	85.48	63.29	2.57
medicalneuro-002	323	0.76	100.00	60.47	75.36	0.32	75.86	89.07	81.94	2.34
bioelectricity-002	266	0.76	75.00	54.55	63.16	0.34	75.36	82.98	78.99	2.41
bioinformethods1-001	235	0.55	56.00	60.87	58.33	0.78	59.67	83.72	69.68	2.36
musicproduction-006	232	0.01	0.00	0.00	0.00	185.00	0.52	50.00	1.03	2.55
comparch-002	132	0.46	47.62	100.00	64.57	1.56	48.57	83.61	61.45	2.37
casebasedbiostat-002	126	0.20	13.33	100.00	23.53	3.54	24.47	92.00	38.66	2.11
gametheory2-001	125	0.19	28.57	28.57	28.57	5.18	18.27	86.36	30.16	2.61
biostats-005	55	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	2.01
Average	529	0.36	39.95	54.80	41.59	17.68	37.64	72.74	45.54	2.36
Weighted Macro Avg	NA	0.40	45.44	61.84	49.04	10.96	42.37	74.11	50.56	2.37

**Table 2: Individual course results for each course in the D14 dataset. Weights  $W$  weigh each +ve class instance  $w$  times as much as a -ve class instance. Performance varies with large variations in Intervention ratio (I-ratio) and # of threads.**

the premise that correct intervention follows the historical pattern of intervention (where instructors already intervene), and may not be where general pedagogy would recommend prediction. We recognize this as a limitation of our work.

To attempt to quantify this problem, we assess whether peer instructors with general teaching background could replicate the original intervention patterns. Three human instructors<sup>9</sup> annotated 13 threads from the musicproduction-006 course. We chose this course to avoid bias due to background knowledge, as none of the annotators had any experience in music production. This course also had near zero interventions; none of the 13 threads in the sample were originally intervened by the instruction staff of the course.

They annotated 6 exam threads and 7 lecture threads. We found that among exam threads annotators agreed on 5 out of 6 cases. Among lecture threads at least two of three annotators always agreed. On 4 out of 7 cases, all three agreed. The apparently high agreement could be because all annotators chose to intervene only on a few threads. This corresponds to a averaged interannotator agreement of  $k = 0.53$ . The annotators remarked that it was difficult to make judgements, that intervention in certain cases may be arbitrary, especially when expert knowledge would be needed to judge whether factual statements made by students is incorrect (thus requiring instructor intervention to clarify). As a consequence, agreement on exam threads that had questions on exam logistics had more agreement at  $k = 0.73$ .

While only indicative, this reveals the subjectiveness of intervention. Replicating the ground truth intervention history may not be feasible – satisfactory performance for the task may come closer to the interannotator agreement levels: i.e.,  $k = 0.53$  corresponding to an  $F_1$  of 53%. We believe this further validates the significance of the prediction improvement, as the upper bound for deciding intervention is unlikely to be 100%.

<sup>9</sup>The last three authors, not involved in the experimentation: two professors and a senior pedagogy researcher.

## 5. DISCUSSION

From handling the threads and observing discussion forum interactions across courses, several issues arise that merit a more detailed discussion. We discuss each in turn, identifying possible actions that may mollify or address these concerns. Specifically:

1. The number of threads per course varies significantly.
2. Intervention decisions may be subjective.
3. Simple baselines outperform learned systems.
4. Previous experimental results are not replicable.

**Issue 1: Variation in the number of threads.** We observed significant variation in the number of threads in different courses, ranging from tens to thousands. Figure 4 shows thread distribution over the D14 dataset for the errata, homework, lectures and exam forums; a similar distribution held for the larger D61 dataset. These distributions are similar to those reported earlier in the large cross-course study of [18]. The difference in number of threads across courses is due to a multitude of factors. These include number of students participating, course structure, assignment of additional credits to participating students, course difficulty, errors in course logistics and materials, etc.

When performing research that cuts across individual MOOCs, this issue becomes important. As we saw, using simple averaging on a per-course basis equates to a macro-averaging: putting each course on par in importance. However, when the decision unit is at the thread (as in our task), it makes more sense to treat individual threads at parity. In such cases, normalization at the thread level (analogous to micro-averaging) may be considered. Such thread-level normalization can affect how we weight information from each course when training in aggregate over data from multiple courses: courses with many threads should carry more weight in both training and evaluation.

**Issue 2: Intervention decisions may be subjective.** Instructor policy with respect to intervention can markedly differ. Instructors may only intervene in urgent cases, preferring students to do peer learning and discovery. Others may want to intervene often,

to interact with the students more and to offer a higher level of guidance. Which policy instructors adopt varies, as best practices for both standard classroom and MOOC teaching have shown both advantages and disadvantages for [12, 11].

Instructors can also manifest in different roles. In Coursera, posts and comments marked as instructor intervened can come from actual instructor intervention as well as participation by helpers, such as community teaching assistants (CTAs). We observe courses with CTAs where CTAs have a higher intervention rate. We hypothesize that such factors decreases agreements.

This plays out in our datasets. We observe that intervention is not always proportional to the number of threads in the course. Some courses such as compilers-004 (see Figure 4) has relatively fewer number of threads than other large courses. Yet its intervention rate is noticeably low. This suggests that other factors inform the intervention decision. Handling this phenomenon in cross-course studies requires an additional form of normalization.

To normalize for these different policies we can upweight (by oversampling) threads that were intervened in courses with fewer interventions. We can continue to randomly oversample a course’s intervened threads until its *intervention density* reaches the dataset average. Note this normalization assumes that the few threads intervened in course with relatively low intervention density are more important; that the threads intervened for a similar high intervention density course would be a proper superset.

Even when a policy is set, intervention decisions may be subjective and non-replicable. Even with our cursory annotation of a course to determine an upper bound for intervention shows the potentially large variation in specific intervention decisions. We believe that automated systems can only approach human performance when such decisions can be subjective. As such, the upper bound for performance (cf Section 4.1) should not be construed as the single gold standard; rather, prediction performance should be calibrated to human performance levels.

**Issue 3: Simple baselines outperform learned models.** We also compared our work with a simple baseline that predicts all threads as needing instructor intervention. This baseline does no work – achieving 100% recall and minimal precision – but is very competitive, outperforming our learned models for courses with high levels of intervention (see Table 4). Diving deeper into the cause, we attribute this difference to the subjective nature of interventions and other extraneous reasons (bandwidth concerns) resulting in high false positive rates. That is, given two threads with similar set of features, one may be intervened while the other is not (e.g., Figure 5). This makes the ground truth and the evaluation less reliable. An alternative evaluation model might be to assign a confidence score to a prediction and evaluate the overlap between the high confidence predictions and the ground truth interventions.

**Issue 4: Previous results are not replicable.** From earlier work [5], intervention prediction seemed to be straightforward task where improvement can be ascribed to better feature engineering. However, as we have discovered in our datasets, the variability in instructor intervention in MOOCs is high, making the application of such previously published work to other MOOCs difficult. This is the perennial difficulty of replicating research findings. Findings from studies over a small corpus with select courses from specific subject categories may not generalise. Published findings are not verifiable due to restricted access to sensitive course data. The

Course	Individual		D14	
	$F_1$	$F_1@100R$	$F_1$	$F_1@100R$
ml-005	64.96	63.79	72.35	61.83
rprog-003	49.62	47.39	48.55	49.31
calc1-003	51.29	74.83	70.63	75.33
smac-001	25.00	34.67	34.15	29.28
compilers-004	14.28	3.28	4.82	4.75
maththink-004	63.56	63.08	61.11	65.49
medicalneuro-002	75.36	88.66	78.06	85.67
bioelectricity-002	63.16	86.84	80.10	85.84
bioinfomethods1-001	58.33	67.65	69.40	71.07
musicproduction-006	0.00	4.35	1.09	1.72
comparch-002	64.57	55.56	60.49	63.21
casebasedbiostat-002	23.53	14.81	38.71	34.25
gametheory2-001	28.57	45.16	27.12	30.56
biostats-005	0.00	0.00	0.00	0.00
Average	41.59	46.43	45.18	47.09
Weighted Macro Avg	49.04	51.51	54.79	53.22

**Table 4: Comparison of  $F_1$  in Table 2 with those of a naïve baseline that classifies every instance as +ve – resulting in 100% recall.**

problem is acute for discussion forum data due to privacy and copyright considerations of students who have authored posts on those forums.

The main challenge is to provision secured researcher access to the experimental data. Even in cases where researchers have access to larger datasets, such prior research [1, 5, 14, 15, 16, 22] have reported findings on each course separately (cf Table 2 “(II) Individual”), shying away from compiling them into a single dataset in their study. Bridging this gap requires cooperation among interested parties. The shared task model is one possibility: indeed, recently Rose *et al.* [17] organised a shared task organised to predict MOOC dropouts over each week of a MOOC. To effectively make MOOC research replicable, data must be shared to allow others to follow and build on published experimentation. Similar to other communities in machine learning and computational linguistics, the community of MOOC researchers can act to legislate data sharing initiatives, allowing suitably anonymized MOOC data to be shared among partner institutes.

We call for the community to seize this opportunity to make research on learning at scale more recognizable and replicable. We have gained the endorsement of Coursera to launch a data-sharing initiative with other Coursera-partnered universities. While we recognize the difficulties of sharing data from the privacy and institutional review board perspectives, we believe that impactful research will require application to a large and wide variety of courses, and that restricting access to researchers will alleviate privacy concerns.

## 6. A FRAMEWORK FOR INTERVENTION RESEARCH

We have started on the path of instructor intervention prediction, using the task formalism posed by previous work by Chaturvedi *et al.* [5]: the binary prediction of whether a forum discussion thread should be intervened, given its lexical contents and metadata. While we have improved on this work and have encouraging results, this binary prediction problem we have tackled is overly constrained and does not address the real-world need for intervention prediction. We outline a framework for working towards the real-world needs of instructor intervention.

Forums / Programming Assignments

## When is hard deadline assignment of PA2? **INTERVENED (+vE)**

Subscribe for email updates.

ProgrammingAssignments × deadline × + Add Tag

10 months ago

On one hand, according to the website it is 6th of June, but the Assignment reads 2nd of July.

0 ↓ · flag

10 months ago

I would assume it is June 6th since that is the date listed on Coursera. It's possible that the July 2nd date was from a previous iteration of the class and the pdf file never got updated.

0 ↓ · flag

+ Comment

**INSTRUCTOR** · 10 months ago

The hard deadline is the last day of the class (June 6). As suspected, the July 2 reference is a missed update from an earlier offering—sorry about that.

1 ↓ · flag

+ Comment

Forums / Programming Assignments

## Deadline for PA4 correct **NOT INTERVENED (-vE)**

Subscribe for email updates.

deadline × pa-4 × + Add Tag

9 months ago

While PA1-PA3 have a hard deadline of "Thu 5 Jun 2014 1:59 AM CEST", PA4 has a hard deadline of Tue 27 May 2014.

Additionally the website tells us for PA4 (source: <https://class.coursera.org/compilers-004/>): "The time to complete this part of the project is roughly the same as the third assignment". Since we had 3 weeks for PA3 this does not fit the PA4 deadline let alone the non-hard deadline.

Thus I ask whether the deadline is really correct?

7 ↓ · flag

9 months ago

Not sure why the first three assignments have a later hard deadline. As for the duration of time given to complete PA3, it was just over 2 weeks (4/26-5/12), while PA4 is exactly 2 weeks. I think what's meant by "The time to complete this project..." is the time it normally takes a student to complete the assignment, not the time between the assigned date and the due date.

2 ↓ · flag

+ Comment

9 months ago

Agree: we could use a clarification on hard deadlines for PAs. (I've seen this done two ways on Coursera: either all fall on the same "last call" closing date of the class or they are two weeks (say) after their respective soft deadlines. Last assignment cutting off before earlier ones doesn't make sense.)

**Figure 5: Interventions are, at times, arbitrary. We show two threads from compilers-001 with similar topics, context, and features that we model (red underline). Yet only one of them is intervened (circled in red).**

We thus propose a framework for investigation that iteratively relaxes our problem to take into account successively more realistic aspects of the intervention problem, with the hope of having a fieldable, scalable, real-time instructor intervention tool for use on MOOC instructors' dashboard as an end result.

**1. Thread Ranking.** We posit that different types of student posts may exhibit different priorities for instructors. A recommendation for intervention should also depend on thread criticality. For example, threads reporting errors in the course material may likely be perceived as critical and hence should be treated as high-priority for intervention. Even with designated errata forums, errata are reported in other forums, sometimes due to the context – *e.g.*, when a student watches a video of a lecture, it is natural for him to report an error concerning it in the lecture forum, as opposed to the proper place in the errata forum. Failure to address threads by priority could further increase the course's dropout rate, a well-known problem inherent to MOOCs [6]. Thread ranking can help to address this problem to prioritize the threads in order of urgency, which the naïve, always classifying all instances as positive, baseline system cannot perform.

**2. Re-intervention.** Threads can be long and several related concerns can manifest within a single thread, either by policy or by serendipity. Predicting intervention at the thread level is insufficient to address this. A recommendation for intervention has to consider not only those threads that had been newly-created but also if older threads that had already been intervened require further intervention or *reintervention*. In other words, intervention decision needs to be made in the light of newly posted content to a thread. We can change the resolution of the intervention prediction problem to one at the post level, to capture re-interventions; *i.e.*, when a new post within a thread requires further clarification or details from instructor staff.

**3. Varying Teaching Roles.** MOOCs require different instruction formats than the traditional course format. One evolution of the MOOC teaching format to adapt to the large scale is to recruit community teaching assistants (CTA)s. Community TAs are volunteer TAs recruited by MOOC platforms including Coursera based on their good performance in the previous iteration of the same MOOC. CTAs, traditional Teaching Assistants and technical staff are all termed as "staff" within the Coursera system. Currently, Coursera only marks threads with a "staff replied" marking, which we use directly in our training supervision in this paper. At a post level, those posted by CTAs, instructor and technical staff are marked appropriately.

We hypothesize that that these various roles differ in the quantum of time and effort, and type of content that they provide in answering posts that they contribute on a forum. It will be important to consider the role of the user while recommending threads to intervene, as the single problem of intervention may lead to  $n$  separate triaging problems for the  $n$  staff types or individual instructors that manage a MOOC.

**4. Real-time.** In the real world, a system needs to be predicting intervention in real-time; as new posts come into a course's forum. With ranking, we can decide when to push notifications to the instruction staff, as well as those less urgent that can be viewed at leisure on the instructor's MOOC dashboard.

With the timestamp metadata in the dataset, we have a transaction log. This allows us to easily simulate the state of a MOOC by "rewinding" the state of the MOOC at any time  $t$ , and make a prediction for a post or thread based on the current state.

This half-solves the problem. For real-world use, we also need to do online learning, by observing actual instructor intervention and adopting our system for the observed behavior. We feel this will be important to learn the instructor's intervention preference, as we have observed the variability in intervention per course, per instructor.

In our work, we focus only on the *instructor's view*, however this set of problems also has an important dual problem set: that of the *student's view*. We believe that solving both problems will have certain synergies but will differ in important ways. For example, solving the student's view will likely have a larger peer and social component than that for instructors, as MOOCs develop more social sensitivity.

## 7. CONCLUSION

We describe a system for predicting instructor intervention in MOOC forums. Drawing from data over many MOOC courses from a wide variety of coursework, we devise several novel features of forums that allow our system to outperform the state-of-the-art work on an average by a significant margin of 10.15%. In particular, we find that knowledge of where the thread originates from (the forum type – whether it appears in a *lecture*, *homework*, *examination* forum) alone informs the intervention decision by a large 2% margin.

While significant in its own right, our study also uncovers issues that we feel must be accounted for in future research. We have described a framework for future research on intervention, that will allow us to account for additional factors – such as temporal effects, differing instructor roles – that will result in a ranking of forum threads (or posts) to aid the instructor in best managing her time in answering questions on MOOC forums.

Crucially, we find the amount of instructor intervention is widely variable across different courses. This variability undermines the veracity of previous works and shows that what works on a small scale may not hold well in large, cross-MOOC studies. Our own results show that for many courses, simple baselines work better than supervised machine learned models when intervention ratios are high. To allow the replicability of research and to advance the field, we believe that MOOC-fielding institutions need to form a data consortium to make MOOC forum data available to researchers.

## 8. ACKNOWLEDGMENTS

The authors would like to thank Snigdha Chaturvedi and her co-authors for their help in answering detailed questions on the methodology of their work.

## 9. REFERENCES

- [1] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with Massive Online Courses. In *Proc. of WWW '14*, pages 687–698. International World Wide Web Conferences Steering Committee, 2014.
- [2] A. Arnt and S. Zilberstein. Learning to Perform Moderation in Online Forums. In *Proc. of WIC '03*, pages 637–641. IEEE, 2003.
- [3] Y. Artzi, P. Pantel, and M. Gamon. Predicting Responses to Microblog Posts. In *Proc. of NAACL '12*, pages 602–606. Association for Computational Linguistics, 2012.
- [4] L. Backstrom, J. Kleinberg, L. Lee, and C. Danescu-Niculescu-Mizil. Characterizing and curating conversation threads: expansion, focus, volume, re-entry. In *Proc. of WSDM '13*, pages 13–22. ACM, 2013.
- [5] S. Chaturvedi, D. Goldwasser, and H. Daumé III. Predicting Instructor's Intervention in MOOC Forums. In *Proc. of ACL '14 (Volume 1: Long Papers)*, pages 1501–1511. ACL, 2014.
- [6] D. Clow. MOOCs and the funnel of participation. In *Proc. of LAK '13*, pages 185–189. ACM, 2013.
- [7] R. Ferguson and M. Sharples. Innovative Pedagogy at Massive Scale: Teaching and Learning in MOOCs. In *Open Learning and Teaching in Educational Communities*, pages 98–111. Springer, 2014.
- [8] J. Kim, J. Li, and T. Kim. Towards identifying unresolved discussions in student online forums. In *Proc. of NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 84–91. ACL, 2010.
- [9] R. Kizilcec and S. Halawa. Attrition and Achievement Gaps in Online Learning. In *Proc. of ACM L@S '15*, Vancouver, Canada, March 14–15 2015. In Press.
- [10] F.-R. Lin, L.-S. Hsieh, and F.-T. Chuang. Discovering genres of online discussion threads via text mining. *Computers & Education*, 52(2):481–495, 2009.
- [11] J. Mackness, S. Mak, and R. Williams. The ideals and reality of participating in a MOOC. 2010.
- [12] M. Mazzolini and S. Maddison. Sage, guide or ghost? The effect of instructor intervention on student participation in online discussion forums. *Computers & Education*, 40(3):237–253, 2003.
- [13] M. Mazzolini and S. Maddison. When to jump in: The role of the instructor in online discussion forums. *Computers & Education*, 49(2):193–213, 2007.
- [14] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Modeling Learner Engagement in MOOCs using Probabilistic Soft Logic. In *NIPS Workshop on Data Driven Education*, 2013.
- [15] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Learning Latent Engagement Patterns of Students in Online Courses. In *Proc. of AAAI '14*, 2014.
- [16] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Understanding MOOC Discussion Forums using Seeded LDA. In *Proc. of 9th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–33. ACL, 2014.
- [17] C. P. Rosé and G. Siemens. Shared task on prediction of dropout over time in massively open online courses. In *Proc. of EMNLP '14*, page 39, 2014.
- [18] L. A. Rossi and O. Gnawali. Language Independent Analysis and Classification of Discussion Threads in Coursera MOOC Forums. In *Proc. of IEEE IRI '14*, 2014.
- [19] G. S. Stump, J. DeBoer, J. Whittinghill, and L. Breslow. Development of a framework to classify MOOC discussion forum posts: Methodology and challenges. In *NIPS Workshop on Data Driven Education*, 2013.
- [20] J. H. Tomkin and D. Charlevoix. Do professors matter?: using an A/B test to evaluate the impact of instructor involvement on MOOC student outcomes. In *Proc. of ACM L@S*, pages 71–78. ACM, 2014.
- [21] L. Wang, S. N. Kim, and T. Baldwin. The Utility of Discourse Structure in Identifying Resolved Threads in Technical User Forums. In *Proc. of COLING '12*, pages 2739–2756, 2012.
- [22] M. Wen, D. Yang, and C. P. Rosé. Linguistic Reflections of Student Engagement in Massive Open Online Courses. In *Proc. of ICWSM '14 (poster)*, 2014.
- [23] D. Yang, D. Adamson, and C. P. Rosé. Question Recommendation with Constraints for Massive Open Online Courses. In *Proc. of ACM RecSys*, pages 49–56. ACM, 2014.

# Investigating how student's cognitive behavior in MOOC discussion forums affect learning gains

Xu Wang, Diyi Yang, Miaomiao Wen, Kenneth Koedinger, Carolyn P. Rosé

School of Computer Science, Carnegie Mellon University

5000, Forbes, Pittsburgh, PA, 15213

{xuwang, diyi, mwen}@cs.cmu.edu, koedinger@cmu.edu, cprose@cs.cmu.edu

## ABSTRACT

While MOOCs undoubtedly provide valuable learning resources for students, little research in the MOOC context has sought to evaluate students' learning gains in the environment. It has been long acknowledged that conversation is a significant way for students to construct knowledge and learn. However, rather than studying learning in MOOC discussion forums, the thrust of current research in that context has been to identify factors that predict dropout. Thus, cognitively relevant student behavior in the forums has not been evaluated for its impact on cognitive processes and learning. In this paper, we adopt a content analysis approach to analyze students' cognitively relevant behaviors in a MOOC discussion forum and further explore the relationship between the quantity and quality of that participation with their learning gains. As an integral part of our approach, we built a computational model to automate the analysis so that it is possible to extend the content analysis to all communication that occurred in the MOOC. We identified significant associations between discourse behavior and learning. Theoretical and practical implications are discussed.

## Keywords

Massive Open Online Courses (MOOC); Cognitive behavior; Content analysis; Discussion forum; Learning gains;

## 1. INTRODUCTION

Despite concerns over their effectiveness, MOOCs (Massive Open Online Courses) have attracted increasing attention both in the popular press and academia, raising questions about their potential to deliver educational resources at an unprecedented scale to new populations of learners. With learning through social processes featuring among the potential impacts of MOOC platforms [5], and discussion forums currently the primary means for supporting social learning in typical MOOC platforms, recent research has begun to focus on interventions that might enrich students' interaction in this context [e.g., 30], with the purpose of providing a more engaging and effective learning experience. Previous studies on learning and tutoring systems have provided evidence that students' participation in discussion [e.g., 2, 9, 12] is correlated with their learning gains in other instructional contexts.

However, whether discussion will also contribute substantially to

learning in a MOOC context, and what aspects of discussion will ultimately matter most to learning in this new context remain important open questions. Considering the significant connection that has been discovered between discussion behaviors in MOOC forums and student commitment, its potential for enabling students to form supportive relationships with other students, and the potential to enhance social learning through interaction, in depth empirical research is needed to uncover the relationship between student discourse patterns and learning gains in MOOCs.

One challenge to assessing learning in MOOCs, even in cases where formal assessments are integrated with the courses, is that students come into a MOOC with a wide variety of backgrounds [15,20], and it is typically unnatural to make a pretest a natural part of the learning process, especially when activities in the MOOC are all voluntary. However, while inconvenient, it is not impossible. The study reported in this paper took place in an unusual MOOC where a pretest was provided and students were aware that the MOOC data would be used for research purposes. This dataset, from a course entitled "Introduction to Psychology as a Science", thus provides a unique opportunity to begin to address the research questions introduced above.

Many student behaviors have been observed in discussion forums, e.g., question answering, self-introduction, complaining about difficulties and corresponding exchange of social support. A very coarse grained distinction in posts could be on vs. off topic. However, the important distinctions do not stop there and may be substantially more nuanced than that. Other than literal topic features, students' cognitively relevant behaviors, which are associated with important cognitive processes that precede learning may also be found in discussion forums. What those behaviors are in this context, and how frequently they occur are two questions we address.

Specifically, we ask the following research questions in this work:

1. Is a higher quantity of participation in MOOC discussion forums associated with higher learning gains?
2. Is on-task discourse associated with more learning gains than off-task discourse?
3. If certain properties of discussion are associated with enhanced learning, why it is so? What are the higher-order thinking behaviors demonstrated in student discourse and their connection with learning?

We consider that answering these questions has important implications for designing discussion interventions in MOOCs.

Some previous studies on MOOC discussion forums analyzed at a macro-level the quantity of participation [e.g., 1], whereas other work [23] pointed out that quantitative indices of participation does not directly imply the quality of conversation and interaction. Others conducted content analysis of thread topics [17] or used rule-based algorithms to extract linguistic markers [28]. However, students' higher-order thinking behaviors are not well represented

or thoroughly and systematically explored in these previous investigations. In this work, we aim to adopt a content analysis approach to hand-code data based on a well-established learning activity classification framework from earlier cognitive science research [8] in an attempt to capture students' discussion behaviors and their underlying cognitive strategies in a MOOC discussion forum. This is the first work we know of that has brought this lens to explore students' discussion behaviors and their association with learning gains in MOOCs.

In particular, we contribute to the existing literature by 1) developing a coding scheme based on Chi's ICAP (Interactive-Constructive-Active-Passive) framework [8] in categorizing students' discussion behaviors in a MOOC context; 2) providing empirical support for the importance of discussion in enhancing learning in a MOOC context. We also contribute to the literature on computer-supported collaborative learning by exploring the relationship between discourse and learning in a multi-user distributed asynchronous discussion environment.

In the remainder of the paper, we first discuss related work and existing theoretical foundations that we leverage in our analysis. Next we introduce our dataset. We then describe our methods, including specifics about the coding scheme, and computational model in the Methods section. We present an extensive correlational analysis and then discuss our interpretation along with caveats and directions for continued work.

## 2. RELATED WORK

### 2.1 Research on MOOC discussion forums

Studies in the field of learning science and computer supported collaborative learning have provided evidence that learners' contribution to discourse is an important predictor of their knowledge construction [2, 12]. In offline environments, studies have suggested, for example, that the number of words per utterance [26] and proportion of words produced [14] are correlated with learning gains. Transitioning from traditional classroom to online learning, computer-mediated conferencing has proved to be a gold mine of information concerning students' psycho-social dynamics and their knowledge acquisition [19]. Investigating the usage of discussion forums in MOOCs has been one major theme for research. To give a few examples, at a participation level, Anderson and colleagues [1] found that students who participated in other platform activities (videos, quizzes, etc.) participated more in the forum as well. They also explored patterns of thread initiators and contributors in terms of specific discussion behaviors in the discussion forum. At a content level, Brinton [5] categorized discussion threads into "small-talk", "course logistics", and "course specific" categories. Gillani [17] adopted a content analysis approach combined with machine learning models to discover sub-communities in a MOOC based on user profiles. Anderson [1] used a lexical analysis to see which words predict the number of assignments a student finally turns in.

These studies have set up a good foundation for analyses in MOOC discussion forums. However, to confirm a relationship between discussion and learning, we need to look closer into what aspects of discussion actually contribute to learning from a cognitive perspective.

### 2.2 Content analysis

We base our work on previous approaches to analyze content of student dialogues in tutoring and computer-supported collaborative learning environments. Chi [6] pointed out the importance of verbal analysis, which is a way to indirectly view student cognitive activity. De Wever [16] further demonstrated

that content analysis has the potential to reveal deep insights about psychological processes that are not situated at the surface of internalized collaboration scripts.

Chi's ICAP framework [8] has been considered to be the strongest evidence for the value of a dialogic approach to learning [25], which has been widely adapted and applied to identify learning activities and explain study results [e.g., 24, 27]. The framework has been utilized to explain classical educational experiments [10] and serve as a theoretical foundation for studies on tutoring and computer-supported collaborative learning, for example in a discourse analysis of different kinds of scaffolds [24].

The framework was created through a meta-analysis of 18 studies in which learning activities were classified into 3 categories, namely, interactive activities that involve discussing and co-constructing with a peer or the learning environment, constructive activities that produce a representation of information that goes beyond the presented information, and finally, active activities that show how students are actively engaged in the learning process. The taxonomy suggests the hypothesis that what are referred to in it as interactive activities should generate more learning outcomes than constructive activities, which in turn should generate more learning outcomes than active activities. [8]

MOOCs provide an emerging environment where computer-supported collaborative learning activities might be provided, and where social presence might reflect cognitive presence [27]. Thus, in this context we aim to apply the ICAP framework to explore the relationship between discussion and learning by coding observed student behaviors in the discussion forum.

## 3. DATASET

In this work, we conducted a secondary analysis of the dataset of the course "Introduction to Psychology as a Science" offered through Coursera collaboratively by Georgia Institute of Technology and Carnegie Mellon University. The course incorporated elements of the OLI (Open Learning Initiative) "Introduction to Psychology" learning environment. One special characteristic of the course was that it administered a pre/post test with the intention to support research.

"Introduction to Psychology as a Science" was designed as a 12-week introductory course. For each week of class, the course targeted a major topic (e.g. Memory, Brain Structures, Nervous System); Course materials include video lectures, assigned MOOC activities, learning activities in the OLI environment, and what are referred to as weekly high-stakes quizzes.

In the first analysis of the dataset [21], researchers found that students who registered for the OLI activities learned more than students who used only the typical MOOC affordances, and further demonstrated that students who did more learning-by-doing activities learn more than students who watch more videos or read more texts. In other words, doing an activity has a much greater effect (6x) on predicted learning outcomes than watching a video or reading a web page. However, students' participation in the discussion forum hasn't been explored yet in that work.

In our preliminary exploration into the dataset, we found that when controlling for students' registration for OLI activities (which serves as a control variable associated with effort and commitment to the course), their quantity of participation in discussion forums significantly predicts learning gains as well. Based on this, we wanted to further explore how students' specific cognitively relevant behaviors in the forums correlate with their learning gains. We observed specific related discourse behaviors in the forum, and present several examples here.

**Active behavior:** “According to the OLI textbook, creative intelligence is ‘the ability to adapt to new situations and create new ideas or practicality’.”

This is an example of the student actively repeating what’s being said in the course materials.

**Constructive behavior:** “When I tell my son to wash the dishes, it’s much more straightforward to explain his refusal or agreement by some behavioral (e.g. Reward or punishment) or cognitive mechanisms than by an innate instinct to wash or not to wash the dishes.”

This is an example of constructive behavior, when the learner produces output, which could be examples, explanations, etc., that go beyond course materials.

**Interactive behavior:** “I agree that language can be an extra difficulty, but it is not a variable with which is counted. Also, depression, work stress...could form extra difficulties for the student in particular.”

This interactive behavior example shows that students not only engage in self-construction, but build their ideas upon their partners’ contributions.

Altogether, there are 27,750 registered users in the dataset, and 7,990 posts and comments in the dataset. For the learners who have both pretest and posttest on record, which is our population of interest, there are 3,864 posts in total and 491 users. In addition to forum records, student clicks with course materials are also recorded in the clickstream data. The course has 1,487,665 student clicks. The clickstream logfile provides us with the opportunity to observe each students’ interaction with course materials.

## 4. METHOD

### 4.1 Unit of analysis

In this paper, our unit of analysis is the message. As proposed in [16], in their review of 15 instruments in doing content analysis of the transcripts of online asynchronous discussion groups, 7 recommended using the message as the unit of analysis.

We first looked at students’ quantity of participation, and distinguished on-task discourses from off-task. We then applied a coding scheme on on-task discourse to capture the cognitive behaviors in the discussion forum. We hand-coded half of the dataset, and trained a machine learning model to replicate that annotation approach in the rest of the dataset.

In a MOOC context, the data we usually deal with is student log data [4, 5, 13], which illustrates their participation process. However, students’ cognitive behaviors are better represented in their discourse displayed in the discussion forum. In this work, we hand-coded a large sample of the dataset, which may reduce noise in this kind of analysis. Thus the result may be more reliable in demonstrating the relationship between students’ cognitive behaviors in the discussion forum and their learning gains.

### 4.2 Quantity of participation

H1: In response to our first research question, we hypothesized that students who participated more in the discussion forum have higher learning gains.

We quantified students’ participation in the discussion forum by the variable PostCountByUser.

PostCountByUser: It is measured by the number of posts a user posted in the discussion forum.

We did not distinguish between posts and comments in this analysis. So the word posts when mentioned in the rest of the paper refers both to posts and comments.

### 4.3 On-task vs. Off-task discourse

H2: in response to our second research question, we distinguished on-task and off-task discourse in the dataset. And we hypothesized that students’ total number of on-task discourse contributions has a positive association with their learning gains.

We distinguished on-task discourse from off-task discourse in the dataset, based on the following definitions. On-task discourse includes posts that talk about course content, the content of quizzes and assignments, comments on course materials, and interaction between students on course content-related issues. Off-task discourse includes posts that talk about administrative issues in the course, e.g., asking for extensions on assignments; technical issues regarding course materials, e.g., asking where to download videos, off-topic self-introductions and social networking.

This feature in the dataset is acquired through hand-coding.

### 4.4 Cognitively Relevant Discussion behavior

H3: In response to our third research question, we want to investigate what discussion behaviors are demonstrated in the discussion forum, their frequencies and their association with learning. In order to capture these discussion behaviors, we developed a coding scheme based on Chi’s ICAP framework [8].

We further hypothesized that students who demonstrated more higher-order thinking behaviors in each of the categories, active discourse, constructive discourse, and interactive discourse have higher learning gains. And according to Chi’s work represented in 18 empirical studies, we hypothesized that the effect follows the pattern interactive>constructive>active.

#### 4.4.1 Coding scheme

Students’ cognitive behaviors are reflected in the MOOC discussion forums, which is not easily mined through rule-based algorithms due to its scale and informal style. This may pose challenges for computational modeling. In this work, we adopt a hand-coding method to capture higher-order thinking behaviors and follow the hand coding with computational modeling.

Within the category of on-task discourse we divide all posts into 3x3 categories as listed in Table 1 according to Chi’s Active-Constructive-Interactive framework [8]. We further offer operational definitions for each category, and provide examples from our dataset. Due to space limitations, we provide abbreviated definitions rather than the full ones provided to the human coders. When defining each category of cognitive behavior, we evaluated how this might contribute to learning. Through empirical observation, we found this coding scheme to be exhaustive of all conditions. The 9 categories are not mutually exclusive. Thus, a post may belong to more than one of these fine-grained categories.

#### 4.4.2 Inter-rater reliability

Two experts separately coded 100 posts randomly selected from the dataset, and applied on- vs. off-task annotation plus the 9 fine-grained categories of discussion behaviors to the sample. The two experts at first reached an agreement statistic of 0.52 (Cohen’s Kappa), which is a moderate level of agreement. The two experts then resolved their disagreements through consensus coding by discussing and clarifying some borderline cases. After higher consensus was achieved, one of the experts coded 2000 posts randomly sampled from the whole dataset (3864 posts).

**Table 1. Coding Examples**

<b>Active Discourse- (1) Repeat</b>	Operational Definition: The learner explicitly repeats information that's already covered in the material, which could be indicated by quotes. <i>E.g. 1: Week 2, I quote from the picture: "The portion of the sensory and motor cortex devoted ... as does the entire trunk of the body."</i>
<b>Active Discourse- (2) Paraphrase</b>	Operational Definition: The learner paraphrases what's covered in course materials, it could be indicated by words like "it's mentioned in the textbook...", "it's said in the video..." <i>E.g. 2: On the chapter about Health Psychology there is a board depicting various factors about Happiness, such as the Inequality of Happiness and then the Inequality Adjusted Happiness.</i>
<b>Active Discourse- (3) Notes-taking</b>	Operational Definition: The learner mentions about note-taking and information seeking. <i>E.g. 3: I use the text files as a basis for my lecture notes.</i>
<b>Constructive Discourse- (1) Ask novel questions</b>	Operational Definition: The learner proposes a novel question or problem based on his/her own understanding. <i>E.g. 4: Violence is throughout our history and have shaped societies, is it really as simple as an observed response? or a throwback of survival instinct?</i>
<b>Constructive Discourse- (2) Justify or provide reasons</b>	Operational Definition: The learner uses examples and evidence to support a claim he/she has made. Reasoning is explicitly demonstrated in the discourse. <i>E.g. 5: It depends on the visual field. Signals from the right visual field come to the left hemisphere, while signals from the left visual field come to the right hemisphere.</i>
<b>Constructive Discourse- (3) Compare or connect</b>	Operational Definition: The learner compares cases, connects or shares links to external resources. <i>E.g. 6: Here's a link to an article about a lady who stopped dreaming after suffering a stroke: [link]</i>
<b>Interactive discourse- (1) Acknowledgement of partners' contribution</b>	Operational Definition: The learner explicitly acknowledges their partners' contribution, which could be indicated by "thanks for pointing that out", "I agree with you there..." <i>E.g. 7: That's an interesting point, and it has made me wonder why this example was chosen.</i>
<b>Interactive discourse- (2) Build on partners' contribution</b>	Operational Definition: The learner makes a point that builds on what their partner has said. <i>E.g. 8: I do agree with what you said to a large degree. Changing a behavior merely to avoid pain or any other form of punishment is not good... Hence it requires a much deeper introspection and understanding...</i>
<b>Interactive discourse- (3) Defend and challenge</b>	Operational Definition: The learner challenges his/her partners' ideas, or defends their own ideas, when there is a disagreement. (Note: The partner here can be either a peer or the learning environment) <i>E.g. 9: I think I understand what you mean (I am currently doing the statistics course as well). However, as I can see from what you've described, you still have the hypothesis in your psychological experiment which is not null - your prediction that something WILL happen.</i>

#### 4.4.3 Computational model and data preparation

In order to better visualize the dataset and potentially apply the model to another context, we trained a computational model based on the coded 2000 posts to predict the cognitively relevant discussion behavior categories and expand the coding to the rest of the dataset.

In our hand-coded dataset, we labeled 9 types of cognitively relevant discussion behaviors, but due to the fact that the occurrences of each single category are relatively sparse, we acquired a low accuracy when using the sample to train a model and apply it to the rest of the dataset. Instead, we aggregated the 9 categories into the three major categories—Active, Constructive, and Interactive. All three are binary variables indicating whether the user has a post under this category. We then built models to predict these labels.

Our classifier is designed to predict whether the cognitive behavior expressed in a post belongs to Active (A), Constructive (C) or Interactive (I) by taking advantage of a bag-of-words representation. However, we have to distinguish between on-task discourse and off-task discourse since learning relevant cognitive behaviors will occur primarily in on-task discussion (Among our coded 2000 posts, 558 are on-task discussions).

For this purpose, we built a two-stage classification model. In the first stage, we designed a logistic regression classifier to predict whether a post is on-task or off-task; in the second stage, we classified the posts that were predicted to be on-task into A, C or I categories. The input for each classifier is a bag-of-words feature representation. In the first step, we used the coded 2000 posts as the training set to train a logistic regression classifier to distinguish on-task discourses and off-task discourse, and in the second step, we used 588 on-task messages as our training set to train three logistic regression classifiers to label on-task

discourses in the three categories (A, C, I). On the training set, we adopted a 10-fold cross-validation approach to evaluate the model. The classification results presented in Table 2 are the average accuracy and Kappa for this cross-validation. The results show that both accuracy and kappa are within a reasonable range for our further analysis of the whole dataset.

Table 3 shows some top-ranked features identified by the classifiers that are used to predict the three cognitive behaviors. From this table, we can see that in active discourse, students more often mentioned “lectures” “page” “notes”, which indicates they’re actively engaged with the course materials. In constructive discourse, students more often mentioned words associated with reasoning, such as “more” “but” “hence” “examples”, and in interactive discourses, students mentioned “your” “agree” “disagree” more often, which implies interaction. These features are consistent with our initial definitions of these distinct categories of discussion behavior and assumptions about their underlying cognitive processes and strategies.

**Table 2. Evaluation metrics of the computational model.**

Evaluation Metrics		Accuracy	Kappa
1 <sup>st</sup> Stage	On-task	82.1%	0.527
<b>On-task Prediction</b>			
2 <sup>nd</sup> Stage	Active	74.3%	0.361
<b>Cognitive Behavior</b>			
	Constructive	75.4%	0.318
	Interactive	75.6%	0.236

**Table 3. Performance of Discussion Behaviors Regressors and Top Ranked Features**

Categories	Active	Constructive	Interactive
<b>Most Important Word Features (Regression Weight)</b>	<i>lecture (1.68)</i>	<i>course (.87)</i>	<i>your (1.56)</i>
	<i>page (1.24)</i>	<i>more (.79)</i>	<i>agree (1.11)</i>
	<i>what (.84)</i>	<i>give (.75)</i>	<i>our (.99)</i>
	<i>text (.83)</i>	<i>trying (.68)</i>	<i>again (.86)</i>
	<i>incorrect (.79)</i>	<i>but (.64)</i>	<i>thanks (.76)</i>
	<i>answer (.72)</i>	<i>hence (.64)</i>	<i>disagree (.6)</i>
	<i>says (.72)</i>	<i>looking (.61)</i>	<i>response (.6)</i>
	<i>notes (.68)</i>	<i>topics (.58)</i>	
		<i>example (.56)</i>	
		<i>because (.56)</i>	

#### 4.4.4 Clickstream Data

In order to explore the relationship between cognitively relevant discussion behaviors and learning, we also need to control for students’ involvement in other activities in the MOOC environment other than the discussion forum. This enables us to isolate, to some extent, the effect of pure effort and engagement in the course from the effects specifically related to discussion behavior. We further generated the following control variables through mining clickstream data of the course.

**Video:** The variable was computed first by summing the number of unique videos the student started to watch (Based on clicks on unique video urls), and then standardizing the sums.

**Quiz:** The variable was computed first by summing the number of unique quizzes the student attempted (Based on clicks on unique quiz urls), and then standardizing the sums.

**OLI\_textbook:** The variable indicates reading the OLI textbook, and it’s calculated by summing the number of clicks the student made in the OLI environment and then standardizing the sums.

## 5. RESULTS

### 5.1 Participation quantity in the discussion forum

In response to the first research question, we fitted linear regression models to explore the relationship between students’ quantity of participation and their learning gains.

In the dataset, there are 1,079 students out of 27,750 students (i.e., students who registered for the course) who have both pre- and post-test scores on record. And among these students, there are 491 students who have posted in the discussion forum, with a total of 3,864 posts. We now introduce the variables we used in these models.

*Dependent variable:*

**Post-test:** The dependent variable in all the following models is students’ posttest score, which is standardized. Post-test score is students’ final exam score composed of 35 multiple-choice questions.

*Control variable:*

**Pre-test:** This is a test students took before the course started, which contains 20 multiple-choice questions. We also standardized the pretest score.

**OLI\_Registration:** This is a binary variable capturing whether the student has registered for OLI or not. 1 means the student registered for OLI. As demonstrated in [21], students who registered for OLI learnt more than students who didn’t.

We also controlled for students’ involvement in other activities, including Video, Quiz and OLI\_textbook.

*Independent variable:*

**Participation:** This is a binary variable indicating whether the student has ever posted in the discussion forum during the course.

**PostCountByUser:** This is the total number of posts a student contributed in the discussion forum during the course.

From Model 1, we see that whether the student has participated in the discussion forum is a significant predictor of the student’s learning gains. The result from Model 2 shows that for those who have participated in the discussion forum, the more they posted, the higher the learning gains they achieved.

**Table 4. Regression results of learning gains on the quantity of participation and on-task discourse**

Control/Indep. Variable	Model 1 (N=1079)	Model 2 (N=491)	Model 3 (N=491)
<b>Participation</b>	0.089**		
<b>PostCountByUser</b>		0.005*	0.006*
<b>OnTaskPercent.</b>			0.123**
<b>Pretest</b>	0.196***	0.254***	0.243***
<b>OLI_registration</b>	0.119**	0.107	0.120
<b>Video</b>	0.056*	0.0001	-0.011
<b>Quiz</b>	-0.008	-0.035	-0.037
<b>OLI_textbook</b>	0.050**	0.048	0.044

(p<0.001\*\*\*, p<0.01\*\*, p<0.05\*)

## 5.2 On-task versus off-tasks discourse

In response to the second research question, we looked at whether students' on-task discourse contributes to their learning gains. In this model, we examine the main effect of on-task discourse, which is represented by the variable OnTaskPercent.

Independent variable:

**OnTaskPercent.** : This is measured by the number of a student's posts that are categorized as on-task divided by the total number of posts the student has made, and the value is standardized.

In this regression model, we also controlled for a student's number of posts, whether they registered for OLI, pretest score, and their involvement in other activities. The regression result is displayed in Table 4-Model 3. The result shows that the quantity of students' on-task discourse in the discussion forum is a significant predictor of their learning gains.

## 5.3 Cognitively relevant discussion behavior analysis

### 5.3.1 Active, Constructive and Interactive behaviors

In this section we examine the relationship between students' discussion behavior and their learning gains and attempt to explain why certain behaviors lead to learning. We built linear regression models to explore the relationship between students' active, constructive and interactive discussion behaviors and their learning gains.

In the whole dataset, the number of instances (N=3864) of active, constructive and interactive activities is respectively 269, 744 and 203. And the number of students (N=491) who have demonstrated active, constructive and interactive activities is respectively 114, 230 and 84.

Our independent variables include:

**Active, Constructive, Interactive:** All three are binary variables indicating whether the student has a post that is categorized in that category.

We also controlled for variables including pretest, the number of posts, whether registered for OLI, students' involvement in other activities, as defined above. The regression result is shown in Table 5.

In Model 4 and Model 5, we found that students who have demonstrated active and constructive behaviors in the discussion forum had significantly more learning gains than students who didn't. From Model 6, we can see that the effect of Interactive discussion behavior is not significant in predicting learning gains. And we then introduced another variable to define whether a user is an active poster by counting the total number of their posts.

**Poster profile:** This nominal variable categorizing users into active poster and inactive poster. If a user has more than 3 posts (including 3), he/she is categorized as an active poster, otherwise categorized as an inactive poster. 3 is the median of the number of posts.

When nesting interactive behaviors with a poster profile, we found that interactive discussion is a significant predictor of learning gains for students who posted less. We think this might be because the number of posts is a basic measure of a student's social engagement in the discussion forum, which overlaps with some behaviors under the Interactive category. We further fitted a regression model to check the correlation between a student's total number of posts and the number of posts that are categorized as Interactive. The result shows that Interactive posts account for

66% of the variance in the total number of posts. We consider this high correlation could lead to the result described above. The results here show that both active and constructive discussion behaviors significantly contribute to students' learning gains, with active behaviors having higher predictive power. For users who posted less in the discussion forum, interactive behaviors strongly predict their learning gains (coefficient=0.515), however, the effect of interactive behavior disappears on the overall user population.

In addition to the occurrence of different discussion behaviors, we also used the frequency of each behavior as independent variables and did a second round of regression, from which we acquired similar results.

**Table 5. Regression results of learning gains on discussion behaviors (part 1, N=491)**

Control/Independ. Variable	Model 4	Model 5	Model 6	Model 7
Active	0.125*			
Constructive		0.112*		
Interactive			0.106	
Interactive [inact. poster]				0.496*
Interactive [act. poster]				0.043
Pretest	0.252***	0.246***	0.254***	0.254***
PostCntByUser	0.004	0.004	0.004	0.004
OLI_registr.	0.125	0.109	0.104	0.115
Video	-0.004	0.015	0.003	0.007
Quiz	-0.039	0.036	-0.038	-0.036
OLI_textbook	0.034	0.044	0.040	0.036

(p<0.001\*\*\*, p<0.01\*\*, p<0.05\*)

### 5.3.2 Specific discussion behaviors

From the hand-coded dataset (N=2000), we summarized the occurrences of the 9 sub-categories of behaviors in Table 6. It shows that the most frequent behavior in the discussion forum is proposing an idea or asking novel questions. And the least frequent behaviors include building on a partner's contribution as well as defending and arguing, which is consistent with our expectation that higher-order thinking behaviors and highly interactive behaviors are relatively rare in MOOC discussion forums, and that the conversations going on in MOOCs are not satisfactorily rich and interactive.

**Table 6. Distribution of 9 categories of discussion behaviors**

Behavior Type	Freq.	Behavior Type	Freq.
Repeat	53	Notes-taking	28
Paraphrase/ask shallow questions	103	Justify or provide reasons	118
Propose an idea/ask novel questions	315	Compare, connect/ Reflect	59
Acknowledge partners' contribution	101	Build on partners' contribution	23
Defend and argue	14		

We also fitted regression models on this more nuanced coded dataset, but due to the fact that the occurrences of each category is relatively sparse, there was not sufficient statistical power to detect a significant effect of every category on learning gains. We display just the 2 significant predictors (out of 9) in Table 7.

Independent variables:

**Constructive-(1):** This is a binary variable indicating whether the student has a post that is categorized as “propose an idea/ ask novel questions”.

**Constructive-(2):** This is a binary variable indicating whether the student has a post that is categorized as “Justify or provide reasons”.

We controlled for pretest, number of posts, and whether the student registered for OLI in the regression models. We also controlled for students’ involvement in other activities, the effects of which aren’t significant in the regression models, so we don’t report them here in Table 7.

**Table 7. Regression results of learning gains on discussion behaviors (part 2, N=399)**

	Model 8	Model 9
Constructive-(1)	0.136*	
Constructive-(2)		0.211**
Pretest	0.205***	0.198***
OLI registration	0.225	0.214
Number of posts	0.007	0.005

( $p < 0.001$ \*\*\*,  $p < 0.01$ \*\* ,  $p < 0.05$ \*)

After fitting regression models of learning outcome and all discussion behaviors as main effects, we found that only two categories are significant in predicting learning gains, as shown in Table 7. We consider higher frequencies of both behaviors might be the reason leading to significant effects on learning. Nevertheless, the processes of proposing an idea or a problem, and providing examples and reasons to justify a claim are considered to be higher-order thinking behaviors that have been proved to be instrumental to learning in several studies [e.g., 7, 18, 22], which could also lead to the significant effects.

## 6. CONCLUSION AND DISCUSSION

In this paper we adopted a content analysis approach and developed a coding scheme to analyze students’ discussion behaviors, which are hypothesized as relating to their underlying cognitive processes in the discussion forum of a MOOC. The learning gains measures available for students in this MOOC enabled us to explore the relationship between students’ discussion behaviors and their learning, and discuss what aspects of discussion appear to contribute to learning.

We observed that students’ active and constructive discussion behaviors are significant in predicting students’ learning gains, with active discussion behaviors possessing better predictive power, which is inconsistent with our hypotheses. Interactive discussion behaviors are significant in predicting learning gains only for students who are less active in the forums. This work also provides insight into how students are discussing in the discussion forum now, what behaviors they demonstrate and what the underlying cognitive processes are.

### 6.1 Active-Constructive-Interactive framework

Based on Chi’s framework [8], we hypothesized that students’ interactive discussion behaviors will produce more learning gains than constructive behaviors, and constructive behaviors will produce more learning gains than active behaviors. However, in this analysis we found that students’ active discussion behaviors are most effective in predicting students’ learning gains (coefficient=0.125). In our categorization of active behavior, students are talking about what is already covered in the materials,

repeating statements that had appeared in the textbook or video lectures, and asking clarifying questions about definitions, implicitly expressing confusion about course materials, etc. According to Chi’s framework [8], constructive activities should provide better learning outcomes than active activities. An example of this is when students need to explain in a constructive condition. However, we consider one reason we may not have seen this pattern in our dataset is that the post-test may not have targeted the skills and concepts students learned from these constructive activities. Assessments of a different nature, for example incorporating more demanding open ended response items, may have been more sensitive to these gains. For example, when the learning task is about design of psychology experiments, an assessment of requiring the students to design an actual experiment might be more telling than multiple-choice questions in measuring students higher-order thinking skills.

## 6.2 Invisible learning practices

In this paper, we looked at students’ overt discussion activities in the forum, however students may be engaged in these higher order thinking activities without articulating their reasoning in a visible discourse. As indicated by [3], reading but not necessarily posting can be a productive practice for some learners. Our estimates of the amount of videos, quizzes and OLI textbook pages attempted could also be improved, for example, using the time spent on each activity, and further details about the attempt of OLI activities could be incorporated, as defined and estimated in [21].

## 6.3 Design implications

As MOOCs evolve, our focus as a community will transition from a primary concern about retaining users to actively improving the pedagogical effectiveness of this learning environment. Thus we need an empirical foundation to base designs for discussion affordances in MOOCs that might facilitate constructive and interactive conversations. Also, we need to come up with better assessment methods to assess and acknowledge students’ higher-order thinking behaviors and skills they acquired through reading others’ ideas, explaining and arguing in a discussion forum.

The paper proposes a manual way to hand-code students discussion behaviors, and offers a machine learning model to predict the corresponding behaviors in all communications of the dataset. We haven’t had the opportunity to test the model in other courses, as few courses have pre- and post-test measures. If the computational model can be applied, we may provide feedback on students’ advanced discussion behaviors in the forum, in terms of their cognitive processes and strategies.

## 7. ACKNOWLEDGEMENTS

This research was funded in part from NSF DATANET grant 1443068 and a grant from Google.

## 8. REFERENCES

- [1] Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2014). Engaging with massive online courses. *In Proceedings of the 23rd international conference on World wide web* (pp. 687-698). International World Wide Web Conferences Steering Committee.
- [2] Barab, S., & Duffy, T. (2000). From practice fields to communities of practice. In D. H. Jonassen & S.M. Land (Eds.), *Theoretical Foundations of Learning Environemnts*.
- [3] Beaudoin, M. F. (2002). Learning or lurking?: Tracking the “invisible” online student. *The internet and higher education*, 5(2), 147-155.

- [4] Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*, 8(1), 13-25.
- [5] Brinton, C., Chiang, M., Jain, S., Lam, H., Liu, Z., & Wong, F. (2013). Learning about social learning in MOOCs: From statistical analysis to generative model, 7(4), 346-359.
- [6] Chi, M. T. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *The journal of the learning sciences*, 6(3), 271-315.
- [7] Chi, M. T. H. (2000). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In R. Glaser (Ed.), *Advances in instructional psychology* (pp. 161-238). Mahwah, NJ: Erlbaum.
- [8] Chi, M. T. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, 1(1), 73-105.
- [9] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, 25, 471-533.
- [10] Chi, M. T. H., & VanLehn, K. a. (2012). Seeing Deep Structure From the Interactions of Surface Features. *Educational Psychologist*, 47(3), 177-188.
- [11] Clow, D. (2013). MOOCs and the funnel of participation. *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (pp.185-189). ACM.
- [12] Cohen, E. G. (1994). Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research*, 64, 1-35.
- [13] Coffrin, C., Corrin, L., de Barba, P., & Kennedy, G. (2014). Visualizing patterns of student engagement and performance in MOOCs. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge - LAK '14* (pp. 83-92). New York, New York, USA: ACM Press.
- [14] Core, M. G., Moore, J. D., & Zinn, C. (2003). The role of initiative in tutorial dialogue. In *Proceedings of the 11<sup>th</sup> Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (pp. 67-74). Morristown, NJ: Association of Computation Linguistics.
- [15] DeBoer, Jennifer, G. S. Stump, D. Seaton, and Lori Breslow. (2013). Diversity in MOOC students' backgrounds and behaviors in relationship to performance in 6.002x. In *Proceedings of the Sixth Learning International Networks Consortium Conference*.
- [16] De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education*, 46(1), 6-28.
- [17] Gillani, N., Eynon, R., Osborne, M., Hjorth, I., & Roberts, S. (2014). Communication communities in MOOCs. *arXiv preprint arXiv:1403.4640*.
- [18] Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American educational research journal*, 31(1), 104-137.
- [19] Henri, F. (1992). Computer conferencing and content analysis. In *Collaborative learning through computer conferencing* (pp. 117-136). Springer Berlin Heidelberg.
- [20] Kizilcec, R. F., Piech, C., & Schneider, E. (2013). Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge* (pp. 170-179). ACM.
- [21] Koedinger, K.R., Kim, J., Jia Z., McLaughlin E., Bier, N. (2015). Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC. *Learning at Scale '15*.
- [22] Mestre, J. P. (2001). Implications of research on learning for the education of prospective science and physics teachers †. *Physics Education*, 36(1), 44-51.
- [23] Meyer, K. (2004). Evaluating online discussions: four different frames of analysis. *Journal of Asynchronous Learning Networks*, 8(2), 101-114.
- [24] Molenaar, I., Chiu, M. M., Slegers, P., & van Boxtel, C. (2011). Scaffolding of small groups' metacognitive activities with an avatar. *International Journal of Computer-Supported Collaborative Learning*, 6(4), 601-624.
- [25] Osborne, J., Simon, S., Christodoulou, A., Howell-Richardson, C., & Richardson, K. (2013). Learning to argue: A study of four schools and their attempt to develop the use of argumentation as a common instructional practice and its impact on students. *Journal of Research in Science Teaching*, 50(3), 315-347.
- [26] Rosé, C. P., Bhembé, D., Siler, S., Srivastava, R., & VanLehn, K. (2003). Exploring the effectiveness of knowledge construction dialogues. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Artificial intelligence in education: Shaping the future of learning through intelligent technologies* (pp. 497-499). Amsterdam, the Netherlands: IOS Press.
- [27] Shea, P., & Bidjerano, T. (2012). Learning presence as a moderator in the community of inquiry model. *Computers & Education*, 59(2), 316-326.
- [28] Wen, M., Yang, D., & Rose, C. P. (2014, May). Linguistic reflections of student engagement in massive open online courses. In *Proceedings of the International Conference on Weblogs and Social Media*.
- [29] Wen, M., Yang, D., & Rosé, C. P. (2014). Sentiment Analysis in MOOC Discussion Forums: What does it tell us. *Proceedings of Educational Data Mining*.
- [30] Yang, D., Adamson, D., & Rosé, C. P. (2014). Question recommendation with constraints for massive open online courses. *Proceedings of the 8th ACM Conference on Recommender Systems - RecSys '14*, 49-56.
- [31] Zimmerman, B. J. (1989). A social cognitive view of self-regulated academic learning. *Journal of Educational Psychology*, 81(3), 329-339.

# Methodological Challenges in the Analysis of MOOC Data for Exploring the Relationship between Discussion Forum Views and Learning Outcomes

**Yoav Bergner**

Educational Testing Service  
Princeton, NJ 08541  
ybergner@ets.org

**Deirdre Kerr**

Educational Testing Service  
Princeton, NJ 08541  
dkerr@ets.org

**David E. Pritchard**

M.I.T.  
Cambridge, MA 02139  
dpritch@mit.edu

## ABSTRACT

Determining how learners use MOOCs effectively is critical to providing feedback to instructors, schools, and policy-makers on this highly scalable technology. However, drawing inferences about student learning outcomes in MOOCs has proven to be quite difficult due to large amounts of missing data (of various kinds) and to the diverse population of MOOC participants. Thus significant methodological challenges must be addressed before seemingly straightforward substantive questions can be answered. The present study considers modeling final exam performance outcomes on early-stage ability estimates, discussion forum viewing frequency, and overall assessment-oriented engagement (AOE, seen as a proxy measure of motivation). These variables require careful operationalization, analysis of which is the principle contribution of this work. This study demonstrates that the effect sizes of discussion forum viewing activities on final exam outcomes are quite sensitive to these choices.

## Author Keywords

MOOCs; discussion forums; social learning.

## INTRODUCTION

Massive open online courses (MOOCs), a recent modality of distance learning wherein course materials are made available online and are freely accessible by anyone with computer access, have been rapidly gaining popularity as new platforms and courses come online. As of August 2014, over 2000 MOOCs were being offered through more than 50 initiatives ([www.mooc-list.com](http://www.mooc-list.com)), and these numbers had more than doubled over the prior year. MOOCs are generally viewed as having great value because they provide expanded opportunities to learn and near-instantaneous feedback and support. Additionally, the large number of enrollees and clickstream interaction logs in any given MOOC provide a vast amount of fine-grained data that can help researchers understand how people learn and how best to support learning in an online environment.

This program of research began with the hope of capitalizing on these properties in order to examine the impact of MOOC discussion forum use on learning outcomes. Simply put, we wanted to study whether viewing discussion board threads while doing homework resulted in

final exam gains attributable to this behavior, i.e. controlling for other factors. It seemed prudent to try to account for enrollees with different levels of prior ability and engagement/motivation, as MOOC students are known to have diverse populations. Thus, final exam performance would be our outcome variable; prior ability, engagement/motivation (or some proxy), and discussion forum usage would be covariate predictors. Along the way, however, we perceived that the challenges of operationalizing all of the variables gained more and more importance to the validity of our inferences.

Indeed, recent work by other authors concentrated on the sensitivity of analytical inferences to operationalization of predictor variables such as time-on-task estimation [18]. In reference to that work, this paper may also be seen as an attempt to “penetrate the black box” of a particular MOOC analysis. Thus, we raise the following auxiliary research questions: Does the method of quantifying discussion forum use significantly impact the analysis of its effect on performance? Given that motivation matters, does the decision of which filter to use to exclude unmotivated students change the results of the analysis? Issues of prior ability estimation are myriad; we discuss these briefly below but get into more details in a separate study [4].

In the remainder of this paper, we examine the impact of methodological decisions on the quality and type of inferences that can be drawn from examining MOOC forum use, focusing specifically on methods of quantifying discussion forum use and filtering unmotivated students.

The organization is as follows. By way of motivating our original substantive questions, we first review related literature on the impact of discussion forums in online learning. We then describe our data set. Next, we turn to the challenges of MOOC analyses, in general and specifically to the variables under consideration. We describe different methods for and results from operationalization choices with regards to discussion forum usage, motivation proxies, and prior ability estimates. Finally, we consider the impact of these variables on performance using multiple linear regression models for final exam score.

## DISCUSSION FORUMS IN ONLINE LEARNING

The impact of discussion forums on learning in MOOCs and other online courses is still not well understood,

although the literature on the subject dates back to the 1990s. While some early research on discussion forums cautioned about the shortcomings of computer-mediated dialogue as compared with face-to-face interactions [25], much of that research explored the benefits of the cognitive processes involved in the use of discussion forums, such as reshaping ideas and constructing meaning with the help of peers [3,21]. Later research (but still prior to the MOOC era) focused on measuring the level and quality of student activity in the forums, for example using data mining and text mining [8]. Cultivation of successful asynchronous discussion was linked to measures of discussion quality [2]. Artificial intelligence approaches for classifying effective synchronous collaborative learning [23] were also applied to asynchronous forums in a graduate level course [24].

Correlations of discussion activity with external performance measures have been the subject of several studies ranging from high school [15] to college [17,19] to graduate school [24], with mixed results. Correlations of 0.51 were found for topical student discussion behaviors (coded by hand) with concept-test performance in a physics course using the learning online network with computer-assisted personalized approach (LON-CAPA) learning management system [17]. Operationalizing discussion behavior purely by counts, [15] found correlations of 0.27-0.44 between project performance and activity volume in the forums for secondary school computer science. [19] performed a multiple regression analysis of quiz scores in two college psychology courses, finding that only content-page-hits were significant, not counts of discussion posts or reads. [24] also found no significant correlations between number of posts and student success in a graduate level course, but success variability was very low and the number of students was only 18.

Prior to MOOCs, the largest number of students in any of these studies was 214 [17]. This is one profound difference in the MOOC era, where tens of thousands of students participate and often thousands complete an online course. More recent analyses of discussion forum use in large MOOCs include the following: one analysis found that superposters elicited more posting from their less prolific peers, but the study did not analyze the impact of posting behavior on performance [14]. A randomized controlled trial comparing students with access to chat and discussion forums to students with access to only discussion forums found no differences in retention or performance between groups [6]. Background characteristics of forum users and the communication networks they formed were analyzed in [12], which found that higher performing students participated more in discussion forums but did not interact exclusively with other higher performing students.

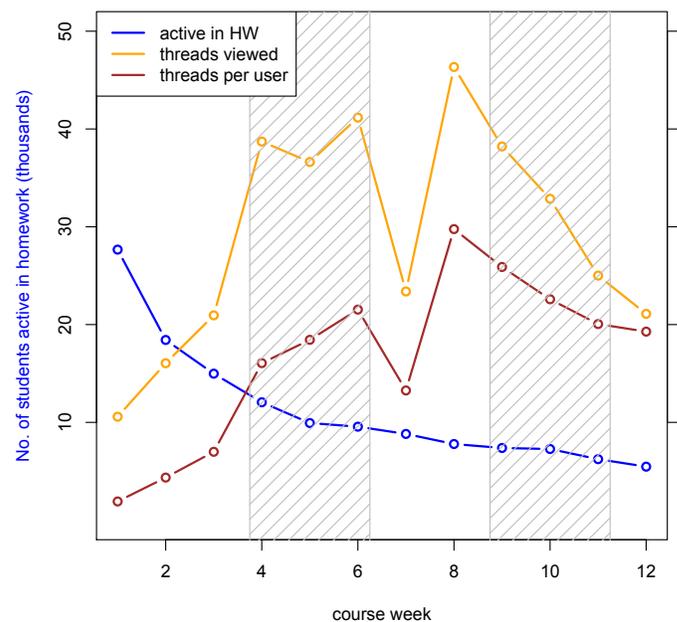
### MOOC DATA SET

The data for this study come from the Spring 2012 Circuits and Electronics MOOC on the MITx platform. Descriptive measures of discussion forum usage, homework

performance, and final exam scores were extracted from the MOOC clickstream logs using parsers written in Python [22]. Over 100,000 students registered for this course, though only half as many attempted to solve at least one problem in the course. Roughly 9000 attempted at least one problem on the final exam, and 7157 earned certificates.

Each access by a student to the discussion forum was recorded in the click-stream logs of the MOOC, as were the times when the student first opened each weekly homework assignment and the time of the last submit (the “homework window”). Thus it was straightforward to enumerate the number of threads viewed each week during the homework window. In this course, the most commonly referenced resource during homework solving was the discussion forum [22], which was structured as a Q&A board with up-voting and search capability (other course resources included lecture videos, an online textbook, and a wiki). Interestingly, most of this activity was “voyeuristic” not contributive: 67% of active students viewed (that is, clicked on—without scroll information and/or eye-tracking sensors, one cannot say for sure whether students read the threads they opened) at least one discussion thread between the first time they opened the homework and their last submission, whereas fewer than 10% posted a question, comment, or answer. Moreover 95% of all discussion activity in this course (by number of events) was viewing, not posting.

Because discussion forum content was generated by students, the forum was not as rich in the first few weeks of the course until participation reached a critical level, as shown in Figure 1.



**Figure 1: MOOC activity over time. Grey bars indicate early stage and late stage intervals on either side of the midterm.**

As seen in this figure, the number of students actively doing homework in our data set (active in homework, blue line)

decays over time, while activity in the forums increases before leveling off (threads per user, brown line). The midterm exam occurred between weeks 7 and 8, which explains the dip and then surge in discussion forum activity, as it was not permitted to post questions or answers about the midterm. The greyed regions of Figure 3 represent two three-week intervals, which we label “early stage”—weeks 4-6, after the discussion forum had fully taken off but before the midterm—and “late stage”—weeks 9-11, after the midterm but before the final exam. To smooth out week-to-week variation, we summed over views within each three-week long interval, as discussed below.

### CHALLENGES IN OPERATIONALIZING PREDICTORS

MOOCs differ from standard courses in a number of ways that make analyzing enrollee behavior difficult. These include higher than usual variability in prior educational attainment [20] and assessment motivation [26], large amounts of missing data, and affordances of multiple attempts on both formative and summative assessments [4]. Due to these issues, several researchers have noted that traditional measures of participation and achievement may need to be reconsidered in the context of MOOCs [5,7,13,16]. In this section, we introduce three sets of challenges, one for each predictor variable:

1. How can *prior ability* be estimated so that performance models can control for prior ability?
2. How should *discussion forum usage* be quantified? Is it a static quantity, or does it change over time?
3. Can we identify students who appear to be *disengaged/unmotivated*? What effect would excluding those students have on the effect size of forum usage?

#### Prior Ability

Enrollees in MOOCs range from high school students to professionals with earned doctorates [20]. Because overall performance is likely to depend on prior ability, this factor should be accounted for in any analysis of “treatment effects” from discussion forum usage. However, prior ability is typically unavailable information. Not all MOOCs survey incoming students, and those that do often survey sparsely. Enrollees in the Spring 2012 Circuits and Electronics MOOC were not given a pretest. Therefore, prior ability had to be inferred from the course data. In this study, we chose to estimate prior ability levels from performance on homework assignments in the first three weeks of the course, when enrollees had just begun to learn the content and before discussion forum use had taken off. The main idea was that early stage ability estimates were not likely to be affected by discussion forum usage, whereas final exam performance might be.

Because homework assignments allowed an unlimited number of attempts, the variability of the eventually correct (EC) score (the official score of record) was quite low. However, scoring items based on whether they were solved correctly on the first attempt (CFA) resulted in a far more

normal distribution (see Figure 2). A host of options for scoring homework in the presence of missing data and multiple attempts was described in [4]. While approaches based on polytomous item response models were most predictive of final exam scores, a reasonable improvement of the EC score was obtained for observed scores based on CFA. For simplicity, we use the mean CFA score, which is the proportion of homework problems attempted by each enrollee in the first three weeks of the course that were solved correctly on the first attempt. Skipped items are ignored, rather than scored as incorrect. For detailed considerations of homework scoring in MOOCs, we refer the reader to [4].

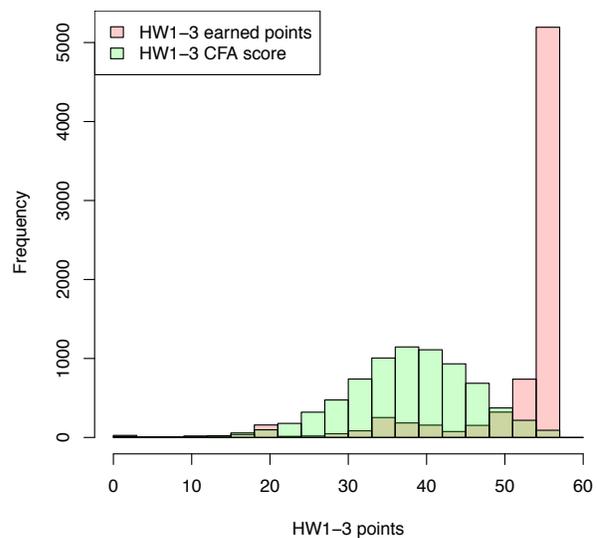


Figure 2: EC and CFA score distributions

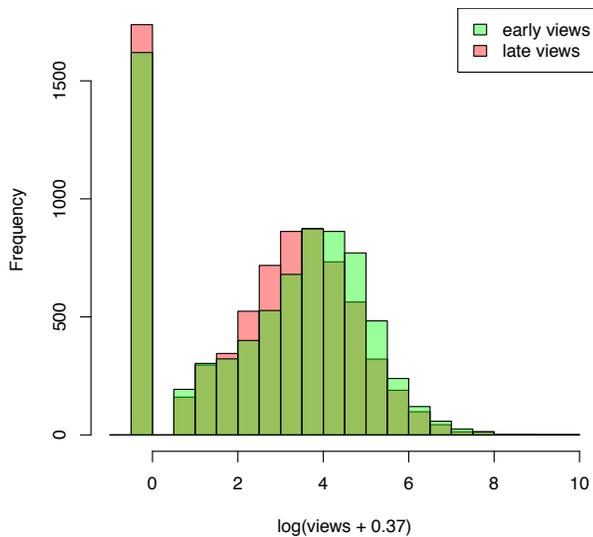
It should be noted that the issues of homework scoring also arise in the final exam, which is our outcome measure. We do not consider alternate scoring options, e.g. CFA scoring or item response theory, for the final exam. Only three attempts were allowed versus unlimited attempts on homework, and we did not want to punish students for strategically using their available attempts. However, there remain issues of examinee motivation, as discussed below.

#### Discussion Forum Usage

The average number of threads viewed per week was shown in Figure 1. We now explore the distribution over MOOC users of the early stage and late stage intervals (grey regions in Figure 1; the purpose of summing was to smooth out week-to-week variation.) We are interested in knowing both the distribution of counts within each interval—e.g. is it simple or bimodal?—as well as across the intervals—i.e. do learners exhibit consistent discussion usage over time, or does it change? These are important considerations for modeling the effect of discussion views. Consider students who purposefully increase their reference to forums after the midterm and reap performance gains as

a result. Modeling their usage as constant over time would distort the positive effect.

As shown in Figure 3, the early/late view count variables are of mixed type: many students do not view any threads, but among those who view at least one, the counts are roughly log-normally distributed. We have added 0.37 to all counts, such that after log-transformation, the students with zero counts appear in the disjoint bin at -1. As seen in the figure, there are roughly 1600 students in this bin for both early stage and late stage counts.



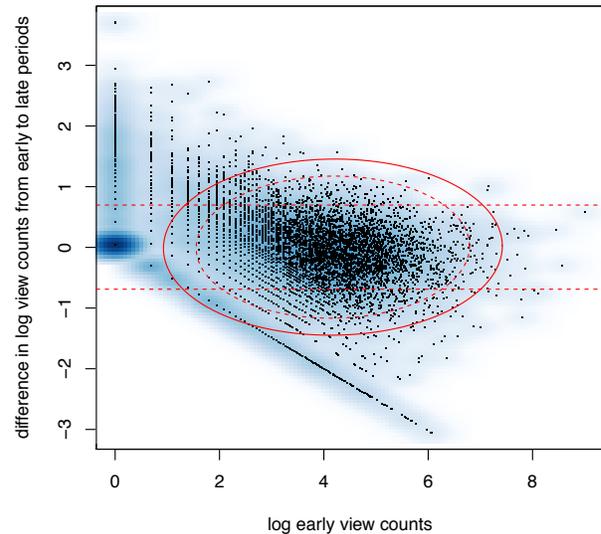
**Figure 3: Distribution of view counts (log-transformed)**

Figure 3 does not reveal whether there are students who significantly increase or decrease their discussion viewing between these time periods. Moreover, determining what amount of change is significant is a subtle point.

To address this question, we plot early view counts (scaled) against the difference between early and late counts (also both scaled) in Figure 4. Scatterplot and point density are both shown. There is a floor effect, which appears as a diagonal lower bound in the figure, representing students who went from a finite number of threads viewed in the early period to zero in the late period. Another salient feature is that for medium to large values of early counts, the change (from early to late counts) seems to be a random effect around zero (no change). This random description does not however fit all of the data. There does appear to be a clump of students on the upper left, whose viewing counts increase from very low levels to moderate levels. And there are some whose viewing decreases beyond the noise threshold. We chose to identify these students as outliers from the random distribution.

We determined empirical means and variances after removing low values and then drew a random sample of 7000 data points from a bivariate normal distribution with center  $\mu = (4.17, -0.27)$  and with covariance matrix  $\Sigma = (1.15, 0, 0, 0.84)$ . Elliptical contours are drawn at the 95%

and 99% confidence level in the figure. We have also included reference lines at the vertical mean value plus and minus  $\log(2)$ . The purpose of this second boundary is to define a criterion for those students whose early view counts were extreme outliers but whose change was still modest. Since the vertical axis is a difference of logarithms (or the log of the ratio), points outside this inner region represent doubling (or halving) in the counts.



**Figure 4: Change in discussion view counts against early counts. Ellipses denote 95% and 99% confidence intervals around a bivariate normal uncorrelated distribution. Dashed lines at  $\pm \log(2)$  denote doubling thresholds.**

As a result of this exploratory analysis, we divided our initial population into an *overall group* ( $N = 6505$ ), whose discussion viewing during homework could be seen as unchanging over time and thus aggregated into a variable  $V_O$ , and a *change group* ( $N = 989$ ), whose viewing change  $V_C$  should be modeled instead.  $V_O$  is the sum of the early and late stage counts, and  $V_C$  is the difference. Each would subsequently be treated as a continuous variable in an overall model or a change model, respectively.

#### Assessment-oriented Engagement and Total Time as Proxy Measures of Motivation

Inferences about ability from standard measures of performance may not always be valid in a MOOC due to differences in enrollees' motivations for taking the course. The expectancy-value model [9] puts the validity problem as follows: achievement motivation is influenced by both the individual's expectancies for success and the subjective value attached to success on the task. If the value of success is low, the examinee's achievement motivation will be low. Motivation thus acts as a source of construct-irrelevant variance and impacts the validity of score-based inferences [10]. In a meta-analysis of twelve empirical studies, [26] found that motivated students scored on average 0.59

standard deviations higher than their unmotivated counterparts. Such a result highlights the need to evaluate examinee motivation and possibly filter data from unmotivated test-takers to strengthen the assumption that a score obtained from an assessment accurately reflects the underlying abilities/traits of interest [1].

Consider the final exam score, which typically counts heavily toward qualification for a certificate (in the course under study, the final counted for 40% of the cumulative grade). However, the MOOC certificate is largely symbolic when it confers no degree credit. Thus, enrollees whose motivations for taking the course do not include certification may well view the final exam as low-stakes. The consequentiality of certificates may, in fact, change as more MOOCs seek accrediting status and even charge fees accordingly.

In the following, we consider three solutions to this problem, which is essentially the problem of whom to include. The first is to use a heuristic cutoff with respect to proportion of items attempted in the initial and final ability assessments. In the second solution, we attempt to filter out unmotivated students using a simple measure that should be relatively insensitive to the initial and final assessments, namely total time spent online in the course. The third and most intricate solution will be to use a latent class cluster analysis to model the course population as a mixture of classes based on cumulative evidence of assessment-oriented engagement (AOE). Thus both AOE and time-on-task are effective proxy measures for motivation, but we continue to use the original term in order to make contact with validity literature.

#### *Motivation heuristic filter on attempts*

Screening out students who attempted less than 60% of the HW1-3 items (which constitute our proxy measure of “prior ability”) or less than 60% of the final exam leaves 6210 students. This proportion is chosen to match the passing grade threshold of the course; in order to achieve this minimum, a student must at the very least attempt the same fraction of assessment items. This cutoff ignores the proportion of attempts on items in between Week 3 and the final exam, which will enter into the latent class analysis.

Although this is a filter based on attempts and not scores, it raises selection bias issues. While low-performing students who at least attempted many items would remain, this filter does, by definition, remove low scoring students. Thus our proxy for motivation is wrapped up in the outcome variable of our analysis. The rationale for solution two is partly a response to the bias of solution one.

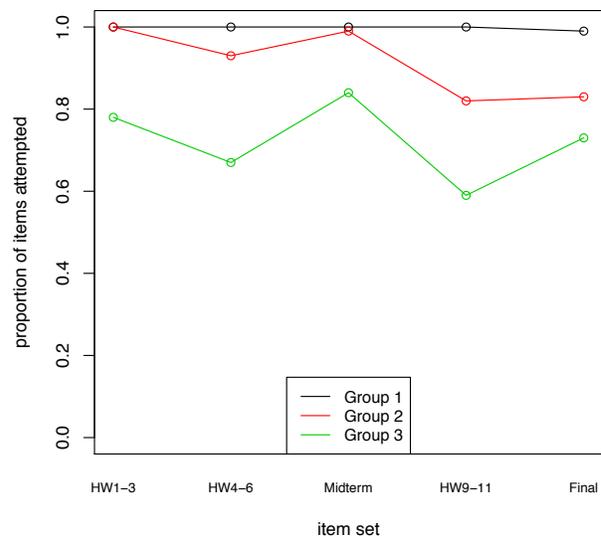
#### *Motivation heuristic filter on time*

What if there were students who invested significant amounts of time and effort in this course but were simply unable to answer many questions and were disinclined to guess? Alternately, what if there were students who carelessly attempted many items, but whose investment in

the course was more accurately reflected in low overall time commitment. Rather than filter on proportion of assessment items, we considered overall time spent in the course as a proxy for motivation. All activity, including video views, was included in this time aggregate, which is roughly log-normally distributed (slightly skewed to the left) with a median value around 100 hours. At a minimum time cutoff of 30 hrs (~1.5 standard deviations below), 679 students would be excluded, leaving 6815.

#### *Motivation via latent class analysis of AOE*

In the third approach, rather than determine whom to include or exclude, we seek to identify self-similar groups of students based on a pattern throughout the course. We could then model the effect of discussion viewing separately for all groups. Our idea is related to the approach in [16], where week-by-week trajectories were clustered. The results of that analysis were largely interpreted in terms of proportion of assessment attempted, so we went directly to that measure as a basis for clustering. We used five measures based on proportion of assessment items attempted: homework in weeks 1-3, homework in weeks 4-6, midterm exam, homework in weeks 9-11, and final exam. Each student’s record of item attempts was thus mapped to a vector of five proportions, and these vectors were clustered using the Gaussian mixture model-based clustering algorithm in the MClust package [11] in R.



**Figure 5: Mean values of proportion of items attempted for three latent class cluster groups.**

The model-based approach used here differs from the clustering method in [16], but the results are consistent. The best fit was at three clusters. Mean values for proportion of items attempted are plotted in Figure 5. Groups 1-3 roughly correspond to what [16] called completing, disengaging, and sampling. Probably because we removed in advance students who did not attempt at least one final exam

problem, we do not have an auditor group, typified by students who watch videos but do not attempt any assessment items.

### SUBSTANTIVE ANALYSES

Having operationalized our predictors, we now turn to modeling the effect of discussion viewing on final exam performance. Using multiple linear regression, we examine the standardized regression coefficient for the discussion viewing term as a probe of effect size. Based on the exploratory analyses described above, discussion viewing was treated differently for those students whose usage levels were consistent overall versus those who changed their viewing amount between the early and late stages. We computed two different variables  $V_o$  and  $V_c$  for these two populations respectively. Variability in motivation was handled both through heuristic attempt-based and time-based filters as well as via latent class analysis.

#### Model and results using motivation filters

Consider the following linear model for predicting the final exam  $Y$  using prior ability  $\theta$  and overall discussion view counts  $V_o$ ,

$$Y = \beta_0 + \beta_1\theta + \beta_2V_o$$

The change model is identical except for the substitution of view change for overall views. Importantly, the populations included for each model are different, as described above.

Table 1 reports standardized regression coefficients  $\beta_2$  for these two models. The first column is the result when including all students who attempted at least one final exam problem and one homework item in weeks 1-3 (HW1-3 performance was the basis for estimating prior ability  $\theta$ ). The middle column shows results when excluding students who spent fewer than 30 hours online. The last column shows results excluding those who did not attempt at least 60% of both the final exam and the weeks 1-3 homework.

**Table 1: Standardized regression coefficients for discussion viewing factor in two models under different data thresholds (white cells  $p < .001$ ; grey cells not significant)**

	No filter	Time > 30h	Attempt > 60%
Overall $\beta_2$	0.24	0.18	-0.01
Change $\beta_2$	0.19	0.19	0.16

The effect of discussion viewing in the overall model (first row of Table 1) appears to be significant when no filter is applied. But this unfiltered population contains hundreds of students who attempted very few assessment items, so these coefficients are not necessarily trustworthy. Indeed, the effect of overall viewing starts to decline as the population is refined in the next two columns. Screening out students who spent comparatively little time in the course reduces the effect but not by much. On the other hand, after

screening out students who did not attempt at least 60% of those assessment items that formed the basis of the prior and outcome performance measures, the effect of discussion viewing disappears entirely.

At the least, it must be said that the effect size of discussion viewing in the overall model is sensitive to selection of students. We note that these models altogether explain only about 10% of the variance in the final exam. The midterm exam, for reference, is more predictive ( $R^2 = 0.22$ ).

The effect of discussion views in the change model (second row), in contrast, appears to be more robust under selection for motivated students. At first glance, it is not clear whether increases in viewing are translating into higher scores or decreases in viewing are translating into lower scores. The latter could be consistent with attrition, for example. However, if attrition were the dominant explanation, then the third column coefficient would also be small, since course droppers would have been screened out. Thus the change model coefficients suggest that increasing discussion views are associated with higher final scores. We believe that interpretation of this effect is improved with reference to the latent class models, described next.

#### Model and results for latent class analysis

**Table 2: Standardized regression coefficients for the overall viewing model with latent class cluster groups (white cells,  $p < .005$ ; grey cells are not statistically significant)**

$Y = \beta_0 + \beta_1\theta + \beta_2V_o + \beta_3G + \beta_4\theta G + \beta_5V_oG$							
	0.75	0.14	-0.09	0	0	0	G=1
				-0.76	0.05	0.05	G=2
				-0.96	0.09	0.53	G=3

In Table 2 we show the model equation and estimated parameters for overall viewing effect with latent class cluster assignments. There were significant interactions between the cluster groups  $G$  and the continuous prior ability and discussion variables for the overall model; therefore we include five coefficients. Group 1, the reference group, attempted almost all assessment items (see Figure 5). Because Group 2 and 3 attempted fewer items, the main effect for those groups ( $\beta_3$ ;  $p < .001$ ) is a lower expected final exam score. Indeed, Group 1 may be thought of as a more restrictive subsample from the third column of Table 1. The interpretation of this small negative  $\beta_2$  is not necessarily that discussion views hurt, of course. Among Group 1 students, more viewing may indicate challenges with homework that transfer into challenges on the final.

Given that students in Group 3 omitted significant numbers of assessment items, why would such students reap more rewards from viewing discussion threads ( $\beta_5$ )? A possible explanation is that discussion viewing is a proxy for activity within Group 3. Indeed, there were positive correlations

between overall views and final exam items *attempted* (0.38) as well as late-stage homework *attempted* (0.53). Students who viewed more also did more assessment items relative to other students in this group.

Finally, Table 3 shows the change model with latent classes. Comparing to the second row of Table 1, we see now that for Group 1, increasing views are no longer associated with higher final exam scores. Recall that this group comprises the most active population with respect to assessment items. Again, a plausible explanation is that increasing discussion views are simply an indication of increasing participation in Groups 2 and 3, for example due to late joiners to the course. The correlation between viewing change and final exam items attempted is low in both cases (roughly 0.06), but the correlation with late homework attempted is moderate (0.27 and 0.33 for Groups 2 and 3, respectively). For the sporadic users of assessment in these groups, the positive association of increasing discussion views over time is there, but it may be linked to increasing engagement with the homework.

**Table 3: Change model including latent class cluster groups (white cells,  $p < .05$ ; grey cells are not statistically significant)**

$Y = \beta_0 + \beta_1\theta + \beta_2V_C + \beta_3G + \beta_4\theta G + \beta_5V_C G$							
	0.76	0.18	-0.05	0	0	0	G=1
				-0.80	-0.06	0.22	G=2
				-1.14	-0.15	0.21	G=3

### CONCLUSIONS AND FUTURE WORK

We started out with a simple goal of studying the learning outcome benefit from viewing discussion threads while doing homework in a MOOC. Along the way, it became clear that operationalizing almost all of the variables in this equation presented challenges. We have considered solutions to several issues that are endemic to MOOCs: estimating prior ability; determining whether to use an overall or a change model of discussion viewing; and screening out unmotivated students for the purpose of increasing the validity of inferences.

In the end, neither overall discussion viewing (for those whose viewing was fairly steady) nor change in discussion view volume appeared to be significant for students who attempted most of the assessment items, i.e. Group 1. The gain that appears from a naïve application of a linear model to the larger student sample (Table 1, column 1) seems to be due to confounding discussing thread viewing with participation, among sporadic participants. More work would need to be done to decouple use of the discussion forum from assessment-oriented engagement, for example by treating the latter as a continuous measure rather than as an indicator on which to filter the population. Moreover, counting discussion thread views is a limited window into usage of the forums. We did not analyze posting or

commenting in this analysis, nor did we discriminate between threads using textual analysis.

We did not say much about why the effect size of discussion viewing seemed insensitive to filtering students by overall time spent online. We suspect this is because there were hundreds of students who scored very highly on the final exam in this course but spent almost no time learning; in other words, these students already knew the content, but took the tests for fun or for the certificate.

As suggested above, we suspect that late joiners—whose increasing viewing over time appeared to associate with score gains—were a foil in this analysis. It would be interesting to dig deeper into how to model students whose trajectories of participation are increasing or decreasing over time. Also, although we used the final exam because it was an obvious choice, it may be possible to model the effect of discussion viewing on homework performance directly. There are subtleties to this, because multiple attempts increase the likelihood of correct responses. From a learning science perspective, looking at how students search the forums to get homework assistance may also be a fruitful direction.

### ACKNOWLEDGMENTS

We are grateful to edX for providing the raw data for this analysis, to Daniel Seaton for critical contributions to the processing of these data, and to helpful suggestions from reviewers. DEP would like to acknowledge support from a Google faculty award and from MIT.

### REFERENCES

1. AERA, APA, NCME. *Standards for educational and psychological testing*. American Educational Research Association, Washington, D.C., 1999.
2. Andresen, M. Asynchronous Discussion Forums: Success Factors, Outcomes, Assessments, and Limitations. *Educational Technology & Society* 12, (2009), 249–257.
3. Bates, A.W.T. *Technology, E-Learning and Distance Education*. Routledge, 1995.
4. Bergner, Y., Colvin, K., and Pritchard, D.E. Estimation of Ability from Homework Items When There Are Missing and/or Multiple Attempts. *Proceedings of LAK 2015*, (2015).
5. Clow, D. MOOCs and the funnel of participation. *Proceedings of the Third International Conference on Learning Analytics and Knowledge Discovery*, (2013), 185–189.
6. Coetzee, D. and Hearst, M.A. Chatrooms in MOOCs : All Talk and No Action. (2014), 127–136.
7. DeBoer, J., Ho, A.D., Stump, G.S., and Breslow, L. Changing “Course”: Reconceptualizing Educational

- Variables for Massive Open Online Courses. *Educational Researcher March*, (2014), 74–84.
8. Dringus, L.P. and Ellis, T. Using data mining as a strategy for assessing asynchronous discussion forums. *Computers & Education* 45, 1 (2005), 141–160.
  9. Eccles, J.S. and Wigfield, A. Motivational Beliefs, Values, and Goals. *Annual Review of Psychology* 53, (2002), 109–132.
  10. Eklof, H. Development and Validation of Scores From an Instrument Measuring Student Test-Taking Motivation. *Educational and Psychological Measurement* 66, 4 (2006), 643–656.
  11. Fraley, C. and Raftery, A.E. Model-based Clustering, Discriminant Analysis and Density Estimation : *Journal of the American Statistical Association* 97, (2002), 611–631.
  12. Gillani, N. and Eynon, R. Communication patterns in massively open online courses. *The Internet and Higher Education* 23, (2014), 18–26.
  13. Ho, A.D., Reich, J., Nesterko, S.O., et al. HarvardX and MITx: The First Year of Open Online Courses, Fall 2012-Summer 2013. *SSRN Electronic Journal*, (2014).
  14. Huang, J., Dasgupta, A., Ghosh, A., Manning, J., and Sanders, M. Superposter behavior in MOOC forums. *Proceedings of the first ACM conference on Learning @ scale conference - L@S '14*, (2014), 117–126.
  15. Kay, R.H. Developing a comprehensive metric for assessing discussion board effectiveness. *British Journal of Educational Technology* 37, 5 (2006), 761–783.
  16. Kizilcec, R., Piech, C., and Schneider, E. Deconstructing Disengagement: Analyzing Learner Subpopulations in Massive Open Online Courses. *Proceedings of the Third International Conference on Learning Analytics and Knowledge Discovery*, (2013).
  17. Kortemeyer, G. Correlations between student discussion behavior, attitudes, and learning. *Physical Review Special Topics - Physics Education Research* 3, 1 (2007), 010101.
  18. Kovanović, V., Gašević, D., Dawson, S., Joksimović, S., Baker, R.S., and Hatala, M. Penetrating the black box of time-on-task estimation. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge - LAK '15*, ACM Press (2015), 184–193.
  19. Ramos, C. and Yudko, E. “Hits” (not “Discussion Posts”) predict student success in online courses: A double cross-validation study. *Computers & Education* 50, 4 (2008), 1174–1182.
  20. Rayyan, S., Seaton, D.T., Belcher, J., Pritchard, D.E., and Chuang, I. Participation And performance In 8.02x Electricity And Magnetism: The First Physics MOOC From MITx. (2013), 4.
  21. Rowntree, D. Teaching and learning online: a correspondence education for the 21st century? *British Journal of Educational Technology* 26, 3 (1995), 205–215.
  22. Seaton, D.T., Bergner, Y., Chuang, I., Mitros, P., and Pritchard, D.E. Who does what in a massive open online course? *Communications of the ACM* 57, 4 (2014), 58–65.
  23. Soller, A. Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in ...* 12, 1 (2001).
  24. Song, L. and McNary, S. Understanding students’ online interaction: Analysis of discussion board postings. *Journal of Interactive Online Learning* 10, 1 (2011), 1–14.
  25. Thomas, M.J.W. Learning within incoherent structures: the space of online discussion forums. *Journal of Computer Assisted Learning* 18, 3 (2002), 351–366.
  26. Wise, S.L. and DeMars, C.E. Low Examinee Effort in Low-Stakes Assessment: Problems and Potential Solutions. *Educational Assessment* 10, 1 (2005), 1–17.

# Influence Analysis by Heterogeneous Network in MOOC Forums: What can We Discover?

Zhuoxuan Jiang<sup>1</sup>, Yan Zhang<sup>2</sup>, Chi Liu<sup>1</sup>, Xiaoming Li<sup>1</sup>  
Institute of Network Computing and Information System  
Peking University, Beijing, China  
<sup>1</sup>{jzhx,liuchi,lxm}@pku.edu.cn, <sup>2</sup>zhy@cis.pku.edu.cn

## ABSTRACT

With the development of Massive Open Online Courses (MOOC) in recent years, discussion forums there have become one of the most important components for both students and instructors to widely exchange ideas. And actually MOOC forums play the role of social learning media for knowledge propagation. In order to further understand the emerging learning settings, we explore the social relationship there by modeling the forum as a heterogeneous network with theories of social network analysis. We discover a specific group of students, named representative students, who feature large engagement in discussions and large aggregation of the majority of the whole forum participation, except the large learning behavior or the best performance. Based on these discoveries, to answer representative students' threads preferentially could not only save time for instructors to choose target posts from all, but also could propagate the knowledge as widespread as possible. Furthermore if extra attention is paid to representative students in the sight of their behavior, performance and posts, instructors could readily get feedback of the teaching quality, realize the major concerns in forums, and then make measures to improve the teaching program. We also develop a real-time and effective visualization tool to help instructors achieve these.

## Keywords

MOOC forum, Coursera, influence, behavior, performance, heterogeneous network

## 1. INTRODUCTION

Comparing with the traditional distance education or online courses, discussion forums in Massive Open Online Courses (MOOC) offer a big and lively venue for communication between students and instructors, which has been proved important for large-scale social learning [1, 7, 9]. However, due to their massiveness, the forums are full of various information relevant and irrelevant to the course [6]. So how to fast and accurately extract valuable information from the large-scale settings has become a problem to which priority should be given.

Considering Twitter, Facebook or StackOverflow, MOOC forums look similar to some kind of social media because of the large number of participants and their interactivity. Every member in the forum may talk about course content, such as asking or answering a question. The intensive interaction between them actually supports the knowledge propagation between members of the learning community. However here comes up a dilemma. In light of knowledge propagation, the proportion of instructors' responses is expected as large as possible in order to resolve students' questions; But considering the scale, instructors could not have enough time to read every thread. In order to cope with this situation, we propose a trade-off solution that extracts influential students from all and recommended them to instructors. Then instructors could make decisions in a much smaller scale and their effort would be amplified based on principles of influence propagation [12, 16, 24].

Although the definition of influence is various from different perspectives, we leave aside others except instructor for the time being in this paper. We conceive in each forum there could be a group of influential students who attract many others to interact with them, just like the verified accounts in Twitter. We call them 'representative students' and they involuntarily undertake the responsibility for knowledge propagation. So instructors could amplify the influence of right answers by preferentially responding to questions of representative students. Thus, many more students who pay attention to representative students' answers would also benefit without actually having a response by the instructor. On the other hand, given that representative students' threads may get a lot of attention, instructors could address the main concerns in the learning community more promptly. Through the rank list of representative students' influence, the chief instructor could also realize whether other instructors (or called TAs) are on duty, since TAs' influence could be calculated meanwhile. As we show later in this paper, representative students' performance is not the best within the learning community, but given their positive motivation and high volume of messages answering promptly their questions is beneficial for the whole learning community.

Since posts irrelevant to the course are unavoidable in such a free forum, for example chatting, making friends or other things, it is not reasonable to directly regard superposter [9] as representative students or merely consider their social relationship. Experiments later in this paper approve the opinion and find post contents are useful. That being the case, since we regard the interaction in MOOC forums as the procedure of knowledge propagation in social media, we could build a heterogeneous network [23] to model the forum with two kinds of entities by leveraging theories of networked entities ranking. Then we can get a rank list of students' in-

fluence from that network with a specially designed algorithm. The higher a student ranks on the list, the more influential she would be. This model could fully utilize the social information and textual messages to avoid outliers or exceptions (e.g. someone who always submits posts irrelevant to the course).

To our knowledge, this is the first work to adopt a heterogeneous network to model social relationship in MOOC forums and extract representative students. We also propose a novel algorithm for ranking students' influence based on graphic theories. Experimental results show the effectiveness and efficiency of the algorithm are both decent. Through the analysis of representative students' log data, we find they engage highly and aggregate much participation except the excellent grades, which suggests they are representative for instructors to watch the class and are the first low hanging fruit for increasing the passing rate. Analysis of historical records of interaction between instructors and students indicates it is time-saving and meaningful for instructors to recommend threads of representative students. Based on those discoveries, we developed a web service of visualization tool as an assistant for instructors to achieve the conception of supervising their class effort-savingly.

## 2. RELATED WORK

In traditional off-line classes, the scale is relatively small and face-to-face Q&A is not a challenge. And in traditional online education or online video class, not only the scale is not large enough but the absence of instructors is very common. However, a widespread viewpoint is that it is quite important for MOOC to make students engage in a social learning environment to guarantee and improve the teaching quality [1, 6, 7, 18].

In view of researches in the field of Community Question Answering (CQA), issues related to this paper are about expert finding and forum search [21]. Recently, several novel methods for finding experts in CQA have been provided [26, 29, 30]. Nevertheless, there would be rare experts in MOOC forum due to the specificity that a MOOC forum is not open to all kinds of discussions and it just belongs to the corresponding course for students to acquire knowledge. Also the definition of representative students here is different from that of experts. On the other hand, the task of discovering representative learners and their posts seems like forum search [3, 19] which develops a mechanism analogous to a search engine. But here we concentrate on just the ranking result and not emphasise the accuracy of retrieval. Except those general forum-related work, recently some researches of MOOC forums have been published from various perspectives. For example, Yang et al. [25] tried thread recommendation for MOOC students with method of an adaptive feature-based matrix factorization framework. Wen et al. [22] analyzed the sentiment in MOOC forums via students' words for monitoring their trending opinions. And Stump et al. [20] proposed a framework to classify forum posts.

The classical PageRank [5] and HITS [14] have been applied on broad problems of networked entities ranking and been promoted to solve problems in heterogeneous network [11, 15, 27]. [17, 28] built a heterogeneous network with two types of nodes to discover the influential authors with scientific repository data, which is similar to our work. The point in common is to discover influential entities with iteration by building a graphic model. In this paper, we leverage that principle and build a new heterogeneous network to model MOOC forum and discover representative students.

Besides, many MOOC log analysis also involve forums. Ander-

**Table 1: Pairs of course code and course title**

Course Code	Course Title
peopleandnetworks-001	Networks and Crowds
arthistory-001	Art History
dsalgo-001	Data Structures and Algorithms A
pkuic-001	Introduction to Computing
aoo-001	The Advanced Object-Oriented Technology
bdsalgo-001	Data Structures and Algorithms B
criminallaw-001	Criminal Law
pkupop-001	Practice on Programming
chemistry-001	General Chemistry (Session 1)
chemistry-002	General Chemistry (Session 2)
pkubioinfo-001	Bioinformatics: Introduction and Methods (Session 1)
pkubioinfo-002	Bioinformatics: Introduction and Methods (Session 2)

**Table 2: Statistics per course**

Course	# threads	# posts	# votes
peopleandnetworks-001	219	1,206	304
arthistory-001	273	2,181	1,541
dsalgo-001	283	1,221	266
pkuic-001	1,029	5,942	595
aoo-001	97	515	204
bdsalgo-001	319	1,299	132
criminallaw-001	118	763	648
pkupop-001	1,085	6,443	977
chemistry-001	110	591	65
chemistry-002	167	715	678
pkubioinfo-001	361	2,139	1,474
pkubioinfo-002	170	942	235
Overall	4,259	24,042	-

son et al. [2] deployed a system of badges to produce incentives for activity and contribution in the forum based on behavior patterns. Huang et al. [9] specially analyzed the behavior of superposter in 44 MOOC forums and found MOOC forums are mostly healthy. Kizilcec et al. [13] did a research on the behavior of students disengagement. Some technical reports and study case papers also involved behavior analysis of MOOC students in forums, such as [8] and [4]. Nevertheless, we believe incentives established on intelligent analysis of various data like social information and textual messages would be more reasonable than on the pure credits mechanism in traditional forums, since the latter only considers the quantity of behavior while not the quality.

## 3. DATASET

We use all the log data of 12 courses from Coursera platform. They were offered in Fall Semester of 2013 and Spring Semester of 2014. There are totally over 4,000 threads and over 24,000 posts. For convenience later in the paper, Table 1 lists the pairs of course code and course title. Table 2 shows the statistics of the dataset per course. Here posts denotes responses including posts and comments. We can see both the subjects and scales range widely.

## 4. MODEL AND ALGORITHM

In order to model MOOC forums as social media, the first challenge is that no explicit post-reply relationship which describes who replies who is recorded. We simplify this problem and assume

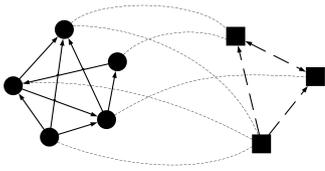
**Table 3: Attributes of the heterogeneous network constructed per course**

Course	$G_S$			$G_K$			$G_{SK}$	
	$n_S$	$ E_S $	$ E_S /n_S^2$	$n_K$	$ E_K $	$ E_K /n_K^2$	$ E_{SK} $	$ E_{SK} /(n_S + n_K)^2$
peopleandnetworks-001	321	3,287	0.032	1,193	104,821	0.074	4,814	0.002
arthistory-001	540	17,022	0.058	3,376	1,019,289	0.089	14,195	0.001
dsalgo-001	295	1,876	0.022	1,152	124,118	0.094	5,009	0.002
pkuic-001	768	19,801	0.034	2,302	302,989	0.057	14,599	0.002
aoo-001	175	1,963	0.064	783	73,208	0.119	2,597	0.003
bdsalgo-001	225	2,369	0.047	781	23,540	0.039	3,133	0.003
criminallaw-001	219	2,971	0.062	1,224	123,737	0.083	4,577	0.002
pkupop-001	628	12,883	0.033	1,748	88,035	0.029	13,807	0.002
chemistry-001	130	886	0.052	1,055	111,026	0.100	2,685	0.002
chemistry-002	125	2,341	0.150	964	61,425	0.066	2,574	0.002
pkubioinfo-001	594	22,275	0.063	686	46,768	0.099	1,946	0.001
pkubioinfo-002	189	1746	0.049	380	16662	0.115	784	0.002

**Table 4: Notations**

Notation	Description
$G = (V, E, W)$	heterogenous network
$G_S = (V_S, E_S, W_S)$	student subnetwork
$G_K = (V_K, E_K, W_K)$	keyword subnetwork
$G_{SK} = (V_{SK}, E_{SK}, W_{SK})$	bipartite subnetwork
$n_S, n_K$	$ V_S ,  V_K $

if two students appear in the same thread, they have the same topic interests and the one whose post is chronologically later replies the other. As mentioned in previous sections, post contents of representative students should be course-related. Thus it may be not enough to cover that demand with only extracting the post-reply relationship. Based on the fact that the most post contents are course-related [9], we add the keywords as another kind of entities into the model to construct the heterogenous network. The keywords here are all meaningful nouns in post contents and they could represent various aspects of topics. Other kinds of parts of speech are unexplored at the present. The role of keywords in the heterogenous network is to help the algorithm reinforce the influence of students who involve more topics, which ensures the need that posts of representative students are course-related. Figure 1 shows the demo of the heterogeneous network, and Table 4 lists the defined notations.

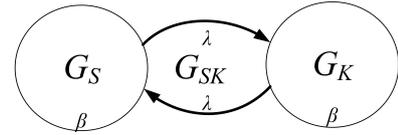


**Figure 1: Demo of the heterogeneous network  $G$ .** Circles denote  $V_S$  and rectangles denote  $V_K$ . Solid lines with arrows denote the co-presence relationship between students in the same thread and arrows denote one whose post is later points to the other. Dash lines with arrows denote the co-presence of keywords in the same thread but directed or bidirectional arrows mean the two keywords are in the different post or not. Dash lines without arrows denote the authorship between students and keywords. The weight values mean the times of co-presence of two entities on corresponding edges. Self co-presence is meaningless and all ignored.

This model captures the characteristic that representative students

would own more latent post-reply relationship and involve more topics. After building the network through log dataset, the basic attributes of graphs per course are calculated (Table 3).

For co-ranking students and keywords, we need an algorithm. We simulates two random surfers jumping and walking in the heterogeneous network and design the algorithm named Jump-Random-Walk (JRW). We assume the weights  $W$  represent the influence between entities and the algorithm's task is to discover the most influential students, namely representative students. Figure 2 shows the framework of JRW algorithm.



**Figure 2: The framework of Jump-Random-Walk algorithm.**  $\beta$  is the probability of walking along an edge within  $G_S$  or  $G_K$ .  $\lambda$  is the probability for jumping from  $G_S$  to  $G_K$  or in reverse.  $\lambda = 0$  means to discover representative students only by using post-reply relationship. We assume the probabilities of each jump or walk are consistent.

Denote  $\mathbf{s} \in \mathbb{R}^{n_S}$  and  $\mathbf{k} \in \mathbb{R}^{n_K}$  are the ranking result vectors, also probability distributions, whose entries are corresponding to entities of  $V_S$  and  $V_K$ , subject to  $\|\mathbf{s}\|_1 \leq 1$  and  $\|\mathbf{k}\|_1 \leq 1$ . Denote the four transition matrixes,  $G_S, G_K, G_{SK}$  and  $G_{KS}$ , for iteration as  $S \in \mathbb{R}^{n_S \times n_S}, K \in \mathbb{R}^{n_K \times n_K}, SK \in \mathbb{R}^{n_{SK} \times n_{SK}}$ , and  $KS \in \mathbb{R}^{n_K \times n_S}$  respectively. Adding the probability of random jumping for avoiding trapped in connected subgraph or set of no-out-degree entities, the iteration functions are

$$\mathbf{s} = (1 - \lambda)(\beta S \mathbf{s} + (1 - \beta) \mathbf{e}_{n_S} / n_S) + \lambda SK \tilde{\mathbf{k}}, \quad (1)$$

$$\mathbf{k} = (1 - \lambda)(\beta K \tilde{\mathbf{k}} + (1 - \beta) \mathbf{e}_{n_K} / n_K) + \lambda KS \mathbf{s}, \quad (2)$$

where  $\mathbf{e}_{n_S} \in \mathbb{R}^{n_S}$  and  $\mathbf{e}_{n_K} \in \mathbb{R}^{n_K}$  are the vectors whose all entries are 1. The mathematical forms of four transition matrixes are

$$S_{i,j} = \frac{w_{i,j}^S}{\sum_i w_{i,j}^S} \quad \text{where } \sum_i w_{i,j}^S \neq 0, \quad (3)$$

$$K_{i,j} = \frac{w_{i,j}^K}{\sum_i w_{i,j}^K} \quad \text{where } \sum_i w_{i,j}^K \neq 0, \quad (4)$$

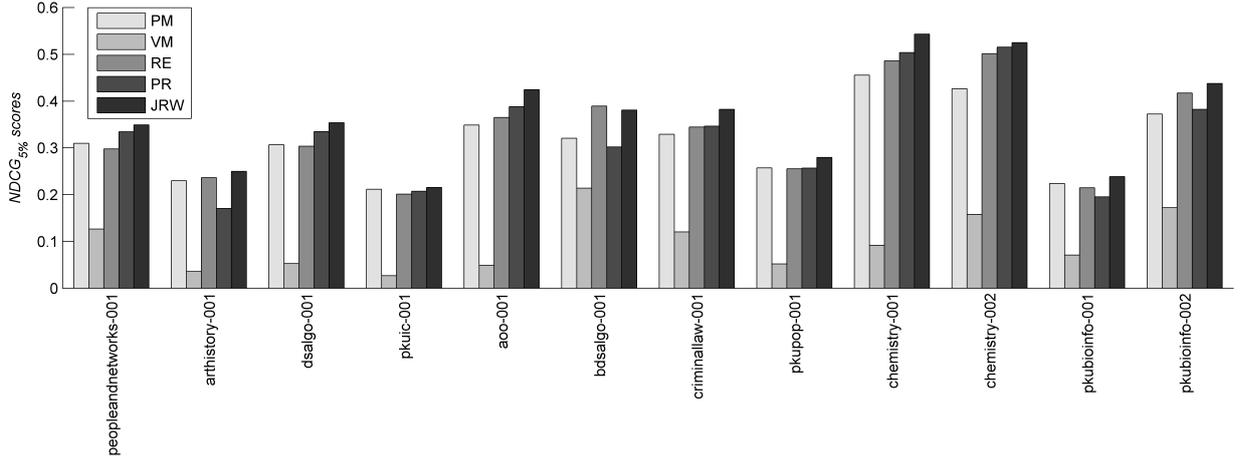


Figure 3:  $NDCG_{5\%}$  scores of different rankings

$$SK_{i,j} = \frac{w_{i,j}^{SK}}{\sum_i w_{i,j}^{SK}}, \quad (5)$$

$$KS_{i,j} = \frac{w_{i,j}^{KS}}{\sum_i w_{i,j}^{KS}} \quad \text{where } \sum_i w_{i,j}^{KS} \neq 0. \quad (6)$$

$w_{i,j}^S$  is the weight of the edge from  $V_i^S$  to  $V_j^S$ ,  $w_{i,j}^K$  is the weight of the edge between  $V_i^K$  and  $V_j^K$ ,  $w_{i,j}^{SK}$  is the weight of the edge between  $V_i^S$  and  $V_j^K$  and  $w_{i,j}^{KS}$  is the weight of the edge between  $V_i^K$  and  $V_j^S$ . Actually  $w_{i,j}^{SK} = w_{j,i}^{KS}$ . When  $\sum_i w_{i,j}^S = 0$ , it means the student  $V_j^S$  is always the last one in a thread. If  $\sum_i w_{i,j}^K = 0$ , it means the keyword  $V_j^K$  always has no peer in a thread. Actually this situation almost never happens in our filtered data.  $\sum_i w_{i,j}^{SK} = 0$  is also impossible, which means every keyword would have at least one author (student). On the contrary, it does not make sure that every student would post at least one keyword, because maybe there is some post having nothing valuable or not containing any nounal keyword. Algorithm 1 shows the detail of JRW algorithm below.

---

**Algorithm 1** Jump-Random-Walk on  $G$

---

**INPUT**  $S, K, SK, KS, \beta, \lambda, \epsilon$

1:  $s \leftarrow \mathbf{e}/n_S$

2:  $\mathbf{k} \leftarrow \mathbf{e}/n_K$

3: **repeat**

4:  $\tilde{s} \leftarrow s$

5:  $\tilde{\mathbf{k}} \leftarrow \mathbf{k}$

6:  $s = (1 - \lambda)(\beta S \tilde{s} + (1 - \beta)\mathbf{e}_{n_S}/n_S) + \lambda SK \tilde{\mathbf{k}}$

7:  $\mathbf{k} = (1 - \lambda)(\beta K \tilde{\mathbf{k}} + (1 - \beta)\mathbf{e}_{n_K}/n_K) + \lambda KS \tilde{s}$

8: **until**  $|s - \tilde{s}| \leq \epsilon$

9: **return**  $s, \mathbf{k}$

---

## 5. EXPERIMENTS

We do not exclude the data of instructors (or TAs) and regard everyone in the forums as ‘students’. So that instructors’ influence can also be evaluated in the uniform framework. Since the courses are all in Chinese and the contents are overwhelmingly most in simple

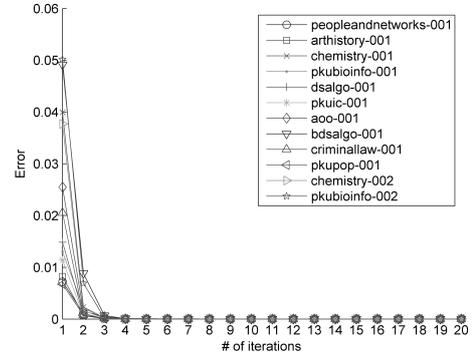


Figure 4: Iteration speed of Jump-Random-Walk

Chinese or traditional Chinese, we filter the non-Chinese contents in the preprocessing step with a tool of Chinese words segmentation which is essential for extracting Chinese keywords. Also we filter the HTML tags irregularly existed. During this process, most spam and valueless posts are filtered incidentally.

To evaluate the effectiveness of JRW, we set some competitors listed below.

- **Post the most (PM)**, for superposter by quantity. The more amount and frequency of posts are submitted, the higher she would rank.
- **Be voted the most (VM)**, for superposter by quality. The larger ratio of the number of votes earned to the average number of votes in a forum, the higher she would rank.
- **Reputation (RE)**, for superposter by reputation. It is a reputation score maintained by the Coursera platform and can be seen as a measure of both the quantity and quality of a forum student’s contribution.
- **PageRank (PR)**, for representative students only by post-reply relationship. It computes each forum student’s influence only in  $G_S$  with PageRank algorithm.

**Table 5: Representative students’ behavior and performance.**  $P(R|T)$  is the proportion of the number of threads initiated by representative students to the all.  $P(R|P)$  is the proportion of the number of posts by representative students to the all. *Over Rate* is the deviation of the average numbers of posts per thread initiated by representative students and the all.  $P(R|V)$  is the proportion of the number of watching video by representative students to the all.  $P(R|Q)$  is the proportion of the number of submitting quiz by representative students to the all.  $P(R|C)$  and  $P(R|C, D)$  are the proportions of certificated representative students and certificated representative students with distinction to the all. *Precise* is the proportion of the number of posts by instructors in threads initiated by representative students to that of all the instructors’ posts. *Recall* is the proportion of the number of threads replied by instructors to that of threads initiated by representative students.

Course	Forum Behavior			Learning Behavior		Performance		Instructor	
	$P(R T)$	$P(R P)$	<i>Over Rate</i>	$P(R V)$	$P(R Q)$	$P(R C)$	$P(R C, D)$	<i>Precise</i>	<i>Recall</i>
peopleandnetworks-001	0.205	0.246	1.182	0.084	0.074	0.126	0.167	0.267	0.556
arthistory-001	0.289	0.335	1.125	0.102	0.074	0.109	0.188	0.453	0.190
dsalgo-001	0.177	0.355	5.961	0.061	0.082	0.075	0.038	0.182	0.540
pkuic-001	0.282	0.444	-0.649	0.077	0.088	0.117	0.151	0.328	0.545
aoo-001	0.247	0.328	1.446	0.090	0.056	0.071	0.042	0.351	0.583
bdsalgo-001	0.210	0.473	0.401	0.110	0.047	0.047	0.054	0.286	0.866
criminallaw-001	0.246	0.326	1.524	0.060	0.067	-	-	0.504	0.793
pkupop-001	0.283	0.428	1.122	0.095	0.091	0.126	0.212	0.356	0.596
chemistry-001	0.082	0.367	1.706	0.050	0.076	0.078	0.079	0.207	1.000
chemistry-002	0.413	0.494	0.707	0.056	0.042	0.071	0.036	0.362	0.696
pkubioinfo-001	0.260	0.332	-0.963	0.097	0.061	0.075	0.061	0.284	0.713
pkubioinfo-002	0.200	0.445	0.282	0.029	0.035	0.028	0.035	0.210	0.706

- **Jump-Random-Walk (JRW)**, for representative students. It co-ranks the influence of both forum students and keywords meanwhile in  $G$ .

In order to compare with superposter, we set the same metric that a student is called a representative student when she is within top 5% of the rank list. Note that other alternative metrics, such as the threshold of an absolute number, are also feasible. The parameters used in JRW are  $\beta = 0.85$ ,  $\lambda = 0.5$  and  $\epsilon = 10^{-6}$ .  $\lambda = 0.2$  and  $\lambda = 0.8$  are also tried, however the differences are tiny. We adopt Normalized Discounted Cumulated Gain (NDCG) [10] as the metric which is applicable for evaluating rankings’ quality. We invited two human judges who both are experienced in MOOC forums. They give the influence of each top 5% student a score by reading all the contents of related threads. Each thread and post here are preprocessed to be anonymous and unordered. Score values include 0, 1, 2 and 3, which denotes strongly disagree, disagree, agree and strongly agree. Finally the two assessments are averaged.

Figure 3 shows the results of human assessment. JRW outperforms others among the majority of courses as well as PR, which suggests the necessity of building such a heterogeneous network for discovering representative students. If instructors would set a rule to incentivize representative students, JRW could also be more objective and fairer than simple rankings based on the quantity of behavior. Here is a phenomenon that students voted the most are not representative. This is maybe by reason that the majority of forum students are actually not used to voting the influential posts while unusual comments earn many. In addition, we carry out the convergence analysis of JRW algorithm. Figure 4 shows this algorithm can converge rapidly and satisfy the requirement of real-time computation in large-scale applications.

## 6. ANALYSIS OF REPRESENTATIVE STUDENTS

In this section, we would explore the characteristics of representative students in two aspects of behavior and performance. Then

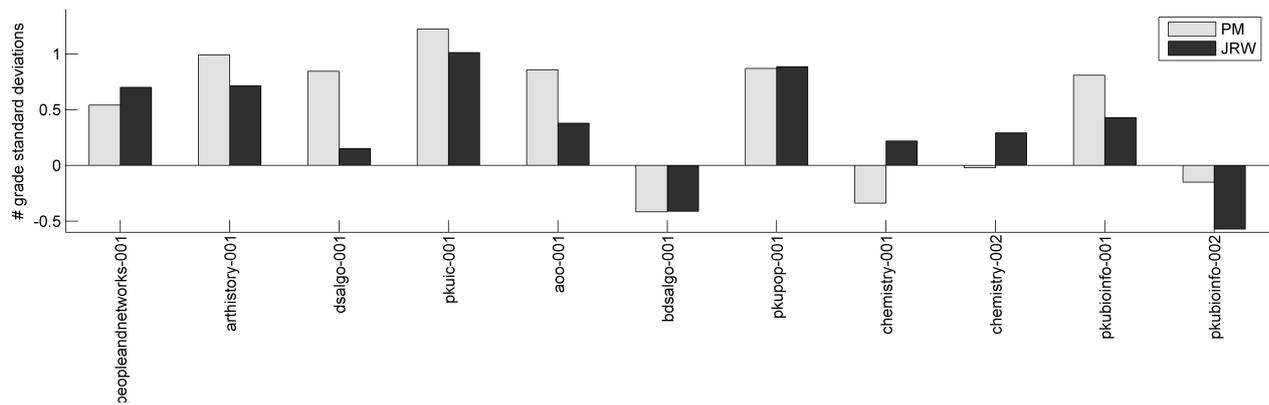
based on the model and algorithm proposed, we developed a web service which can help instructors supervise not only the behavior and performance of each student, but also their relative position compared with the average level of the whole class. This service could be competent for instructors to gain feedback of the teaching quality.

### 6.1 Behavior and Performance

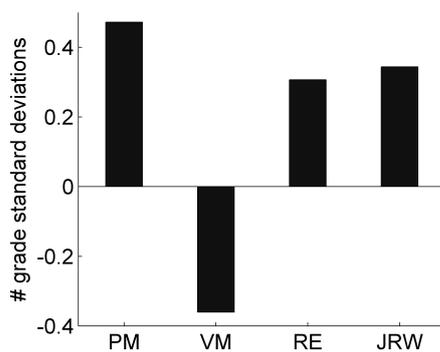
Firstly, we analyze the difference of behaviors between representative and non-representative students from a statistic view. Table 5 shows the proportions of various behavior of representative students to the whole forum students per course. The column of Forum Behavior contains three indicators, among which  $P(R|T)$  and  $P(R|P)$  reflect the degree of representative students’ participation in forums. *Over Rate* indicates if the value is over zero, it means representative students’ threads are more popular than the average, and vice versa. The values of the three indicators suggest in most forums representative students’ participation is relatively high considering their low ratio, only 5%, and their threads are more popular. In other words, the result here manifests threads of representative students initiate the majority of discussions, not counting in the possible sub-discussions initiated by them within a thread.

The column of Learning Behavior shows the behavior of watching video and submitting quiz by representative students. The values of the two indicators,  $P(R|V)$  and  $P(R|Q)$ , suggest the degree of learning behavior of representative students is relatively low compared with their participation, but still larger than 5%. So we can infer that representative students’ learning behavior is just above the average. This also suggests their motivation is positive by judging from the value of  $P(R|Q)$  which is related to the final certificate.

The column of Instructor demonstrates the necessity of preferentially answering the threads of representative students. *Precise* suggests instructors spent almost one third energy on answering representative students’ questions, while *Recall* suggests instructors have answered about two third, up to overall, threads initiated by



**Figure 6: # of standard deviations of representative students outperforming non-representative students on grades per course, comparing with superposters by quantity.**



**Figure 5: # of standard deviations of representative students outperforming non-representative students on grades averaged over all courses.**

representative students. The historical records explain it is necessary for instructors to discover the representative students and their posts, since the range and time cost of choosing which post to reply from all are both reduced. The indicator of *Over Rate* also implies preferentially answering the threads of representative students means more audience would be indirectly beneficial, without actually having a response by the instructor.

Then we would analyze the performance of representative students in the forums. Still in Table 5, the column of Performance denotes the proportions of certificated representative students.  $P(R|C)$  and  $P(R|C, D)$  are indicators of the passed and the excellent representative students respectively. The values indicate representative students have the higher proportion among the excellent students than the passed students in most courses. However it is potential to improve the proportion of passing rate considering the large forum participation and positive motivation of representative students. So they are worthy being paid extra attention by instructors.

Figure 5 shows the standard deviations, that are averaged z-score grades, to illustrate whether representative students' averaged grade outperforms that of non-representative students among all courses, comparing four different ranking metrics. Superposter by quantity (PM), superposter by reputation (RE) and representative students by JRW (JRW) outperform their peers. However, the score of JRW

is lower than that of PM. This may suggest representative students' performance is better than the peers, but not the group with best scores, and the top 5% students who post the most have the higher average score.

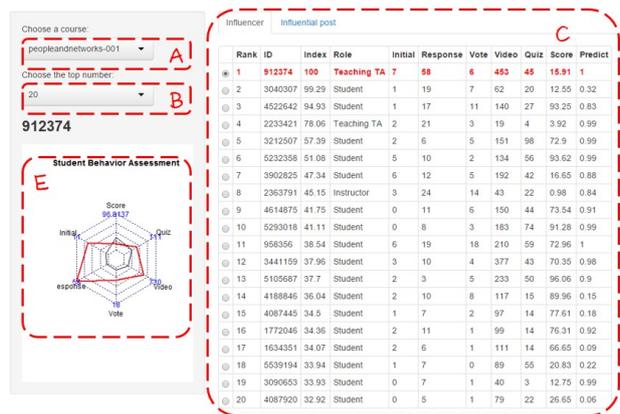
From the perspective of each course, representative students' performances are various. Figure 6 exhibits the same standard deviations per course. We can see representative students do not outperform their peers in some courses. Superposter and representative students almost show the consistent trends except for General Chemistry. Representative students' grade is lower than that of superposters by quantity in most courses, which also suggests representative students may have better performance above the average but not the best. This phenomenon could be explained that maybe similar to off-line class, representative students hard to master course content would involve more questions and need more instructions, while superposters by quantity are ones good at the course and always answer questions. So representative students are characterised by large participation of discussions, moderate learning behavior, and above-average performance but not the best.

## 6.2 Visualization Tool for Instructor

With the various forms of data, an open-and-shut visualization tool could be helpful for instructor to evaluate representative students and supervise their behavior. In order to apply the model proposed in previous sections to an actual function, we scale the final ranking scores to 0-100 as an index score, and developed such a web service whose interface looks as Figure 7.

Here we present the typical usage scenario of the service. Instructors could choose which course to see (Figure 7 A). Surely we would add role and permission administration to protect privacy in the future while here is just the demo of use cases. Then instructors could choose to see how many top students, at most overall (Figure 7 B). Instructors can also select to see the representative students' behavior (Figure 7 C) or their post contents (Figure 7 D). In the main exhibition area (Figure 7 C) where is a table list, instructors can realize the top students' various behavior, including forum participation, learning behavior and performance, students' influence index, and role in the forum. If instructors select to see 'influential post', the main area would be replaced by the post contents composed by representative students (Figure 7 D). We conceive that Figure 7 D should provide functions for instructors to re-

Forum Students Influence Index Rank List



Forum Students Influence Index Rank List

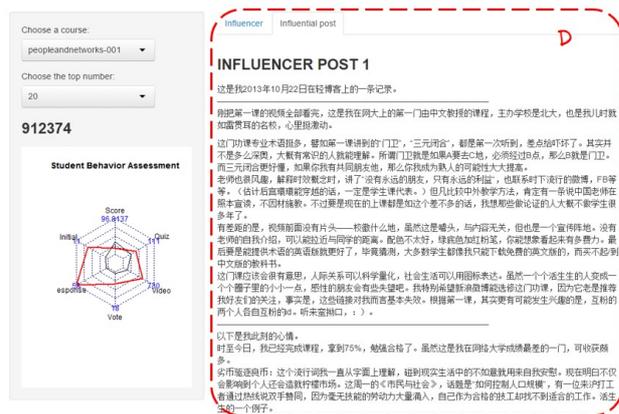


Figure 7: Web service interface

spond, rate, provide feedback and/or other post-related operations like those in the normal forum discussion settings in the future. Given the menu tab 'Influencer' selected, if instructors click the radio button ahead each record of the list, the behavior of corresponding student would also be presented in the radar chart (Figure 7 E). The radar chart displays six dimensions about students' behavior, that are quiz submission, video watching, vote, response, initiated thread, and final score. The scale of each dimension ranges from the minimum to the maximum of each class. Actually there are two closed hexagons on the radar chart. The fixed one in the middle denotes the average values in the whole class while the other, changed with trigger of radio click corresponding to each student, indicates the behavior of individual student. This radar chart can help instructors evaluate the behavior of each student comparing with the whole class under different dimensions.

In our observation and interview, this web service offers instructors the way to realize the class macroscopically and get feedback of main concerns in the forum promptly. Note that due to the rapid speed of our algorithm, this web service can real-time refresh with changes of students' forum behavior.

## 7. CONCLUSION AND FUTURE WORK

In the MOOC forum settings, different participants may consider the influence as different definitions. We stand at the side of instructors and assume the influencers in MOOC forums are representative students who stimulate and attract much forum participation. They are actually characterized by lively engagement in forum discussions but unexpected learning behavior and performance, comparing with superposter. They are worthy being paid extra attention from instructors thereby to improve the course passing rate. Since they aggregate much discussion, they could be helpful to amplify instructors' answers and play the latent roles of knowledge propagation. Through representative students' influence, instructors can time-savingsly realize the hot topics concerned by the most students. TAs' workload can be evaluated incidentally. In general, it is meaningful for instructors to preferentially read and answer representative students' threads.

In this paper, we leverage methods and algorithms of social network analysis to model MOOC forums in order to further understand the MOOC social learning settings and provide bases for in-

structors to intervene the social learning. This model has the advantages of fully utilizing social information and textual messages to identify and rank students' influence. Thus based on their behavior, performance and post contents, instructors may make measures to improve the teaching quality, better with that web service of visualization tool as an assistant.

Nevertheless, we have much future work to refine the discoveries in this paper. We would attempt other kinds of heterogeneous networks with more forum information and explore the effect of parameters. Some other random walk algorithms, such as HITS and topic based ones, would be more effective. Furthermore, by integrating our visualization tool into a practical platform, whether the amplification of knowledge propagation via representative students is effective and whether the teaching quality could be promoted still need to be verified through subsequent courses specifically designed in the future.

## 8. ACKNOWLEDGMENTS

This research was supported in part by 973 Program with Grants No.2014CB340405, NSFC with Grants No.61272340, No.61472013 and No.61370054.

## 9. REFERENCES

- [1] P. Adamopoulos. What makes a great mooc? an interdisciplinary analysis of student retention in online courses. In *Proceedings of the 34th International Conference on Information Systems, ICIS '14*, 2014.
- [2] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pages 687–698. ACM Press, 2014.
- [3] S. Bhatia and P. Mitra. Adopting inference networks for online thread retrieval. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI '10*, pages 1300–1305. AAAI Press, 2010.
- [4] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton. Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*, 8(1):13–25, 2013.
- [5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th*

- International Conference on World Wide Web, WWW '1998*, pages 107–117. Elsevier Science Publishers, 1998.
- [6] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong. Learning about social learning in MOOCs: From statistical analysis to generative model. *IEEE Transactions on Learning Technologies*, 7(4):346–359, 2014.
- [7] W. Cade, N. Dowell, A. Graesser, Y. Tausczik, and J. Pennebaker. Modeling student socioaffective responses to group interactions in a collaborative online chat environment. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM '14*, pages 399–400. Chapman & Hall/CRC Press, 2014.
- [8] HarvardX and MITx: The first year of open online courses, Fall 2012–Summer 2013. [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2381263](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2381263).
- [9] J. Huang, A. Dasgupta, A. Ghosh, J. Manning, and M. Sanders. Superposter behavior in mooc forums. In *Proceedings of the first ACM Conference on Learning @ Scale Conference, L@S '14*, pages 117–126. ACM Press, 2014.
- [10] K. Jarvelin and J. Kekalainen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00*, pages 41–48. ACM Press, 2000.
- [11] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 1298–1306. ACM Press, 2011.
- [12] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pages 137–146. ACM Press, 2003.
- [13] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge, LAK '13*, pages 170–179. ACM Press, 2013.
- [14] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.
- [15] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, pages 199–208. ACM Press, 2010.
- [16] Q. Liu, B. Xiang, E. Chen, H. Xiong, F. Tang, and J. X. Yu. Influence maximization over large-scale social networks: A bounded linear approach. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14*, pages 171–180. ACM Press, 2014.
- [17] Q. Meng and P. J. Kennedy. Discovering influential authors in heterogeneous academic networks by a co-ranking method. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13*, pages 1029–1036. ACM Press, 2013.
- [18] T. Schellens and M. Valcke. Fostering knowledge construction in university students through asynchronous discussion groups. *Computers & Education*, 46(4):349–370, 2006.
- [19] A. Singh, D. P. and D. Raghu. Retrieving similar discussion forum threads: A structure based approach. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, pages 135–144. ACM Press, 2012.
- [20] G. S. Stump, J. DeBoer, J. Whittinghill, and L. Breslow. Development of a framework to classify mooc discussion forum posts: Methodology and challenges. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 2013.
- [21] H. Wang, C. Wang, C. Zhai, and J. Han. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11*, pages 435–444. ACM Press, 2011.
- [22] M. Wen, D. Yang, and C. Rose. Sentiment analysis in mooc discussion forums: What does it tell us? In *Proceedings of the 7th International Conference on Educational Data Mining, EDM '14*, pages 130–137. Chapman & Hall/CRC Press, 2014.
- [23] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pages 266–275. ACM Press, 2003.
- [24] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang. Pagerank with priors: An influence propagation perspective. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13*, pages 2740–2746. AAAI Press, 2013.
- [25] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM '14*, pages 257–260. Chapman & Hall/CRC Press, 2014.
- [26] R. Yeniterzi and J. Callan. Analyzing bias in cqa-based expert finding test sets. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14*, pages 967–970. ACM Press, 2014.
- [27] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14*, pages 283–292. ACM Press, 2014.
- [28] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles. Co-ranking authors and documents in a heterogeneous network. In *Proceedings of the 7th IEEE International Conference on Data Mining, ICDM '07*, pages 739–744. IEEE Press, 2007.
- [29] G. Zhou, S. Lai, K. Liu, and J. Zhao. Topic-sensitive probabilistic model for expert finding in question answer communities. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 1662–1666. ACM Press, 2012.
- [30] H. Zhu, H. Cao, H. Xiong, E. Chen, and J. Tian. Towards expert finding by leveraging relevant categories in authority ranking. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 2221–2224. ACM Press, 2011.

# Modeling Learners' Social Centrality and Performance through Language and Discourse

Nia M. Dowell  
Department of Psychology  
Institute for Intelligent Systems  
University of Memphis  
365 Innovation Drive  
Memphis, TN 38152  
+1 901-678-5102  
ndowell@memphis.edu

Arthur C. Graesser  
Department of Psychology  
Institute for Intelligent Systems  
University of Memphis  
365 Innovation Drive  
Memphis, TN 38152  
+1 901-678-5102  
a-graesser@memphis.edu

Thieme A. Hennis  
Delft Extension School  
Delft University of Technology  
2628 BX, Delft  
+31651855220  
t.a.hennis@tudelft.nl

Oleksandra Skrypyk  
School of Education  
University of South Australia  
Adelaide, Australia  
+61 402918694  
olesandra.skrypyk@mymail.u  
nisa.edu.au

Shane Dawson  
Learning and Teaching Unit  
University of South Australia  
Adelaide, Australia  
+61 883027850  
shane.dawson@unisa.edu.au

Pieter de Vries  
Systems Engineering Department  
Participatory Systems Design  
Delft University of Technology  
2628 BX Delft, Netherlands  
+31651517278  
pieter.devries@tudelft.nl

Srećko Joksimović  
School of Interactive Arts and  
Technology  
Simon Fraser University  
Burnaby, Canada  
+1604-375-2496  
sjoksimo@sfu.ca

Dragan Gašević  
Schools of Education and Informatics  
University of Edinburgh  
Edinburgh, United Kingdom  
+44 131 651 6243  
dragan.gasevic@ed.ac.uk

Vitomir Kovanović  
Schools of Education and Informatics  
University of Edinburgh  
+1604-375-2496  
v.kovanovic@ed.ac.uk

## ABSTRACT

There is an emerging trend in higher education for the adoption of massive open online courses (MOOCs). However, despite this interest in learning at scale, there has been limited work investigating the impact MOOCs can play on student learning. In this study, we adopt a novel approach, using language and discourse as a tool to explore its association with two established measures related to learning: traditional academic performance and social centrality. We demonstrate how characteristics of language diagnostically reveal the performance and social position of learners as they interact in a MOOC. We use Coh-Matrix, a theoretically grounded, computational linguistic modeling tool, to explore students' forum postings across five potent discourse dimensions. Using a Social Network Analysis (SNA) methodology, we determine learners' social centrality. Linear mixed-effect modeling is used for all other analyses to control for individual learner and text characteristics. The results indicate that learners performed significantly better when they engaged in more expository style discourse, with surface and deep level cohesive integration, abstract language, and simple syntactic structures. However, measures of social centrality revealed a different picture. Learners garnered a more significant and central position in their social network when they engaged with more

narrative style discourse with less overlap between words and ideas, simpler syntactic structures and abstract words. Implications for further research and practice are discussed regarding the misalignment between these two learning-related outcomes.

## Keywords

Social Centrality, Learning, Discourse, Coh-Matrix, MOOCs

## 1. INTRODUCTION

Advances in educational technologies and a desire for increased access to learning, are enabling the development of pedagogical environments at scale, such as Massive Open Online Courses (MOOCs) [41]. Open online courses have the potential to advance education on a global level, by providing the masses with broader access to lifelong learning opportunities. Additionally, the insulated nature of the MOOC web-based platforms allows valuable learning dynamics to be detailed at unprecedented resolution and scale. As such, the digital traces left by learners are regarded as a gold mine that can offer powerful insights into the learning process, resulting in the advancement of educational sciences and substantially improved learning environments.

While the scale of the data has grown, making sense of data from the learning environments is not a novel effort. Prior to the arrival of MOOCs, similar endeavors were undertaken at smaller scale in the domains of computer-supported collaborative learning and intelligent tutoring systems, among others. The volume of student behavior and performance data produced in those interactions motivated the fields of educational data mining (EDM) and learning analytics (LA) [37]. Both of these research communities have leveraged this fine-grained data and aligned with educational

\*Textbox for copyright information\*

theory. The EDM community offer methods for exploring learners and educational settings, while LA focuses on the measurement, collection, and analyses that aim at optimizing the learning process [38]. That said, inquiring into MOOCs and other unexplored learning environments requires inputs from both communities. Direct application of methodologies, theoretical frameworks, and established analytics require deeper understanding of the relationships between parts of the whole, to enable drawing the relevant parallels with existing research.

Drawing on this, this paper adopts a novel approach, which uses language and discourse as a tool to explore its association with two established measures of learning, namely traditional academic performance and social centrality. Specifically, we are investigating the extent to which characteristics of language diagnostically reveal the performance and social position of students as they interact in a MOOC. As a methodological contribution, we adopt a theoretically grounded computational linguistics modeling approach to explore students' forum posting, within a MOOC, across five potent discourse dimensions. In line with current practice, we implement a Social Network Analysis (SNA) methodology to monitor and detect learners' social centrality. Students' performance in the course, i.e. course grade, is represented by an aggregate measure combining scores for the essays submitted during the MOOC, and a final peer-evaluated, open-ended written-assignment. Linear mixed-effects modeling approach is used for all other analyses to control for individual learner and text characteristics. This design allows us to contrast the linguistic profiles of high performing learners and centrally situated learners. Consequently, we gain insights into the qualitative differences between these two different learning-related outcomes. Finally, we explored whether the discourse features characterizing learning-related outcomes varied within different learner population, namely across all learners in the MOOC and within a subset of active learners.

The subsequent sections of the paper are organized as follows. First, we provide a brief overview of language and discourse situated within the contexts of psychological frameworks of comprehension and learning. Then, the following two sections address the traditional application of social network analysis, including theoretical foundations, as well as interpretations applied in MOOCs research. We then move into the methodological features of the current investigation, and conclude the paper with a detailed discussion of the results in the context of theory, as well as a general discussion of the theoretical, methodological, and practical implications for the EDM and LA community.

## 2. THEORETICAL BACKGROUND

### 2.1 Language and Discourse

Across academic fields, there has been a burgeoning literature demonstrating the usefulness of language and discourse in predicting a number of psychological, affective, cognitive, and social phenomena, ranging from personality to emotion to learning to successful group interactions (e.g. [6,10,26]). Within the educational contexts, there are many critical learning-related constructs that cannot be directly measured, but can be inferred from measurable signals like language and other behavioral patterns. Working with these barriers, we are continually pushing beyond the boundaries of established implementation. In that realm, it is particularly important that these endeavors be guided by established theory. A number of psychological models of discourse comprehension and learning, such as the construction-integration, constructionist, and indexical-embodiment models,

lend themselves nicely to the exploration of learning related phenomena in computer-mediated educational environments. These psychological frameworks have identified the representations, structures, strategies, and processes at multiple levels of discourse [16,23,40]. Five levels have commonly been offered in these frameworks: (1) words, (2) syntax, (3) the explicit textbase, (4) the situation model (sometimes called the mental model), and (5) the discourse genre and rhetorical structure (the type of discourse and its composition). In the learning context, learners can experience communication misalignments and comprehension breakdowns at different levels. Such breakdowns and misalignments have important implications for the learning process. In this paper we adopt this multilevel approach to the analysis of language and discourse.

With regard to analytical approaches, there has been extensive knowledge gleaned from manual content analyses of learners' discourse during educational interactions, however, these methods are no longer a viable option with the increasing scale of educational data. As such, researchers have been incorporating automated linguistic analysis, including more shallow level word counts and deeper level discourse analysis approaches. Both levels of linguistic analysis are informative. Content analysis using word-counting methods allows getting a fast overview of learners' participation levels, as well as assessing specific words. For instance, a study by Wen and colleagues [43] is an example of incorporating word counts (LIWC) of theory-informed and carefully selected words with manual message coding. Their work links specific (and thus identifiable and countable) words used by the students with the degree of their engagement and commitment to remain in the course.

To extend analysis of learning-related phenomena beyond the shallow level word counts, one needs to conduct a deeper level discourse analysis employing sophisticated natural language processing techniques, e.g. syntactic parsing and cohesion computation. For example, Dowell and colleagues [11] explored the possibility of using discourse features to predict student performance during collaborative learning interactions. Their results indicated that students who engaged in deeper cohesive integration and generated more complicated syntactic structures performed significantly better. In line with this, Cade and others [3] demonstrated that cognitive linguistic cues can be used in detecting students' socio-affective attitudes towards fellow students in CMCL environments. As a whole, these studies highlight the critical and complex role of language and discourse. This is, perhaps, not surprising, since language is a primary means for expressing and communicating information in computer-mediated learning environments.

### 2.2 Social Network Analysis in Educational Research

Social Network Analysis (SNA) is a methodology that is increasingly being used for analyzing learning-related phenomena, especially in online settings [25]. SNA has gained popularity with researchers who view social relationships between students as an aspect influencing overall educational experience and learning outcomes (i.e. [33]). Its methodology is grounded in systematic empirical data [4:8], as well as "motivated by a relational intuition based on ties connecting social actors" (*ibid.*). Studies that employ SNA, aim at revealing the role of social relationships in learning, around such issues as *who is central in a social learning network*, *who is talking to whom*, and *who is participating peripherally* and *how those interaction patterns influence learning* [4,25,42]. Due to such focus, SNA provides the

theoretical and methodological tools to understand activities and social processes that students and teachers engage with. [25,31]

Traditionally, the analyses of social networks of learners have been derived from participation in discussion forums in formal online courses. The relationship between learners' position in a social network and student academic performance is well documented, in this context [5,14,33]. The general finding in this literature shows more centrally situated learners tend to get higher final grades [33]. Moreover, Russo and Koesten [34] showed that network centrality (measured as in-degree and out-degree) is a significant predictor of cognitive learning outcome. Rizzuto and others [32] found that network density significantly predicted the scores reflecting course material comprehension. Reflective of the finding from these studies a students' position in a network also influences their overall sense of community [9]. These studies suggest, in the context of formal online learning, individuals who are centrally positioned in their network perform better, and feel a stronger sense of connection than students that are more peripheral in the network structure.

In the context of MOOCs, SNA is increasingly used to explore learning-related phenomena [13]. For example, Gilliani et al. [15] applied SNA to capture broad trends in communication and the roles of individuals in facilitating discussions [15]. Another example of SNA in MOOCs is a study by Yang and colleagues [44], which suggests that learners who join forums (i.e. networks of learners) earlier are likely to persist in the course, in contrast to their counterparts who joined later and found it difficult to form social bonds. This finding is parallel to prior findings in the domain of traditional online learning revealing that learners central to the social network tend to have a higher sense of belonging to the group [8]. However, there is research that suggests the interpretation of SNA in MOOCs requires further attention. For example, the relationship between student centrality in MOOC discussion forums and their academic performance (i.e., final grade), has been shown to be context dependent [21]. Jiang and colleagues [21] demonstrated that in Algebra MOOC, betweenness and degree centrality yielded significant correlation with the final grade, while none of the metrics analyzed (i.e., closeness, degree, and betweenness centrality) was significantly correlated with the learning outcome in a Financial Planning MOOC.

Automated linguistic analysis of student interactions, within computer-mediated learning environments, can compliment SNA techniques by adding rich contextual information to the structural patterns of learner interactions. However, the combination of these two analytical methods is relatively sparse in the literature, beyond a few noteworthy exceptions [22,36]. Similar to the current work, is Joksimović and colleagues' [22] analysis of students' interaction patterns in a distributed MOOC, i.e. learner interactions take place via social media, and the course is based on connectivist pedagogy. Their findings pinpoint specific discourse features that were predictive of a learners' accumulation of social capital.

### 2.3 Research Questions

To summarize, SNA is a widely used tool for exploring learning processes that take place in MOOCs, largely due to its theoretical foundation and established application in formal educational contexts. However, given the open nature of scaled online courses, the interpretation of SNA in MOOCs requires further attention. This study approaches language as the primary means for communication and a window into inferring learning-related phenomena. We apply discourse analysis as a proxy for providing

qualitative information about the position of learners in the network and their performance. The analysis focuses around the following research questions: Which characteristics of language diagnostically reveal the performance and social position of students as they interact in a MOOC? And do these features operate similarly with different learner populations, namely across all learners in a MOOC and within a subset of active learners?

## 3. METHODS

### 3.1 Participants

The study analyzed forum discussion posted on the edX platform, within the course NG1101x Next Generation Infrastructures (NGIx). It ran for 8 weeks in the period of April 22 – July 8, 2014. The subject area of the analyzed MOOC fell under the domain of applied non-life soft sciences [2]; the course objective was to introduce the complexity of infrastructure systems, familiarize students with the main concepts within the area, as well as with the practical approaches to the infra-systems analysis. In total 16,091 participants enrolled and 517 received certificate of completion (passed). To pass the course the students needed to receive a score of 0.7 (out of 1) or higher. The grade was derived from the submission of 3-6 open-ended papers (60% of the grade) and a final issue paper (40% of the grade) that was peer assessed by several co-learners. The dataset for the analysis in this study included 1,754 participants ( $N_{post}=7,244$ ,  $M=4.13$ ,  $SD=9.85$ ,  $Q1=1.0$ ,  $Q3=4.0$ ,  $Min=1.0$ ,  $Max=180$ ), i.e. all those who used the course forum. Forum data was collected from the edX platform in the JSON format, and included all the information specified within the edX discussion forums data documentation<sup>1</sup>.

### 3.2 Analyses

#### 3.2.1 Social Network Analysis

Although other approaches have been proposed, the most common approach for extracting social networks from online discussions is to consider each message as directed to the previous one in the thread [25,31]. In the current study, we followed the approach suggested in [24,25,31], among others. Specifically, social graph representing interaction within the discussion forum included all the students who posted a message(s). For example, author A1 initiated the discussion, and author A2 posted a message directly into the thread, in reply to A1's initial thread message, we would add directed edge A2->A1. Then, if author A3 replied to the message posted by author A2, we would include a direct edge A3->A2 to the graph. If author A4 started a nested discussion as a reply to A1's initial post, then A4 would have a direct edge to A1. The concept of centrality has been commonly used to assess the importance of an individual node within a social network [12,42]. The following well-established SNA measures [42], that capture various notions of a graph structural centrality, were calculated for each learner in the social network extracted:

- **Degree Centrality** – the number of edges a node has in a network;
- **Closeness Centrality** – the distance of an individual node in the network from all the other nodes;
- **Betweenness Centrality** – the number of shortest paths between any two nodes that pass via a given node.

Degree centrality is generally used to capture the “potential for activity in communication” [12:219] or the *popularity* [31] of a node in a social network. Betweenness centrality, on the other hand, represents a *potential for influence* over the information

<sup>1</sup> [http://devdata.readthedocs.org/en/latest/internal\\_data\\_formats/discussion\\_data.html](http://devdata.readthedocs.org/en/latest/internal_data_formats/discussion_data.html)

flow, as it *bridges* the parts of the network that were disconnected otherwise [12,31,42]. Finally, the concept of closeness centrality refers to the distance between a learner and the other participants of the network. In a MOOC, closeness centrality can be interpreted as the extent to which a learner is in the middle of what is happening on the forum. The relationship between students' linguistic properties and their position in the social network, measured through the three properties described above, has been investigated in this study. The social network variables were analyzed using *igraph 0.7.1* [7], a comprehensive R software package for complex social network analysis research.

### 3.2.2 Coh-Matrix Analyses

Prior to Coh-Matrix analyses, the logs were cleaned and parsed to facilitate a student level evaluation. Thus, text files were created that included all contributions from a single learner, yielding a total of 1,754 text files, one for each student. All files were then analyzed using Coh-Matrix. Coh-Matrix ([www.cohmetrix.com](http://www.cohmetrix.com)) is a computational linguistics facility that provides measures of over 100 measures of various types of cohesion, including co-reference, referential, causal, spatial, temporal, and structural cohesion [18,26]. Coh-Matrix also has measures of linguistic complexity, characteristics of words, and readability scores. Currently, Coh-Matrix is being used to analyze texts in K-12 for the Common Core standards and states throughout the U.S. More than 50 published studies have demonstrated that Coh-Matrix indices can be used to detect subtle differences in text and discourse [26].

There is a need to reduce the large number of measures provided by Coh-Matrix into a more manageable number of measures. This was achieved in a study that examined 53 Coh-Matrix measures for 37,520 texts in the TASA (Touchstone Applied Science Association) corpus, which represents what typical high school students have read throughout their lifetime [17]. A principal components analysis was conducted on the corpus, yielding eight components that explained an impressive 67.3% of the variability among texts; the top five components explained over 50% of the variance. Importantly, the components aligned with the language-discourse levels previously proposed in multilevel theoretical frameworks of cognition and comprehension [16,23,40]. These theoretical frameworks identify the representations, structures, strategies, and processes at different levels of language and discourse, and thus are ideal for investigating trends in learning-oriented conversations. Below are the five major dimensions, or latent components, that may be useful for understanding trends in learning-oriented, but inherently social, conversations:

- **Narrativity.** The extent to which the text is in the narrative genre, which conveys a story, a procedure, or a sequence of episodes of actions and events with animate beings. Informational texts on unfamiliar topics are at the opposite end of the continuum.
- **Deep Cohesion.** The extent to which the ideas in the text are cohesively connected at a deeper conceptual level that signifies causality or intentionality.
- **Referential Cohesion.** The extent to which explicit words and ideas in the text are connected with each other as the text unfolds.
- **Syntactic Simplicity.** Sentences with few words and simple, familiar syntactic structures. At the opposite pole are structurally embedded sentences that require the reader to hold many words and ideas in working memory.
- **Word Concreteness.** The extent to which content words that are concrete, meaningful, and evoke mental images as opposed to abstract words.

### 3.2.3 Data Preparation

The students' performance, linguistic and network data were merged to facilitate subsequent statistical analyses. Following this, the scores were centered and normalized by removing any outliers. Specifically, the normalization procedure involved Winsorising the data based on each variable's upper and lower percentile. Finally, we were interested in exploring whether the discourse features characterizing learning-related outcomes varied within different learner population, namely across all learners in the MOOC and within a subset of active learners. To enable this analysis, we created two datasets. The *All Learner* dataset contained data for the full 1,754 students that participated in the MOOC. We operationalized active students as those learners who made 4 or more posts in the MOOC. The cut-off point was chosen because the top 25% of learners made 4 or more posts. The resulting *Active Learner* dataset contained the data for those top 471 learners.

### 3.2.4 Statistical analyses

A mixed-effects modeling approach was adopted for all analyses due to the structure of the data (e.g., inter-individual and word count variability) [30]. Mixed-effects models include a combination of fixed and random effects and can be used to assess the influence of the fixed effects on dependent variables after accounting for any extraneous random effects. The primary analyses focused on identifying the association between the discourse features, namely, Narrativity, Deep Cohesion, Referential Cohesion, Syntax Simplicity, and Word Concreteness and the learning outcomes, measured through learners' social centrality and grades. Therefore, we identified two sets of dependent measures in the present analyses: (1) learners' social centrality (Closeness, Degree, and Betweenness) and (2) learners' performance in the course (the final grade). The independent variables in all models were the five discourse features of interest.

Additionally, the influence of language on learning and social capital might vary depending on relevant learner characteristics. For instance, discourse may play a more meaningful role, for student performance and social position in a network, for more active learners than less active learners [25]. This would be in line with Gillani and others [15] conclusion that suggests the social network extracted from the learner interactions "was a noise-corrupted version of the "true" network" (p.2). Thus, we decided to further refine our analysis and create social graph only for those learners who actively participated in discussions (for the cut-off point see Section 4.2). This resulted in an additional four models, labeled as *Active Learners*, exploring the influence of language on learners' social centrality (three models) and performance (one model) for the most active participants in the course.

It is important to note that in addition to constructing the models with the five discourse features as fixed effects, *null models* with the random effects (*learner* and *word count*) but no fixed effects were also constructed. A comparison of the null random effects only model with the fixed-effect models allows us to determine whether discourse predicts social centrality and performance above and beyond the random effects. Akaike Information Criterion (AIC), Log Likelihood (LL) and a likelihood ratio test were used to determine the best fitting and most parsimonious model. In addition, we also estimate effect sizes for each model, using a pseudo  $R^2$  method, as suggested by Nakagawa and Schielzeth [28]. For mixed-effects models,  $R^2$  can be characterized into two varieties: marginal  $R^2$  and conditional  $R^2$ . Marginal  $R^2$  is associated with variance explained by fixed factors, and conditional  $R^2$  is can be interpreted as the variance explained

by the entire model, namely random and fixed factors. Both marginal ( $R^2_m$ ) and conditional ( $R^2_c$ )  $R^2$  convey unique and relevant information regarding the model fit and variance explained, and so we report both here. The lme4 package in R [1] was used to perform all the required computation.

## 4. RESULTS AND DISCUSSION

### 4.1 Discourse and Learning

First, we assessed the relationship between learners discourse patterns and performance in the MOOC. The likelihood ratio tests indicated that both the *All Learner* and *Active Learner* models yielded a significantly better fit than the null model with  $\chi^2(5) = 82.57, p = .001, R^2_m = .05, R^2_c = .93$ , and  $\chi^2(5) = 85.44, p = .001, R^2_m = .21, R^2_c = .95$ , respectively. A number of conclusions can be drawn from this initial model fit evaluation and inspection of  $R^2$  variance. First, the model comparisons imply that the discourse features were able to add a significant improvement in predicting the learners' performance above and beyond individual participant characteristics. Second, for the *All Learner* model, discourse and individual participant features explained about 93% of the predictable variance, with 5% of the variance being accounted for by the discourse features. However, the discourse features alone were able to explain a total of 21% of predictable variance in active learners' performance. The observed difference in variance suggests discourse features are more accurate at predicting active learners' performance than that of learners who are less active in the course. It is important to note that the difference in the explained variance for the *All Learner* and *Active Learner* models is not a result of the students simply being more prolific, because we controlled for number of words. Instead the findings might be reflecting a more substantive difference for the active students' potency of thought integration, complexity and communication style, beyond the observation that they are communicating more, compared to the overall learner population. Table 1 shows the discourse features that were predictive of learning performance for both the *All Learner* and *Active Learner* models. As can be seen from Table 1, all five levels of discourse were predictive of learning performance for the *All Learner* models, and four of the five levels were predictive of learning in the *Active Learner* models. Specifically, learners who engaged in more expository style discourse with referential and deep level cohesive integration, abstract language, and simple syntactic structures performed significantly better in the course.

Narrative discourse expresses events and actions performed by characters that unfold over time, as is typical in everyday oral communication, folktales, drama, and short stories [35]. In contrast to narrative, expository language is decontextualized and generally informs the audience about new concepts, broad truths, and technical material as in the case of academic articles and college textbooks. The genre of a text can be particularly revealing with regard to its difficulty. For example, narrative text is substantially easier to read, comprehend, and recall than informational or expository text [16]. From a constructionist theory [19,20] view, this is because expository discourse frequently presents abstract categories and less familiar information that require learners to have extensive background knowledge about the topics in order to generate the inferences necessary for comprehension [39]. As a reminder, our measure of narrativity/expository is a single continuum, wherein higher numbers indicate narrative style discourse and lower numbers indicate expository style discourse. Thus, the negative findings for Narrativity (Table 1) can be extrapolated to conclude that learners who articulated their responses in a more expository style,

mirroring the informational nature of their class material, extracted enough information about the subject to generate inferential processing. Such interpretation is in line with other research showing knowledgeable students develop more comprehensive representations from material than less knowledgeable students [27], and can inferentially relate the information they derive from text better than readers with less background knowledge.

In line with Kintsch's [23] construction-integration theory, Coh-Metrix distinguishes between multiple types of cohesion which fall under two main forms, namely textbase (i.e. referential cohesion) and situation model cohesion (i.e. deep cohesion). Referential or textbase cohesion is primarily maintained through the bridging devices, i.e. the overlap in words, or semantic references. In this context, the findings for referential cohesion suggest that learners who perform better, construct their messages using more bridging devices

A theory of situation model cohesion has been described by [45] that characterizes it as knowledge elaborations that are product of incorporating information derived from the explicit texts with background world knowledge. Coh-Metrix analyzes the situation model dimension on causation, intentionality, space, and time [26]. With regard to the findings for deep cohesion, this suggests that students who are learning are engaging in deeper integration of topics with their background knowledge, generating more inferences to address any conceptual and structural gaps, and consequentially increasing the probability of comprehension. The results for syntax show that simple syntactic structures were associated with better performance. However, this finding was not significant in the *Active Learner* model.

**Table 1.** Descriptive Statistics and Mixed-Effects Model Coefficients for Predicting Performance with Language

Measure	All Learner Model				Active Learner Model			
	M	SD	$\beta$	SE	M	SD	$\beta$	SE
Narrativity	0.00	1.00	-.20**	.02	-0.23	0.69	-.60**	.07
Deep Cohesion	0.00	1.00	.08**	.02	0.27	0.55	.19*	.08
Referential Cohesion	0.00	1.00	.08**	.02	-0.26	0.64	.35**	.07
Syntax Simplicity	0.00	1.00	.07**	.02	0.36	0.67	.08	.07
Word Concreteness	0.00	1.00	-.13**	.02	-0.25	0.51	-.35**	.09

Note: \*  $p < .05$ ; \*\*  $p < .001$ . Mean (**M**). Standard deviation (**SD**). Fixed effect coefficient ( **$\beta$** ). Standard error (**SE**). All Learner Model  $N=1754$ , Active Learner Model  $N=471$ .

Coh-Metrix measures psychological dimensions of words that influence language complexity. As a reminder, our measure of word concreteness is a single continuum, wherein scores are higher when a higher percentage of the content words are concrete, are meaningful, and evoked mental images – as opposed to being abstract. Thus, the negative findings for word concreteness show learners who engaged using more abstract language performed significantly better in the course. There are interesting interpretations from the view of Petty and Cacioppo's Elaboration Likelihood Model (ELM) [29]. The ELM outlines several factors that affect both the ability and motivation to elaborate on arguments contained in messages. If ability to process is impaired, or motivation to process is low, the elaboration and thought density of the learners' communication

would likely suffer. With the exception of syntax ease, the findings suggest students who adopt central route linguistic characteristics perform significantly better than those who use peripheral linguistic features.

## 4.2 Discourse and Social Centrality

Next, we investigated the relationship between learners' discourse patterns and their position in the social network. The likelihood ratio tests indicated that the *All Learner* models for Closeness, Betweenness and Degree yielded a significantly better fit than the null random effects only models with  $\chi^2(5) = 135.74, p = .001, R^2_m = .07, R^2_c = .93, \chi^2(5) = 25.63, p = .0001, R^2_m = .01, R^2_c = .91,$  and  $\chi^2(5) = 62.19, p = .0001, R^2_m = .02, R^2_c = .94,$  respectively. Similarly, for the *Active Learner* models, the likelihood ratio tests indicated that Closeness, Betweenness and Degree yielded a significantly better fit than the null models with  $\chi^2(5) = 38.39, p = .0001, R^2_m = .08, R^2_c = .94, \chi^2(5) = 45.92, p = .0001, R^2_m = .09, R^2_c = .94,$  and  $\chi^2(5) = 63.78, p = .0001, R^2_m = .12$  and  $R^2_c = .96,$  respectively. Similar to the results for performance, the model comparisons imply that the discourse features were able to add a significant improvement in predicting the learners' social centrality above and beyond participant characteristics. In line with this, across the three *All Learner* models, our features explained about 92% of the predictable variance, with 10% of the variance being accounted for by the linguistic features. However, the discourse features were able to explain a total of 29% of predictable variance in active learners' social centrality. Again, this suggests discourse more accurately predicts active learners' position than less active learners. The details of the *All Learner* and *Active Learner* models are reported in Table 2 and Table 3. Interestingly, the pattern of discourse features associated with learners' social centrality differed from the one observed for students' performance in the MOOC. Instead, learners who garnered central positions in the network engaged in narrative discourse with lower referential cohesion, abstract words and simple syntactic structures. With the exception of word abstractness, this pattern is indicative of informal communication.

Across all learners, higher closeness centrality is characterized by more narrative style discourse with less overlap between words and ideas (i.e. low referential cohesion), simple syntactic structures and abstract words. For active learners, the pattern is similar, with only narrativity and referential cohesion being significant. The conventional interpretation of closeness centrality indicates the efficiency of an individual in passing the information directly onto all other individuals in the social network [12]. Due to the nature of MOOC centralized forums, it can be inferred that shorter distance to all the learners can be obtained, if the individual participates in many various discussion threads. Therefore, individuals who are more active and initiate more topical messages yielding replies from many other learners, or reply to many other discussions, would use language characterized by simpler structures, narrative style, and lower referential cohesion. Similar pattern for higher narrativity and lower referential cohesion has been observed in the discourse of learners with high degree and betweenness centrality in a distributed MOOC – a course where learner interactions take place on social media, rather than on the course platform [22]. Although conventionally betweenness centrality is associated with the brokering of information between sub-groups, this is questionable in the context of an online open centralized discussion forum.

These results suggest that learners who attained a more prominent social centrality position used more conversational style discourse. Most noteworthy is that these results do not mirror the

pattern observed for high performing learners. On the contrary, linguistic profiles of high performing learners are characterized by formal discourse that uses expository style language (i.e. negative relationship with narrativity), and more surface and deep level cohesive integration (i.e. positive relationship with referential and deep cohesion) (Table 1).

**Table 2.** All Learner Mixed-Effects Model Coefficients for Predicting Social Network Centrality with Language

Measure	Closeness		Betweenness		Degree	
	$\beta$	SE	$\beta$	SE	$\beta$	SE
Narrativity	.070*	.03	.03	.03	.07**	.02
Deep Cohesion	.008	.02	.01	.02	-.02	.02
Referential Cohesion	-.15**	.03	-.02	.03	-.06**	.02
Syntax Simplicity	.13**	.03	.09*	.03	.06*	.02
Word Concreteness	-.09**	.03	-.03	.02	-.05*	.02

Note: \*  $p < .05$ ; \*\*  $p < .001$ . Mean (*M*). Standard deviation (*SD*). Fixed effect coefficient ( $\beta$ ). Standard error (*SE*).  $N = 1754$ .

**Table 3.** Active Learner Mixed-Effects Model Coefficients for Predicting Social Network Centrality with Language

Measure	Closeness		Betweenness		Degree	
	$\beta$	SE	$\beta$	SE	$\beta$	SE
Narrativity	.32**	.07	.17*	.07	.21**	.06
Deep Cohesion	-.06	.08	.02	.08	.05	.08
Referential Cohesion	-.33**	.07	.11	.07	.09	.07
Syntax Simplicity	.07	.07	.42**	.07	.47**	.07
Word Concreteness	.14	.09	-.07	.09	-.06	.09

Note: \*  $p < .05$ ; \*\*  $p < .001$ . Mean (*M*). Standard deviation (*SD*). Fixed effect coefficient ( $\beta$ ). Standard error (*SE*).  $N = 471$ .

## 5. GENERAL DISCUSSION

This paper adopted a novel approach, which uses language and discourse as a tool to explore its association with two established measures of learning, namely traditional academic performance and social centrality. Specifically, we explored the extent to which characteristics of discourse diagnostically reveal the performance and social position of learners as they interact in a MOOC. The findings present some methodological, theoretical, and practical implications for the educational data mining and learning analytics communities. First, as a methodological contribution, we have highlighted the rich contextual information that can be gleaned from combing deeper level linguistic analysis and SNA. Particularly, discourse features add a significant improvement in predicting both the performance and social network positioning in MOOC forums.

Secondly, the results pose some important theoretical and practical implications for transferring analytic approaches to scaled environments without careful consideration. The results indicate that learners who performed significantly better engaged in more expository style discourse, with surface and deep level cohesive integration, abstract language, and simple syntactic structures. However, linguistic profiles of the centrally positioned learners differed from the high performers. Learners with a more significant and central position in their communication network engaged using a more narrative style discourse with less overlap between words and ideas, simpler syntactic structures and abstract words. In other words, high performers and those with central

positions in the network are not necessarily the same individuals. The misalignment between the linguistic features associated with improved performance and more centrally located network positions is captured by the discrepant pattern for narrative, referential and deep cohesion. These three discourse features are inversely related with high performance and centrality in networks. This difference has important implications because these linguistic dimensions are strongly associated with comprehension according to construction-integration and constructivist theories.

The study also suggests that in open online environments two established measures of learning: traditional academic performance and social centrality reflect different learning outcomes. Academic performance represents a snapshot of students' mastery of the subject, and is one way of accessing the state of subject comprehension. Positioning in social network represents a snapshot of the participation processes and social learning activities. In this study, we demonstrate that the skills associated with these two learning-related outcomes differ.

It could be speculated that the observed misalignment between linguistic performance and social network position in the analyzed open online course, shows the difference in communication patterns of formal and informal learning environments. Formal learning environments have a clearer start and end, and often require participation related to the subject matter, as embedded in tasks, or course design. In open learning environments, adult learners can opt in and opt out of the learning situations. The issue is further complicated by the discussions being held by the learners on MOOC forums on various topics: from subject matter, to technical troubleshooting, or clarification of administrative issues. Centralized forums of MOOCs are more than a social learning space; they are also a communication space. As a result, learners' high activity on a number of issues during one or two weeks of the course may result in a more central position in the network of learners, but may not necessarily indicate that the learners engaged with the content, or demonstrated the required understanding of the subject at the end of the course.

It is unclear from this study what relationship should be deduced between learning and social centrality measures within in the open online environments. At the minimum, the findings suggest that the social positioning in a network of learners in a MOOC may not be equivalent with measured academic performance. Further research is needed to understanding what analytical approaches, such as SNA, are reflecting in emerging educational environments.

## 6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (REC 0106965, ITR 0325428, HCC 0834847) and (DRK-12-0918409), the Institute of Education Sciences (R305G020018, R305A080589), The Gates Foundation, U.S. Department of Homeland Security (Z934002/UTAA08-063), Natural Sciences and Engineering Research Council of Canada (356029), Social Sciences and Humanities Research Council of Canada (435-2013-1708), and Canada Research Chairs Program. Any opinions, findings, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

## 7. REFERENCES

[1] Bates, D., Maechler, M., Bolker, B., et al. *lme4: Linear mixed-effects models using Eigen and S4*. 2014.

- [2] Biglan, A. The characteristics of subject matter in different academic areas. *Journal of Applied Psychology* 57, (1973), 195–203.
- [3] Cade, W.L., Dowell, N.M., Graesser, A.C., Tausczik, Y.R., and Pennebaker, J.W. Modeling student socioaffective responses to group interactions in a collaborative online chat environment. In J. Stamper, Z. Pardos, M. Mavrikis and B.M. McLaren, eds., *Proceedings of the 7th International Conference on Educational Data Mining*. Springer, Berlin, 2014, 399–400.
- [4] Carolan, B.V. *Social Network Analysis Education: Theory, Methods & Applications*. SAGE Publications, Inc. SAGE Publications, Inc., 2014.
- [5] Cho, H., Gay, G., Davidson, B., and Ingrassia, A. Social networks, communication styles, and learning performance in a CSCL community. *Computers & Education* 49, 2 (2007), 309–329.
- [6] Chung, C.K. and Pennebaker, J.W. Using Computerized Text Analysis to Track Social Processes. In T. Holtgraves, ed., *Oxford Handbooks Online*. Oxford, 2014.
- [7] Csardi, G. and Nepusz, T. The igraph Software Package for Complex Network Research. *InterJournal Complex Systems*, (2006), 1695.
- [8] Dawson, S. Online forum discussion interactions as an indicator of student community. *Australasian Journal of Educational Technology* 22, 4 (2006), 495–510.
- [9] Dawson, S. A study of the relationship between student social networks and sense of community. *Educational Technology & Society* 11, 3 (2008), 224–238.
- [10] D’Mello, S. and Graesser, A.C. Language and Discourse Are Powerful Signals of Student Emotions during Tutoring. *IEEE Transactions on Learning Technologies* 5, 4 (2012), 304–317.
- [11] Dowell, N.M., Cade, W.L., Tausczik, Y.R., Pennebaker, J.W., and Graesser, A.C. What works: Creating adaptive and intelligent systems for collaborative learning support. In S. Trausan-Matu, K.E. Boyer, M. Crosby and K. Panourgia, eds., *Twelfth International Conference on Intelligent Tutoring Systems*. Springer, Berlin, 2014, 124–133.
- [12] Freeman, L.C. Centrality in social networks conceptual clarification. *Social networks* 1, 3 (1979), 215–239.
- [13] Gasevic, D., Kovanovic, V., Joksimovic, S., and Siemens, G. Where is research on massive open online courses headed? A data analysis of the MOOC Research Initiative. *The International Review of Research in Open and Distributed Learning* 15, 5 (2014).
- [14] Gašević, D., Zouaq, A., and Janzen, R. “Choose Your Classmates, Your GPA Is at Stake!”: The Association of Cross-Class Social Ties and Academic Performance. *American Behavioral Scientist*, (2013).
- [15] Gillani, N., Yasseri, T., Eynon, R., and Hjorth, I. Structural limitations of learning in a crowd: communication vulnerability and information diffusion in MOOCs. *Scientific reports* 4, (2014).
- [16] Graesser, A.C. and McNamara, D.S. Computational Analyses of Multilevel Discourse Comprehension. *Topics in Cognitive Science* 3, 2 (2011), 371–398.

- [17] Graesser, A.C., McNamara, D.S., and Kulikowich, J.M. Coh-Metrix: Providing Multilevel Analyses of Text Characteristics. *Educational Researcher* 40, 5 (2011), 223–234.
- [18] Graesser, A.C., McNamara, D.S., Louwerse, M.M., and Cai, Z. Coh-metrix: analysis of text on cohesion and language. *Behavior research methods, instruments, & computers: a journal of the Psychonomic Society, Inc* 36, 2 (2004), 193–202.
- [19] Graesser, A.C., Singer, M., and Trabasso, T. Constructing Inferences during Narrative Text Comprehension. *Psychological Review* 101, 3 (1994), 371–95.
- [20] Graesser, A.C. and Wiemer-Hastings, K. Situation models and concepts in story comprehension. In S.R. Goldman, A.C. Graesser and P. van den Broek, eds., *Narrative comprehension, causality, and coherence*. Mahwah, NJ, 1999, 77–92.
- [21] Jiang, S., Fitzhugh, S.M., and Warschauer, M. Social Positioning and Performance in MOOCs. Proceedings of the Workshops held at Educational Data Mining 2014, co-located with 7th International Conference on Educational Data Mining (EDM 2014), (2014), 14.
- [22] Joksimović, S., Dowell, N.M., Skrypnik, O., et al. How do you connect? Analysis of Social Capital Accumulation in connectivist MOOCs. In *Proceedings from the 5th International Learning Analytics and Knowledge (LAK) Conference*. Poughkeepsie, New York, 2015.
- [23] Kintsch, W. *Comprehension: A Paradigm for Cognition*. Cambridge University Press, Cambridge, U.K., 1998.
- [24] Kovanović, V., Joksimović, S., Gašević, D., and Hatala, M. What is the Source of Social Capital? The Association between Social Network Position and Social Presence in Communities of Inquiry. *Proceedings of the Workshops held at Educational Data Mining 2014, (EDM 2014)*, (2014), 1–8.
- [25] De Laat, M., Lally, V., Lipponen, L., and Simons, R.-J. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for Social Network Analysis. *International Journal of Computer-Supported Collaborative Learning* 2, 1 (2007), 87–103.
- [26] McNamara, D.S., Graesser, A.C., McCarthy, P.M., and Cai, Z. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press., Cambridge, M.A., 2014.
- [27] McNamara, D.S., Kintsch, E., Songer, N.B., and Kintsch, W. Are Good Texts Always Better? Interactions of Text Coherence, Background Knowledge, and Levels of Understanding in Learning from Text. *Cognition and Instruction* 14, 1 (1996), 1–43.
- [28] Nakagawa, S. and Schielzeth, H. A general and simple method for obtaining R<sup>2</sup> from generalized linear mixed-effects models. *Methods in Ecology and Evolution* 4, 2 (2013), 133–142.
- [29] Petty, R.E., Cacioppo, J.T., Strathman, A.J., and Priester, J.R. To Think or Not to Think: Exploring Two Routes to Persuasion. In T.C. Brock and M.C. Green, eds., *Persuasion: Psychological insights and perspectives, 2nd ed.* Sage Publications, Inc, Thousand Oaks, CA, US, 2005, 81–116.
- [30] Pinheiro, J.C. and Bates, D.M. *Mixed-effects models in S and S-Plus*. Springer, 2000.
- [31] Rabbany k., R., Takaffoli, M., and Zañane, O.R. Social Network Analysis and Mining to Support the Assessment of On-line Student Participation. *SIGKDD Explor. Newsl.* 13, 2 (2012), 20–29.
- [32] Rizzuto, T., LeDoux, J., and Hatala, J. It's not just what you know, it's who you know: Testing a model of the relative importance of social networks to academic performance. *Social Psychology of Education* 12, 2 (2009), 175–189.
- [33] Romero, C., López, M.-I., Luna, J.-M., and Ventura, S. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education* 68, (2013), 458–472.
- [34] Russo, T.C. and Koesten, J. Prestige, centrality, and learning: A social network analysis of an online class. *Communication Education* 54, 3 (2005), 254–261.
- [35] Sanford, A.J. and Emmott, C. *Mind, Brain and Narrative*. Cambridge University Press, Cambridge, 2012.
- [36] Scholand, A.J., Tausczik, Y.R., and Pennebaker, J.W. Assessing Group Interaction with Social Language Network Analysis. In S.-K. Chai, J.J. Salerno and P.L. Mabry, eds., *Advances in Social Computing*. Springer Berlin Heidelberg, 2010, 248–255.
- [37] Siemens, G. and Baker, R.S. Learning analytics and educational data mining: towards communication and collaboration. *Proceedings of the 2nd international conference on learning analytics and knowledge*, ACM (2012), 252–254.
- [38] Siemens, G. and Gašević, D. Special Issue on Learning and Knowledge Analytics. *Educ Technol Soc* 15, 3, 1–2.
- [39] Singer, M. and O'Connell, G. Robust inference processes in expository text comprehension. *European Journal of Cognitive Psychology* 15, 4 (2003), 607–631.
- [40] Snow, C.E. *Reading for Understanding: Toward a Research and Development Program in Reading Comprehension*. Rand Corporation, Santa Monica, CA, 2002.
- [41] Walsh, T. and Bowen, W.G. *Unlocking the Gates: How and Why Leading Universities Are Opening Up Access to Their Courses*. Princeton University Press, Princeton, 2011.
- [42] Wasserman, S. and Faust, K. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge; New York, 1994.
- [43] Wen, M., Yang, D., and Rose, C. Linguistic Reflections of Student Engagement in Massive Open Online Courses. In *Proceedings 14th International Conference on Web and Social Media*. AAAI, Ann Arbor, MI, 2014, 525–534.
- [44] Yang, D., Wen, M., Kumar, A., Xing, E., and Rose, C. Towards an Integration of Text and Graph Clustering Methods as a Lens for Studying Social Interaction in MOOCs. *The International Review of Research in Open and Distributed Learning* 15, 5 (2014).
- [45] Zwaan, R.A. and Radvansky, G.A. Situation models in language comprehension and memory. *Psychological Bulletin* 123, 2 (1998), 162–185.

# You are Your Words: Modeling Students' Vocabulary Knowledge with Natural Language Processing Tools

Laura K. Allen  
Tempe, AZ, USA  
Arizona State University  
LauraKAllen@asu.edu

Danielle S. McNamara  
Tempe, AZ, USA  
Arizona State University  
Danielle.McNamara@asu.edu

## ABSTRACT

The current study investigates the degree to which the lexical properties of students' essays can inform stealth assessments of their vocabulary knowledge. In particular, we used indices calculated with the natural language processing tool, TAALES, to predict students' performance on a measure of vocabulary knowledge. To this end, two corpora were collected which contained essays from early college and high school students, respectively. The lexical properties of these essays were then calculated using TAALES. The results of this study indicated that two of the linguistic indices were able to account for 44% of the variance in the college students' vocabulary knowledge scores. Additionally, the significant indices from this first corpus analysis were able to account for a significant portion of the variance in the high school students' vocabulary scores. Overall, these results suggest that natural language processing techniques can inform stealth assessments and help to improve student models within computer-based learning environments.

## Keywords

Intelligent Tutoring Systems, writing, Natural Language Processing, feedback

## 1. INTRODUCTION

Writing is a complex cognitive and social process that is important for both academic and professional success [1]. As contemporary societies grow increasingly reliant on text sources to communicate ideas (e.g., emails, text messages, online reports, blogs), the importance of developing proficiency in this area is more important than ever. Unfortunately, acquiring writing skills is no simple task – as evidenced by the many students who underachieve each year on national and international assessments of writing proficiency [1, 2, 3, 4]. Indeed, this text production process is complex and relies on the development of both lower and higher-level knowledge and skills, ranging from a strong knowledge of vocabulary to the strategies necessary for tying their ideas together [5, 6, 7].

To develop the skills that are required to produce high-quality

texts, students need to be provided with comprehensive instruction that targets their individual strengths and weaknesses. In particular, this instruction should explicitly describe and demonstrate the skills and strategies that will be necessary during each of the phases of the writing process. Additionally, it should offer students opportunities to receive summative and formative feedback on their work, while engaging in deliberate practice. This form of *deliberate* practice is an important factor in students' development of strong writing skills [8, 9], because it can promote self-regulation of the planning, generation, and reviewing processes [9]. Unfortunately, however, deliberate practice inherently relies on individualized writing feedback. This is often difficult for teachers to provide, as they are faced with large class sizes and do not have the time to provide thorough comments on every essay that a student writes.

As a result of these classroom needs, researchers have developed computer-based writing systems that can provide students with feedback on their writing [10]. These systems have been used for both classroom assignments and high-stakes writing assessments to ease the burden of individualized essay scoring [11]. Specifically, *automated essay scoring* (AES) systems evaluate the linguistic properties of students' essays to assign them holistic scores [12, 13]. These systems use a multitude of natural language processing (NLP) and machine learning methodologies to provide these essay scores, and previous research suggests that they are often comparable to human raters [11, 13, 14, 15].

To provide students with greater context for the scores on their essays, AES systems are commonly incorporated into educational learning environments, such as *automated writing evaluation* (AWE) systems [16, 17] and *intelligent tutoring systems* (ITSs) [18]. These systems not only provide students with summative feedback on their essays (i.e., holistic scores), they also provide formative feedback and writing instruction. In order to be successful, these systems must contain algorithms that can provide individualized feedback that is relevant to students' individual skills.

Importantly, these computer-based writing environments rely on linguistic features to assess the *quality* of the individual essays submitted to the systems. Although the scores are generally valid and reliable, the systems rarely consider student-level information (e.g., their knowledge, skills, or affect) when providing feedback based on these scores. This can pose critical problems when developing adaptive components for the systems. As an example, consider two students, Mary and John, who both write essays that receive holistic scores of "3" from an AWE system. While Mary is able to clearly argue her point in the thesis and topic sentences,

her essay is weakened by simplistic language and sentence constructions. John, on the other hand, employs sophisticated vocabulary and eloquent sentences throughout his essay; however, he does a poor job of explaining his position on the argument. In this example, both students received the same score from the system; however, their essays were affected by different student-level strengths and weaknesses. Mary may have suffered from lower vocabulary knowledge and general language skills, whereas John may not have developed adequate planning and organization strategies.

One way to accommodate these individual differences is to develop user models based on students' characteristics, beyond simply their scores on essays. These models can provide more specific instruction and feedback that are tailored to students' strengths and weaknesses. One individual difference that may be particularly important to consider in these student models is *vocabulary knowledge*. Previous studies have shown that vocabulary knowledge plays a major role in the writing process, as it is strongly correlated with the scores assigned to students' essays [5, 19]. In the current paper, we examine the efficacy of NLP techniques to inform stealth assessments of this knowledge. In particular, we examine whether the lexical properties of students' essays can accurately model their scores on a standardized measure of vocabulary knowledge. Ultimately, our aim is to use these measures to provide more individualized tutoring to student users.

### 1.1 Stealth Assessments

In order to provide a more personalized learning experience (e.g., individualized instruction and feedback), computer-based learning environments must rely on repeated assessments of performance as students interact with the system. These measures can provide important information about students' knowledge states and learning trajectories, which can help to increase the adaptivity of these systems. Despite the importance of these assessments, however, they are not particularly conducive to robust student learning. In particular, constantly exposing students to questionnaires and tests can disrupt their learning flow [20] and subsequently harm their performance on later tasks.

As a response to this assessment problem, researchers have placed an emphasis on the development of methods that can accumulate information about student users without persistently disrupting the learning task [20, 21]. In particular, researchers have proposed the development of *stealth assessments*. These assessments are intended to measure students' performance and knowledge without requiring any explicit testing. Typically, these stealth assessments are embedded within the learning task itself and, as a result, are not able to be detected by students [22].

Within the context of computer-based learning environments, these stealth assessments can be informed by a wealth of information that can be easily logged in the system. These data can range from the speed at which someone is typing to the trajectories of their mouse movements. Snow and colleagues (2014), for example, developed stealth assessments of agency within a reading comprehension tutoring system [23]. They found that students who exhibited more systematic patterns of behavior in the system produced higher quality self-explanations compared to students who were more disordered in their choice patterns. They stated that this measure of behavior patterns could serve as a stealth assessment of agency in adaptive learning environments. Overall, stealth assessments can serve as a viable solution to the

assessment problem, as they can be informed by a wide variety of data types to model the characteristics of student users (e.g., their skills, attitudes, etc.) [23, 24].

Importantly, after they have been developed, these stealth assessments can be used to enhance student models. Models of students' performance and attitudes are typically embedded in ITSs as a means to provide more individualized instruction and feedback [25]. In these systems, student users are represented by continuously updating models that are representative of their own knowledge and performance in the system. Thus, once the system has the ability to reliably assess students' particular skill sets, it can adapt in precise ways that can enhance the overall efficacy of the instruction [26].

### 1.2 Natural Language Processing

Natural language processing (NLP) tools provide a means through which researchers can develop stealth assessments of student characteristics [24]. In addition, these tools can help researchers to investigate the relationships between individual differences and the learning process at a more fine-grained size. By calculating indices related to multiple levels of the text (e.g., lexical, syntactic, discourse), researchers can look beyond simple measures of holistic quality (i.e., essay scores) and begin to examine and model the components of the writing process more thoroughly [27]. These models of student performance can then allow researchers and educators to provide students with more effective instruction that specifically targets their individual needs.

Broadly, NLP involves the automated calculation of linguistic text features using a computer program (or programming language) [28]. Thus, the focus of NLP primarily rests on the use of computers to understand, process, and produce natural language text for the purpose of automating certain communicative acts (e.g., providing technical support) or for studying communicative processes (e.g., examining the linguistic properties of readable texts). This technique can serve as a powerful methodological approach for researchers who are interested in examining particular aspects of the writing process [27] or for many other domains in which students produce natural language.

Researchers have employed NLP techniques within a variety of domains and contexts for the purpose of developing a better understanding the learning process [7, 24, 29, 30, 31]. For example, Varner, Jackson and colleagues (2013) used NLP tools to calculate the extent to which students' self-explanations of complex science texts contained cohesive elements [31]. Results from this study indicated that better readers produced more cohesive self-explanations than less skilled readers, indicating that automated indices of cohesion could potentially serve as a proxy for the coherence of students' mental text representations. In another study, Graesser and colleagues (2011) developed multiple components of text readability using NLP tools [29]. These components related to different dimensions of text complexity, such as narrativity, concreteness, and referential cohesion. Through the use of NLP tools, these researchers were able to develop components that provide multidimensional information about texts and the specific properties that influence students' ability to comprehend these texts successfully.

### 1.2.1 NLP and Writing

With regards to the writing process, NLP can serve as a particularly beneficial tool, as it can provide explicit information about students' processes and performance on the learning task. Accordingly, these NLP techniques have been used in previous research on writing, primarily with the goal of modeling human ratings of text quality [14, 30, 32]. In one particular study, Crossley and McNamara (2011) examined the linguistic indices that were significantly related to quality ratings of timed, prompt-based essays. Results of this study revealed that higher quality essays contained more sophisticated language, greater lexical diversity, more complex sentence constructions, and less frequent words. In a similar analysis, Varner and colleagues (2013) investigated differences between the linguistic indices associated with teachers' ratings of essay quality and students' self-assessments of their own essays [30]. This analysis suggested that students were less systematic in their self-assessments than teachers, at least in relation to the linguistic characteristics of the essays. Additionally, students' ratings were related to different linguistic features than the essay ratings of their teachers.

Overall, the results of these (and many other) studies suggest that NLP can serve as a powerful resource with which researchers can model the writing process at a more fine-grained size. In particular, NLP tools can potentially help researchers to develop better models of the individual differences that are important to writing proficiency (e.g., vocabulary knowledge), as well as for any other domain in which students produce natural language.

### 1.3 The Writing Pal

The Writing Pal (W-Pal) is an intelligent tutoring system (ITS) that was designed to provide explicit writing strategy instruction and practice to high school and early college students [18, 33]. Unlike typical AWE systems, W-Pal places a strong emphasis on the instruction of writing strategies, as well as multiple forms of practice (i.e., strategy-specific practice and holistic essay writing practice).

The strategy instruction in W-Pal covers all three phases of the writing process: prewriting, drafting, and revising. Within W-Pal, these strategies are taught in individual instructional modules, which include: *Freewriting* and *Planning* (prewriting); *Introduction Building*, *Body Building*, and *Conclusion Building* (drafting); and *Paraphrasing*, *Cohesion Building*, and *Revising* (revising; see Figure 1 for a screenshot of the main W-Pal interface). Each of these instructional modules contains multiple lesson videos, which are each narrated by an animated pedagogical agent. In these videos, the agent describes and provides examples of specific strategies that are important for writing.

After viewing these lesson videos, students unlock multiple mini-games, which allow them to practice the strategies in isolation before applying them to complete essays. Within the W-Pal system, students can engage with identification mini-games, where they are asked to select the best answer to a particular question, or generative mini-games, where they produce natural language (typed) responses related to the strategy they are practicing.

One of the key features of the W-Pal system is its AWE component (i.e., the essay practice component). This system contains a word processor where students can write essays in response to a number of SAT-style prompts (teachers also have the option of adding in their own prompts to assign to students). Once a student has completed an essay, it is submitted to the W-Pal system. The W-Pal algorithm [14] then calculates a number of linguistic features related to the essay and provides summative and formative feedback to the student (see Figure 2 for a screenshot of the W-Pal feedback screen). The summative feedback in W-Pal is a holistic essay score that ranges from 1 to 6. The formative feedback in W-Pal provides information about strategies that students can employ in order to improve their essays. Once they have read the feedback, students have the option to revise their essays based on the feedback that they were assigned.



Figure 1. Main Interface of the W-Pal System

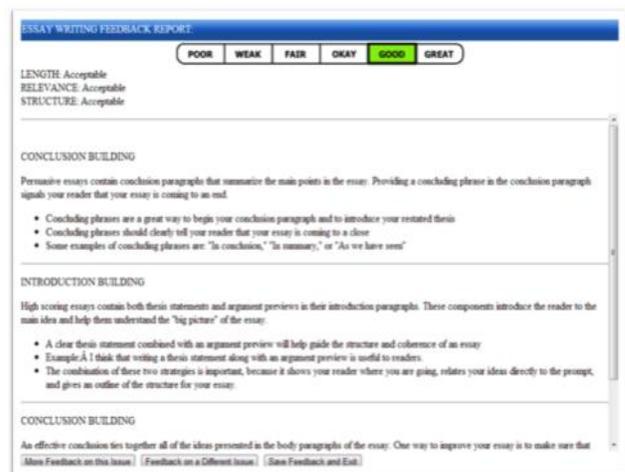


Figure 2. Example of W-Pal Feedback

## 2. CURRENT STUDY

The purpose of the current study is to investigate the degree to which the lexical properties of students' essays can inform stealth assessments of their vocabulary knowledge. Ideally, these assessments will serve to inform student models in the Writing Pal system and contribute to its adaptability in the form of more

sophisticated scoring algorithms, feedback, and adaptive instruction. To this end, two corpora were collected which contained essays from early college and high school students, respectively. The lexical properties of these essays were then calculated using the Tool for the Automatic Analysis of Lexical Sophistication (TAALES) [34]. TAALES is an automated text analysis tool that provides linguistic indices related to the lexical sophistication of texts. We used this tool in the current study so that we could investigate the relationships between students' vocabulary knowledge and the lexical properties of the essays. We hypothesized that these lexical indices would be significantly related to vocabulary knowledge and that they would provide reliable measures of vocabulary knowledge across two distinct student populations.

## 2.1 Primary Corpus

The primary corpus for this study is comprised of 108 essays written by college students from a large university campus in Southwest United States. These students were, on average, 19.75 years of age (range: 18-37 years), with the majority of students reporting a grade level of college freshman or sophomores. Of the 108 students, 52.9% were male, 53.7% were Caucasian, 22.2% were Hispanic, 10.2% were Asian, 3.7% were African-American, and 9.3 % reported other ethnicities. All students wrote a timed (25-minute), prompt-based, persuasive essay that resembled what they would see on an SAT. Students were not allowed to proceed until the entire 25 minutes had elapsed. These essays contained an average of 410.44 words ( $SD = 152.50$ ), ranging from a minimum of 84 words to a maximum of 984 words.

## 2.2 Vocabulary Knowledge Assessment

Students' vocabulary knowledge was assessed using the Gates-MacGinitie (4<sup>th</sup> ed.) reading comprehension test (form S) level 10/12 [35]. This assessment is a 10-minute task, which is comprised of 45 simple sentences that each contains an underlined vocabulary word. Students were asked to read each sentence and then select the most closely related word (from a list of five choices) to the underlined word within the sentence.

## 2.3 Text Analyses

To assess the lexical properties of students' essays, we utilized the Tool for the Automatic Analysis of Lexical Sophistication (TAALES). TAALES is an automated text analysis tool that computes 135 indices that correspond to five primary categories of lexical sophistication: *word frequency*, *range*, *n-gram frequencies*, *academic language*, and *psycholinguistic word information* [34]. These categories are discussed in greater detail below (see 34 for more thorough information).

*Word frequency* indices are indicative of lexical sophistication, because high frequency words are typically learned earlier in life, are processed more quickly, and are indicative of writing quality (i.e., with high frequency words indicating lower quality writing). There are two primary forms of frequency measures: frequency bands and frequency counts. Frequency bands measure the percentage of a text that occurs in particularly frequency bands (e.g., whether they are in the most frequent 1,000 words, 2,000 words in a frequency list, etc.). Frequency counts employ reference corpora and calculate the frequency of the words in a target text within the reference corpus.

*Range* indices are indicative of how widely used a particular word or family of words is. Thus, unlike frequency indices, range

indices do not simply calculate a raw count of a word in a particular list or corpus. Rather, range indices measure the number of individual documents that contain that word in order to determine the extent that it is used broadly. Range has been used to successfully distinguish the frequent verbs produced by L2 speakers of English from the frequent verbs produced by native English speakers [36].

*N-gram frequencies* emphasize units of lexical items rather than single words. In particular, n-grams consist of combinations of *n* number of words (e.g., the bigram "years ago") that frequently occur together. Bigram lists have been shown to be predictive of a speaker or writer's native language, as well as the quality of a given text.

*Academic language* indices measure the degree to which a text contains words that are found infrequently in natural language corpora, but frequently in academic texts. A number of academic word lists have been calculated to measure the words that are commonly used in academic texts, such as textbooks and journal articles. Thus, these indices provide a measure of how academic a text is compared to more typical texts.

*Psycholinguistic word* indices provide information about the specific characteristics of the words used in texts. These properties have been shown to be related to lexical decision times, lexical proficiency, and writing quality. TAALES focuses on five particular properties of words: *concreteness* (i.e., perceptions of how abstract a word is), *familiarity* (i.e., judgments of how familiar words are to adults), *imageability* (i.e., judgments of how easy it is to imagine a word), *meaningfulness* (i.e., judgments of how related a word is to other words), and *age of acquisition* (i.e., judgments of the age at which a word is typically learned).

## 2.4 Statistical Analyses

Statistical analyses were conducted to investigate the role of lexical properties in assessing and modeling students' vocabulary knowledge scores. Pearson correlations were first calculated between students' scores on a vocabulary knowledge measure and the lexical properties of their essays (as assessed by TAALES). The indices that demonstrated a significant correlation with vocabulary knowledge scores ( $p < .05$ ) were retained in the analysis. Multicollinearity of these variables was then assessed among the indices ( $r > .90$ ). When two or more indices demonstrated multicollinearity, the index that correlated most strongly with vocabulary knowledge scores was retained in the analysis. All remaining indices were finally checked to ensure that they were normally distributed.

A stepwise regression analysis was conducted to assess which of the remaining lexical indices were most predictive of vocabulary knowledge. For this regression analysis, a training and test set approach was used (67% for the training set and 33% for the test set) in order to validate the analyses and ensure that the results could be generalized to a new data set. To additionally avoid overfitting the model, we chose a ratio of 15 essays to 1 predictor, which allowed 7 indices to be entered, given that there were 108 essays included in the analysis.

A final linear regression analysis was conducted to determine the extent to which these indices could model the vocabulary knowledge of students in a different population. In particular, we investigated whether the lexical sophistication indices that were retained in the previous regression model (i.e., the regression

model for the college students) accounted for a significant amount of the variance in a second set of students' (i.e., the high school students) vocabulary knowledge.

### 3. RESULTS

#### 3.1 Vocabulary Knowledge Analysis for the Primary Corpus

Pearson correlations were calculated between the TAALES indices and students' Gates-MacGinitie vocabulary knowledge scores to examine the strength of the relationships among these variables. This correlation analysis revealed that there were 45 linguistic measures that demonstrated a significant relation with vocabulary knowledge scores and did not demonstrate multicollinearity with each other. To avoid overfitting the model, we only selected the 7 indices that were most strongly correlated with vocabulary knowledge. These 7 indices are listed in Table 1 (see Kyle & Crossley for explanations of each variable) [34].

A stepwise regression analysis was calculated with these 7 TAALES indices as the predictors of students' vocabulary knowledge scores for the students in the training set. This regression yielded a significant model,  $F(2, 76) = 29.296, p < .001, r = .660, R^2 = .435$ . Two variables were significant predictors in the regression analysis and combined to account for 44% of the variance in students' vocabulary knowledge scores: mean age of acquisition log score [ $\beta = .92, t(2, 76) = 6.423, p < .001$ ] and normed count for all academic word lists [ $\beta = -.36, t(2, 76) = -2.539, p = .013$ ]. The regression model for the training set is presented in Table 2. The test set yielded  $r = .600, R^2 = .360$ , accounting for 36% of the variance in vocabulary knowledge scores.

**Table 1. Correlations between Gates-MacGinitie vocabulary knowledge scores and TAALES linguistic scores**

TAALES variable	<i>r</i>	<i>p</i>
Mean age of acquisition log score	.614	<.001
Mean range (number of documents that a word occurs in) log score	-.562	<.001
Spoken bigram proportion	-.511	<.001
Mean unigram concreteness score	-.492	<.001
Mean frequency score (bigrams)	-.488	<.001
Mean frequency log score	-.476	<.001
Normed count for all academic word lists	.402	<.001

**Table 2. TAALES regression analysis predicting Gates-MacGinitie vocabulary knowledge scores**

Entry	Variable added	$R^2$	$\Delta R^2$
Entry 1	Mean age of acquisition log score	.387	.387
Entry 2	Normed count for all academic word lists	.435	.048

The results of this regression analysis indicate that the students with higher vocabulary scores produced essays that were more lexically sophisticated. The essays contained words that were

acquired at a later age, such as the words *vociferous* or *ubiquitous*, which are predicted to be learned later than words such as *toy* and *animal*. The essays also contained a greater proportion of academic words that are frequently found in academic texts, such as *financier* or *contextualized*, rather than household words such as *bread* and *house*. Hence, better writers use words that are found in academic, written language, rather than more common, mundane language. Notably, these two indices, age of acquisition, and academic words, are likely to correlate with indices related to the frequency or familiarity of words in language. However, in this case, they more successfully captured students' vocabulary knowledge from their writing samples compared to simple frequency or familiarity indices.

#### 3.2 Generalization to a New Data Set

Our second analysis specifically tested the ability of the linguistic indices to predict the Gates-MacGinitie vocabulary knowledge scores of students in a completely separate population. To address this question, we collected a test corpus of essays written by high school students and analyzed the lexical properties of these essays. Specifically, we calculated the *mean age of acquisition log score* and the *normed count for all academic word lists*, as these were the two indices retained in the previous regression model. These indices were then used as predictors in a regression model to predict students' vocabulary knowledge.

#### 3.3 Test Corpus

The test corpus in this paper was collected as part of a larger study ( $n = 86$ ), which compared the complete Writing Pal system to the AWE component of the system. Here, we focus on the pretest essays produced by these participants. All participants were high-school students recruited from an urban environment located in the southwestern United States. These students were, on average, 16.4 years of age, with a mean reported grade level of 10.5. Of the 45 students, 66.7% were female and 31.1% were male. Students self-reported ethnicity breakdown was 62.2% were Hispanic, 13.3% were Asian, 6.7% were Caucasian, 6.7% were African-American, and 11.1% reported other. All students wrote a timed (25-minute), prompt-based, argumentative essay that resembled what they would see on the SAT. Students were not allowed to proceed until the entire 25 minutes had elapsed. These essays contained an average of 340.84 words ( $SD = 124.31$ ), ranging from a minimum of 77 words to a maximum of 724 words. Finally, these students completed the same vocabulary knowledge assessment as the students in the previous corpus.

#### 3.4 Vocabulary Knowledge Analysis for the Test Corpus

The two TAALES indices (i.e., *mean age of acquisition log score* and the *normed count for all academic word lists*) were entered as predictors of students' Gates-MacGinitie vocabulary knowledge scores. This regression yielded a significant model,  $F(2, 83) = 8.521, p < .001, r = .413, R^2 = .170$ . Only one of the variables was a significant predictor in the regression analysis: mean age of acquisition log score [ $\beta = .54, t(2, 83) = 3.666, p < .001$ ]. This model suggests that the regression model generated with the primary corpus partially generalized to a new data set. One of the indices accounted for a significant amount of the variance in students' vocabulary knowledge scores. However, this variance was smaller than the variance accounted for in the primary corpus.

## 4. DISCUSSION

Computer-based writing systems provide students with learning environments in which they can receive writing instruction and engage in deliberate practice [10]. One of the major difficulties that developers of these systems face, however, is the ability to provide instruction and feedback that is *personalized* to individual student users. Developers of these systems often rely on NLP techniques to assess the quality of individual essays; however, it has been relatively unclear whether these NLP techniques can be used to assess relevant individual differences among students.

In the current study, we used NLP techniques to develop stealth assessments of students' vocabulary knowledge. Vocabulary knowledge is an important component of the writing process [5, 19]; thus, our aim was to determine whether we could assess and model individual differences in this knowledge by calculating the lexical sophistication of students' essays. Specifically, an automated text analysis tool was used to analyze the lexical properties of the essays. This tool (TAALES) provided information about the lexical sophistication of the essays at multiple levels (e.g., *word frequency, range, n-gram frequencies, academic language, and psycholinguistic word information*). The results revealed that these indices were able to significantly model students' vocabulary knowledge scores. Additionally, these findings were able to predict students' vocabulary scores on a separate data set.

The TAALES correlation analysis revealed that there were 45 lexical sophistication indices that significantly correlated with students' vocabulary knowledge. This is important, because it indicates that individual differences in students' vocabulary knowledge could be detected by analyzing the lexical items that students used in their essays. Further, the regression analyses revealed that the *psycholinguistic word information* and *academic language* indices provided the most predictive power in the model (as opposed to simple measures of word frequency or familiarity), with indices of age of acquisition and academic words accounting for 44% of the variance in the vocabulary scores. Thus, students with greater vocabulary knowledge tended to produce essays with words that are judged to be acquired later in life and were more academic in nature.

Importantly, the follow-up regression analysis revealed that these two TAALES indices accounted for a significant amount of the variance in vocabulary scores for a separate corpus of student essays. In particular, the age of acquisition variable was able to account for approximately 17% of the variance in students' vocabulary knowledge scores. This finding provides confirmation that the automated lexical sophistication indices could be used across two separate data sets to model vocabulary knowledge.

It is important to note, however, that this variable accounted for a significantly smaller amount of the variance in this test corpus than in our primary corpus. This suggests that individual differences may manifest in the properties of students' essays in different ways depending on the specific context. For instance, in this study, the students who produced essays for the two corpora were in college and high school, respectively. Thus, variations in vocabulary knowledge might have influenced the high school and college students' writing process differentially based on the other knowledge, skills or strategies that they had available to them. The results of this follow-up analysis suggest, therefore, that computer-based learning environments may need to rely on

separate models for students from different populations. Although the same techniques may be able to be used for all student groups (e.g., the use of NLP), the specific indices in the models may need to be modified across different populations.

Overall, the results from the current study suggest that NLP indices can be utilized to develop stealth assessments of students' skills. When taken together, two indices of lexical sophistication accounted for nearly half of the variance in students' vocabulary knowledge scores. These findings are important, because they indicate that students' individual differences can manifest in the ways that they produce essays. Thus, linguistic analyses of essays (and any other natural language input) may provide useful information about individual students' knowledge and skills. Here, we only analyzed students' vocabulary knowledge at pretest (i.e., before they received any training or feedback). In the future, additional studies will be conducted to specifically examine how these stealth assessments of vocabulary knowledge will change throughout training and how they will serve to inform consistently updating student models.

An additional area for future research lies in the assessment of other individual difference variables. In the current study, we solely analyzed the lexical properties of students' essays because we were focusing on one particular individual difference measure: vocabulary knowledge. In future studies, however, it will be important to consider additional linguistic indices that may be related to other specific constructs of interest. For instance, if we aim to model students' attitudes during writing practice, lexical sophistication indices may provide little valuable information. Instead, we may turn to measures of semantic information, such as the tone or themes found in the essays. Similarly, if we are assessing students' reading comprehension skills, it may be more fruitful to include cohesion indices, which describe the degree to which information in a text is explicitly connected.

In conclusion, the current study utilized the NLP tool, TAALES, to investigate the efficacy of NLP techniques to inform stealth assessments of vocabulary knowledge. Eventually, we expect that this stealth assessment will enhance our student models within the W-Pal system and allow us to provide students with more pointed feedback and instruction. More broadly, the current study suggests that NLP techniques can (and should) be used to help researchers and system developers build stealth assessments and student models in computer-based learning environments. These models can ultimately be used to provide more personalized and adaptive computer-based instruction for students.

While a wealth of studies awaits to answer myriad questions on *how* to construct the most powerful models of individual differences without having to administer the tests, this is a strong step forward in demonstrating the feasibility of such stealth measures.

## 5. ACKNOWLEDGEMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080589 to Arizona State University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

## 6. REFERENCES

- [1] National Commission on Writing. 2003. *The Neglected "R."* College Entrance Examination Board, New York.

- [2] Baer, J. D., and McGrath, D. 2007. The reading literacy of U.S. fourth-grade students in an international context: Results from the 2001 and 2006 Progress in International Literacy Study (PIRLS). National Center for Educational Statistics, Institute of Education Sciences, U.S. Department of Education.
- [3] National Assessment of Educational Progress. 2009. The Nation's Report Card: Writing 2009. Retrieved Nov. 5, 2012, nces.ed.gov/nationsreportcard/writing.
- [4] National Assessment of Educational Progress. 2011. The Nation's Report Card: Writing 2011. Retrieved Nov. 5, 2012, nces.ed.gov/nationsreportcard/writing.
- [5] Allen, L. K., Snow, E. L., Crossley, S. A., Jackson, G. T., and McNamara, D. S. 2014. Reading comprehension components and their relation to the writing process. *L'année psychologique/Topics in Cognitive Psychology*, 114, (2014) 663-691.
- [6] Flower, L. and Hayes, J. 1981. Identifying the organization of writing processes. In L. Gregg and E. Steinberg (Eds.), *Cognitive processes in writing*. Erlbaum & Associates, Hillsdale, NJ, 3-30.
- [7] Allen, L. K., Snow, E.L., and McNamara, D. S. 2014. The long and winding road: Investigating the differential writing patterns of high and low skilled writers. In J. Stamper, S. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining* (London, UK, July 4 -7, 2014). Heidelberg, Berlin, Germany: Springer, 304-307.
- [8] Johnstone, K.M., Ashbaugh, H., and Warfield, T.D. 2002. Effects of repeated practice and contextual writing experiences on college students' writing skills. *Journal of Educational Psychology* (2002), 94, 305-315.
- [9] Kellogg, R., and Raulerson, B. 2007. Improving the writing skills of college students. *Psychonomic Bulletin and Review*, 14, (2007), 237-242.
- [10] Allen, L. K., Jacovina, M. E., and McNamara, D. S. in press. Computer-based writing instruction. In C. A. MacArthur, S. Graham, and J. Fitzgerald (Eds.), *Handbook of Writing Research*.
- [11] Dikli, S. 2006. An overview of automated scoring of essays. *Journal of Technology, Learning, and Assessment*, (2006), 5.
- [12] Deane, P. 2013. On the relation between automated essay scoring and modern views of the writing construct. *Assessing Writing*, 18, (2013), 7-24.
- [13] Shermis, M., and Burstein, J. (Eds.). 2003. *Automated essay scoring: A cross-disciplinary perspective*. Erlbaum, Mahwah, NJ.
- [14] McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., and Dai, J. 2015. Hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23, (2015), 35-59.
- [15] Warschauer, M., & Ware, P. 2006. Automated writing evaluation: defining the classroom research agenda. *Language Teaching Research*, 10, (2006), 1-24.
- [16] Attali, Y., and Burstein, J. 2006. Automated essay scoring with e-rater V.2. *Journal of Technology, Learning, and Assessment*, 4, (2006), 3.
- [17] Crossley, S. A., Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Using automated indices of cohesion to evaluate an intelligent tutoring system and an automated writing evaluation system. In K. Yacef et al (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED)*. Springer, Heidelberg, Berlin, 269-278.
- [18] Roscoe, R. D., Varner, L. K., Weston, J. L., Crossley, S. A., and McNamara, D. S. 2014. The Writing Pal intelligent tutoring system: Usability testing and development. *Computers and Composition*, 34 (2014), 39-59.
- [19] Crossley, S. A., Allen, L. K., Snow, E. L., & McNamara, D. S. in press. Pssst...textual Features... there is more to automatic essay scoring than just you! In *Proceedings of the 5th International Learning Analytics and Knowledge Conference (LAK)*, Poughkeepsie, NY.
- [20] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. In S. Tobias & J. D. Fletcher (Eds.), *Computer Games and Instruction*. Information Age Publishers, Charlotte, NC, 503-524.
- [21] Shute, V. J., and Kim, Y. J. 2013. Formative and stealth assessment. In J. M. Spector, M. D. Merrill, J. Elen, and M. J. Bishop (Eds.), *Handbook of Research on Educational Communications and Technology (4th Edition)*. Lawrence Erlbaum Associates, Taylor & Francis Group, New York, NY, 311-323.
- [22] Shute, V. J., Ventura, M., Bauer, M. I., and Zapata-Rivera, D. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In U. Ritterfeld, M. Cody & P. Vorderer (Eds.), *Serious games: Mechanisms and effects*. Routledge, Mahwah, NJ, 295-321.
- [23] Snow, E. L., Jacovina, M. E., Allen, L. K., Dai, J., and McNamara, D. S. 2014. Entropy: A stealth assessment of agency in learning environments. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining*, (London, UK, July 4 -7, 2014), Springer Berlin Heidelberg, 241-244.
- [24] Allen, L. K., Snow, E. L., and McNamara, D. S. in press. Are you reading my mind? Modeling students' reading comprehension skills with natural language processing techniques. In *Proceedings of the 5th International Learning Analytics and Knowledge Conference (LAK)*, Poughkeepsie, NY.
- [25] Brusilovsky, P. 1994. The construction and application of student models in intelligent tutoring systems. *Journal of Computer and Systems Science International*, 23, (1994), 70-89.
- [26] Vanlehn, K. 2006. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16 (2006), 227-265.
- [27] Crossley, S. A., Allen, L. K., Kyle, K., and McNamara, D. S. 2014. Analyzing discourse processing using a simple natural

- language processing tool (SiNLP). *Discourse Processes*, 51, 511-534.
- [28] Crossley, S. A. 2013. Advancing research in second language writing through computational tools and machine learning techniques: A research agenda. *Language Teaching*, 46 (2013), 256-271.
- [29] Graesser, A. C., McNamara, D. S., and Kulikowich, J. M. 2011. Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, 40, (2011), 223-234
- [30] Varner, L. K., Roscoe, R. D., and McNamara, D. S. 2013. Evaluative misalignment of 10<sup>th</sup>-grade student and teacher criteria for essay quality: An automated textual analysis. *Journal of Writing Research*, 5, (2013), 35-59.
- [31] Varner, L. K., Jackson, G. T., Snow, E. L., & McNamara, D. S. (2013). Does size matter? Investigating user input at a larger bandwidth. In C. Boonthum-Denecke & G. M. Youngblood (Eds.), Proceedings of the 26th Annual Florida Artificial Intelligence Research Society (FLAIRS) Conference (pp. 546-549). Menlo Park, CA: The AAAI Press.
- [32] Crossley, S. A., and McNamara, D. S. Text coherence and judgments of essay quality: Models of quality and coherence. In L. Carlson, C. Hoelscher, & T.F. Shipley (Eds.), Proceedings of the 33rd Annual Conference of the Cognitive Science Society. (pp. 1236-1231). Austin, TX: Cognitive Science Society.
- [33] Roscoe, R. D., and McNamara, D. S. 2013. Writing Pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105, (2013), 1010-1025.
- [34] Kyle, K. and Crossley, S. A. in press. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly* (in press).
- [35] MacGinitie, W.H., MacGinitie, R.K., Maria, K., and Dreyer, L.G.: Gates-MacGinitie Reading Test (4th ed.). The Riverside Publishing Company, Itasca, 2000.
- [36] Crossley, S. A., Cobb, T., and McNamara, D. S. 2013. Comparing count-based and band-based indices of word frequency: Implications for active vocabulary research and pedagogical applications. *System*, 41, (2013), 965-981.

# Automatic Identification of Nutritious Contexts for Learning Vocabulary Words

Jack Mostow, Donna Gates, Ross Ellison, Rahul Goutam

Project LISTEN ([www.cs.cmu.edu/~listen](http://www.cs.cmu.edu/~listen)), School of Computer Science, Carnegie Mellon University

RI-NSH 4103, 5000 Forbes Avenue, Pittsburgh, PA 15213-3980, USA

011 (412) 268-1330

[mostow@cmu.edu](mailto:mostow@cmu.edu), [dmg@alumni.cmu.edu](mailto:dmg@alumni.cmu.edu), [rpelliso@andrew.cmu.edu](mailto:rpelliso@andrew.cmu.edu), [rgoutam@cmu.edu](mailto:rgoutam@cmu.edu)

## ABSTRACT

Vocabulary knowledge is crucial to literacy development and academic success. Previous research has shown learning the meaning of a word requires encountering it in diverse informative contexts. In this work, we try to identify “nutritious” contexts for a word – contexts that help students build a rich mental representation of the word’s meaning. Using crowdsourced ratings of vocabulary contexts retrieved from the web, AVER learns models to score unseen contexts for unseen words. We specify the features used in the models, measure their individual informativeness, evaluate AVER’s cross-validated accuracy in scoring contexts for unseen words, and compare its agreement with the human ratings against the humans’ agreement with each other. The automated scores are not good enough to replace human ratings, but should reduce human effort by identifying contexts likely to be worth rating by hand, subject to a tradeoff between the number of contexts inspected by hand, and how many of them a human judge will consider nutritious.

## Keywords

Vocabulary learning, crowdsourcing, automated scoring, regression models.

## 1. INTRODUCTION

Years of research on vocabulary learning have found that vocabulary is a bottleneck to comprehension [1], shown that vocabulary instruction benefits students’ word learning and text comprehension [2-5], and identified several principles of effective vocabulary instruction [6-12]. The principle relevant here is that vocabulary learning requires exposure to diverse informative example contexts in order to develop a rich mental representations of word meanings and their relations to other words.

This paper describes AVER (“Automatic Vocabulary Example Rater”), an attempt to automatically identify “nutritious” contexts – example uses of a word that should help in learning its meaning. (*Aver* is itself a vocabulary word that means *assert*.) This work is part of a larger project that supplied our training and test data in the form of target vocabulary words, example contexts in which they occur, and human ratings of their nutritiousness. The

contexts were retrieved from the web by DictionarySquared.com, an online high school vocabulary tutor that searches the web for a given target word in order to find candidate contexts that contain it. DictionarySquared aims to pick contexts a few dozen words long, preferring to start and end at boundaries between sentences, paragraphs, or HTML blocks.

This paper describes how AVER trains and evaluate models to predict the nutritiousness of such contexts, based on human ratings crowdsourced using Amazon Mechanical Turk.

Ideally AVER would identify a set of examples that maximizes the amount of actual student learning from a given number of contexts, taking into account the diversity of multiple contexts for the same word, and possibly even their relation to example contexts for other target vocabulary words to learn. However, this paper focuses on the initial problem of predicting the suitability of individual contexts, using crowdsourced human estimates instead of students’ subjective ratings of contexts, or objective measures of their actual learning gains.

### 1.1 Relation to Prior Work

Some previous work has addressed the problem of finding suitable example contexts to support vocabulary learning, but differed in one or more respects from the work reported here. REAP [13] selected examples from an already-vetted corpus, based on specified selection criteria such as student interests. VEGEMATIC [14] constructed 9-word contexts centered on a given target vocabulary word by concatenating overlapping 5-grams from the Google *n*-gram corpus, based on heuristic constraints and preferences; only some of them were good enough to use, but hand-vetting them was faster than composing good examples by hand. Follow-on work [15] extended VEGEMATIC to generate contexts for a particular sense of a target word. AVER also seeks to identify example contexts suitable for vocabulary learning, but addresses a different goal than both these projects: instead of applying explicit hand-crafted heuristics, AVER learns to predict crowdsourced ratings by human judges.

The rest of the paper is organized as follows. First we describe our data set. Then we describe the features we used, tried but dropped, or identified but didn’t implement. Next we describe and evaluate how AVER rates contexts. Finally we conclude.

## 2. DATA SET

The data for this work consists of a vocabulary word and a context that contains at least one instance of the vocabulary word and that illustrates usage of the vocabulary word. The overall data set includes 75,844 contexts for 1,000 vocabulary words, comprising 100 words from each of 10 difficulty bands based on their Standardized Frequency Index [16], a measure of log frequency in a text corpus, adjusted by dispersion across multiple domains.

Dr. Margaret G. McKeown, an international expert on vocabulary learning and instruction, rated 93 contexts based on three criteria – the typicality of the usage of the vocabulary word in the context, the degree to which the context constrains the meaning of the vocabulary word, and the comprehensibility of the context for students. Thus the expert provided three ratings of each context, one on each criterion, ranging from 1 (very poor) to 5 (very good). These data helped in developing a rating scale. However, it would have been infeasible to obtain expert ratings of enough contexts to train good models.

Therefore, using Amazon Mechanical Turk, 13,270 contexts were each rated by 10 amateur workers who passed a brief test of their performance on this task: “Based on context, rate how helpful the text is for helping a high school student understand the meaning of the target word. A helpful context is one that reinforces a word’s meaning and is understandable to high school students.” Contexts ranged in length from 18 to 137 words, with median 63.

Raters differed in how many contexts they rated, ranging from several to hundreds. They rated contexts on a 5-point scale:

- 4 = Very Helpful: After reading the context, a student will have a very good idea of what this word means.
- 3 = Somewhat Helpful
- 2 = Neutral: The context neither helps nor hinders a student’s understanding of the word’s meaning.
- 1 = Bad: The context is misleading or too difficult.
- 0 = Otherwise inappropriate for high school students.

We used the mean of their 10 ratings to label our training and testing data. Inter-rater standard deviation averaged 0.81, so standard error averaged 0.27. We labeled the 4107 contexts with mean rating at or above 3 as “good,” and the 9150 contexts with mean rating below 3 as “bad.”

### 3. FEATURES USED

The remaining 62,574 contexts were not rated by humans. To rate their nutritiousness automatically, AVER uses the human-labeled data to train and test regression models to predict the ratings of unseen contexts for unseen words, or to predict the probability that a context is “good,” i.e., its rating is greater than or equal to 3.

To train these models, we extract features of the vocabulary word and context we consider likely to be informative in predicting its human rating. We normalize every feature as a z-score by subtracting the mean value for that feature and dividing by its standard deviation. By translating all feature values onto a common scale, normalization makes their regression coefficients comparable. Normalization does not affect a feature’s correlation with Turker ratings or other features because correlation is invariant under constant addition or multiplication. We assign a z-score of zero to features with undefined values, so that they have no impact on model output.

To describe various types of features, illustrate their values, explain their meaning, and discuss the intuition underlying them, we will use the following example context for the vocabulary word *alleviate*, with mean Turker rating = 3.7, i.e. quite good:

*It is ironic that students are pressured to do well in school in order to continue participating in extracurricular activities, yet these after school activities are just what they need to relieve stress. Sports clubs and*

*even being involved in student government can help alleviate stress. They allow us to get away from school pressure and enjoy ourselves.*

### 3.1 Comprehensibility

Our goal is to help students learn the typical usage of a vocabulary word by providing them with example contexts. If the example contexts are too difficult to understand, they will not be very helpful to students. Thus indicators of comprehensibility are useful features in predicting the rating of a context.

Rarer words are typically harder. The log frequency of *alleviate*, i.e., the log of its unigram count (1,596,620) divided by the total number of tokens (1,024,908,267,229) in the Google *n*-grams corpus, is -13.4 (z-score = -0.090), placing it in the third most common of 10 word bands (z-score = 0.150). This feature of the target word is the same for all its contexts, but helps control for target word frequency in general models to predict context ratings.

The more and longer the words in a context, the harder it is to understand. The example context has 58 words (z-score = -0.235, which on average are 5.1 letters long (z-score = 0.358), not counting spaces or punctuation.

Flesch-Kincaid scores for reading ease and grade level are widely used to assess readability, and we compute them for contexts:

Reading ease =

$$206.835 - 1.015 \times \frac{\text{total words}}{\text{total sentences}} - 84.6 \times \frac{\text{total syllables}}{\text{total words}}$$

Grade level =

$$0.39 \times \frac{\text{total words}}{\text{total sentences}} + 11.8 \times \frac{\text{total syllables}}{\text{total words}} - 15.59$$

A higher reading ease score characterizes text as easier to read and understand. The reading ease score ranges from 0 to 100. The reading ease score for our example context is 47.18, indicating that it is moderately difficult (z-score = -0.015). Flesch-Kincaid scores depend on how syllables, words, and sentences are counted, and hence differ from one implementation to another, but not by much. Thus Microsoft Word reports a reading ease of 48.6 for this paragraph.

A higher grade level score indicates a context that is more difficult to read and understand. The grade level roughly translates to the number of years of education required to understand the context. The grade level score for our example context is 11.48 (z-score = -0.217), compared to 11.2 in Microsoft Word.

Mean human ratings correlated 0.009 with log of target frequency, 0.023 with word band, -0.082 with context length, -0.039 with average word length, 0.043 with reading ease, and -0.030 with grade level.

### 3.2 Local Predictability

AVER extracts local predictability features from a 9-word context centered on the target word (e.g. *student government can help alleviate stress . They allow*). They estimate the probability of the target word given a local context containing the target word. Five of these local contexts are 5 words long, four are 4 words long, three are 3 words long, two are 2 words long, and the target itself can be considered a 1-word context, so there are 15 probabilities. The submitted version of this paper used all 15 of these probabilities as features.

To estimate these probabilities, AVER uses the Google  $n$ -grams tables [16] based on over a trillion words from the web. These tables specify the frequency of every word unigram, bigram, trigram, 4-gram, and 5-gram with at least 40 occurrences. Thus AVER can use them to estimate such conditional probabilities up to a context length of 5 words. For example, it would estimate the conditional probability of *alleviate* given the 5-word local context *government can help \_\_\_\_ stress* as a fraction whose numerator is the frequency of the 5-gram *government can help alleviate stress* and whose denominator is the summed counts of all 5-grams of the form *government can help \* stress*.

AVER log-transforms the probability estimates to reduce their enormous dynamic range, and normalizes the log probabilities as z-scores, which it uses as features to measure local predictability.

If the numerator is zero, AVER smoothes it to 1. The numerator is zero for 88% to 93% of the 5-word contexts, varying by the position of the target word. E.g., *help alleviate stress . They* is not in the 5-gram table. The numerator is zero for 68% to 78% of the 4-word contexts, 33% to 44% of the 3-word contexts, and 8% to 9% of the 2-word contexts.

What if the denominator is zero (e.g. no 5-grams of the form *government can help \* stress* are listed in the 5-gram table)? The denominator is zero for 82% to 86% of our 5-word contexts that contain the target word; the percentage varies by its position in the context. Likewise, the denominator is zero for 47% to 57% of the 4-word contexts, and 33% to 44% of the 3-word contexts.

In the submitted version of this paper, we translated the resulting undefined probability into a z-score of zero, so that it would neither increase nor decrease the output of our predictive models. However, the effect was that some features, especially for 5-grams, were mostly zero in the training data. Could we do better?

Inspired by a reviewer comment, we implemented a new version, called AVER.b (b for “backoff”) based on an idea from statistical language modeling: in the absence of data about a particular  $n$ -gram, back off to successively shorter  $n$ -grams. For instance, if the denominator is zero because no 5-grams of the form *government can help \* stress* are in the 5-gram table, AVER.b looks for 4-grams of the form *government can help \* or can help \* stress*. If AVER.b finds both, it backs off to whichever yields a higher probability for the target word, on the assumption that it is more informative. If it finds neither, it backs off to trigrams, then bigrams, then finally the unigram *alleviate*.

For our example, 5-word contexts of the form *can help \* stress .* are the only ones listed in the 5-gram table, with frequency 109 for *alleviate*, 455 for *reduce*, 329 for *relieve*, and 49 for *with*. The numerator 109 and denominator 942 yield log probability  $-2.16$ .

For the other 4 positions, AVER.b backs off to 4-grams. Its 4-gram table yields non-zero denominators for 4-word contexts of the form *help \* stress .* (4829), *can help \* stress* (6484), and *government can help \** (6765). It yields non-zero numerators for *help alleviate stress .* (330) and *can help alleviate stress* (325) but zero for *government can help alleviate*, which it smoothes to 1, yielding respective log probabilities of  $-2.68$ ,  $-2.99$ , and  $-8.82$ .

AVER.b finds no 4-grams of the form *\* stress . They*, so it backs off to 3-grams, using the count of *alleviate stress .* (2120) as numerator and the number of 3-grams of the form *\* stress .* (1599767) as denominator, yielding log probability  $-6.63$ .

To speed up such computations, we had years earlier indexed each table by various sequences of  $n$ -gram positions designed to

quickly retrieve all rows matching the values specified for any subset of positions. Table 1 lists these indexes, which took weeks of computer time to build because the tables have so many rows.

**Table 1: Indexes constructed for Google  $n$ -grams tables**

Table:	# rows:	Indexed by:
unigram	13,588,391	1, frequency
bigram	314,843,401	12, 21
trigram	977,069,902	123, 312, 23
4-gram	1,313,818,354	1234, 234, 314, 412, 24, 34
5-gram	1,176,470,663	12345, 5432, 3145, 2541, 1523, 432

For instance, to look up the count of the 5-gram *government can help alleviate stress* efficiently, both versions of AVER use the index 12345. This count is the numerator for estimating the probability of *alleviate* at word 4 given a 5-word context. To find all 5-grams of the form *government can help \* stress*, AVER uses the index 1523. If it finds any, it sums their frequencies as the denominator. If not, AVER.b backs off as described above. It then uses the index 1234 to look up the 4-grams *government can help alleviate* and *can help alleviate stress* as well as 4-grams of the form *government can help \**. AVER uses the index 412 to find 4-grams of the form *can help \* stress*.

This method if necessary estimates the conditional probability of *alleviate* given the local bigram context *help \_\_\_\_* as the bigram frequency of *help alleviate* divided by the summed frequency of all bigrams of the form *help \**. However, there are 28,578 bigrams of this form, and it takes non-trivial time to retrieve them in order to compute their summed frequency of 270,480,813. Instead, both versions of AVER would approximate this sum as the unigram frequency of *help*, namely 271,840,666, which it can retrieve quickly from a single row of the Google unigram table. This over-estimate includes all bigrams of the form *help \** that occurred fewer than 40 times in the Google  $n$ -grams corpus and hence do not appear in the Google bigrams table. This approximation is possible only if the blank falls at the start or end of the  $n$ -gram. Thus it can approximate the number of trigrams of the form *can help \** or *\* stress .*, but not *help \* stress*. The approximation was not necessary for 4- or 5-grams because they typically have many fewer rows in the  $n$ -gram table.

A target word can occur at  $n$  different positions in a word window of size  $n$ , with a separate probability for each window size and position within the window, represented as a log probability. Consequently, original AVER’s local predictability features consist of  $1 + 2 + 3 + 4 + 5 = 15$  different log probabilities. For our example context, their respective z-scores are  $-0.090$ ;  $-0.120$ ,  $0.740$ ;  $0.431$ ,  $1.340$ ,  $-6.775$ ;  $0$ ,  $0.972$ ,  $0.909$ ,  $-0.351$ ; and  $0$ ,  $0.603$ ,  $0$ ,  $0$ . The z-scores of zero reflect the sparsity of  $n$ -grams as  $n$  increases.

The relative weights of these 15 z-scores reflect the overall local predictability of the target word *alleviate* in the local context *student government can help alleviate stress . They allow*. AVER sets these weights empirically as part of optimizing the weights for all our features, not just these 15. Correlations of the 15 features with human ratings range from 0.138 for  $\log P(\text{target } w_1 | \_ w_1)$  to  $-0.009$  for  $\log P(\text{target } w_1 w_2 w_3 w_4 | \_ w_1 w_2 w_3 w_4)$ . I.e., before *stress*, *alleviate* is likelier to occur, but before *stress . They allow*, the word *alleviate* is a bit less likely to occur.

In contrast, AVER.b uses just five local predictability features, one for each position in a 5-word context. In our example, their respective z-scores are 0.071, 1.006, 1.157, 0.944, and -0.457. The third value is largest, i.e. *can help* — *stress* . is the 5-word context that most strongly predicts *alleviate*. The five features correlate with mean Turker ratings at 0.055, 0.038, 0.065, 0.042, and 0.062.

To estimate the probability of the target word at word  $i$  given a 5-word window, AVER.b uses  $n$ -grams whose length  $n_i$  varies by the amount of backoff. To reflect the relative specificity of the evidence for each estimated probability, we tried weighting it by

$$n_i / \sum_{i=1}^5 n_i$$

but it made model fit slightly worse, so we decided not to weight by  $n$ -gram length. Perhaps weighting it differently would help.

### 3.3 Topicality

Topicality features measure relatedness of the target vocabulary word to other content words in the context. The intuition behind using such features is that a context containing a typical usage of the target vocabulary word is likely to contain other content words that co-occur frequently with the target vocabulary word or are distributionally similar to it, i.e. tend to co-occur with the same words that the target word co-occurs with. The DISCO tool [17] at [www.linguatools.de](http://www.linguatools.de) measures the co-occurrence of two words within 3 words of each other (“S1”) and their distributional similarity (“S2”) in a specified corpus, such as the British National Corpus (BNC), which contains 119 million tokens and 122,000 unique content words in “samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of British English, both spoken and written, from the late twentieth century” [18]. AVER uses DISCO to compute co-occurrence and distributional similarity between the target vocabulary word and each content word in the context.

To score the overall topicality of a context for the target word, we must aggregate the relatedness scores for the individual context words. Typically only a few of the context words are strongly related to the target word. Consequently, the overall average relatedness of the context dilutes their influence. Instead, AVER averages relatedness over just the most related  $k$  words of the context. In informal tests of different values of  $k$ , the average of the top 5 relatedness scores did best at predicting human ratings.

Thus AVER computes two topicality scores for a context. The co-occurrence z-score for our example context is 5.063. Context words that tend to co-occur with the target vocabulary word ‘*alleviate*’ include ‘*pressure*’ and ‘*stress*’. The distributional similarity z-score for our example context is 1.497. The context word with the highest distributional similarity to ‘*alleviate*’ is ‘*relieve*’. DISCO’s S1 and S2 scores based on BNC correlated with mean human context ratings at 0.060 and 0.025, respectively.

## 4. FEATURES TRIED BUT ABANDONED

We now discuss several features that we experimented with but do not use in AVER, either because they hurt predictive accuracy in informal small experiments, or because they were too complex to compute efficiently.

### 4.1 Topicality Based on Google $N$ -grams

As explained above, AVER computes context topicality using DISCO co-occurrence and similarity scores based on the British National Corpus. These scores suffer from data sparsity in the

case of less-frequent words. In contrast, the Google  $n$ -grams corpus is based on over 10,000 times as much text, namely a trillion words of Web text. Not only is this corpus four orders of magnitude larger than BNC, it is also more relevant to the example contexts because they too consist of Web text.

Although the Google  $n$ -grams corpus is already in the form of  $n$ -grams rather than the text they are based on, its size makes it computationally expensive to compute similarity scores from it, so in previous work we had precomputed and indexed a table of the number of  $n$ -grams containing a given pair of words at a distance of 1, 2, 3, or 4 words, and those  $n$ -grams’ summed frequency. However, this table has 921,643,327 rows. Despite efficient indexing, a target word’s co-occurrences take considerable time to look up – over 30 seconds for *alleviate*. To compute distributional similarity with reasonable speed, we therefore estimated it from the first few hundred rows. Unfortunately, the resulting feature harmed rather than helped model accuracy. To compute more predictive estimates of co-occurrence and distributional similarity based on Google  $n$ -grams, it might help to sample them more judiciously, and to adjust better for differences among target words to make estimates comparable.

### 4.2 Language Model Probability

To quantify the likelihood of a given context occurring in English, we used a language model trained on English text using the NLTK language model package at [www.nltk.org](http://www.nltk.org). The motivation for this feature was to penalize contexts that contain ill-formed or incomplete sentences. We dropped this feature because it did not improve predictive accuracy, but maybe other variants of it might.

### 4.3 Weighted Human Ratings

Apart from different features that we tried out but did not include in the final model, we also investigated methods to improve the accuracy of the labels computed by averaging 10 raters’ ratings of each context. These methods weighted the average based on each rater’s degree of agreement with expert ratings of other contexts. The more closely the rater agreed with the expert on the contexts they both rated, the more accurately we expected the rater to rate contexts that the expert did not rate.

However, most raters did not overlap with the expert in terms of which contexts they rated. We therefore extended the method transitively to rate such raters based on their degree of agreement with raters who had non-zero overlap with the expert, and on how closely those raters agreed with the expert on the contexts they both rated.

We also used the overlapping contexts to train a model to predict a rater’s *expected* degree of agreement with the expert, based on features of the rater such as the total number of contexts he or she had rated. We hoped to use this model to predict agreement with the expert even for raters with zero overlap. However, the expert rated only 93 contexts, so very few raters overlapped with the expert. Even they overlapped too little to accurately estimate the rater’s agreement with the expert. We therefore abandoned the approach of rating raters by their actual or expected agreement with the expert, and using it to weight the individual ratings averaged to rate a given context. Rating raters might be effective given a larger sample of expert ratings, and greater overlap of the raters with the expert.

## 5. FEATURES FOUND BUT NOT USED

Based on expert linguistic analysis of over 200 contexts whose human and automated ratings differed drastically, we identified

some syntactic and semantic features not exploited by the current models, and likely to improve them.

## 5.1 Syntactic Features

Additional syntactic features of a context could be computed by parsing it with the Stanford parser, and extracting them from the parse tree with Tsurgeon and Tregex, using the tools at [nlp.stanford.edu/software/corenlp.shtml](http://nlp.stanford.edu/software/corenlp.shtml) [19]. [commondatastorage.googleapis.com/books/syntactic-ngrams/index.html](http://commondatastorage.googleapis.com/books/syntactic-ngrams/index.html) [20] is a corpus of syntactic  $n$ -grams that provides counts of dependency tree fragments, which could be used to rate the plausibility of the parse and to infer likely dependency relations among context words. If for some reason part-of-speech tagging the context is feasible but parsing it is not, its dependency relations could be inferred from its part-of-speech  $n$ -grams [21].

Informative syntactic features include the direct object of a target verb, e.g. *abdicate* in *Edward abdicated the throne*, and the objects of prepositions following a target word, e.g. *keen* in *They are very keen on education*. Another syntactic feature comes from coordinate constructions, e.g., *it is characterized by inconsistency and vagary*. The coordinated conjuncts are likely to be semantically similar or even synonymous.

It might also be useful to incorporate syntactic information into the current  $n$ -gram features. In particular, disaggregating  $n$ -gram features by the target word's part of speech in the context would exploit systematic statistical differences between parts of speech. For instance, if the target word is a verb, its subject is likely to precede it, and shed semantic light on what sorts of agents can perform the verb. Conversely, if the target word is an adjective, the noun phrase after it illustrates what the adjective can modify.

## 5.2 Semantic Features

Our analysis of misrated contexts found that spuriously low similarity ratings are often caused by lack of co-occurrences due to sparse data for less-frequent words. This deficiency might be addressed by augmenting BNC data with definitions, Wordnet gloss examples, and Google  $n$ -grams, provided the computational issues discussed earlier are satisfactorily addressed. For example, if we use Google  $n$ -gram features only where BNC data is too sparse, they might not pose such computational bottlenecks. Likewise, we could complement DISCO metrics of semantic similarity with features based on WordNet links from a target word to any of its synonyms, antonyms, hypernyms, and hyponyms that occur in the context.

## 6. AUTOMATED RATING OF CONTEXTS

AVER and AVER.b use the features described above in two types of models to rate contexts automatically for a given target word. The linear regression model predicts the mean human rating of a context. The logistic regression model is a binary classifier: it predicts whether a context is "good" (rated 3 or above) or "bad" (below 3).

We could run these models on all 75,844 contexts, but we can evaluate the models only on the 13,270 contexts rated by humans. To estimate the performance of both models on unseen data, we therefore use 5-fold cross-validation: We split the target words randomly into 5 equal subsets so as to partition the contexts into 5 subsets ("folds") with no overlap in target words between folds. For each fold we train both models on the other 4 folds, measure their performance on the held-out fold, and average over the held-

out folds to estimate predictive accuracy on unseen target words – including the 62,574 unrated contexts, assuming they're similar.

To estimate performance fairly on unseen target words, it is essential to avoid overlap in target words between folds. Otherwise even if contexts do not overlap across folds, overlap in target words causes overfitting and inflates estimated performance on unseen data, especially if the training and test sets contain very similar contexts. Our initial results suffered from this problem before we eliminated overlap in target words across folds.

For the original AVER, the correlation between predicted and actual mean human ratings is 0.180 for the linear model and 0.178 for the logistic model. The Area Under Curve (AUC) for the original AVER is 0.600, significantly better than the 0.5 expected from a random baseline.

The linear model predicts mean human ratings, so it optimizes the correlation of predicted to actual ratings. The logistic model classifies contexts as good or bad, so it optimizes the number of misclassified contexts. Consequently correlation is higher for the linear model, whereas AUC is higher for the logistic model.

Unfortunately, AVER.b fared considerably worse. Its predictions correlated with actual ratings at only .093, with AUC only 0.563. Accordingly we focus on the results for the original AVER.

Table 2 shows the original AVER linear model's coefficients for each normalized feature. According to this analysis, the features in **boldface** are reliable at  $p < .05$  (\*), .005 (\*\*), or .0005 (\*\*\*)

**Table 2: Coefficients of linear model for (original) AVER**

Feature	Coefficient
WordBand	-.5691
<b>Flesch-Kincaid Reading Ease</b>	*** <b>.1220</b>
<b>Flesch-Kincaid Grade</b>	*** <b>.0627</b>
<b>Average word length</b>	*** <b>.0520</b>
<b>Unigram logP(t)</b>	* <b>-1.017</b>
<b>Bigram logP(t w1   __ w1)</b>	*** <b>.0621</b>
<b>Bigram logP(w1 t   w1 __)</b>	** <b>.0188</b>
<b>Trigram logP(t w1 w2   __ w1 w2)</b>	*** <b>-.0394</b>
Trigram logP(w1 t w2   w1 __ w2)	.0070
<b>Trigram logP(w1 w2 t   w1 w2 __)</b>	*** <b>-.0053</b>
4gram logP(t w1 w2 w3   __ w1 w2 w3)	.0088
4gram logP(w1 t w2 w3   w1 __ w2 w3)	.0213
4gram logP(w1 w2 t w3   w1 w2 __ w3)	-.0109
<b>4gram logP(w1 w2 w3 t   w1 w2 w3 __)</b>	*** <b>.0398</b>
<b>5gram logP(t w1 w2 w3 w4   __ w1 w2 w3 w4)</b>	* <b>-.0297</b>
5gram logP(w1 t w2 w3 w4   w1 __ w2 w3 w4)	-.0002
5gram logP(w1 w2 t w3 w4   w1 w2 __ w3 w4)	.0193
<b>5gram logP(w1w2w3 t w4   w1 w2 w3 __ w4)</b>	* <b>-.0283</b>
5gram logP(w1 w2 w3 w4 t   w1 w2 w3 w4 __)	.0017
<b>Co-occurrence (DISCO S1)</b>	*** <b>.0340</b>
<b>Distributional Similarity (DISCO S2)</b>	*** <b>.0674</b>
<b>Intercept</b>	*** <b>2.5079</b>

As Table 2 shows, unigram log probability of the target word was by far the strongest predictor of human ratings, and negative:

contexts for rarer words get lower ratings, which may reflect that the less frequently the target word appears in the Google  $n$ -grams corpus, the less likely it is to have good example contexts on the web. As expected, Reading Ease is a positive predictor: readable example contexts are likelier to help students. Surprisingly, the coefficients for word length and grade level are positive even though in isolation they correlate negatively with ratings. Perhaps they reflect positive effects exposed after other predictors account for the negative effects, or are simply artifacts of including correlated predictors in the model. Several  $n$ -gram based metrics of local predictability in the form of conditional probability of the target given the surrounding context are significant, but it is not clear why some are positive and others are negative. Fewer features based on longer  $n$ -grams are significant, presumably due to sparseness in the corpus. Finally, both topicality indicators are significant positive predictors: contexts relevant to a target word are likelier to be nutritious for learning it.

Although AVER.b's results were worse, they are easier to interpret, and differ from the original AVER. Table 3 shows AVER.b linear model's coefficients for each normalized feature. According to this analysis, the features in **boldface** are reliable at  $p < .05$  (\*) or .0005 (\*\*); one feature is suggestive at  $p < .1$  (.).

**Table 3: Coefficients of linear model for AVER.b**

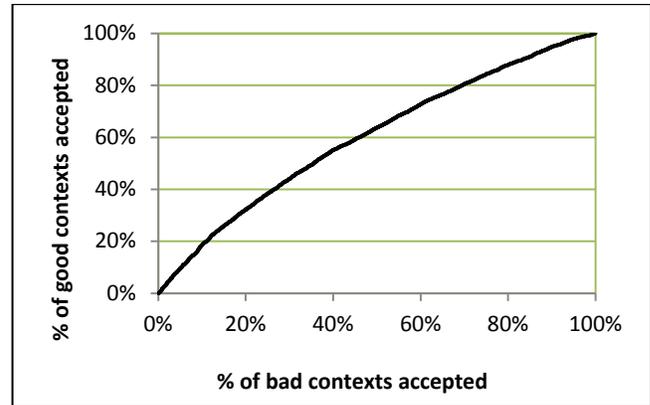
Feature	Coefficient
<b>WordBand</b>	*** <b>0.0508</b>
<b>Flesch-Kincaid Reading Ease</b>	*** <b>0.0567</b>
<b>Flesch-Kincaid Grade</b>	* <b>0.0328</b>
<b>Average word length</b>	* <b>-0.0199</b>
Unigram logP(t)	0.0052
<b>logP(t w1 w2 w3 w4   __ w1 w2 w3 w4)</b>	*** <b>0.0241</b>
logP(w1 t w2 w3 w4   w1 __ w2 w3 w4)	-0.0039
<b>logP(w1 w2 t w3 w4   w1 w2 __ w3 w4)</b>	*** <b>0.0415</b>
logP(w1w2w3 t w4   w1 w2 w3 __ w4)	. -0.0152
<b>logP(w1 w2 w3 w4 t   w1 w2 w3 w4 __)</b>	*** <b>0.0321</b>
<b>Co-occurrence (DISCO S1)</b>	*** <b>0.0483</b>
Distributional Similarity (DISCO S2)	0.0031
<b>Intercept</b>	*** <b>2.5823</b>

For AVER.b, WordBand is significant and Unigram is not, just the opposite of the original AVER. One reason may be that the AVER.b's context probabilities back off to unigram probability for the 8%-9% of 2-word contexts not listed in the bigram table. Reading Ease, Grade, and Word Length are significantly positive in both models. The five context probabilities show a striking pattern: the first, middle, and last positions in a 5-word context are highly predictive, whereas the other two are not. One candidate explanation is that target words tend to be adjacent to function words that provide much less specific information about them. However, the five features have similar correlations with Turker ratings, ranging from 0.038 to 0.065. A simpler explanation is that successive contexts make correlated predictions, and regression assigns the shared variance to just one.

Finally, DISCO S1 was highly significant in both models, but DISCO S2 was significant in the original AVER but not AVER.b. It is not obvious how to explain this difference based on the

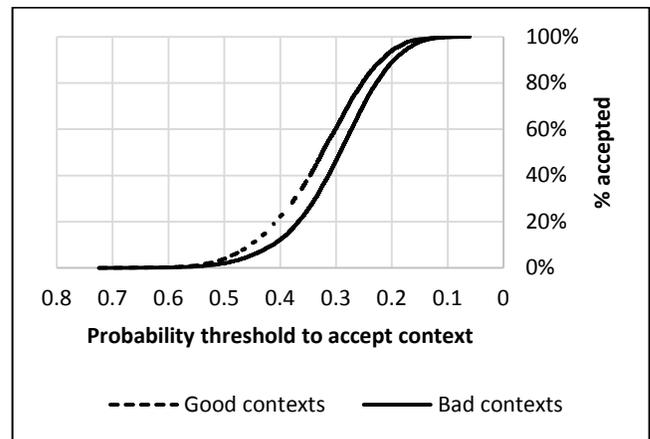
difference in representation of local context features, i.e., how backoff would steal variance from distributional similarity.

To compare the cross-validation results for the original AVER to a random baseline, Figure 1 shows the ROC for the percentage of good contexts (rated 3 or above) accepted against the percentage of bad (rated below 3) contexts accepted, as the acceptance threshold on the logistic model's output probability varies.



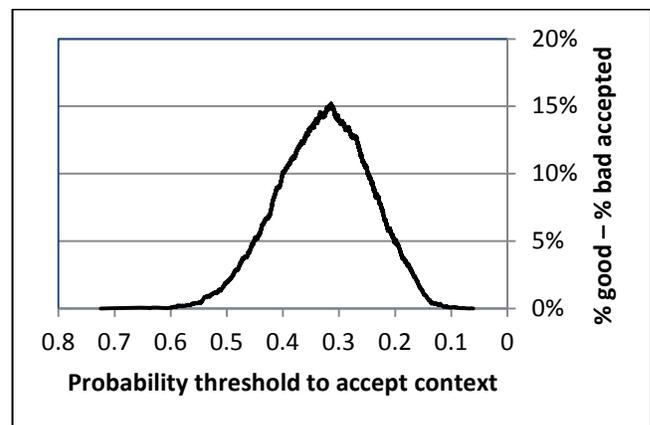
**Figure 1: ROC curve for % good vs. % bad contexts accepted**

Figure 2 plots the percentages of all the good and bad contexts accepted as the probability threshold decreases from 0.8.



**Figure 2: % of contexts accepted vs. probability threshold**

As Figure 3 shows, the difference in percentages peaks at 15.2%:



**Figure 3: % good - % bad vs. probability threshold**

However, bad contexts outnumber good ones, so even when the percentage accepted out of all the good contexts exceeds the percentage accepted out of all the bad contexts, the accepted contexts contains a higher percentage of bad than good contexts, and this imbalance worsens as the threshold decreases, as Figure 4 shows.

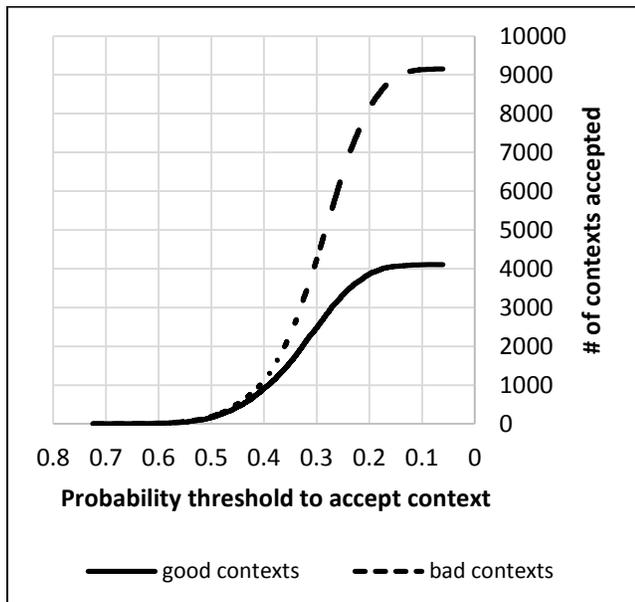


Figure 4: # of contexts accepted vs. probability threshold

As Figure 5 shows, at a threshold of 0.476, the ratio of good to bad contexts reaches a local peak of 0.911 – over twice as high as 0.449, the overall baseline ratio of good contexts to bad contexts. However, at such a high threshold, only 4.4% of the contexts are accepted: 278 (6.8%) of the 4107 good contexts and 305 (3.3%) of the 9150 bad contexts. Thus there is a tradeoff between the number and quality (% good) of the accepted contexts.

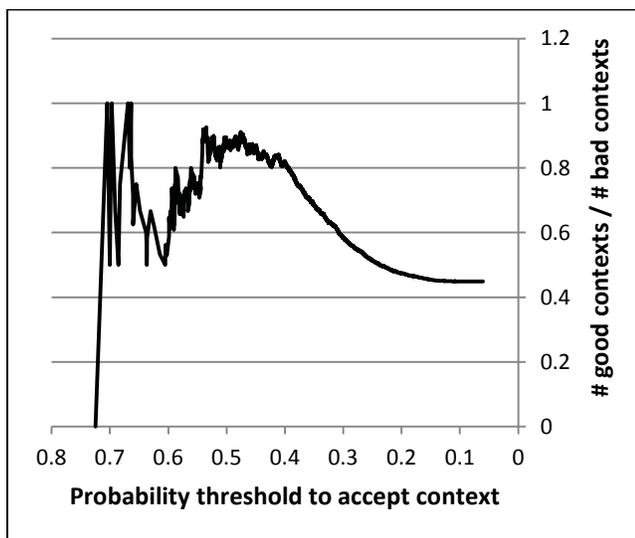


Figure 5: Ratio of good to bad contexts accepted

Visualizing the accuracy of the predicted ratings requires a different type of plot because predicting ratings is not a classification task. Accordingly, Figure 6 shows the distribution of errors in rating good and bad contexts as a histogram of

predicted minus actual ratings, binned to the nearest 0.1. Figure 6 reflects the fact that there are many more bad than good contexts. It shows that almost all the errors in ratings are less than 1 in size.

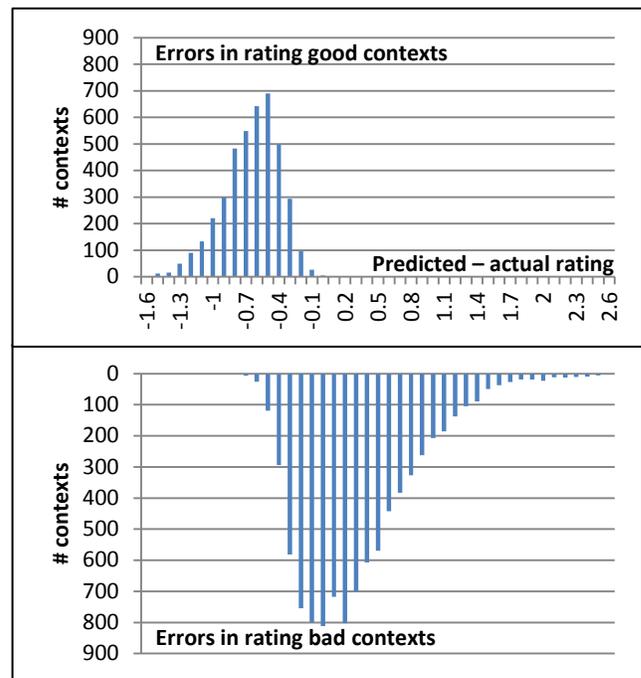


Figure 6: Histogram of errors in rating contexts

## 7. CONCLUSION

This paper presented and evaluated two models for predicting human ratings of example contexts for learning vocabulary. In contrast to prior work that used manually specified, explicitly operationalized criteria to evaluate contexts, both models approximate the implicit criteria underlying human judgments. Given the wide range of phenomena in language, the diversity of criteria that affect the nutritiousness of example contexts, and humans' limited ability to articulate these criteria explicitly and operationalize them precisely, models trained on human ratings have the potential to surpass hand-crafted models, just as machine learning has surpassed hand-crafted classifiers in other domains.

The AVER system reported here is just an initial step toward this goal: it rates contexts reliably more accurately than chance, but not by very much. Its features are shallow, based on local or bag-of-words statistics rather than deeper linguistic structures such as dependency graphs. Future work should develop more sophisticated features. Our analysis of example contexts with large discrepancies between actual and predicted ratings exposed some promising syntactic and semantic features, informed by human understanding of what makes particular contexts useful to learners or not.

Second, supervised learning from labeled data is only as good as the quality of the labels. The larger project of which this work is a part has already revised the training and selection of raters. However, even expert labels are only a proxy for what actually helps real students. Definitive labels should be grounded empirically in data on how much different students learn about different words from different example contexts. To be practical, this approach will require considerable amounts of data – even more so if it tries to model individual differences among students, not just what works well overall on average.

Third, we rated example contexts in isolation, but learning a word's meaning requires encountering it in diverse contexts, not just repeated encounters in the same context, because students learn different aspects from different contexts. Optimizing the entire sequence of encounters will require identifying what those different aspects are, what sorts of contexts help in learning which aspects, and how learning is affected by their order and how they are related.

Besides accelerating the practical task of selecting good example contexts to teach vocabulary, machine-learned models may eventually shed new light on what properties make example contexts nutritious for learning vocabulary, thereby improving our understanding of human vocabulary learning and instruction.

## 8. ACKNOWLEDGMENTS

The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A130467 ("Developing an Online Tutor to Accelerate High School Vocabulary Learning") to University of South Carolina (Suzanne Adlof, PI) and its subcontracts to Carnegie Mellon University (Jack Mostow, PI), and University of Pittsburgh (Charles Perfetti, PI). The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education. We thank DictionarySquared founder Adam Kapelner for the contexts, Margaret McKeown for expert ratings, Suzanne Adlof and Julie Byard for Turker ratings, and the reviewers for helpful comments.

## 9. REFERENCES

- [1] Stanovich, K., R. West, and A.E. Cunningham. Beyond phonological processes: Print exposure and orthographic processing. In S. Brady and D. Shankweiler, Editors, *Phonological Processes in Literacy*. Lawrence Erlbaum Associates: Hillsdale, NJ, 1992.
- [2] Baumann, J.F., E.J. Kame'enui, and G.E. Ash. Research on vocabulary instruction: Voltaire redux. In J. Flood, et al., Editors, *Handbook of research on teaching the English language arts*, 752-785. Erlbaum & Associates: Mahwah NJ, 2003.
- [3] Graves, M.F. Vocabulary learning and instruction. In E.Z. Rothkopf, Editor, *Review of Research in Education*, 91-128 1986.
- [4] Mezynski, K. Issues concerning the acquisition of knowledge: Effects of vocabulary training on reading comprehension. *Review of Educational Research*, 1983. **53**: p. 253-279.
- [5] Stahl, S.A. and M.M. Fairbanks. The effects of vocabulary instruction: A model-based meta-analysis. *Review of Educational Research*, 1986. **56**(1): p. 72-110.
- [6] Graves, M.F. A Vocabulary Program to Complement and Bolster a Middle-Grade Comprehension Program. In B.M. Taylor, M.F. Graves, and P. van den Broek, Editors, *Reading for Meaning: Fostering Comprehension in the Middle Grades. Language and Literacy Series*, 116-135. International Reading Association: Newark, DE, 2000.
- [7] Biemiller, A. and C. Boote. An effective method for building meaning vocabulary in primary grades. *Journal of Educational Psychology*, 2006. **98**(1): p. 44-62.
- [8] Stahl, S.A. and W.E. Nagy. *Teaching Word Meanings*. Literacy Teaching Series. 2006, Mahwah, NJ: Lawrence Erlbaum Associates. ix+220.
- [9] Beck, I.L., M.G. McKeown, and L. Kucan. *Bringing Words to Life: Robust Vocabulary Instruction*. 2002, NY: Guilford.
- [10] Pavlik Jr., P.I. and J.R. Anderson. Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect. *Cognitive Science*, 2005. **29**(4): p. 559-586.
- [11] Aist, G.S. Helping Children Learn Vocabulary During Computer-Assisted Oral Reading. *Educational Technology and Society*, 2002. **5**(2): [http://ifets.ieee.org/periodical/vol\\_2\\_2002/aist.html](http://ifets.ieee.org/periodical/vol_2_2002/aist.html).
- [12] Reinking, D. and S.S. Rickman. The effects of computer-mediated texts on the vocabulary learning and comprehension of intermediate-grade readers. *Journal of Reading Behavior*, 1990. **22**(4).
- [13] Brown, J. and M. Eskenazi. Retrieval of Authentic Documents for Reader-Specific Lexical Practice. *Proceedings of InSTIL/ICALL 2004 Symposium on Computer Assisted Learning*, paper 006. 2004. Venice, Italy.
- [14] Liu, L., J. Mostow, and G.S. Aist. Generating Example Contexts to Help Children Learn Word Meaning. *Journal of Natural Language Engineering*, 2013. **19**(2): p. 187-212.
- [15] Mostow, J. and W. Duan. Generating Example Contexts to Illustrate a Target Word Sense. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, J. Tetreault, J. Burstein, and C. Leacock, Editors. 2011, Association for Computational Linguistics, Stroudsburg, PA: Portland, OR, p. 105-110. At <http://aclweb.org/anthology-new/W/W11/W11-14.pdf>.
- [16] Franz, A. and T. Brants. All Our N-gram are Belong to You. 2006. At <http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html>.
- [17] Kolb, P. Disco: A multilingual database of distributionally similar words. *Proceedings of KONVENS-2008 (Konferenz zur Verarbeitung natürlicher Sprache)* 2008. Berlin.
- [18] BNC Consortium. The British National Corpus, version 3 (BNC XML Edition). 2007, Oxford University Computing Services. At <http://www.natcorp.ox.ac.uk/>.
- [19] Surdeanu, M., J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55-60. 2014. Baltimore, MD.
- [20] Goldberg, Y. and J. Orwant. A dataset of syntactic-ngrams over time from a very large corpus of english books. *Second Joint Conference on Lexical and Computational Semantics (\*SEM)*, 241-247. 2013.
- [21] Jang, H. and J. Mostow. Inferring Selectional Preferences from Part-of-Speech N-grams. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 2012: Avignon, France, p. 377-386.

# Mining a Written Values Affirmation Intervention to Identify the Unique Linguistic Features of Stigmatized Groups

TRAVIS RIDDLE <sup>‡</sup>, SOWMYA SREE BHAGAVATULA<sup>1</sup>, WEIWEI GUO<sup>1</sup>, SMARANDA MURESAN<sup>1</sup>,  
GEOFF COHEN<sup>2</sup>, JONATHAN E. COOK<sup>3</sup>, AND VALERIE PURDIE-VAUGHNS<sup>1</sup>

<sup>1</sup>Columbia University

<sup>2</sup>Stanford University

<sup>3</sup>Pennsylvania State University

## ABSTRACT

Social identity threat refers to the process through which an individual underperforms in some domain due to their concern with confirming a negative stereotype held about their group. Psychological research has identified this as one contributor to the underperformance and underrepresentation of women, Blacks, and Latinos in STEM fields. Over the last decade, a brief writing intervention known as a values affirmation, has been demonstrated to reduce these performance deficits. Presenting a novel dataset of affirmation essays, we address two questions. First, what linguistic features discriminate gender and race? Second, can topic models highlight distinguishing patterns of interest between these groups? Our data suggest that participants who have different identities tend to write about some values (e.g., social groups) in fundamentally different ways. These results hold promise for future investigations addressing the linguistic mechanism responsible for the effectiveness of values affirmation interventions.

## Keywords

Interventions, Natural Language Processing, Achievement Gap

## 1. INTRODUCTION

In the American education system, achievement gaps between Black and White students and between male and female students persist despite recent narrowing. This is true in STEM fields in particular, with the underachievement leading in turn to problems with underemployment and underrepresentation more generally. Women, for example, make up a scant 28% of the STEM workforce [1].

While we acknowledge that the reasons for underachievement

<sup>‡</sup>tar2119@columbia.edu; Corresponding Author

ment and underrepresentation are numerous and complex, *social identity threat* has consistently been shown to be one factor which contributes to these problems and features a psychological basis [32]. Social identity threat refers to the phenomenon in which an individual experiences stress due to concerns about confirming a negative stereotype held about his or her social group. For instance, Black students are stereotyped to be less capable in academic settings than White students. Therefore, a Black student who is aware of this stereotype may feel psychologically threatened, leading to changes in affect, physiology, and behavior [17, 35, 27, 5].

The description of a psychological process that partly accounts for these achievement gaps opens the door to possible psychological interventions. Indeed, a brief, relatively simple intervention derived from self-affirmation theory known as a *values affirmation* has been shown to diminish these achievement gaps - especially when delivered at key transitional moments, such as the beginning of an academic year [6, 4]. The values-affirmation intervention instructs students to choose from a series of values, and then reflect on why this value might be important to them. The intervention draws on self-affirmation theory, which predicts that a fundamental motivation for people is to maintain self-integrity, defined as being a good and capable individual who behaves in accordance with a set of moral values [31].

Accumulating evidence indicates that this intervention is effective in reducing the achievement gap. For instance, students who complete the intervention have shown a blunted stress response [8] and improved academic outcomes longitudinally [4], as well as in the lab [13, 26]. There is also evidence that these affirmations reduce disruptive or aggressive behavior in the classroom [33, 34].

In short, research has definitively shown that values affirmations can reduce achievement gaps. However, the content of the essays themselves has not been as thoroughly examined. While some studies have examined the content of expressive writing for instances of spontaneous affirmations [7], or examined affirmations for instances of certain pre-defined themes (e.g., social belonging [28]), these efforts have been on a relatively small scale, and have been limited by the usual constraints associated with hand-annotating (e.g., experimenter expectations, annotator bias, or excessive time

requirements).

The goal of this paper is to explore the *content of values affirmation essays* using *data mining techniques*. We explore the differences in the content of affirmation essays as a function of ethnic group membership and gender. We are motivated to address these questions because ethnicity and gender, in the context of academic underperformance and the affirmation intervention, are categorical distinctions of particular interest. Identifying as Black or as a woman means that one is likely to contend with negative stereotypes about intelligence, which in turn puts the individual at risk of experiencing the negative effects of social identity threat. The content of the essays produced by individuals under these different circumstances could lead to insights on the structure of threat or the psychological process of affirmation. Additionally, we hope to eventually use information from this initial study to create affirmation prompts which are tailored to individual differences. That is, it may be beneficial to structure the values-affirmation in different ways depending on the particular threatening context or identity of the writer.

We will explore these issues from two different perspectives. First, we investigate the latent topics of essays using Latent Dirichlet Allocation (LDA) [2], which is a generative model that uncovers the thematic structure of a document collection. Using the distribution of topics in each essay, we will present examples of topics which feature strong and theoretically interesting between-group differences. Second, we approach the question of between-group differences in text as a classification problem. For instance, given certain content-based features of the essays (e.g., topics, n-grams, lexicon-based words), how well can we predict whether an essay was produced by a Black or White student? This approach also allows us to examine those features which are the most strongly discriminative between groups of writers. Finally, classification will allow us to closely compare the relative strength of each model's features with respect to differences between groups.

## 2. DATA

Our data come from a series of studies conducted on the effectiveness of values affirmations. For the datasets that have resulted in publications, detailed descriptions of the subjects and procedures can be found in those publications [4, 5, 27, 28]. The unpublished data follow nearly identical procedures with respect to the essay generation.

As an illustrative example of the essay generation process, we describe the methods from Cohen et. al [4]. This study, conducted with seventh-graders, featured a roughly equal number of Black and White students who were randomly assigned to either the affirmation condition or a control condition. The affirmation intervention was administered in the student's classrooms, by teachers who were blind to condition and hypothesis. Near the beginning of the fall semester, students received closed envelopes from their teachers, who presented the work as a regular classroom exercise. Written instructions inside the envelope guided students in the affirmation condition to choose their most important values (or, in study 2, their top two or three most important values) from a list (athletic ability, being good at art, being smart or

getting good grades, creativity, independence, living in the moment, membership in a social group, music, politics, relationships with friends or family, religious values, and sense of humor), while control students were instructed to select their least important value (two or three least important values in study 2). Students in the affirmation condition then wrote about why their selected value(s) are important to them, while students in the control condition wrote about why their selected values might be important to someone else. All students quietly completed the material on their own.

The other samples in our data include both lab and field studies and feature methods largely similar to those just described. Across all studies, participants completing the affirmation essays are compared with students who do not suffer from social identity threat as well as students who complete a control version of the affirmation. Our datasets feature students of college age, as well as middle school students. Below we show two examples of *affirmation essays* (one from a college student and one from a middle school student) and a *control essay* (middle school student):

**Affirmation Essay (college student):** My racial/ethnic group is most important to me when I am placed in situations that are alienating or dangerous or disrespectful. Since coming to Yale a school much larger than my former school where I feel my minority status that much more sharply or feel like people are judging me because I have dark skin I have placed a much higher value on being black. I work for the Af-Am House. I am involved in Black groups and most of my friends are Black. But often being black holds me down and depresses me because people are surprised at how much like them I can be and I dont think Im pretty. Its stressful to have to avoid stereotypes like being late or liking to dance or being sexual. I dont want people to put me in a box labeled black Girl 18. I am my own person.

**Affirmation Essay (middle school student):** Being smart and getting good grades is important to me because it is my path to having a succesful life. Independence is also important because I don't want to be like everybody else. I want to be special in my own way. I want to be different.

**Control Essay:** I think that being good in art can be important to someone else who likes and enjoys art more than I do. I also think this because there are people who can relate and talk about art by drawing and stuff like that but I don't.

In total, we were able to obtain 6,704 essays. Of these, our analyses included all essays which met the following criteria:

1. The essay was an *affirmation* essay (not control). We opted to exclude control essays because the psycholog-

ical process behind the generation of a control essay is fundamentally different from the process that generates an affirmation essay. We are interested in the *affirmation* process, and including control essays in a topic model, for instance, would only add noise to the signal we are interested in exploring.

2. The writing prompt did not deviate (or deviated only slightly) from the writing prompt most widely used across various studies [4]. For example, most of the essays used prompts mentioned above (e.g., athletic ability, religious values, independence). We excluded prompts such as reflection on President Obama’s election, since they are of a different nature.

Including only the essays which met the above criteria resulted in a final dataset of 3,097 essays. Given that some individuals wrote up to 7 essays over the period of their participation, the 3,097 essays came from 1,255 writers (425 Black, 473 White, 41 Asian, 174 Latino, 9 other, 83 unrecorded; 657 females, 556 males, 42 unrecorded). The majority of these writers ( $n = 655$ ) were from a field study in which 8 cohorts of middle school students were followed over the course of their middle school years. The remainder were from several lab-based studies conducted with samples of college students. Before modeling, all essays were preprocessed by removing stop words and words with frequency counts under four. We also tokenized, lemmatized, and automatically corrected spelling using the jazzy spellchecker [11].

The essays varied in length (median number of words = 39, mean = 44.83, SD = 35.85). Some essays are very short (e.g., 2 sentences). As we describe in the next section, this posed some interesting opportunities to test different methods of modeling these essays, especially with regard to using topic models.

### 3. MODELS FOR CONTENT ANALYSIS

To explore the differences in the content of affirmation essays as a function of ethnic group membership and gender we used several methods to model essay content.

*Latent Dirichlet Allocation (LDA)*. Graphical topic models such as LDA [2] have seen wide application in computational linguistics for modeling document content. Such topic models assume that words are distributed according to a mixture of topics and that a document is generated by selecting a topic with some mixture weight, generating a word from the topic’s word distribution, and then repeating the process. LDA specifies a probabilistic procedure by which *essays* can be generated: the writer chooses a topic  $z_n$  at random according to a multinomial distribution ( $\theta$ ), and draws a word  $w_n$  from  $p(w_n|z_n, \beta)$ , which is a multinomial probability conditioned on the topic  $z_n$  ( $\theta \sim Dir(\alpha)$ ). The topic distribution  $\theta$  describes the portion of each topic in a document. One drawback of the current LDA framework is that it assumes equal contribution of each word to the topic distribution of a document  $\theta$ . Since many of our writers tended toward using repetitive language (e.g., miming the essay prompt), we used a modified version of LDA to model our essays, which uses a tf-idf matrix instead of the

My racial/ethnic group is most important to me when I am placed in situations that are alienating or dangerous or disrespectful. Since coming to Yale a school much larger than my former school where I feel my minority status that much more sharply or feel like people are judging me because I have dark skin I have placed a much higher value on being black. I work for the Af-Am House. I am involved in Black groups and most of my friends are Black. But often being black holds me down and depresses me because people are surprised at how much like them I can be and I dont think Im pretty. Its stressful to have to avoid stereotypes like being late or liking to dance or being sexual. I dont want people to put me in a box labeled black Girl 18. I am my own person.

Figure 1: An example essay from a college-aged writer. Words have been highlighted to show their topic assignments

standard word-count matrix [21]. This allows words that are more unique in their usage to take on greater weight in the topic model. We settled on a model with 50 topics, as this provided a good fit to our data, and topics with good subjective interpretability. Given that a primary goal of our analysis was to investigate the topics, we prioritized interpretable topics over statistical fit when necessary. Figure 1 shows the affirmation essay written by the college student given in Section 2, where words are highlighted to show their topic assignments. This example includes three topics, one of which is clearly related to ethnic group (red text), while the other two are somewhat more ambiguous. Section 4 shows some of the learned topics, an analysis of the topic distributions as a function of gender and race, and the results of using the topic distributions as additional features for classification experiments (gender, ethnicity, and gender-ethnicity).

*Weighted Textual Matrix Factorization (WTMF)*. Topic models such as LDA [2] have been successfully applied to relatively lengthy documents such as articles, web documents, and books. However, when modeling short documents (e.g., tweets) other models such as Weighted Textual Matrix Factorization (WTMF) [10] are often more appropriate. Since most of our essays are relatively short (2-3 sentences), we use WTMF as an additional method to model essay content. The intuition behind WTMF is that it is very hard to learn the topic distribution only based on the limited observed words in a short text. Hence Guo and Diab [10] include unobserved words that provide thousands more features for a short text. This produces more robust low dimensional latent vector for documents. However, while WTMF is developed to model latent dimensions (i.e., topics) in a text, a method for investigating the most frequent words of these latent dimensions is not apparent (unlike LDA). We therefore use this content analysis method only for the classification tasks (gender, ethnicity, gender-ethnicity), with the induced 50 dimensional latent vector as 50 additional features in classification (Section 4).

*Linguistic Inquiry and Word Count (LIWC)*. Pennebaker et al.’s LIWC (2007) dictionary has been widely used both in psychology and computational linguistics as a method for content analysis. The LIWC lexicon consists of a set of 64

**Table 1: Top 10 words from select LDA topics**

Topic3	Topic22	Topic33	Topic43	Topic47
relationship	time	group	religion	religious
life	spring	black	church	god
feel	play	white	religious	faith
independent	hang	racial	god	religion
family	talk	identify	treat	jesus
support	help	race	sunday	believe
time	friend	ethnic	believe	belief
friend	family	certain	famous	church
through	homework	culture	stick	christian
help	school	history	lord	earth

word categories grouped into four general classes organized hierarchically: 1) Linguistic Processes (LP) [e.g., Adverbs, Pronouns, Past Tense, Negation]; 2) Psychological Processes (PP) [e.g., Affective Processes [Positive Emotions, Negative Emotions [Anxiety, Anger, Sadness]], Perceptual Processes [See, Hear, Feel], Social Processes, etc]; 3) Personal Concerns (PC) [e.g., Work, Achievement, Leisure]; and 4) Spoken Categories (SC) [Assent, Nonfluencies, Fillers]. LIWC’s dictionary contains around 4,500 words and word stems. In our analysis we used LIWC’s 64 categories as lexicon-based features in the classification experiments (Section 4).

## 4. RESULTS

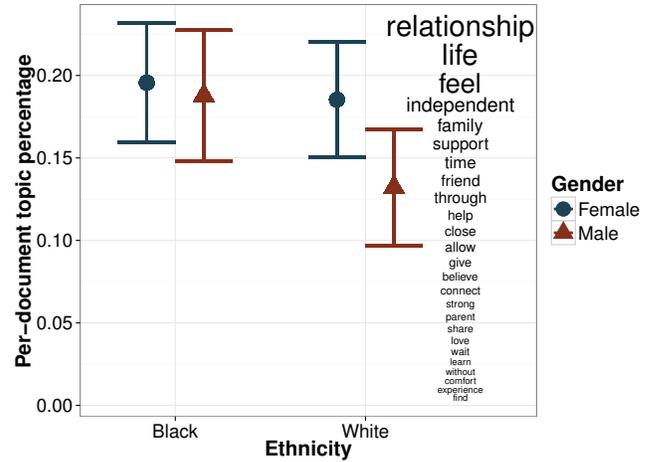
One of our primary questions of interest is whether we can discover between-group differences in the content of the essays. In order to examine this idea in a straightforward way, we limit the analyses to only those individuals who identified as Black or White (2,392 essays from 897 writers). While there are stereotypes suggesting that Asians and Latinos should perform well and poorly in academic domains, respectively, many individuals in our samples who identify with these groups are born in other countries, where the nature of prevailing stereotypes may be different. This is not true to the same extent of individuals who identify as Black or White. We thus exclude Asians and Latinos (as well as those who identified as “other” or declined to answer) for our between-group differences analyses and classification experiments. Inferential analyses were conducted using R [20], and figures were generated using the ggplot2 package [36].

### 4.1 Interpreting Topic Models

We first describe the results of using LDA to see whether we can detect topics that feature strong and theoretically interesting between-group differences. Accurately interpreting the meaning of learned topics is not an easy process [14] and more formal methods are needed to qualitatively evaluate these topics. However, our initial investigation suggests that participants use common writing prompts to write about values in different ways, depending on the group to which they belong.

Table 1 provides the top 10 words from several learned LDA topics<sup>1</sup>. Manually inspecting the topics, we noticed that LDA not only learned topics related to the values given, but it seemed to be able to learn various aspects related to these

<sup>1</sup>As noted in section 3, we are unable to investigate WTMF models in the same fashion.

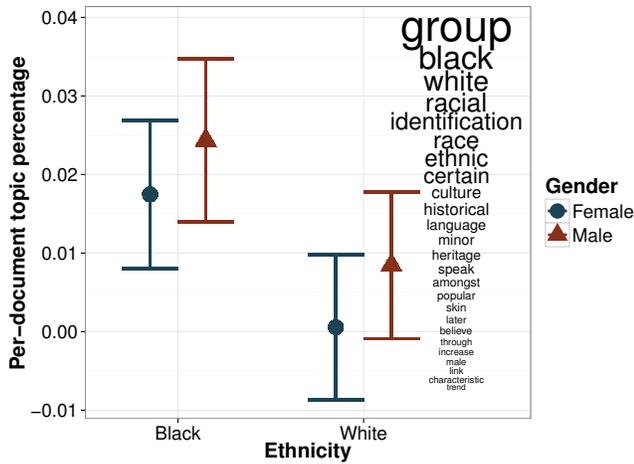


**Figure 2: Topic3: Most prominent topic. Points represent fixed effect estimates. Error bars represent represent +/- 1.96 standard errors. Word size represents weighting in the topic**

values. For example, Topic43 and Topic47 both relate to religious values but Topic43 refers to religion as it pertains to elements of the institution (including words such as church, sunday, and catholic), while Topic47 seems to focus more on the content of faith itself (indicated by words such as faith, jesus, and belief). A similar interpretation can be given to Topic3 and Topic22 — they both refer to relationship with family and friends, but one focuses on the support and help aspect (Topic3), while the other seems to refer to time spent together and hanging out (Topic22). Finally, Topic33 shows an example where the topic learned is about ethnic group, even if ethnicity was not a specific value given as a prompt (rather the more general value of ‘membership in a social group’ was given). Figure 1 shows an example of an essay and the word-topic assignments, where Topic33 is one of the topics (ethnic group, shown in red).

In order to identify interesting between-group differences in topic distributions, we fit a series of mixed-effects linear regressions, with each of the 50 topics as the outcomes of interest. For each model, we estimated effects for gender, ethnicity, and the interaction between the two. For the random effects component, we allowed the intercept to vary by writer. Across the 50 models and excluding the intercept, we estimated a total of 150 effects of interest. Of these, 23 reached the threshold for statistical significance. This proportion is greater than would be expected by chance ( $p < .01$ ). Having established that there are real and meaningful between-groups differences, we more closely examined topics which had theoretically interesting insights.

For example, Figure 2 shows the most frequent words from the most prominent topic (Topic3; relationships with family and friends as basis of support/help) across all essays, along with differences between groups. The model for this topic yielded marginal effects of gender ( $B = .02$ ,  $SE = .01$ ,  $p = .08$ ), with female writers devoting a greater proportion of their writing to the topic ( $M = .12$ ,  $SD = .27$ ) than males ( $M = .09$ ,  $SD = .24$ ). There was also a marginal effect of



**Figure 3: Topic33: effect of ethnicity.** Points represent fixed effect estimates. Error bars represent  $\pm 1.96$  standard errors. Word size represents weighting in the topic

ethnicity, ( $B = .02$ ,  $SE = .01$ ,  $p = .10$ ), with black writers ( $M = .11$ ,  $SD = .26$ ) devoting more of their writing to the topic than white ( $M = .10$ ,  $SD = .25$ ) writers.

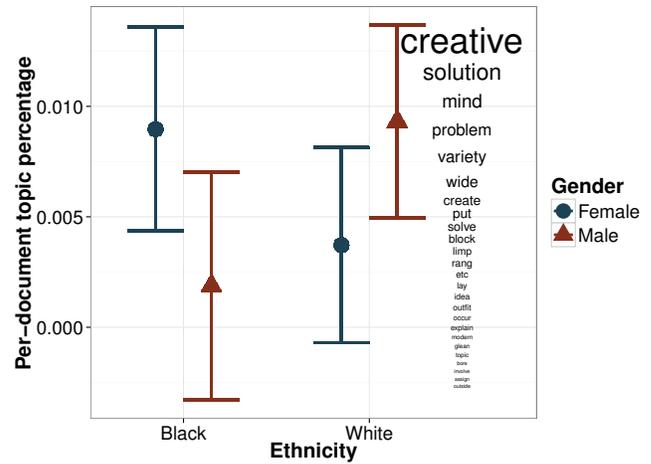
There were also topics which strongly discriminated between ethnicities. Figure 3 presents findings from one such topic (Topic33; ethnic group). The model for this topic revealed the expected main effect of ethnicity ( $B = .008$ ,  $SE = .02$ ,  $p < .01$ ), with black writers devoting a greater proportion of their writing to the topic ( $M = .01$ ,  $SD = .07$ ) than white writers ( $M = .003$ ,  $SD = .03$ ).

The LDA model also estimated topics that were utilized differently by black and white writers, depending on if they happened to be males or females. For instance, Figure 4 presents a topic which is related to problem-solving. Modeling this topic showed that the interaction between gender and ethnicity was significant ( $B = .003$ ,  $SE = .01$ ,  $p < .01$ ). Specifically, for black writers, women wrote more about this topic ( $M = .009$ ,  $SD = .07$ ) than males did ( $M = .001$ ,  $SD = .02$ ,  $p < .05$ ). For white writers, the difference is in the opposite direction, and marginally significant, with males using more of their writing on this topic ( $M = .009$ ,  $SD = .08$ ) than women ( $M = .004$ ,  $SD = .03$ ,  $p = .08$ ). Similarly, the difference for black and white males is statistically significant ( $p < .05$ ), whereas the difference is reversed and marginal for black and white females ( $p = .11$ ).

The findings from the LDA topic modeling show that there are between-group differences emerging from the affirmation essays. To investigate further, in the next section we present the results of a study where we approach the question of between-group differences as a classification problem.

## 4.2 Classification: Gender, Ethnicity, Gender-Ethnicity

Given certain content-based features of the essays (e.g., distribution of topics, LIWC categories, n-grams), these exper-



**Figure 4: Topic23: Interaction between Gender and Ethnicity.** Points represent fixed effect estimates. Error bars represent  $\pm 1.96$  standard errors. Word size represents weighting in the topic

iments aim to classify essays based on the writer's ethnicity and/or gender: Black vs. White (Ethnicity classification), Female vs. Male (Gender classification), and Black-Male vs White-Male and Black-Female vs. White-Female (Ethnicity-Gender classification). In all classification experiments we use a linear Support Vector Machine (SVM) classifier implemented in Weka (LibLINEAR) [9]. We ran 10-fold cross validation and for all results we report weighted F-1 score. As features we used TF-IDF (words weighted by their TF-IDF values)<sup>2</sup>; LDA (topic distributions are used as additional features); WTMF (the 50 dimensional latent vector used as 50 additional features) and LIWC (LIWC's 64 word categories are used as features).

The classification results are displayed in Table 2. We notice that all features give similar performance per classification task. In general, the results were better for the gender classification task (best results 74.09 F1 measure), while the worse results seems to be for the ethnicity classification (best result 66.37 F1). None of the classification tasks showed significant differences as a function of the included features ( $p > .05$ ).

However, the aspect we were more interested in was to analyze the most discriminative features for each classification task with the hope of discovering interesting patterns for between-groups differences. The top 10 discriminating features from each classification type on the TF + LDA + LIWC features are presented in Table 3. There are several interesting observations when analyzing these results. First, supporting the results of the classification experiment, we see that unigrams feature prominently. We also note that LIWC features are largely missing from the top ten, with the only exception being the 10th feature for males in the gender classification. LDA topics, on the other hand, appear as strongly distinguishing in 3 of the 4 classification tasks. Further, in terms of content, the discriminative features sup-

<sup>2</sup>We experimented with presence of n-grams but using TF-IDF gives better results.

**Table 2: SVM Results - cell contents are number of P/R/F1**

Features	Classification			
	Gender	Ethnicity	Bl vs Wh Female	Bl vs Wh Male
TF-IDF	73.38/73.38/73.33	71.34/67.91/65.13	73.43/69.70/67.97	75.26/70.76/67.29
TF-IDF + LDA	73.48/73.46/73.40	<b>70.54/68.41/66.37</b>	73.29/69.62/67.90	74.72/70.85/67.63
TF-IDF + WTMF	73.52/73.46/73.37	71.72/68.00/65.11	<b>73.11/70.02/68.55</b>	74.62/70.59/67.23
TF-IDF+LIWC	74.07/74.0/73.92	72.07/68.08/65.10	73.49/69.78/68.07	75.20/70.85/67.45
TF-IDF+LDA+LIWC	<b>74.09/74.09/74.04</b>	71.38/68.58/66.24	73.49/69.78/68.07	<b>74.98/71.02/67.82</b>

**Table 3: Most discriminative features from classifiers with TF-IDF+LDA+LIWC as features**

Gender		Ethnicity	
Female	Male	Black	White
softball	very	race	Topic15-relationship, creative
jump	available	result	Topic25-music, play, enjoy
swim	football	heaven	younger
happier	Topic26-play, soccer	barely	less
horse	score	disappoint	weird
cheerleader	language	romantic	Topic17-humor, sense, laugh
doctor	lazy	NBA	larger
Topic14-music, relax	moreover	outdoor	rock
boyfriend	baseball	africa	tease
reason	LIWC27-affect	double (game double dutch)	heavy
Females		Males	
Black	White	Black	White
double (game double dutch)	decorate	Topic22-spring, hangout	Topic25-music, play, enjoy
above	rock	NBA	Topic17-humor, sense, laugh
ill	guitar	race	Topic2-reply, already, told
race	peer	head	larger
thick	horse	motive	sit
south	handle	health	cheer
option	grandparents	apart	rock
lord	saxophone	phone	skate
result	crowd	award	handy
york	less	famous	holiday

port some of the results from the topic model analysis. For instance, topic 33 (ethnic group) is the most discriminative, non-unigram feature for ethnicity, and is the 56th most strongly associated feature with Black writers overall. It is also the most discriminative, non-unigram feature for the female-ethnicity classification, as the 44th most strongly associated feature with Black female writers. However, this topic does not show up for the Black vs White male classification. The topic results (Figure 3) also indicate a somewhat stronger relationship for Black vs. White Females.

We also notice that there are strong effects related to sports. In particular, some of the most discriminative features are consistent with social expectations regarding participation in various types of sports. Females, for instance, are more likely to write about softball, swimming, and jumping rope, whereas males are more likely to write about football and baseball. Similar differences can be seen for ethnicity (NBA, double dutch), and gender-ethnicity classifications (females: double dutch, horse; males: NBA, skate).

## 5. RELATED WORK

As mentioned in the introduction, there have been some smaller-scale investigations into the content of affirmation

essays. For instance, Shnabel et al.[28] hand-annotated a subset of the data presented here for presence of social belonging themes. They defined social belonging as writing about an activity done with others, feeling like part of a group because of a shared value or activity, or any other reference to social affiliation or acceptance. Their results indicate that the affirmation essays were more likely to contain such themes than control essays, and that Black students who wrote about belonging themes in their affirmation essays had improved GPAs relative to those who did not write about social belonging. A subsequent lab experiment confirmed this basic effect and strengthened the hypothesized causal claim. The data here are consistent with the idea that social themes are a dominant topic in these essays. Indeed, the most prominent topic (Topic3) seems to be a topic that directly corresponds to social support (see Table 1). Further, even a cursory glance at the topics we have included here will show that references to other people feature prominently - a pattern that is also true for the topics we have not discussed in this paper.

One other finding of interest concerns the discriminative ability of LIWC. Only for the gender classification did LIWC categories appear among the discriminative features. There

are many studies that show gender differences in LIWC categories [25, 19, 24, 16], to say nothing of the broader literature on differences in language use between men and women [15, 12]. However, there is far less consistent evidence for differences in LIWC categories as a function of ethnicity [18]. That our results indicate features from LDA are more discriminative for ethnicity suggests the utility of a bottom-up approach for distinguishing between these groups. However, it should be noted that, in general, classification performance on ethnicity was not as good as classification on gender.

Finally, we also note that this is one of a small, but growing number of studies directly contrasting LIWC and LDA as text modeling tools [30, 22, 25]. While this other work tends to find that LDA provides additional information which results in improvements to classification performance in comparison to LIWC, our do not display this pattern. It is not clear why this may be, although we suspect that frequent misspellings present in our data could lead to some of the discrepancy.

## 6. CONCLUSIONS

We used data mining techniques to explore the content of a written intervention known as a *values affirmation*. In particular, we applied LDA to examine latent topics that appeared in students' essays, and how these topics differed as a function of whether the group to which the student belonged (i.e., gender, ethnicity) was subject to social identity threat. We also investigated between-groups differences in a series of classification studies. Our results indicate that there are indeed differences in what different groups choose to write about. This is apparent from the differences in topic distributions, as well as the classifier experiments where we analyzed discriminative features for gender, ethnicity and gender-ethnicity.

Why might individuals coping with social identity threat write about different topics than those who are not? Some literature shows that racial and gender identity can be seen as a positive for groups contending with stigma [29]. The model of optimal distinctiveness actually suggests that a certain degree of uniqueness leads to positive outcomes [3]. This suggests that if an individual from a stigmatized group perceives their identity to be unique, it may be a source of pride. In the current context, this could be reflected in an increase of writing devoted to the unique social group students are a part of (i.e., African American). On the other hand, there is some evidence that individuals downplay or conceal identities they perceive to be devalued by others [23]. This work would suggest that students in our data would choose to write about what they have in common with others. Our work here seems to provide some support for the former, but we have not addressed these questions directly, and so cannot make any strong claims.

Looking forward, we intend to investigate the relationship between essay content and academic outcomes. Do stigmatized students who write about their stigmatized group experience more benefit from the affirmation, as would be suggested by the optimal distinctiveness model? This work could provide data that speak to this issue. Furthermore, we hope to model the trajectory of how the writing of an indi-

vidual changes over time, especially as a function of whether they completed the affirmation or control essays. Given that values affirmations have been shown to have long-term effects, and our data include some individuals who completed multiple essays, exploration of longitudinal questions about the affirmation are especially intriguing. We also intend to model the essays using supervised-LDA, which would allow us to jointly model the topics with the grouping information. Last but not least we plan to investigate whether there are differences between the middle school students and the college-level students.

## 7. ACKNOWLEDGMENTS

We would like to thank Robert Backer and David Watkins for assistance with this project. This work was supported in part by the NSF under grant DRL-1420446 and by Columbia University's Data Science Institute through a Research Opportunities and Approaches to Data Science (ROADS) grant.

## 8. REFERENCES

- [1] Women, minorities, and persons with disabilities in science and engineering. Technical Report NSF 13-304, National Science Foundation, National Center for Science and Engineering Statistics, Arlington, VA., 2013.
- [2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [3] M. B. Brewer. The social self: On being the same and different at the same time. *Personality and Social Psychology Bulletin*, 17(5):475–482, 1991.
- [4] G. L. Cohen, J. Garcia, N. Apfel, and A. Master. Reducing the racial achievement gap: A social-psychological intervention. *Science*, 313(5791):1307–1310, 2006.
- [5] G. L. Cohen, J. Garcia, V. Purdie-Vaughns, N. Apfel, and P. Brzustoski. Recursive processes in self-affirmation: Intervening to close the minority achievement gap. *Science*, 324(5925):400–403, 2009.
- [6] J. E. Cook, V. Purdie-Vaughns, J. Garcia, and G. L. Cohen. Chronic threat and contingent belonging: Protective benefits of values affirmation on identity development. *Journal of Personality and Social Psychology*, 102(3):479, 2012.
- [7] J. D. Creswell, S. Lam, A. L. Stanton, S. E. Taylor, J. E. Bower, and D. K. Sherman. Does self-affirmation, cognitive processing, or discovery of meaning explain cancer-related health benefits of expressive writing? *Personality and Social Psychology Bulletin*, 33(2):238–250, 2007.
- [8] J. D. Creswell, W. T. Welch, S. E. Taylor, D. K. Sherman, T. L. Gruenewald, and T. Mann. Affirmation of personal values buffers neuroendocrine and psychological stress responses. *Psychological Science*, 16(11):846–851, 2005.
- [9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear - a library for large linear classification, 2008. The Weka classifier works with version 1.33 of LIBLINEAR.
- [10] W. Guo and M. Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational*

- Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics, 2012.
- [11] M. Idzelis. Jazzy: The java open source spell checker, 2005.
- [12] R. T. Lakoff. *Language and woman's place: Text and commentaries*, volume 3. Oxford University Press, 2004.
- [13] A. Martens, M. Johns, J. Greenberg, and J. Schimel. Combating stereotype threat: The effect of self-affirmation on women's intellectual performance. *Journal of Experimental Social Psychology*, 42(2):236–243, 2006.
- [14] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 490–499. ACM, 2007.
- [15] A. Mulac, J. J. Bradac, and P. Gibbons. Empirical support for the gender-as-culture hypothesis. *Human Communication Research*, 27(1):121–152, 2001.
- [16] M. L. Newman, C. J. Groom, L. D. Handelman, and J. W. Pennebaker. Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45(3):211–236, 2008.
- [17] H.-H. D. Nguyen and A. M. Ryan. Does stereotype threat affect test performance of minorities and women? A meta-analysis of experimental evidence. *Journal of Applied Psychology*, 93(6):1314, 2008.
- [18] M. Pasupathi, R. M. Henry, and L. L. Carstensen. Age and ethnicity differences in storytelling to young children: Emotionality, relationality and socialization. *Psychology and Aging*, 17(4):610, 2002.
- [19] J. W. Pennebaker and L. A. King. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6):1296, 1999.
- [20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
- [21] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. *ICWSM*, 5(4):130–137, 2010.
- [22] P. Resnik, A. Garron, and R. Resnik. Using topic modeling to improve prediction of neuroticism and depression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1348–1353. Association for Computational Linguistics, 2013.
- [23] L. M. Roberts. Changing faces: Professional image construction in diverse organizational settings. *Academy of Management Review*, 30(4):685–711, 2005.
- [24] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.
- [25] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.
- [26] J. R. Shapiro, A. M. Williams, and M. Hambarchyan. Are all interventions created equal? A multi-threat approach to tailoring stereotype threat interventions. *Journal of Personality and Social Psychology*, 104(2):277, 2013.
- [27] D. K. Sherman, K. A. Hartson, K. R. Binning, V. Purdie-Vaughns, J. Garcia, S. Taborsky-Barba, S. Tomassetti, A. D. Nussbaum, and G. L. Cohen. Deflecting the trajectory and changing the narrative: How self-affirmation affects academic performance and motivation under identity threat. *Journal of Personality and Social Psychology*, 104(4):591, 2013.
- [28] N. Shnabel, V. Purdie-Vaughns, J. E. Cook, J. Garcia, and G. L. Cohen. Demystifying values-affirmation interventions writing about social belonging is a key to buffering against identity threat. *Personality and Social Psychology Bulletin*, 39(5):663–676, 2013.
- [29] T. B. Smith and L. Silva. Ethnic identity and personal well-being of people of color: a meta-analysis. *Journal of Counseling Psychology*, 58(1):42, 2011.
- [30] A. Stark, I. Shafran, and J. Kaye. Hello, who is calling?: Can words reveal the social nature of conversations? In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 112–119. Association for Computational Linguistics, 2012.
- [31] C. M. Steele. The psychology of self-affirmation: Sustaining the integrity of the self. *Advances in Experimental Social Psychology*, 21:261–302, 1988.
- [32] C. M. Steele, S. J. Spencer, and J. Aronson. Contending with group image: The psychology of stereotype and social identity threat. *Advances in Experimental Social Psychology*, 34:379–440, 2002.
- [33] S. Thomaes, B. J. Bushman, B. O. de Castro, G. L. Cohen, and J. J. Denissen. Reducing narcissistic aggression by buttressing self-esteem: An experimental field study. *Psychological Science*, 20(12):1536–1542, 2009.
- [34] S. Thomaes, B. J. Bushman, B. O. de Castro, and A. Reijntjes. Arousing "gentle passions" in young adolescents: Sustained experimental effects of value affirmations on prosocial feelings and behaviors. *Developmental Psychology*, 48(1):103, 2012.
- [35] G. M. Walton and G. L. Cohen. Stereotype lift. *Journal of Experimental Social Psychology*, 39(5):456–467, 2003.
- [36] H. Wickham. *ggplot2: Elegant graphics for data analysis*. Springer New York, 2009.

# Automatic Classification of Question & Answer Discourse Segments from Teacher's Speech in Classrooms

Nathaniel Blanchard  
University of Notre Dame  
384 Fitzpatrick Hall  
Notre Dame, IN 46556, USA  
nblancha@nd.edu

Sidney D'Mello  
University of Notre Dame  
384 Fitzpatrick Hall  
Notre Dame, IN 46556, USA  
sdmello@nd.edu

Andrew M. Olney  
University of Memphis  
365 Innovation Drive  
Memphis, TN 38152, USA  
aolney@memphis.edu

Martin Nystrand  
University of Wisconsin-Madison  
685 Education Sciences  
Madison WI, 53706-1475  
mnystrand@ssc.wisc.edu

## ABSTRACT

Question-answer (Q&A) is fundamental for dialogic instruction, an important pedagogical technique based on the free exchange of ideas and open-ended discussion. Automatically detecting Q&A is key to providing teachers with feedback on appropriate use of dialogic instructional strategies. In line with this, this paper studies the possibility of automatically detecting segments of Q&A in live classrooms based solely on audio recordings of teacher speech. The proposed approach has two steps. First, teacher utterances were automatically detected from the audio stream via an amplitude envelope thresholding-based approach. Second, supervised classifiers were trained on speech-silence patterns derived from the teacher utterances. The best models were able to detect Q&A segments in windows of 90 seconds with an AUC (Area Under the Receiver Operating Characteristic Curve) of 0.78 in a manner that generalizes to new classes. Implications of the findings for automatic coding of classroom discourse are discussed.

## Keywords

Dialogic instruction, teacher feedback, professional development, live classrooms, speech, learning

## 1. INTRODUCTION

Dialogic instruction, a form of classroom discourse based around the free exchange of ideas and open-ended discussion, is considered to be an important pedagogical approach to increase student engagement [11] and improve student achievement [24]. However, the quality of implementation of dialogic instruction in classrooms varies widely. Recent research has demonstrated the importance of formative assessment of teacher use of dialogic instruction in classrooms [10]. Providing formative feedback based on what actually occurs in classrooms allows teachers to focus their efforts on improving the quality of dialogic instruction over time. Providing formative feedback efficiently, accurately,

and automatically on a day-to-day basis will ensure that teachers receive the feedback they need to better incorporate dialogic instructional practices into their classrooms. However, large-scale efforts to assess the quality of classroom discourse have relied on manual, labor-intensive, and expensive excursions into classrooms. The automation of classroom discourse analysis to inform personalized formative assessment and training programs has the potential to transform teachers' use of dialogic instruction and thereby improve student outcomes. This is the overarching goal of the current project, called CLASS 5.

The CLASS 5 project is focused on automatically analyzing classroom discourse as a means of providing feedback to teachers. CLASS 5 is intended to be a modern adaptation of the traditional model of requiring trained observers to manually code classroom discourse, an unsustainable task for providing day-to-day feedback for professional development. The automated analyses are grounded in the coding scheme of Nystrand and Gamoran [6,19], who observed thousands of students across hundreds of middle and high school English Language Arts classes. They found that the overall dialogic quality of classroom discourse through teacher's use of authentic questions (questions without prescribed responses), uptake (integration of previous speaker's ideas into future questions), and classroom discussion had positive effects on student achievement. The Nystrand and Gamoran coding scheme has been validated in multiple studies across a multitude of classrooms [2,7,17,18], hence, we are optimistic that by automating this coding scheme, we will replicate the well substantiated results of finding positive effects of dialogic instruction on student achievement. In the remainder of this section, we provide a brief overview of the Nystrand and Gamoran coding scheme, review prior work on automated classroom discourse analysis, and provide a brief overview of the present study, which is focused on automatically detecting question-answer (Q&A) segments via audio recordings of teachers during normal classroom instruction.

### 1.1 Coding Classroom Discourse

The Nystrand and Gamoran [6,19] coding scheme can be subdivided into three key 'tracks,' of increasingly fine granularity: 1) episodes, which refer to the activity/topic being addressed by the teacher; 2) segments, seventeen categories that represent possible techniques used to implement the episode; and 3) questions asked by teachers or students embedded within segments [19]. Each track can be further understood by its own nuance and properties. For example, many classes typically begin

and end with procedural episodes (i.e., “getting started”; “preparing to leave”) with one or more instructional episodes permeating the core of the class. All episodes consist of one or more segments, which can be broadly subdivided into four categories: classroom management activities, direct instruction, seatwork, and tests and quizzes. Questions are coded along dimensions of authenticity, uptake, and cognitive level as elaborated in [19].

Our current focus is on classifying key *segments* in classroom discourse. Of the seventeen segment categories the most frequent segments are lecture (including film, music, or video), Q&A, reading aloud, supervision/helping, and small group work [19]. Lecture incorporates instances where a teacher speaks for at least 30 seconds on a topic unrelated to the procedural aspects of running a class (discussing assignment instructions, for example, would not be considered lecture). Q&A segments include a question or series of questions which are non-rhetorical, non-procedural, and non-discourse management questions. Reading aloud segments consist of students reading aloud. Supervised/helping segments occur when teachers help students complete individual work. Small group work segments occurs when a group of students participates in some activity.

Discussions constitute an important, but rare, segment of particular relevance to dialogic instruction. According to the coding scheme, discussion segments consist of a free exchange among three or more participants that lasts longer than 30 seconds. Discussions typically include relatively few questions. Questions that are asked tend to focus on clarification of ideas. Discussions are typically initialized when a student makes an observation, rather than asking a question, and another student or a teacher asks for clarification on that observation. In contrast, Q&A segments usually consist of three parts – an initiation, a response, and an evaluation (IRE). The most common example of these parts begins with a teacher question, followed by a student answer, and then a teacher response to the student’s answer. The teacher’s response is often perfunctory (e.g. ‘right’ or ‘wrong’) – and sometimes non-vocalized (i.e., a nod) [16,18].

Q&A and discussion segments have traditionally positively correlated with achievement, and it is recommended that teachers should attempt to maximize use of these segments [19]. As mentioned above, discussion segments are rare in classrooms. In Nystrand’s observations there was on average less than one minute of discussion per class [19]. Traditionally Q&A segments have dominated between 30% - 42% of class time [19]. In fact, when discussion does occur it tends to do so in the midst of Q&A segments. Therefore, the present study focuses on the automated detection of Q&A segments as an initial approach to automating the coding of classroom discourse.

## 1.2 Related Work

The closest work in this area stems from research by Wang and colleagues. In particular, Wang et. al. [26] used teacher and student speech features obtained by the Language Environment Analysis system (LENA) [5] to analyze discourse profiles from 1<sup>st</sup> to 4<sup>th</sup> grade math classes. LENA is a wearable system which records and measures the quality of language produced by and directed at young children. Wang et. al. had two trained coders listen to 30-second audio windows and classify if the window represented discussion, lecture, or group work. Coders also provided their confidence in their annotation on a scale of 1 to 3

(1 indicating a lack of confidence and 3 indicating very confident).

LENA was adapted to assess when teachers were speaking, students were speaking, speech was overlapping, or there was silence. Wang et al. [25] previously found that LENA coded many student utterances as teacher utterances and modified LENA to improve its voice detection accuracy by changing the categorization algorithm to account for volume as an indicator of the distance between the speaker and the microphone. Their precision for teacher speech detection ranged from 0.95 – 0.99 and their precision for student speech detection ranged 0.70 to 0.86.

They then trained a random-forest classifier to classify the 30-second windows based on the results of speech segmentation. They used one coder’s confidence labels of 3 for training data. This constituted 62% of the windows. They validated their model on all of the windows (including the training windows), but with the annotations provided by a different coder. The coders agreed on 83% (Kappa 0.72) of the annotations, so there was considerable overlap between training and testing data. Their model achieved an accuracy of 83% (Kappa of 0.73) in discriminating between lecturing, discussion, and group work.

Although Wang et. al. [26] reported success at classifying classroom discourse at course-grained levels, their audio solution was focused on what occurred in the context of individual windows, rather than using the broader classroom context to code segments. Further, according to Wang’s coding, discussion occurred approximately 33% of the time, indicating their definition of discussion was much more inclusive than the Nystrand & Gamoran coding scheme [6,19]. Their definition of discussion, which involved students and teachers having conversations about the learning content on the whole class level (the conversation should be accessible to the majority of students in class), is not incorrect, but more closely aligns with our definition of Q&A segments. In addition, their validation method did not include an independent class-level hold-out set, thus evidence for generalizability to new classes is unclear.

## 1.3 Current Study

The present study takes inspiration from Wang et al.’s pioneering work, but also differs from it in significant ways. The LENA system is a research-grade solution and is thereby cost prohibitive and might not be scalable. This raises the question of whether classroom discourse can be automatically analyzed using more cost effective consumer-grade sensors. Of particular interest is addressing which signals are needed for accurate automatic classification of classroom discourse. Teachers lead dialogic instruction and one possibility is the only signals needed to capture classroom activity are signals that capture teacher activity. Since teachers may be anywhere in a classroom, data needs to be collected from a device that accompanies their movements with high fidelity. One attractive candidate for such a sensor is a microphone to record teacher speech, which is the approach adopted here.

Recording teacher speech is not a difficult task, but distilling the signal into appropriate features for classification of Q&A segments is more complicated. Thus, we first focused our efforts on teacher utterance detection in an attempt to find the onsets and offsets of teacher speech. Features extracted from these onsets and offsets, signaling periods of speech and rest, were then used to train classifiers to discriminate Q&A segments from all other

segments combined (i.e., Q&A vs. “other” discriminations). Note that all classification is done by analyzing these utterance onsets and offsets in an attempt to establish the accuracy of Q&A segment classification using a minimalistic approach.

The key differences between the present approach and Wang’s previous work include: (a) our use of a consumer-grade microphone rather than the LENA system; (b) segments are coded during live classrooms, so that the overarching classroom context can be incorporated in the coding; (c) we study Q&A segment classification by exclusively focusing on the teacher speech signal; and (d) our models are validated across class sessions, thereby ensuring generalizability to new classes.

The remainder of the paper is organized as follows. First, we discuss our data collection, which involved coders trained in Nystrand’s coding scheme collecting data from three teachers in 21 class sessions over the course of a semester (Section 2). We recorded teacher speech using a headset microphone and the audio signal was temporally synchronized with the human codings. Next, we developed an amplitude envelop-based utterance detection approach to segment the teacher audio into periods of speech and rest (Section 3). Then, supervised classifiers were used to detect Q&A segments from features extracted by the utterance detection algorithm (Section 4). Implications of our findings towards the broader goal of automating the analysis of classroom discourse at multiple-levels are discussed (Section 5).

## 2. Data Collection

Audio recordings were collected at a rural Wisconsin middle school during literature, language arts, and civics classes. The recordings were of three different teachers: two males – Speaker 1 and Speaker 2 – and one female – Speaker 3. The recordings spanned classes of about 45 minutes each on 9 separate days over a period of 3-4 months. Due to the occasional missed session, classroom change, or technical problem, a total of 21 classroom recordings were available for analyses. During each class session, teachers wore a Samson AirLine 77 ‘True Diversity’ UHF wireless headset microphone that recorded their speech, with the headset hardware gain adjusted to maximum. This microphone was chosen for its high noise-cancelling ability and is not cost-prohibitive (\$300 per unit). Audio files were saved in 16 kHz, 16-bit mono .wav format. Teachers were recorded naturalistically while they taught their class as usual.

Two observers trained in Nystrand et. al.’s dialogic coding technique [19,20] were present in the classroom during recordings. Observers used a specialized coding software developed by Nystrand [15] to mark episodes, segments, and teacher’s dialogic questions with the appropriate labels, as well as start and stop times as the class progressed. Later, these same observers reviewed the recordings to ensure labels were accurate and engaged in discussion until all discrepancies were resolved.

Table 1 lists the proportion of time spent on each of the segments. We note that Q&A segments were the most frequent, while discussions were highly infrequent. Other somewhat frequent segments include small group work, supervised/helping, and lecture/film/video/music. The subsequent analyses focus on detecting the 28.6% Q&A segments from all other segments combined.

**Table 1. Proportion of class time on each segment**

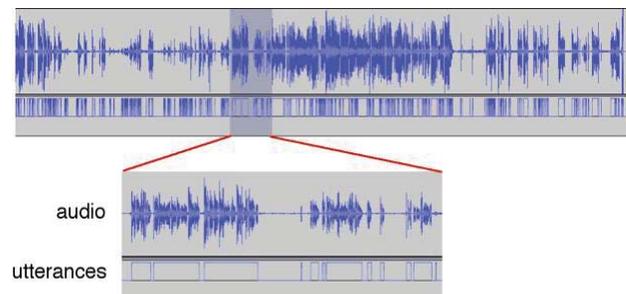
Segment	Proportion
<b>Question/answer</b>	<b>0.286</b>
Small Group Work	0.160
Supervised/helping	0.158
Lecture/film/video/music	0.150
Reading Aloud	0.093
Procedures and directions	0.091
Supervised/monitoring	0.019
Silent Reading	0.017
Other	0.012
Unsupervised seatwork	0.006
Class interruption	0.003
Game	0.002
Discussion	0.001

## 3. TEACHER UTTERANCE DETECTION

Our overall objective was to use teacher speech to detect instances of question-and-answer using recorded audio from classrooms. Before this could be done, recorded audio needed to be distilled into instances of teacher speech vs. rest (silence or no speech). Thus, we developed and validated an utterance detection method as discussed below.

### 3.1 Method

Our first assumption was that all sound was voice because teacher speech was recorded from a high-quality noise-canceling headset microphone, all sound was voice and that no advanced voice activity detection (VAD) techniques were required<sup>1</sup>. Thus, a simple binary procedure was used for utterance detection. The amplitude envelope of the teacher’s low-pass filtered speech was passed through a threshold function in 20 millisecond increments. Where the amplitude envelope was above threshold, the teacher was considered to be speaking. Where the amplitude envelope was below threshold, the teacher was assumed to not be speaking. Any time speech was detected, that speech was considered part of an utterance, meaning there was no minimum threshold for how short an utterance could be. Utterances were marked as complete when speech stopped for 1000 milliseconds (1 second). A typical result of this automatic utterance labeling method is depicted in Figure 1.



**Figure 1:** A 45-minute class recording (top) is depicted, while a small portion of the recording is enlarged for a detailed view (bottom). The upper track visualizes the .wav form of the audio. The lower track visualizes detected utterances.

<sup>1</sup> We also experimented with off-the-shelf voice activity detection algorithms [22], with comparable, if not slightly inferior, results.

The speech delimiter and threshold were both low to ensure all speech was detected, resulting in no known cases of missed speech. This process resulted in 8662 utterances, which we call *potential speech utterances*. An examination of a subset of these potential speech utterances indicated that there were a large number of false alarms. These were mainly attributed to instances of background noise permeating the audio. Common examples of background noise that the microphone picked up included voices of students who were being exceptionally loud, sounds from a film or audio clip being played in the classroom, and sounds of the teacher’s breathing.

A two-step filtering approach was taken to eliminate the false alarms. First, potential utterances less than 125 milliseconds in length (12% in all) were deemed to be too short to contain meaningful speech and were eliminated. Second, the remaining potential speech utterances were submitted through an automatic speech recognizer (Bing Speech) in an effort to identify the false alarms. Bing Speech [13] is a freely available, cloud-based automatic speech recognition service which supports seven languages. Bing returns a recognition result and a confidence score for that speech. Instances where Bing rejected the speech or where it returned no transcribed text were considered to be false alarms. After eliminating the false alarms, we were left with a total of 5502 utterance (64% of the 8662 potential utterances).

### 3.2 Validation

A small study was conducted to evaluate the aforementioned utterance detection method. A random sample of 500 potential utterances was selected and manually annotated for speech/non-speech. Speech was defined to include all articulations (i.e., “um”, “hm”, “sh”, etc) in addition to normal spoken segments. Potential speech utterances that included noise (i.e., loud students) in addition to teacher speech, the utterance was deemed as being a spoken utterance since it contained teacher speech. In total, 63% of potential utterances contained teacher speech and 37% did not. Thus, the effective false alarm rate prior to discarding utterances less than 125 milliseconds in length and accepted by Bing Speech was 37%.

Table 2 presents the confusion matrix obtained when using the 125 millisecond utterance duration threshold and Bing Speech to eliminate false alarms in the sample of 500 potential utterances. The filtering approach was highly successful, resulting in a kappa of 0.93 (agreement between computer-detected teacher utterances and human-detected teacher utterances). We note a substantially high hit and correct rejection rates and very low false alarms and miss rates. This was deemed to be sufficiently accurate for the present goal of detecting Q&A segments from teacher speech.

**Table 2. Descriptive Statistics of Utterances**

	Predicted	
Actual	Speech	Non-Speech
Speech	0.96 (hit)	0.04 (false alarm)
Non-Speech	0.03 (miss)	0.97 (correct rejection)

## 4. CLASSIFYING Q&A SEGMENTS

Segments were coded in the classrooms of three teachers in 21 classes by trained coders over the course of a semester. Our goal was to differentiate Q&A segments, which are key for dialogic instruction, from all other types of segments (a binary Q&A segment vs. “other” classification task). Features for Q&A

segment classification were obtained from the automated teacher speech utterance detection approach discussed above.

## 4.1 Method

### 4.1.1 Creating and labeling instances

Audio was sectioned into non-overlapping windows of 30, 45, 60, 75, and 90 seconds in length. Each window was assigned a label of “Q&A” or “other” based on the annotations by the trained coders (see Section 2). In some cases, there was overlap, defined as a window with multiple segment labels (e.g., first 20 seconds are Q&A and the last 10 seconds are lecture). For windows with overlap, the label of “Q&A” or “other” was assigned based on the label of the majority segment (e.g., Q&A in the example above).

Table 3 presents the number of windows and the proportion of windows that contain overlap for each window size. As expected, the proportion of windows with overlap increases as window size is increased.

**Table 3. Number of instances and proportion of instances with overlap**

Window	N	N (with overlap)	Proportion with overlap
30 seconds	1886	163	0.09
45 seconds	1253	145	0.12
60 seconds	937	126	0.13
75 seconds	748	126	0.17
90 seconds	620	112	0.18

Note: N = Total number of windows in a dataset

### 4.1.2 Feature Engineering

Features were based on teacher utterance detection as discussed in Section 3. The features attempt to capture the temporal speech patterns that teachers use in Q&A segments as defined by the initiation (speech), response (rest), and evaluation (speech) pattern of Q&A discussed in Section 1.1. They include: 1) number of utterances, 2) mean utterance duration, 3) standard deviation of utterance duration 4) minimum utterance duration 5) maximum utterance duration, 6) number of rests, 7) mean rest duration (rests were the intervals of silence between utterances), 8) standard deviation of rest duration, 9) minimum rest duration, 10) maximum rest duration, and 11) window number, the number of windows into a class session.

### 4.1.3 Model Building

Supervised classifiers were built using the Waikato Environment for Knowledge Analysis (WEKA) [9] an open source data mining tool. Models were cross validated on the class level to ensure generalizability across class sessions. In each fold, a random 67% of the classes were used for training and the remaining 33% were used for testing. This process was repeated for 25 iterations and the classification accuracy metrics was averaged across these iterations. A large number (N = 43) of standard classifiers were tested because of a lack of knowledge regarding what classifier works best for this type of data.

Various data treatments were applied in order to determine which combination resulted in the best model. First, tolerance analysis was used to eliminate features that exhibited multicollinearity [1]. Second, four feature selection algorithms: 1) Information Gain Ratio (Info-Gain) [14], 2) RELIEF-F [12], 3) Gain-Ratio [21], and 4) Correlation-based Feature Selection (CFS) [8] were used

(on training data only) to select either 25%, 50%, or 75% of the top features (the specific percentage of features was another parameter). Third, the data was Winsorized by setting outliers greater than 3 standard deviations from the mean to the corresponding value 3 standard deviations from the mean. Finally, synthetic minority oversampling technique (SMOTE) [4] was applied to the training data by creating synthetic instances of the minority Q&A class until the classes were balanced. Testing data was not sampled.

## 4.2 Results

### 4.2.1 Best Models

Classification accuracy was evaluated with area under the receiver operating characteristic curve (AUC), a metric bounded on [0, 1] with 1 indicating perfect classification and 0.5 indicating chance level classification. Table 4 presents an overview of the AUCs associated with the best models for each window size. The mean AUC across all windows was 0.73 (SD = 0.05). Classification accuracy was greater for longer window sizes with the best results obtained for the 90 second window. This model used a logistic regression classifier and had 5 features (discussed below). Table 5 presents the confusion matrix for this 90 second window model. The main source of errors appear to be misses rather than false alarms.

**Table 4. AUC for best models at each window size**

Window Size	AUC
30 secs	0.67 (0.04)
45 secs	0.69 (0.05)
60 secs	0.75 (0.04)
75 secs	0.75 (0.04)
90 secs	0.78 (0.05)

Note: Standard Deviation in parenthesis

**Table 5. Confusion matrix for best model using class-level cross-validation**

	Predicted		
Actual	Q&A	Other	Priors
Q&A	0.78 (hit)	0.22 (false alarm)	0.26
Other	0.36 (miss)	0.64 (correct rejection)	0.74

### 4.2.2 Robustness to Overlap

One concern was whether classification accuracy was degraded due to instances where Q&A segments overlapped other segments within a window. As presented in Section 4.1, the larger the window size, the greater proportion of instances that contain overlap. To study the effect of overlap, we built another set of models with overlapping segments removed.

Performance of models without overlapping windows was consistent compared to models with overlapping windows (see Table 4). Mean AUC for the models built without overlap was 0.74 (SD = 0.04) compared with mean AUC from Section 4.2.1: 0.73 (SD = 0.05). Thus, our best models were robust to instances where Q&A segments overlapped with other segments within a window.

### 4.2.3 Feature Analysis

We analyzed the five features used in the best model (90 second window). These features were 1) number of utterances, 2) mean utterance duration, 3) maximum utterance duration, 4) mean rest duration, 5) maximum rest duration. Table 6 presents the mean and standard deviation for these top features across the four most frequent segments (see Table 1). All non-Q&A segments included a fewer number of utterances, shorter utterance durations, and fewer silences (rest). For lecture/media this was likely a result of the all-inclusiveness nature of lecture/media which could include instances of only speech, a traditional lecture, or instances of no speech (e.g., when a film is played). For group work, this was likely because speech consisted of clarifying instructions or addressing individual group concerns. Supervised/helping was likely similar to group work, but rather than group concerns, individual concerns were addressed.

**Table 6. Mean and standard deviation for features across most frequent segments**

Feature	Q&A	Lecture/ Media	Small Group Work	Supervised/ Helping
Number of utterances	10.45 (4.82)	4.86 (5.16)	8.90 (4.32)	7.38 (4.46)
Mean utterance duration	5.19 (4.15)	3.23 (4.37)	2.76 (1.83)	2.80 (1.92)
Maximum utterance duration	14.62 (9.85)	7.77 (9.44)	7.80 (5.69)	8.14 (7.02)
Mean rest duration	5.40 (4.67)	38.71 (37.26)	12.22 (19.23)	17.57 (24.77)
Max rest duration	15.92 (11.71)	50.42 (33.53)	27.91 (22.31)	35.51 (25.60)

Note: Standard Deviation in parenthesis

## 5. General Discussion

Dialogic instruction is considered to be an important pedagogical approach for promoting learning and engagement in classrooms. However, analyzing the effective use of dialogic instruction in classrooms has traditionally required the presence of trained live coders and is inherently non-scalable. In the present paper, we considered the possibility of automating the coding of classroom discourse. As an initial step, we focused on automatically detecting question-and-answer (Q&A) segments, an important component of dialogic instruction, using teacher speech. We were able to detect instances of Q&A from teacher speech with moderate success in live classrooms. In this section, we compare our results to previous work in this area, discuss major findings, limitations of the present study, and consider next steps with this research.

### 5.1 Comparing with Previous Work

Our goal was to compare our approach, which only uses features from teacher speech, with models from Wang et al. [26], which were based on teacher speech, student speech, overlapping speech, and silence. A perfect comparison is complicated due to many differences across approaches, most importantly with

respect to how classroom activities were coded and how the models were validated. In particular, coders in the Wang et al. study annotated their data using 30-second intervals and specified a confidence level for each annotation. This allowed them to train their models on only the high-confidence labels. In comparison, we used a variety of different window sizes and our labels did not include a confidence level.

Our best model, which used a logistic regression classifier, had a kappa of 0.32, which is much lower than Wang et al.'s kappa of 0.77. To equate models, we also experimented with using a random forest model [3], used by Wang et al. Using a random forest model and validating at the class-level resulted in an AUC of 0.71 (SD = 0.04) and a lower kappa of 0.25 (SD = 0.07). However, we noted that Wang et al. validated their data using both training and testing data, while our models were validated on held-out class sessions. In other words, 62% of their testing data contained training instances. We attempted to replicate their validation approach by randomly selecting 62% of training instances for inclusion in the testing data. This drastically increased the AUC to 0.87, with a Kappa of 0.57.

In conclusion, although our model's performance is lower than Wang et al.'s, there are many possible reasons for this difference. For example, differences in our definitions of Q&A, their coding of each window devoid of context (which could lead to misinterpreting a window due to lack overall of context), different recording setups (LENA vs. microphone), different class structures (elementary mathematics vs. middle-school literature, language arts, and civics classes), and so on. Future work needs to equate these differences so the two approaches can be compared more equitably.

## 5.2 Major Findings

We were moderately successful in detecting Q&A segments despite considerable challenges associated with automatically recording classroom discourse using only teacher speech recorded via a headset microphone. Our major contribution is the use of consumer grade equipment to filter teacher utterances from non-teacher utterances in a noisy classroom environment. We found that we could use those utterances to develop and validate Q&A segment detectors in classrooms using only teacher speech.

Our approach consisted of two steps. Step 1 involved segmenting teacher utterances and Step 2 involved analyzing speech-silence dynamics from this segmentation to train classifiers suitable for discriminating Q&A segments from all other coded segments. For utterance detection, we used an amplitude enveloping approach to identify a large subset of potential teacher utterances and filtered them based on both duration and by submitting them to a web-based automatic speech recognizer (Bing Speech). We validated the utterance detection approach using a sample of 500 potential speech utterances randomly sampled from three teachers and 21 class sessions. We reliably and accurately discriminated speech from non-speech (kappa of 0.93) and this was accomplished despite the complexities of teacher utterance detection in noisy classrooms such as loud student speech, classroom disruptions, the use of media (i.e., video, music), and non-articulations of the teacher (such as breathing).

For Step 2, we built models to classify instances of Q&A from other instructional activities using speech-silence dynamics from the utterance segmentation. The best model was a logistic regression classifier trained on speech and silence features in 90 second windows which yielded an AUC of 0.78 when validated at

the class-level. We also built models without overlap in order to determine their effect. The models without overlap were equitable to models with overlap, indicating our models were robust to this issue. Finally, we analyzed the top features from our best model and the main finding was that Q&A segments were associated with more teacher speech and fewer rests compared to the other segments.

## 5.3 Limitations and Future Work

This study was not without its limitations. First, data was collected from three teachers who taught different subjects. However, this is a small number of teachers and all taught at the same school, so replication with a larger and more diverse sample is warranted. Second, discussion is a key indicator of dialogic discourse in classrooms [19], but our data set had only one instance of discussion, which lasted 77 seconds. Thus models could not be built for this key activity. Finally, our method focuses on a coarse-grained measure of classroom discourse. Future research is needed before a fine-grained analysis of the types of questions being asked in Q&A segments can be done (see Samei et al. [23]). When we use Bing to filter speech, it returns recognition results which could potentially be used for these fine-grained analysis. This is an important item for future work.

In general, future data collection should include more teachers, schools, social environments, and class diversity. Future work should also consider ways to capture student speech in an equally cost effective way. One possibility would be to record the entire room with a boundary microphone. However, it should be noted that every additional sensor increases the complexity of data collection and raises the threshold of adaptation in terms of cost and complexity of use. For example, if using a boundary microphone to capture student speech, a teacher needs to learn where best to position the microphone. However, a headset microphone only requires a teacher to turn it on and wear it. Nevertheless, we anticipate much improved results in Q&A detection when student speech is available.

## 5.4 Concluding Remarks

The overall purpose of this research was to automate the coding of classroom discourse and the present paper made some advances in this direction. As Nystrand et al. found [19], professional development activities focused on increasing the quality of dialogic instruction can have measurable effects on student achievement. The automated classroom discourse analysis techniques developed here can contribute to this goal by providing daily feedback to teachers for their professional development. Although this feedback alone may allow teachers to better reflect on their classroom instruction, it remains to be seen whether this increases their use of appropriate techniques for dialogic instruction. If not, tracking key components of dialogic instruction allows for interventions to increase dialogic instruction in classrooms. The research presented here represents an important initial step toward these goals, the next step involving an analysis of individual question-events at a more fine-grained level.

## 6. ACKNOWLEDGMENTS

We would like to thank Dr. Michael Brady for the amplitude envelope processing method.

This research was supported by the Institute of Education Sciences (IES) (R305A130030). Any opinions, findings and

conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of the IES.

## 7. REFERENCES

1. Allison, P.D. *Multiple regression: A primer*. Pine Forge Press, 1999.
2. Applebee, A.N., Langer, J.A., Nystrand, M., and Gamoran, A. Discussion-Based Approaches to Developing Understanding: Classroom Instruction and Student Performance in Middle and High School English. *American Educational Research Journal* 40, 3 (2003), 685–730.
3. Breiman, L. Random forests. *Machine learning* 45, 1 (2001), 5–32.
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. SMOTE: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, (2011).
5. Ford, M., Baer, C.T., Xu, D., Yapanel, U., and Gray, S. *The LENA Language Environment Analysis System*. Technical Report LTR-03-2. Boulder, CO: LENA Foundation, 2008.
6. Gamoran, A. and Kelly, S. Tracking, instruction, and unequal literacy in secondary school English. *Stability and change in American education: Structure, process, and outcomes*, (2003), 109–126.
7. Gamoran, A. and Nystrand, M. Background and instructional effects on achievement in eighth-grade English and social studies. *Journal of Research on Adolescence* 1, 3 (1991), 277–300.
8. Hall, M.A. *Correlation-based Feature Selection for Machine Learning*. 1999.
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (2009), 10–18.
10. Juzwik, M.M., Borsheim-Black, C., Caughlan, S., and Heintz, A. *Inspiring Dialogue: Talking to Learn in the English Classroom*. Teachers College Press, 2013.
11. Kelly, S. Classroom discourse and the distribution of student engagement. *Social Psychology of Education* 10, 3 (2007), 331–352.
12. Kononenko, I. Estimating attributes: analysis and extensions of RELIEF. *Machine Learning: ECML-94*, Springer (1994), 171–182.
13. Microsoft. *The Bing Speech Recognition Control*. 2014. <http://www.bing.com/dev/en-us/speech>. Accessed 14 Jan 2015
14. Mitchell, T.M. *Machine learning*. 1997. Burr Ridge, IL: McGraw Hill 45, (1997).
15. Nystrand, M. *CLASS (Classroom language assessment system) 2.0: A Windows laptop computer system for the inclass analysis of classroom discourse*. Wisconsin Center for Education Research, Madison, 1988.
16. Nystrand, M. *CLASS 4.0 user's manual*. The National Research Center on, (2004).
17. Nystrand, M. Research on the Role of Classroom Discourse as It Affects Reading Comprehension. *Research in the Teaching of English* 40, 4 (2006), 392–412.
18. Nystrand, M. and Gamoran, A. Instructional discourse, student engagement, and literature achievement. *Research in the Teaching of English*, (1991), 261–290.
19. Nystrand, M., Gamoran, A., Kachur, R., and Prendergast, C. Opening dialogue. *Teachers College, Columbia University, New York and London*, (1997).
20. Nystrand, M., Wu, L.L., Gamoran, A., Zeiser, S., and Long, D.A. Questions in time: Investigating the structure and dynamics of unfolding classroom discourse. *Discourse processes* 35, 2 (2003), 135–198.
21. Quinlan, J.R. Induction of decision trees. *Machine learning* 1, 1 (1986), 81–106.
22. Rouvier, M., Dupuy, G., Gay, P., Houry, E., Merlin, T., and Maignier, S. *An open-source state-of-the-art toolbox for broadcast news diarization*. Idiap, 2013.
23. Samei, B., Olney, A., Kelly, S., et al. Domain independent assessment of dialogic properties of classroom discourse. *Proceedings of the 7th International Conference on Educational Data Mining*, Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (2014), 233–236.
24. Sweigart, W. Classroom Talk, Knowledge Development, and Writing. *Research in the Teaching of English* 25, 4 (1991), 469–496.
25. Wang, Z., Miller, K., and Cortina, K. *Using the LENA in Teacher Training: Promoting Student Involvement through automated feedback*. na, 2013.
26. Wang, Z., Pan, X., Miller, K.F., and Cortina, K.S. Automatic classification of activities in classroom discourse. *Computers & Education* 78, (2014), 115–123.

# Topic Transition in Educational Videos Using Visually Salient Words

Ankit Gandhi\*  
Xerox Research Centre India  
Ankit.Gandhi@xerox.com

Arijit Biswas\*  
Xerox Research Centre India  
Arijit.Biswas@xerox.com

Om Deshmukh  
Xerox Research Centre India  
Om.Deshmukh@xerox.com

## ABSTRACT

In this paper, we propose a visual saliency algorithm for automatically finding the topic transition points in an educational video. First, we propose a method for assigning a saliency score to each word extracted from an educational video. We design several mid-level features that are indicative of visual saliency. The optimal feature combination strategy is learnt from a Rank-SVM to obtain an overall visual saliency score for all the words. Second, we use these words and their saliency scores to find the probability of a slide being a topic transition slide. On a test set of 10 instructional videos (12 hours), the F-score of the proposed algorithm in retrieving topic-transition slides is 0.17 higher than that of Latent Dirichlet Allocation (LDA)-based methods. The proposed algorithm enables demarcation of an instructional video along the lines of ‘table of content’/‘sections’ for a written document and has applications in efficient video navigation, indexing, search and summarization. User studies also demonstrate statistically significant improvement in across-topic navigation using the proposed algorithm.

## Keywords

visual word saliency, ranking, topic transition, educational videos, video demarcation and indexing

## 1. INTRODUCTION

The rapid growth of online courses and Open Educational Resources (OER) is considered to be one of the biggest turning points in education technology in the last few decades. Many top-ranked universities and educational organizations across the world are making thousands of video lectures available online for no cost either in the form of Massively Open Online Courses (MOOCs) or as open access material. A few national governments have also formulated policies to record classroom lectures from top-tier colleges and make them freely available online (e.g., National Program of

\*Equal contribution.

Technology Enhanced Learning (NPTEL)[1] in India). This online content can either assist classroom teaching in educational institutions with limited resources or aid out-of-class learning by the students.

As the amount of this online material is increasing rapidly (tens of thousands of hours of video currently), it is important to develop methods for efficient consumption of this multimedia content. Developing methods for summarization [2, 3], navigation [4] and topic transition[5, 6, 7, 8], for educational videos are now active areas of research.

One of the most challenging areas of research is to automatically identify time instances where a particular topic ends and a new one begins (i.e., topic transitions) in an educational video. Consider this real-classroom example: Professors often teach multiple topics within a lecture (of, say, 60-75 minutes). For example, in a lecture video<sup>1</sup> on support vector machine (SVM), the professor might cover the definition of version space, motivation for SVM, primal formulation, dual formulation, support vectors and perhaps end the lecture with kernel formulation. When a student is viewing this video lecture s/he might only be interested in the part where the professor is discussing, say, the dual formulation for SVM. This frequently happens when only a few topics of the video are relevant for the student or when the student wants to revise particular concepts for an upcoming assessment. In such a situation the student would typically ‘guesstimate’ the location with multiple back and forth navigations of the video. [Indeed, in a large-scale study on the EdX platform, authors in [9] found that certificate earning students, on an average, spend only about 4.4 minutes on a 12-15 minute-long video and skip about 22% of the content.] Finding these topic transition points in long videos can be extremely difficult and time-consuming. On the other hand, if the lecture videos can be automatically annotated with the locations where the topic is changing (e.g., dual formulation start point, primal formulation start point, etc.), the student can easily navigate through these locations and find the topics of interest efficiently.

A human expert familiar with the topic of a lecture can manually go through each lecture video and label the topic transition points. However as the quantity of online video lectures increases, manually labelling topic transition points for all of them is going to be a highly time consuming and expensive process. Demarcating these topic transitions is straightforward in written documents as the authors tend to

<sup>1</sup><https://www.youtube.com/watch?v=eHsErIPJWUU>

create table of contents or sections and subsections. Video lectures, by the very nature of the medium, don't have such demarcation. It is the goal of this research work to automatically identify these topic transitions in educational videos and highlight these 'sections' to the end user.

In this paper, we propose a novel approach where the visual content of a lecture video is analyzed to determine the transition points. In the proposed approach, the visually salient or important words are extracted from the frames of an educational video and these words along with their saliency scores are used to identify the points where the topic is changing in the video. Two major novel contributions of this work are:

1. **Visual saliency of words:** Since we use the visual content in an educational video to find out the topic transition points, one major challenge was to figure out the visual cues that are most important for determining the transition points. Intuitively it is clear that the words used in the slide frames<sup>2</sup> and their distribution can be used to determine the change of topics. However we also figured out that how a word is used in a particular slide provides significant cues regarding the word's significance in topic transition. For example, if a word is bold and located towards the top or left of the page, they contribute more in the topic transition than words which are located at the bottom right corner of a slide. An underlined word is usually more important than other words in a slide frame. To capture these visual characteristics, we propose seven novel mid-level features for the words present in educational videos. These features are called *underlineness*, *boldness*, *size*, *capitalization*, *isolation*, *padding*, and *location*. Once we extract all of these features for a word they are combined using a weight vector to create a saliency score corresponding to every word in the video. To learn this optimal weight vector we propose a novel formulation of the Rank-SVM algorithm [10] on human-annotated salient words (described in Section 4).
2. **Topic transition:** Once we extract the words and their corresponding saliency scores from a video, the next step is to find the topic change points. The saliency scores are used to estimate (a) how many novel yet salient words are introduced in each slide (referred to as Salient-Word-Novelty), and (b) number of lower saliency words in earlier slides that occur with higher saliency (referred to as Relative Saliency), for a particular slide. We propose novel methods for visual content-based across-slide computation of these two features for every slide and formulate a posterior model to estimate the probability that a given slide is a topic transition slide.

Note that the proposed approach is applicable for educational videos where slides are fully or at least partially used as word recognition accuracy for hand-written text in images is extremely poor and still an open research problem. We observed that a sizable majority of the OER is based on slideware.

<sup>2</sup>Throughout the paper by slide frame/slide we mean the frames of an education video where the teacher is displaying a slide. We also assume that the power point (.ppt) slide file is not separately available along with the video.

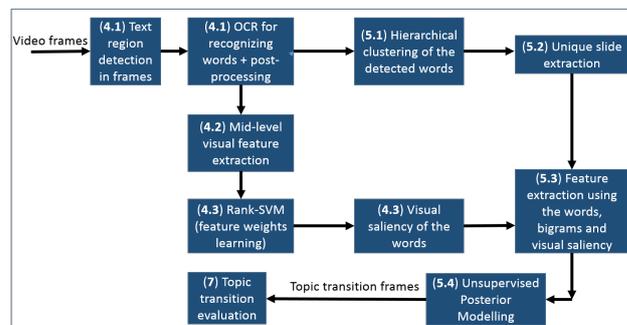


Figure 1: Pipeline of the proposed system. Figure also shows the corresponding section numbers where details of each component are explained.

The performance of the proposed approach in identifying topic transition locations was evaluated on 10 different lecture videos with a total duration of 12 hours chosen from the NPTEL set. The proposed approach outperforms the topic transition points derived using the well-known topic modelling approach [11] by an F-score of 0.17 (0.6 to 0.77 where the maximum possible F-score is 1). User studies demonstrate statistically significant improvement in across-topic navigation using the proposed algorithm.

## 2. RELATED WORK

Topic segmentation of instructional videos is an active area of research. All the work however focuses on analysing the filming aspects of the video and not the educational content.

Authors in [5] proposed a method for high level segmentation of topics in an instructional video using the variation in the content density function. The key contributing factors which manipulate the content density function are shot length, motion and sound energy. This work is extended in [6], where a thematic function is introduced to capture the frequency of appearance of the narrator, frequency of the superimposed text and narrator's voice over. The thematic function is used along with the content density function in a two tiered hierarchical algorithm for segmenting the topics. The authors in [7] propose hidden markov model (HMM) based approaches for topic transition detection. First audio-visual features are extracted from shots in a video and each shot is classified into one of the five classes: direct-narration, assisted-narration, voice-over, expressive-linkage and functional linkage. Direct-narration/assisted-narration/voice-over implies segments where the narrator is seen in the video or not. Functional linkage is captured by large superimposed text or music playing in background. Expressive linkage is used to create the mood for the subject being presented, e.g., houses with fire images in fire safety videos. Then a two level HMM is trained using a training dataset and topic transition points are found out.

All of these approaches were developed mainly for videos used in industries to train people and to convey instructions and practices, e.g., fire safety video. However OER videos, where the teacher goes over the content of slides, are very different from these kinds of videos. The camera captures the teacher and the content interchangeably with the content being more on focus. OER videos do not have music playing in background, images for mood creation, variation in sound energy or significant amount of motion. Thus all

of these prior methods will not be applicable for the educational videos of our interest. More importantly, none of these methods capture the actual content or their characteristics like saliency to model the topic change.

The proposed solution for topic transition will also drive other applications related to educational videos such as non-linear navigation [4] and summarization [2, 3] which are also active areas of research.

### 3. SYSTEM OVERVIEW

A pipeline of the proposed system is shown in Figure 1. In the next two sections (Section 4 and Section 5), we describe the technical detail of each of the components shown in the figure. The input to the system is uniformly sampled frames extracted from an educational video.

### 4. VISUAL SALIENCY

In this section, we discuss the steps involved in assigning visual saliency scores to words present in slides.

#### 4.1 Word Recognition and Text Post-processing

The first step of our pipeline is to recognize words in frames from an educational video. Recognizing text from images [12] is an extremely hard problem and continues to be an active area of research in computer vision/image processing. Words recognition usually involves two steps, first, localization of text in the frame, and then identification of text in the localized regions. In our proposed approach, we have used the algorithm proposed by Neumann *et al* [13] for localizing text in frames and the open source OCR engine Tesseract [14] to identify or recognize the words in the localized regions. The recognized words and their corresponding locations will serve as the input to the next part of our system. We perform stop words removal and words stemming as a text post-processing step on the recognized words. Stop words ('and', 'it', 'the', etc.) [15] do not contribute towards the context or topic of the document. Thus removing them reduces the complexity of system without affecting any downstream processing. Also, all words are stemmed to obtain their base or root form (e.g., stemming the words 'played', 'playing', 'player' to 'play') to further reduce the complexity.

#### 4.2 Saliency Feature Computation

In this step, we compute the visual features of words that helps in determining their saliency. For computing visual features, OCR outputs, i.e., the recognized words and their locations (bounding boxes) are used. Based upon the analysis of several educational videos (different from the ones used in experiments) taken from NPTEL and edX, we formulated several visual features such as location, boldness, underlineness, capitalization, isolation, padding and size, that are indicative of visual saliency. In this section, we provide a way to quantize them and in the next section, a formal framework is proposed that combines them to predict the overall visual saliency of a word. The visual feature extraction procedure for each of the words is described below:

- **Location feature ( $u_1$ ):** This feature captures the location information of a word in a slide. Generally, words which are located towards the top and left of a page are more important than the words located at the bottom and

right corner of a page. We use two one dimensional Gaussian distributions ( $f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ ) to compute this feature. The mean of the first Gaussian distribution is set to be the left most point of an image (giving maximum score to left-most words) and the mean of the second Gaussian distribution is set to be the top most point of an image (giving maximum score to the top most words). The variance is chosen as 0.25 times the width of image and 0.16 times the height of image respectively for the two Gaussian distributions. These parameters are selected using a small validation set. For each word, top-left corner (X-Y coordinate) of its bounding box is chosen as variables in the Gaussian distributions. The location feature is given by the product of the scores obtained from the two Gaussian distributions. If a word moves away from the top left corner of an image, the location feature value gradually decreases.

- **Boldness feature ( $u_2$ ):** It is usually true that if a word in a slide is relatively bolder than other words in the slide it is an important word. For computing boldness feature, first the word image is binarized. Then, the number of pixels which are foreground (i.e., the pixels which are part of the written text) are found. The pixel count is normalized with the number of characters present in the word to obtain the boldness feature. Thus, the boldness feature captures the average number of pixels occupied per character in a word.
- **Underlineness feature ( $u_3$ ):** A word is underlined in a slide if the teacher wants to highlight that particular word. In this work, we use Hough Transform [16] of an image to detect line segments present in that image. Since we are only interested in horizontal or near-horizontal line segments, all other line segments are removed from consideration. We use another post-processing step to remove all the horizontal line segments which are too close to the margin. Then, all the words which are immediately above the remaining horizontal/near-horizontal line segments are assigned a non-zero score for the underlineness feature. Note that the underlineness feature for a word is binary denoting whether an underline is present below the word or not.
- **Capitalization feature ( $u_4$ ):** If all the characters of a word are in upper case, then a word is assigned a non-zero score for the capitalization feature. This feature is also binary.
- **Isolation feature ( $u_5$ ):** The isolation feature represents how isolated a word is in the slide. The hypothesis is that fewer the number of words in a slide, the more important the words present in it and similarly, the fewer the number of words in a line of a slide, more important the words in that line. For example, often in title slides only a title word or a phrase is present in the center of the slide. And, the title word instances are more important than their corresponding instances elsewhere. Suppose, a word  $w$  is present in line  $l$  of a slide, then the isolation feature for word  $w$  is computed as follows -

$$u_5(w) = \frac{1}{\text{No. of lines in a slide} \times \text{No. of words in line } l}$$

- **Padding feature ( $u_6$ ):** In educational slides teachers often end a concept and start talking about another concept starting at the same slide. In those cases, they tend to keep usually more space before or after the title line of the new concept. We introduce a novel feature called padding to capture that information. For a word, padding feature is computed as the amount of empty space available below and above the line in which the word is present. Free space above is computed as number of pixels present between the current line and the previous line. Similarly, free space below is computed as the number of pixels present between the considered line and the next line. The sum is then normalized by the height of the image (slide) and the average line gap in the slide.
- **Size feature ( $u_7$ ):** This feature captures the size of word in the slide. Words appearing with larger font are generally more important than the words appearing with relatively smaller fonts. We denote the size of a word (size feature) as the height of the smallest character present in that word.

We normalize each of the visual features using 0-1 normalization across the entire video. The weighted sum of the normalized scores represents overall saliency of the words in frame. The weights are obtained using Rank-SVM[10], which we describe in the next subsection.

### 4.3 Learning to Rank Using Rank-SVM

In this subsection, we learn the relative importance of the visual features to predict the overall saliency of words. The weights determine how much each visual feature contributes to the overall saliency of a word. The weights were learnt by collecting a training dataset from 10 users over 5 videos. 10 slides were randomly selected from each video (hence, total of 50 slides) to collect the training set. Each slide has been shown to 3 users and thus, a single user provides data for 15 unique slides. For each slide, the user was asked the following question - "What are the salient words present in that slide that describe the overall content of the slide?". Generally, the number of salient words per slide vary between 2-12 depending upon the user and the slide. To overcome inter-user subjectivity, a word is accepted as salient only if it is marked as salient by atleast 2 users. Since in each slide users considered the selected words more salient than the words which were not selected, we can consider them as pairwise preferences. These pairwise preferences can be used in a Rank-SVM framework to learn the corresponding feature weights.

Let  $\mathbf{u} = [u_1 u_2 \dots u_7]$  denote the visual saliency feature vector and  $\mathbf{w} = [w_1 w_2 \dots w_7]$  denotes the weight vector to be learnt for a particular word. Also, let  $\mathcal{D}$  denotes the set of words and  $\mathcal{D}_s$  denotes the set of salient words present in slide  $S$ . Consider two words  $i$  and  $j$  such that  $i \in \mathcal{D}_s$  and  $j \in \mathcal{D} - \{\mathcal{D}_s\}$  and their visual features are  $\mathbf{u}_i$  and  $\mathbf{u}_j$  respectively. Then the weights learnt should satisfy the saliency ordering constraints (pairwise preferences by users):  $\mathbf{w}^T \mathbf{u}_i > \mathbf{w}^T \mathbf{u}_j, \forall i, j$ . For each slide  $S$ , we will have  $|\mathcal{D}_s| \times |\mathcal{D} - \{\mathcal{D}_s\}|$  number of constraints. Our goal is to learn saliency ranking function  $r(\mathbf{u}) = \mathbf{w}^T \mathbf{u}$  such that the maximum number of the following pairwise constraints are satisfied:

$$\mathbf{w}^T \mathbf{u}_i > \mathbf{w}^T \mathbf{u}_j, \forall (i, j) \in (\mathcal{D}_s, \mathcal{D} - \{\mathcal{D}_s\}), \forall S \quad (1)$$

While the above optimization problem is a NP-hard problem, it can be solved approximately by introducing negative slack variables similar to SVM classification. This leads to the following optimization problem:

$$\begin{aligned} \min \quad & \left( \frac{1}{2} \|\mathbf{w}^T\|_2^2 + C \sum \xi_{ij}^2 \right) \quad (2) \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{u}_i > \mathbf{w}^T \mathbf{u}_j + 1 - \xi_{ij}; \forall (i, j) \in (\mathcal{D}_s, \mathcal{D} - \{\mathcal{D}_s\}), \forall S \\ & \xi_{ij} \geq 0 \end{aligned}$$

The above formulation is very similar to the SVM classification problem but on pairwise difference vectors, where  $C$  is the trade-off between maximizing the margin and satisfying the pairwise relative saliency constraints. The primal form of above optimization problem is solved using Newton's method [10, 17]. It should be noted that the above optimization problem learns a function that explicitly enforces a desired ordering on the saliency of words provided as training data. Now for any new word with feature vector  $\mathbf{u}$ , the saliency score can be obtained by computing the dot product of  $\mathbf{u}$  with  $\mathbf{w}$  (i.e.,  $\mathbf{w}^T \mathbf{u}$ ). Some example frames from different videos with the detected words and their corresponding saliency scores are shown in Figure 2. Note that the words 'Torsional' and 'Waves' are part of the title of the slide in Figure 2a and are visually more salient. Hence, they have received higher scores. Similarly, in Figure 2b, the word 'Concepts' has received the highest saliency score.

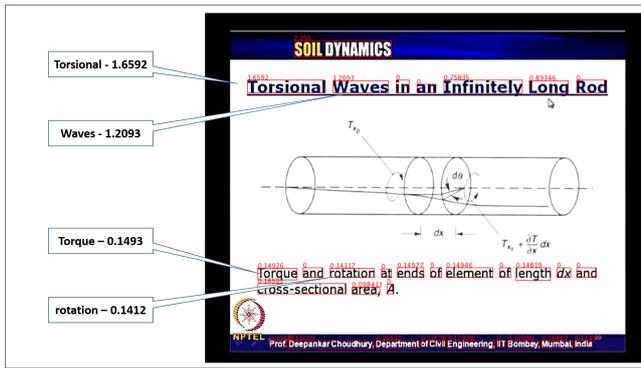
## 5. TOPIC TRANSITION

In this section, we discuss the steps of the topics transition part of our proposed approach. Words from different slides are clustered and unique slides are extracted before we compute probability that given slide is a topic transition slide.

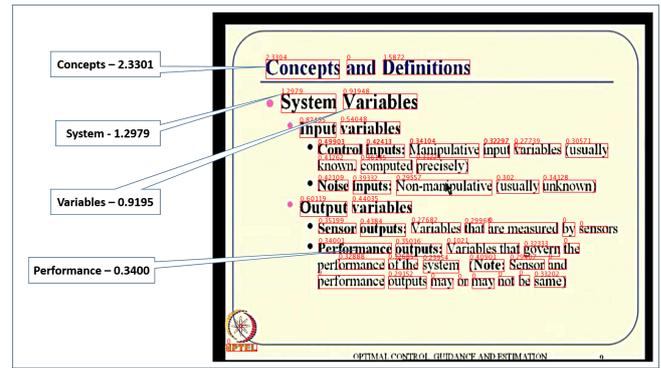
### 5.1 Clustering of Recognized Words

The text localization and recognition in uncontrolled/wild settings is an extremely hard problem to solve. In case of educational videos, word recognition result is not always perfect and is inconsistent across the slides due to changes in lighting conditions, poor frame quality (noise and low resolution), positioning of mouse pointer over frames, presence of special symbols, punctuation, typography due to italics, spacing, underlining, shaded background and unusual typefaces. For e.g., Word 'algorithm' is recognized as 'algorithm' in one slide and 'algorithm' in another slide. One simple approach to tackle this problem is to use a vocabulary and force the words to be one of the in-vocabulary words. However in many practical scenarios, it is often difficult to come up with a vocabulary of all words which can be present in the video (some of the technical words and proper nouns may not be present in the vocabulary). So, instead of using a vocabulary, we propose to use agglomerative hierarchical based clustering approach to cluster words that are same but recognized differently across slides.

Agglomerative hierarchical clustering [18] is a bottom-up clustering method and involves the following steps: (i) assigning each word to a different cluster, (ii) evaluation of all pair-wise distances between clusters, (iii) finding the pair of clusters with the shortest distance, (iv) merging the pair of clusters, (v) updating the distance matrix, i.e., computing the distances of this new cluster to all the other clusters, and (vi) repeat until a pair of clusters can be found with distance less than a predetermined threshold. In our system,



(a) A frame from Video1



(b) A frame from Video2

Figure 2: Figure showing the visual saliency scores of words on few of the slides sampled from NPTEL educational videos. Note that the words which are visually more salient based on boldness, underlineness, size, location, isolation, padding and capitalization have received higher scores.

we have used Damerau-Levenshtein distance [19] normalized by the product of the length of the two words as the distance metric (substitution, deletion and insertion cost used in the Damerau-Levenshtein distance are 1). To measure the distance between a pair of clusters, we compute the average distance (average-link hierarchical clustering) between all possible pairs of words in two clusters. Also, it must be noted that the words belonging to the same cluster will be considered as the same word for any further processing.

## 5.2 Unique Frame Extraction

One more novel contribution of this paper is to find out unique frames from an educational video. Unique frame extraction step finds all the unique frames (slides) in an educational video. Unique frames are identified from uniformly sampled frames of a video based on a criterion defined using pixel difference and the number of words (i.e., word clusters) matched. In case of educational videos, unique slides cannot be directly extracted by just comparing the adjacent slides as the same slide may be present in later portions of the video also (for e.g., in a typical video lecture, there will be frames of a slide followed by frames of a professor discussing the slide and then, again few frames of the same slide). Instead we compare each frame (beginning from start frame) with all the previous frames of a video and mark it as duplicate if the pixel difference threshold is less than  $\gamma$  or more importantly if the words overlap ratio is greater than threshold  $\rho$  with any of the previous slides. If a frame is found to be duplicate to a previous slide, it is removed from the set of possible unique slides. The pseudo code of our unique frame detection approach is provided in Algorithm 1. Using words overlap ratio along with pixel difference as the similarity metric makes our algorithm robust to change in lighting conditions, partial occlusions by the teacher and noisy video capturing methods. We note that our pipeline ignores all non-content (lecturer) frames in the video, where no text region is detected using the text detection algorithm. Hence, the output of the unique frame selection algorithm is all the unique slides present in the actual video. From this section onwards, term ‘slides’ or ‘frames’ will be used to refer to the unique slides in the video.

## 5.3 Content-based Features for Slides

In this subsection, we describe features which we propose to determine the topic transition probabilities. We have stud-

---

### Algorithm 1 Finding unique frames in a video

---

**Input:** Uniformly sampled frames  $\{S_m\}$ ,  $m = 1, 2, \dots, M$   
**Output:** Unique frames  $\{S_t\}$ ,  $t = 1, 2, \dots, T$  and  $t \in \{1, 2, \dots, M\}$

**Approach:**

```

uniqueFrames  $\leftarrow$  []
for i  $\leftarrow$  1...m do
  isUnique  $\leftarrow$  true
  for j  $\leftarrow$  1...i - 1 do
    if pixelDiff( $S_i, S_j$ )  $\leq$   $\gamma$  OR wordsOverlap( $S_i, S_j$ )  $\geq$ 
       $\rho$  then
      isUnique  $\leftarrow$  false
      break
    end if
  end for
  if isUnique AND detectedWordList( $S_i$ )  $\neq$   $\emptyset$  then
    uniqueFrames.append( $S_i$ )
  end if
end for

```

---

ied an extensive number of educational videos from different resources such as NPTEL, Coursera and EdX to figure out how a new topic is introduced in educational videos. There are two most common methods to introduce a new topic. Often a teacher while introducing a new topic, uses a few salient and novel words (the name of the new topic) in the slide. For example, the name of the new topic might be bold, placed on top of the page or might be underlined. Thus saliency of novel words definitely indicates how likely a new topic will start in a slide. Our first feature **salient word novelty** tries to capture how many novel but salient words are introduced in a slide.

Sometimes the teacher also refers to the names of the topics to be discussed later in the video by either enlisting all the topics in the video or in context with some other topics. However these occurrences usually happen with relatively lower saliency. Eventually when the topic discussion begins, the name of that topic is introduced with much higher saliency. Although these words are not novel they can still indicate topic change. Our second feature **relative saliency** is designed to capture if a word which was present earlier with lower saliency reappears in a particular

slide with higher saliency. We have found that these two features extensively cover the topic change scenarios in MOOC videos. We quantify these two features as follows:

Let us denote the unique slides obtained from previous step as set,  $\mathcal{S} = \{S_1, S_2, S_3, \dots, S_T\}$  and the words present in slide  $S_t$  as set,  $\mathcal{W}_t = \{w_1^t, w_2^t, w_3^t, \dots, w_{|\mathcal{S}_t|}^t\}$ . Also, consider a function,  $V : \mathcal{W} \times \mathcal{S} \rightarrow \mathbb{R}$  (where,  $\mathcal{W} = \bigcup_j \{\mathcal{W}_j\}$ ) that takes a word and a slide as input, and returns the saliency of the corresponding word as output. For each slide, salient word novelty and relative saliency features (described below) are computed based upon the saliency of novel and non-novel words present in the slide. A word is novel with respect to a slide if it is not present in the previous few slides of a given slide, and non-novel if it is present in the previous few slides. Those previous few unique slides constitute the neighbourhood of a given slide (for e.g, if neighbourhood size is 4, then  $S_2, S_3, S_4, S_5$  will constitute the neighbourhood of slide  $S_6$ ). Let us denote the neighbourhood of slide  $S_t$  by  $\mathcal{N}_t = \bigcup_{(t-|\mathcal{N}_t|) \leq j < t} \{S_j\}$  and the words present in neighbourhood as  $\mathcal{W}_{\mathcal{N}_t} = \bigcup_{j \in \mathcal{N}_t} \{\mathcal{W}_j\}$ . We have used  $|\mathcal{N}_t| = 4$  for all the videos in our experiments.

*Salient Word Novelty* ( $f_1$ ) (for novel words): This feature is computed using only saliency of novel words present in the slide. Lets define a vector  $F_t = \{V(v_1, t), V(v_2, t), V(v_3, t), \dots\}$  such that  $v_j \in \mathcal{W}_{\mathcal{N}_t} \cap \mathcal{W}_t$  and  $V(v_j, t) \geq V(v_{j+1}, t)$ , i.e,  $F_t$  is the ordered list of only novel words sorted by their saliency scores. Then the feature  $f_1^t$  corresponding to slide  $S_t$  is computed as follows:

$$f_1^t = \mathbf{z}F_t \quad (3)$$

where  $\mathbf{z}$  is weight vector. We wanted to take the number of novel words as well as their visual saliency both into account while designing this feature. We noted that the initial few (2-4) words' saliency matter most in determining new topics. If the number of novel words is high, we want our feature to ignore the saliency of all words except the first few high saliency novel words. Thus, we have used  $\mathbf{z}$  as an exponential decay function which makes it more generalizable than just taking the average or maximum or sum of novel word saliency scores.

*Relative Saliency* ( $f_2$ ) (for non-novel words): This feature is computed using relative saliency of non-novel words present in the slide. Lets define a set  $\mathcal{F}_t = \{v \mid v \in \mathcal{W}_{\mathcal{N}_t} \cap \mathcal{W}_t\}$  containing non-novel words for slide  $S_t$ , then the feature  $f_2^t$  is computed as follows:

$$f_2^t = \sum_{v \in \mathcal{F}_t} \frac{\max\{V(v, j) \mid j \in \mathcal{N}_t\}}{V(v, t)} \quad (4)$$

where  $\max\{V(v, j) \mid j \in \mathcal{N}_t\}$  denotes the maximum saliency of word  $v$  in neighbourhood  $\mathcal{N}_t$  of slide  $S_t$ . Lower the value of this feature, higher is the chance that new topic begins here. Lower value of this feature implies that a word is present in this slide with higher saliency as compared to its neighbourhood. This feature is designed in such a way that if higher number of words reappear in a slide we reduce the topic change probability for that slide.

**Bigrams.** For computing features  $f_1$  and  $f_2$ , we also use bigrams along with the individual words present in slide. A

bigram is a sequence of any two adjacent words in a slide. We denote the visual saliency of a bigram as the maximum visual saliency of the two words that form the bigram. Then a bigram is treated just as another word with some saliency score, and the notion of novel and non-novel word is applicable to bigrams as well. Use of bigrams helps us in treating phrases in a systematic way.

## 5.4 Posterior Modelling

Once we have the 2-dimensional feature ( $f^t = [f_1^t, f_2^t]$ ,  $1 \leq t \leq T$ ) extracted from each of the unique slides, posterior probability of each slide being a topic transition slide is computed. We label the topic transition slides as 1 and non topic-transition slides as 0. We use Gaussian distribution to model the likelihood. Thus, the poster distribution of a slide  $S_t$  being a topic transition slide given observation  $f^t$  is given below. First we define two Gaussian distributions which we will use to compute the posterior probability.

- $\mathcal{N}(\mu_1, \sigma_1)$ : Since we want to maximize the first feature we define a Gaussian distribution centred around the maximum value of  $f_1^t$ . So  $\mu_1 = \max_t(f_1^t)$  and  $\sigma_1$  is set to be twice the standard deviation of  $f_1^t$ .
- $\mathcal{N}(\mu_2, \sigma_2)$ : Since we want to minimize the second feature another Gaussian distribution is defined centred around the minimum value of  $f_2^t$ . So  $\mu_2 = \min_t(f_2^t)$  and  $\sigma_2$  is also set to be twice the standard deviation of  $f_2^t$ .

We compute the final probability as:

$$P(S_t = 1 | f^t) = \frac{P(f^t | S_t = 1) \times P(S_t = 1)}{P(f^t)} \quad (5)$$

$$\cong P(f^t | S_t = 1) \times P(S_t = 1)$$

(assuming feature independence and uniform prior over slides)

$$= P(f_1^t | S_t = 1) \times P(f_2^t | S_t = 1)$$

$$= P(f_1^t | \mu_1, \sigma_1) \times P(f_2^t | \mu_2, \sigma_2)$$

where  $P(f_1^t | \mu_1, \sigma_1)$  denotes the probability of obtaining  $f_1^t$  from  $\mathcal{N}(\mu_1, \sigma_1)$  and  $P(f_2^t | \mu_2, \sigma_2)$  denotes the probability of obtaining  $f_2^t$  from  $\mathcal{N}(\mu_2, \sigma_2)$ . Intuitively this implies that if  $f_1^t$  is higher and  $f_2^t$  is lower for a particular slide, the posterior probability of that slide being a topic transition slide will also be higher.

## 6. BASELINE METHODS

In this section, we discuss the LDA based topic modelling techniques [11] that can be used for detecting topic transition points. We have used two different versions of LDA:

- **LDA:** Latent Dirichlet Allocation (LDA) is a generative model that explains the set of observations using hidden topics. In LDA, each document can be considered as a mixture of topics. In our work, each unique slide is used as a document and the visual words present in it are used as words. Each slide is assigned a topic by maximizing over the topic likelihoods obtained from LDA. Then, we find out the slides where the topic is changing from the last slide.
- **LDA with proposed saliency:** We also compare with another version of LDA where the saliency scores obtained by our approach (Section 4) are used as the weights of the words in the slides. We refer to this method as LDA with proposed saliency.

## 7. EXPERIMENTAL RESULTS

In this section, we evaluate our approach to detect topic transition points on publicly available NPTEL educational videos. We compare the proposed approach with well-known Latent Dirichlet Allocation based topic modelling technique [11]. We also perform a user study to evaluate the efficiency and effectiveness of our approach for finding topic starting points in educational videos and provides a quick way of navigating through videos in a non-linear fashion.

### 7.1 Dataset

The experiments were conducted on 10 NPTEL educational videos. The duration of each of these videos is around 1-1.5 hours; giving us total 12 hours of video content for experiments. NPTEL videos usually have a large amount of diversity. Lighting conditions, slide orientations and style, camera angle, video resolution, and lecturer positioning in the slides (for e.g., on few occasions lecturer occupies bottom right part of the slide and sometimes full frame) vary significantly across the NPTEL videos. In few of the videos, the lecturer uses printed text instead of using slides. Also, in 4 of the selected videos, along with slides, lecturer also uses handwritten text in the presentation. In 2 other videos, the lecturer writes on slides during the presentation. All these scenarios make word recognition and thus, the identification of topic transition points extremely challenging and difficult. Examples of few of the slides from different educational videos can be seen in Figure 2. Ground truth annotation of the topic transition points in this dataset are obtained from humans who are experts in the respective topics.

### 7.2 Evaluation

The proposed approach in this paper assigns a visual saliency score to each word in the video. The mid-level visual features extracted in Section 4.2 are combined using the weight vector obtained in Section 4.3. The weights obtained using our training set are 1.1250 (boldness), 1.0015 (location), 0.6605 (underlineness), 0.6050 (size), 0.4612 (capitalization), 0.2291 (isolation), 0.0232 (padding). We observe that boldness and location features have higher weights compared to the other feature weights indicating that these two features are perhaps more important in determining the overall visual saliency.

Next, we use these saliency scores to assign a probability for each unique slide being a topic transition slide. We generate the ranked list of slides sorted by their 'being a topic transition slide' probabilities. We compute the precision and recall for all top n elements of the ranked list, where n varies from 1 to the length of the ranked list. In our analysis, we have used F-Score to measure the performance. F-Score considers both precision and recall of the method while scoring. In this context, precision is the number of correct topic transition points retrieved (within the top n elements of the ranked list) divided by the total number of retrieved topic transition points, and recall is the number of correct topic transition points retrieved divided by the total number of ground truth topic transition topics. The F-score is defined as the harmonic mean of precision and recall:

$$\text{F-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

While the recall measures how well the system can retrieve the true ground truth topic transitions, and high precision

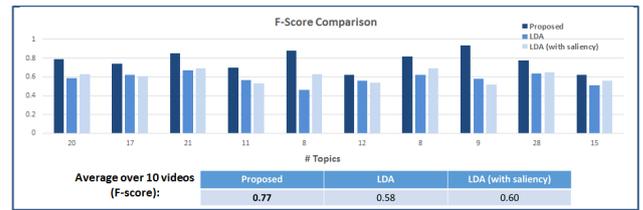


Figure 3: Comparison of proposed approach with LDA and LDA (with visual saliency) topic modelling techniques over 10 NPTEL videos. The proposed method significantly outperforms LDA by 17%.

ensures that it does not over-predict the true topic transitions, the F-Score measures the overall performance of the approach. F-Score is 1 in the ideal case (when the algorithm is perfect and when both precision and recall are 1). Following the norm regarding F-score usage[7], we also report the best F-score obtained from the ranked list. Similarly, for LDA and LDA with proposed saliency, we compute the precision and recall of the topic transitions with respect to ground truth topic transition points and get the F-Score.

In Figure 3, we provide the comparison of our approach with the LDA based techniques. We find our approach gives an average F-score of 0.77 where LDA gives an F-Score of  $0.58 \pm 0.018$  and LDA with proposed saliency gives an F-Score of  $0.60 \pm 0.021$  over 10 videos. The standard deviation values reported show the variation in LDA performance due to different number of topics. We vary the number of topics from 3 to 8 for both versions of LDA. Our method achieves an absolute improvement of 0.17 (relative improvement 28%) over state-of-the-art topic modelling technique LDA for topic transition detection in educational videos. Statistical significance of the improvement was also estimated using t-tests ( $t(10) = 4.31$ ,  $p = 0.0003$ ). This clearly shows the importance of visual saliency of words present in slides and how they can be used to detect topic transitions. We distinguish slides based on the relative saliency of their words, thus the temporal progression of saliency captures the transitions more accurately. We have also observed that the combination of two features novel word saliency and relative saliency performs the best and absence of any one of them deteriorates the performance.

### 7.3 User Study

We conducted a 6-participant 3-video user study to evaluate effectiveness and efficiency of the proposed system and compared it with the baseline transcript+youtube style rendering based interface (similar to the EdX interface) where the text is hyperlinked with the corresponding location in the video where it is spoken.

All the 6 participants had engineering degrees, exposure to online videos and had not seen these videos. The three videos were of 60, 49 and 56 minutes each. We design the video interface where we show the markers for topic transition points in the video timeline (Figure 4). For each video, we show the top-15 topic transition points obtained using the proposed approach. Each topic transition marker in Figure 4 corresponds to the first occurrence of the corresponding topic transition slide in the video. Each participant was presented one video with the proposed interface and one other video with the baseline interface. Thus, each video + interface combination was evaluated by two differ-

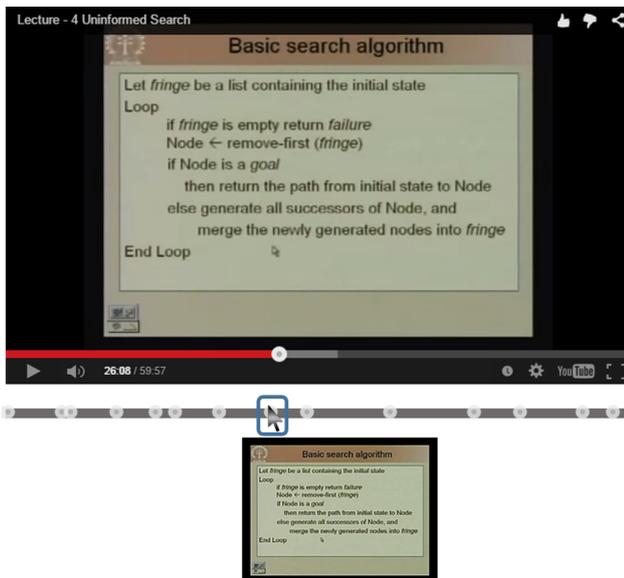


Figure 4: Proposed video interface which shows the markers for topic transition points in the video timeline. Hovering the mouse over a marker shows the thumbnail of the corresponding topic transition slide.

ent users. For each video, the users were given a list of 5 topics and asked to navigate to the starting point of each of these topics. They were allowed to go back and forth in the video multiple times to identify these topic locations. These 5 topics were randomly chosen from the ground truth topics given by the human experts (Section 7.1).

The total time taken by the participant to answer all the questions along with the number of correctly answered questions was measured. The answer is considered to be correct if the timestamp given by the participant is within a window of  $\pm 10$  seconds of the ground truth location. We observed that the average time taken by the participants to correctly answer one question is  $50.07 \pm 14.38$  sec using our interface and  $98.75 \pm 47.75$  sec using the baseline interface. The proposed interface leads to statistically significant time savings in navigating to required topics as compared to the baseline interface ( $t(6) = -2.78$ ,  $p = 0.027$ ). The percentage of correctly answered questions using our interface is 76.67% (out of 30 question instances) as compared to only 60% in baseline interface. Thus, the proposed interface shows both efficiency and effectiveness of our system.

## 8. CONCLUSION

In this paper, we propose a system for automatically detecting topic transitions in educational videos. The proposed algorithm has two novel contributions: (a) a method to assign saliency score to each word on each slide, and (b) a method to combine across-slide word saliency to estimate the posterior probability of a slide being a topic transition point. The proposed method shows a F-Score improvement of 0.17 for detecting topic transition points as compared to the LDA-based topic modelling technique. We also demonstrate the efficiency and effectiveness of the proposed method in a video navigation interface to navigate through various topics discussed in a video.

While the focus of this work is to analyze the visual content to identify topic transitions, the text transcript of the

videos can also be analyzed. In the absence of manually generated text transcripts, Automatic Speech Recognition (ASR) techniques can be used. The accuracy of ASR outputs, especially given the wide variety of speaker accent and topics will be a bottleneck in their use of downstream analysis. We are currently working on combining these multiple modalities of video, speech and text to further improve the topic transition estimation.

## 9. REFERENCES

- [1] <http://nptel.ac.in/>.
- [2] C. Choudary and T. C. Liu. Summarization of visual content in instructional videos. *IEEE Transactions on Multimedia*, 9(7):1443–1455, November 2007.
- [3] T. C. Liu and C. Choudary. Content extraction and summarization of instructional videos. In *ICIP*, pages 149–152, 2006.
- [4] Kuldeep Yadav et al. Content-driven multi-modal techniques for non-linear video navigation. In *ACM IUI*, 2015.
- [5] Dinh Q. Phung, Svetha Venkatesh, and Chitra Dorai. High level segmentation of instructional videos based on content density. In *ACM Multimedia*, 2002.
- [6] Dinh Q. Phung, Svetha Venkatesh, and Chitra Dorai. Hierarchical topical segmentation in instructional films based on cinematic expressive functions. In *ACM Multimedia*, 2003.
- [7] Dinh Q. Phung, Thi V. Duong, Svetha Venkatesh, and Hung Hai Bui. Topic transition detection using hierarchical hidden markov and semi-markov models. In *ACM Multimedia*. ACM, 2005.
- [8] Ying Li, Youngja Park, and Chitra Dorai. Atomic topical segments detection for instructional videos. In *ACM Multimedia*. ACM, 2006.
- [9] Philip J. Guo and Katharina Reinecke. Demographic differences in how students navigate through moocs. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14. ACM, 2014.
- [10] Olivier Chapelle and S. Sathya Keerthi. Efficient algorithms for ranking with SVMs. *Inf. Retr.*, 13(3):201–215, 2010.
- [11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.
- [12] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, 2011.
- [13] Lukáš Neumann and Jiří Matas. Scene text localization and recognition with oriented stroke detection. In *ICCV 2013*. IEEE, 2013.
- [14] <https://code.google.com/p/tesseract-ocr/>.
- [15] <http://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html>.
- [16] R. S. Wallace. A modified hough transform for lines. In *CVPR*, pages 665–667, 1985.
- [17] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, pages 503–510. IEEE, 2011.
- [18] A. Lukasova. Hierarchical agglomerative clustering procedure. *Pattern Recognition*, 11(5-6):365–381, 1979.
- [19] [http://en.wikipedia.org/wiki/Damerau%E2%80%9393Levenshtein\\_distance](http://en.wikipedia.org/wiki/Damerau%E2%80%9393Levenshtein_distance).

# YouEDU: Addressing Confusion in MOOC Discussion Forums by Recommending Instructional Video Clips

Akshay Agrawal  
Stanford University  
akshayka@cs.stanford.edu

Jagadish Venkatraman  
Stanford University  
jagadish@cs.stanford.edu

Shane Leonard  
Stanford University  
shanel@stanford.edu

Andreas Paepcke  
Stanford University  
paepcke@cs.stanford.edu

## ABSTRACT

In Massive Open Online Courses (MOOCs), struggling learners often seek help by posting questions in discussion forums. Unfortunately, given the large volume of discussion in MOOCs, instructors may overlook these learners' posts, detrimentally impacting the learning process and exacerbating attrition. In this paper, we present YouEDU, an instructional aid that automatically detects and addresses confusion in forum posts. Leveraging our Stanford MOOC-Posts corpus, we train a set of classifiers to classify forum posts across multiple dimensions. In particular, classifiers that target sentiment, urgency, and other descriptive variables inform a single classifier that detects confusion. We then employ information retrieval techniques to map confused posts to minute-resolution clips from course videos; the ranking over these clips accounts for textual similarity between posts and closed captions. We measure the performance of our classification model in multiple educational contexts, exploring the nature of confusion within each; we also evaluate the relevancy of materials returned by our ranking algorithm. Experimental results demonstrate that YouEDU achieves both its goals, paving the way for intelligent intervention systems in MOOC discussion forums.

## 1. INTRODUCTION

During recent years, many universities have experimented with online delivery of their courses to the public. Hundreds of thousands of learners across the world have taken advantage of these Massive Open Online Courses (MOOCs). While MOOCs are certainly more accessible than physical classes, the virtual domain brings with it its own challenges.

Lacking physical access to teachers and peer groups, learners resort to discussion forums in order to both build a sense of belonging and to better understand the subject matter at hand. Indeed, these forums could in theory be rich reflections of learner affect and academic progress. But, with MOOC enrollments so high, forums can seem unstructured and might even inhibit, rather than promote, community [17]. It becomes intractable for instructors to effectively monitor and moderate the forums. Learners seeking to clarify concepts might not get the attention that they need, as the greater sea of discussion drowns out their posts. The lack of responsiveness in forums may push learners to drop out of courses altogether [27].

The unattended, confused learner might revisit instructional videos in order to solidify his or her understanding. Yet video, a staple of MOOCs, is tyrannically linear. No table of contents or hyperlinks are available to access material in an organized fashion. Often presented with more than one hundred ten-to-fifteen-minute videos, learners might become discouraged when they realize that they will have to re-view footage to patch holes in their knowledge.

We concerned ourselves with solving the problems related to discussion forums and videos that arise when confusion goes unaddressed. In this paper, we present YouEDU, a unified pipeline that automatically classifies forum posts across multiple dimensions, staging intelligent interventions when appropriate. In particular, for those posts in which our classifier detects confusion, our pipeline recommends a ranked list of one-minute-resolution video snippets that are likely to help address the confusion. These recommendations are computed by using subsets of post contents as queries into closed caption files. That the snippets be short is important; [10] found that, regardless of video length, learners' median engagement time with videos did not exceed six minutes. Individual learners may watch beyond the minute we recommend, should they wish.

In order to enable YouEDU's classification phase, we hired consultants to tag 30,000 posts from three categories of Stanford MOOCs: Humanities and Sciences, Medicine, and Education. The set, dubbed the Stanford MOOCPosts Dataset, is available to researchers on request [2]. Besides describing the extent of confusion, each entry in the MOOCPosts set indicates whether a particular post was a *question*, an *answer*, or an *opinion*, and gauges the post's *sentiment* and *urgency* for an instructor to respond. In detecting confusion, our classifier takes into account the predictions of five other constituent classifiers, one for each of the variables (save confusion itself) encoded in our dataset.

The online teaching platforms that Stanford uses to distribute its public courses gather tracking log data comprising hundreds of millions of learner actions. We use a subset of these data as features for our confusion classification. Some of these data are also available in anonymized form to researchers upon request [1]. Until very recently, the data requisite for our classification approach—the MOOCPosts corpus and this additional metadata—simply did not exist.

The remainder of this paper is organized as follows. We examine related work in Section 2, present the MOOCPosts corpus in Section 3, and sketch the architecture of YouEDU in Section 4. In Sections 5 and 6 we detail, evaluate, and discuss YouEDU’s classification and recommendation phases. We close with a section on future work and a conclusion.

## 2. RELATED WORK

Stephens-Martinez, et al. [21] find that MOOC instructors highly value understanding the activity in their discussion forums. The role of instructors in discussion forums is investigated in [22], which finds that learners’ experiences are not appreciably affected by the presence or absence of (sparse) instructor intervention. The study did not, however, allow for instructors to regularly provide individual feedback to learners. Instructors interviewed in [12] stress the need for better ways to navigate MOOC forums, and one instructor emphasizes in particular the benefits to be reaped by using natural language processing to reorganize forums.

Wen, et al. [24] explore the relationship between attrition and sentiment, using a sentiment lexicon derived from movie reviews. Yang, et al. [27] conduct an investigation into the relationship between attrition and confusion. While [27] also presents a classifier for confusion, our classification approach differs from theirs in that it operates on a larger dataset and uses a different set of features, including those generated by other classifiers. Chaturvedi, et al. [7] predict instructor intervention patterns in forums. Our work is subtly different in that we predict posts that coders—who carefully read every post in a set of courses—deemed to be urgent, rather than learning from posts that the instructors themselves had responded to. The classification of documents by opinion and sentiment is treated in [20] and [4].

Yang, et al. [26] propose a recommendation system that matches learners to threads of interest, while Shani, et al. [19] devise an algorithm to personalize the questions presented to learners. The need for intervention systems to address confusion in particular is highlighted in [27]. Closed caption files were used in the Informedia project [23] to index into television news shows. To the best of our knowledge, the same has not been done in the context of MOOCs.

## 3. THE STANFORD MOOCPOSTS CORPUS

Given that no requestable corpus of tagged MOOC discussion forum posts existed prior to our research, we set out to create our own. The outcome of our data compilation and curation was the Stanford MOOCPosts Dataset: a corpus composed of 29,604 anonymized learner forum posts from eleven Stanford University public online classes. Available on request to academic researchers, the MOOCPosts dataset was designed to enable computational inquiries into MOOC discussion forums.

Each post in the MOOCPosts dataset was scored across six dimensions—confusion, sentiment, urgency, question, answer, and opinion—and subsequently augmented with additional metadata.

### 3.1 Methodology: Compiling the Dataset

We organized the posts by course type into three groups: Humanities/Sciences, Medicine, and Education, with 10,000,

10,002, and 10,000 entries, respectively. Humanities/Sciences contains two economics courses, two statistics courses, a global health course, and an environmental physiology course; Medicine contains two runs of a medical statistics course, a science writing course, and an emergency medicine course; Education contains a single course, *How to Learn Math*.

Each course set was coded by three independent, paid oDesk coders. That is, three triplets of coders each worked on one set of 10,000 posts. No coder worked on more than one course set. Each coder attempted to code every post for his or her particular set. All posts with malformed or missing scores in at least one coder’s spreadsheet were discarded. This elision accounts for the difference between the 29,604 posts in the final set, and the original 30,002 posts.

Coders were asked to score their posts across six dimensions:

- Question: Does this post include a question?
- Opinion: Does this post include an opinion, or is its subject matter wholly factual?
- Answer: Is this post an answer to a learner’s question?
- Sentiment: What sentiment does this post convey, on a scale of 1 (extremely negative) to 7 (extremely positive)? A score of 4 indicates neutrality.
- Urgency: How urgent is it that an instructor respond to this post, on a scale of 1 (not urgent at all) to 7 (extremely urgent)? A score of 4 indicates that instructors should respond only if they have spare time.
- Confusion: To what extent does this post express confusion, or the lack thereof, on a scale of 1 (expert knowledge) to 7 (extreme confusion)? A score of 4 indicates neither knowledge nor confusion.

Coders were given examples of posts in each category. The following was an example of an extremely urgent post:

*The website is down at the moment https://class.stanford.edu/courses/Engineering/Networking/Winter2014/courseware seems down and I’m not able to submit the Midterm. Still have the “Final Submit” button on the page, but it doesn’t work. Are the servers congested? thanks anyway*

And

*Double colons “::” expand to longest possible 0’s. If the longest is 0, will the address be considered valid? (even if it doesn’t make sense and there is no room for adding 0’s) Can someone please answer? Thanks in advance*

was given as an example of a post that was both confused (6.0) and urgent (5.0).

We created three gold sets from the coders’ scores, one for each course type. We computed inter-rater reliability using Krippendorff’s Alpha [11]. For a given post and Likert variable, the post’s gold score was computed as an unweighted average of the scores assigned to it by the subset of two coders who expressed the most agreement on that particular variable. Gold scores for binary variables were chosen

	Humanities	Medicine	Education
Urgency	0.657	0.485	0.000*
Sentiment	-0.171	-0.098	-0.134
Opinion	-0.193	-0.097	-0.297
Answer	-0.257	-0.394	-0.106
Question	0.623	0.459	0.347

Table 1: Correlations with Confusion. The urgency and question variables are strongly correlated with confusion. All correlations, save the one denoted by \*, were significant, with p-values < 0.01.

by majority votes across all three coders. We refer readers to our write-up in [2] for a more detailed treatment of our procedure and the complete inter-rater reliability results.

### 3.2 Discussion

We found significant correlations between confusion and the other five variables. In the humanities and medicine course sets, confusion and urgency were correlated with a Pearson’s correlation coefficient of 0.657 and 0.485, respectively. In all three subdivisions of the dataset, confusion and the question variable were positively correlated (0.623, 0.459, and 0.347), while the sentiment, opinion, and answer variables were negatively correlated with confusion. Table 1 reports the entire set of correlations.

That questions and confusion were positively correlated supports the finding in [25] that confusion is often communicated through questions. The negative correlations can be understood intuitively. Confusion might turn into frustration and negative sentiment; as discussed in [16], confusion and frustration sometimes go hand-in-hand. If a learner is opining on something, then it seems less likely that he or she is discussing course content. And we would hope that learners providing answers are not themselves confused.

## 4. YOUEDU: DETECT AND RECOMMEND

YouEDU<sup>1</sup> is an intervention system that recommends educational video clips to learners. Figure 1 illustrates the key steps that comprise YouEDU. YouEDU takes as input a set  $P$  of forum posts, processing them in two distinct phases: (I) detection and (II) recommendation. In the first phase, we apply a classifier to each post in  $P$ , outputting a subset  $P_c$  consisting of posts in which the classifier detected confusion. The confusion classifier functions as a *combination* classifier in that it combines the predictions from classifiers trained to predict other post-related qualities (Section 5).

The second phase takes  $P_c$  as input and, for each confused post  $p_m \in P_c$ , outputs a ranked list of educational video snippets that address the object of confusion expressed in  $p_m$ . In particular, for a given post, the recommender produces a ranking across a number of one-minute video clips by computing a similarity metric between the post and closed caption sections. In an online system, of course, learners may choose to watch beyond the end of the one-minute snippet—the snippets effectively function as a video index.

## 5. PHASE I: DETECTING CONFUSION

We frame the problem of detecting confusion as a binary one. Posts with a confusion rating greater than four in the MOOCPosts dataset fall into the “confused” class, while all

<sup>1</sup>Our entire implementation is open-source.

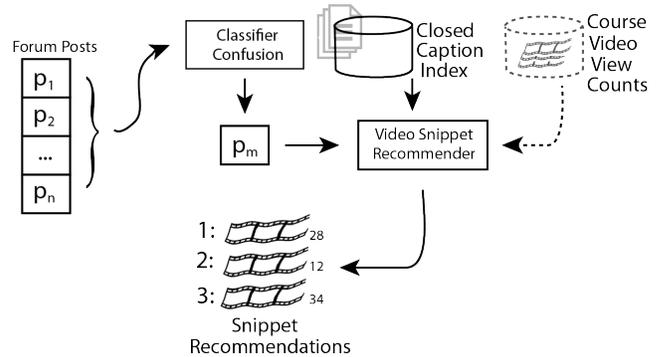


Figure 1: YouEDU Architecture. YouEDU consists of two phases: post classification and video snippet recommendation. The dotted-line module is under construction (see Section 7).

other posts fall into the “not confused” class. We craft a rich feature space that fully utilizes the data available in our MOOCPosts dataset, choosing logistic regression with  $l_2$  regularization as our model.

## 5.1 Feature Space and Model Design

Our feature space is composed of three types of inputs, those derived from the post body, post metadata, and other classifiers. The confusion classifier we train functions as a combining layer that folds in the predictions of other classifiers; these classifiers are trained to predict variables correlated with confusion. We expand upon each type of input here.

### 5.1.1 Bag-of-Words

We take the bag-of-words approach in representing documents, or forum posts. The unigram representation, while simple, pervades text classification and often achieves high performance [6]; we employ  $l_2$  regularization to prevent overfitting [18]. Each document is represented in part as a vector of indicator variables, one for each word that appears in the training data. A word is a sequence of one or more alphanumeric characters or a single punctuation mark (one of { . , ; ! ? }).

Documents are pre-processed before they are mapped to vectors. We use a subset of the stop words published by the Information Retrieval Group at the University of Glasgow [14]. Words omitted from the stop word list include, but are not limited to, interrogatives, words that identify the self (“I”, “my”), verbs indicating ability or the lack thereof, negative words (“never”, “not”), and certain conjunctions (“yet”, “but”). We ignore alphabetic case and collapse numbers, L<sup>A</sup>T<sub>E</sub>X equations, and URLs into three unique words.

### 5.1.2 Post Metadata

The feature vector derived from unigrams is augmented with post metadata, including:

- The number of up-votes accumulated by the post. We rationalized that learners might express interest in posts that voiced confusion that they shared.
- The number of reads garnered by the post’s thread.
- Whether the poster elected to appear anonymous to his or her peers or to the entire population. It has been shown that anonymity in educational discussion forums enables learners to ask questions without fear of

judgement [9], and our dataset demonstrates a strong correlation between questions and confusion.

- The poster’s grade in the class at the time of post submission, where “grade” is defined as the number of points earned by the learner (e.g., by correctly answering quiz questions) divided by the number of points possible. The lower the grade, we hypothesized, the more likely the learner might be confused.
- The post’s position within its thread—we hypothesized that learners seeking help would create new threads.

### 5.1.3 Classifier Combination

In Section 3, we demonstrated that confusion is significantly correlated with questions, answers, urgency, sentiment and opinion. As such, in predicting confusion, we take into account the predictions of five distinct classifiers, one for each of the correlates. The outputs of these five classifiers are fed as input to a *combination function* [3]—that is, a classifier for confusion—that determines the confusion class for posts.

For a given train-test partition, let  $D_{train}$  be the training set and  $D_{test}$  be the test set. Let  $H_q$ ,  $H_a$ ,  $H_o$ ,  $H_s$ , and  $H_u$  be classifiers for the question, answer, opinion, sentiment, and urgency variables, respectively. We call these classifiers *constituent* classifiers. Each constituent is trained on  $D_{train}$ , taking as input bag-of-words and post metadata features.

Let  $H_c$ , a binary classifier for confusion, be our combination function. Like the constituent classifiers,  $H_c$  is trained on  $D_{train}$  and takes as input bag-of-words and metadata features. Unlike the constituents, when training,  $H_c$  also treats the ground-truth labels for the question, answer, opinion, sentiment, and urgency variables as features. When testing  $H_c$  on an example  $d \in D_{test}$ , the constituent classifiers each output a prediction for  $d$ . These five predictions—and not the ground-truth values—are appended to the vector  $v$  of bag-of-words and metadata features derived from  $d$ . In particular, if  $v_h$  is a vector of length five encoding the predictions of the constituent classifiers, then the concatenation of  $v$  and  $v_h$  is the final feature vector for  $H_c$ .

A few subtleties:  $H_s$  uses an additional metadata feature that the other classifiers do not—the number of negative words (e.g., “not”, “cannot”, “never”, etc.).  $H_q$ ,  $H_a$ ,  $H_u$ , and  $H_c$  treat the number of question marks as an additional feature, given the previously presented correlations; [27] also used question marks in predicting confusion. And while  $H_q$ ,  $H_a$ , and  $H_o$  are by nature binary classifiers,  $H_s$  and  $H_u$  are multi-class. They predict values corresponding to negative (score < 4), neutral (score = 4), and positive (score > 4), providing  $H_c$  with somewhat granular information. Going forward, we refer to the confusion classifier that uses all the features described in this section as the *combined* classifier.

## 5.2 Evaluation and Discussion

In this section, we evaluate and interpret the performance of the combined classifier in contrast to confusion classifiers with pared-down feature sets, reporting insights gleaned about the nature of confusion in MOOCs along the way.

We quantify performance primarily using two metrics:  $F_1$  and Cohen’s Kappa. We favor the Kappa over accuracy be-

cause the former accounts for chance agreement [8]. Unless stated otherwise, reported metrics represent an average over 10 folds of stratified cross-validation.

Table 2 presents the performance of the combined classifier on the humanities and medicine course sets. As mentioned in Section 3, both sets are somewhat heterogeneous collections of courses, with a total of nearly 10,000 posts in each set. In our dataset, not-confused posts (that is, posts with a confusion score of at most 4) outnumber confused ones—only 23% of posts exhibit confusion in the humanities course set, while 16% exhibit confusion in the medicine course set.

### 5.2.1 The Language of Confusion Across Courses

Table 3 presents the performance of the combined classifier on select courses, sorted in descending order by Kappa. Our classifier performed best on courses that traded in highly technical language. Take, for example, the following post that was tagged as confused from *Managing Emergencies*, the course on which our classifier achieved its highest performance (Kappa = 0.741):

*At what doses is it therapeutic for such a patient because at high doses it causes vasoconstriction through alpha1 interactions, while at low doses it causes dilation of renal veins and splachnic vessels.*

The post is saturated with medical terms. A vocabulary so technical and esoteric is likely only used when a learner is discussing or asking a question about a specific course topic. Indeed, inspecting our model’s weights revealed that “systematic” was the 11<sup>th</sup> most indicative feature for confusion (odds ratio = 1.23) and “defibrillation” was the 15<sup>th</sup> (odds ratio = 1.22). Similarly, in *Statistical Learning*, “solutions” was the sixth most indicative feature (odds ratio = 1.75), and “predict” was the ninth (odds ratio = 1.65).

A glance at Table 3 suggests that our classifier’s performance degrades as the discourse becomes less technical. Posts like the following were typical in *How to Learn Math*, an education course about the pedagogy of mathematics:

*I am not sure if I agree with tracking or not. I like teaching children at all levels ... In a normal class setting the lower level learners can learn from the higher learners and vice versa. Although I do find it very hard to find a middle ground. There has to be an easier way.*

The above post was tagged as conveying confusion. The language is more subtle than that seen in the posts from *Managing Emergencies*, and it is not surprising that we saw our lowest Kappa (0.359) when classifying *How to Learn Math*. In this course, learners tended to voice more confusion about the structure of the class than the content itself—“link”, “videos”, and “responses” were the fourth, fifth, and seventh most indicative features, respectively.

Examining the feature weights learned from the humanities and medicine course sets provides us with a more holistic view onto the language of confusion. Domain-specific words take the backseat to words that convey the learning process. For example, in both course sets, “confused” was the

Course Set	Not Confused			Confused			Kappa
	Precision	Recall	$F_1$	Precision	Recall	$F_1$	
Humanities	0.898	0.943	0.919	0.778	0.642	0.700	0.621
Medicine	0.924	0.946	0.935	0.699	0.589	0.627	0.564

Table 2: Combined Confusion Classifier Performance, Course Sets.

Course	# Posts (% Confused)	$F_1$ : Not Confused	$F_1$ : Confused	Kappa
Managing Emergencies	279 (18%)	0.963	0.771	0.741
Statistical Learning	3,030 (30%)	0.909	0.767	0.677
Economics 1	1,583 (23%)	0.933	0.741	0.675
Statistics in Medicine (2013)	3,320 (21%)	0.916	0.671	0.589
Women’s Health	2,141 (15%)	0.933	0.506	0.445
How to Learn Math	9,878 (6%)	0.970	0.383	0.359

Table 3: Combined Confusion Classifier Performance, Individual Courses. Our classifier performed best on courses whose discourse was characterized by technical diction, like statistics or economics. In courses like *How to Learn Math* that facilitated open-ended and somewhat roaming discussions, our model found it more difficult to implicitly define confusion.

word with the highest feature weight (odds ratios equal to 3.19 and 2.97 for humanities and medicine, respectively). In the humanities course set, “?”, “couldn’t”, “report”, “question”, “haven’t”, and “wondering” came next, in that order. The importance of question-related features in particular is consistent with [25] and with the correlations in the MOOC-Posts dataset. In medicine, the next highest ranked words were “explain”, “role”, “understand”, “stuck”, and “struggling”. Table 4 displays the most informative features for the humanities and medicine course sets, as well as *How to Learn Math* and *Managing Emergencies*.

### 5.2.2 Training and Testing on Distinct Courses

We ran a series of experiments in which we trained the combined classifier on posts from one course and then tested it on posts from another one, without cross-validation. The results of these experiments are tabulated in Table 5.

Our highest Kappa (0.629) was achieved when training on *Statistics in Medicine 2013* and testing on *Statistics in Medicine 2014*; this makes sense, since they comprise two runs of the same course. Many instructors plan to offer the same MOOC multiple times [12]. Ideally, an instructor would tag but one of those runs, allowing an online classifier to truly shine. Yet even if such tagging were infeasible, our experience learning and testing on similar courses, such as two different statistics courses, suggests that an online classifier might well exhibit good performance. Performance might suffer, however, if the domains of the training and test data are non-overlapping, as is the case in the last two experiments in Table 5.

### 5.2.3 Constituent Classifiers and Post Metadata

Figure 2 illustrates the performance of each constituent classifier when cross-validating on the humanities and medicine course sets, as well as on the education course. The constituent question classifier outperformed all the others by a large margin, likely because the structure of questions is fairly consistent. Note that the constituent classifiers were not themselves fed by a lower level of classifiers; if we were attempting to predict, say, sentiment instead of confusion, we could try to improve over the performance shown here by creating a sentiment combination function that was informed by its own set of constituent classifiers.

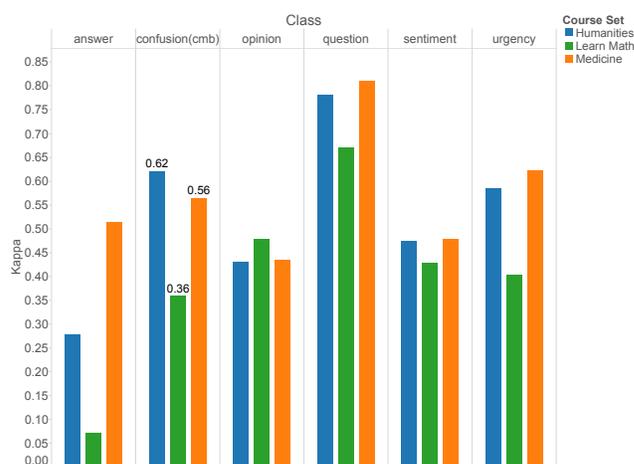


Figure 2: Constituent Classifier Performance. Confusion(cmb) is the combined classifier.

The combining function of our combined classifier consistently determined that the constituent classifiers for the question and urgency variables were particularly indicative of confusion (see Table 4). Figure 3 shows the results of an ablative analysis in which one constituent classifier was removed from the combined classifier at a time, until we were left with a classifier with no constituent classifiers (call it a *flat* classifier). The flat classifier performed worse than the combined classifier in the two course sets and the education course. For both course sets, the urgency constituent seemed to be the most helpful of the five constituents—we would expect that instructors would prioritize posts in which learners were struggling to understand the course material. However, the same was not true for *How to Learn Math*, which is consistent with the fact that no significant correlation between confusion and urgency was found (see Section 3).

The post position metadata feature also contributed positively to the classifier’s performance—removing it from the flat classifier for medicine dropped the Kappa by 0.03. The other metadata features, however, did not appear to consistently or appreciably affect classifier performance, and so we chose to omit them from our ablative analysis. (Though Table 4 shows that the number of question marks was an

Humanities	Medicine	How to Learn Math	Managing Emergencies
constituent:urgency (6.59)	constituent:question (4.05)	constituent:question (6.64)	constituent:urgency (2.47)
constituent:question (3.47)	confused (2.98)	constituent:urgency (2.13)	constituent:question (2.34)
confused (3.20)	explain (2.71)	hoping (1.94)	? (1.73)
? (3.14)	role (2.41)	link (1.76)	metadata:#? (1.54)
couldn't (2.40)	understand (2.36)	available (1.63)	hope (1.40)
report (2.23)	stuck (2.27)	responses (1.62)	what (1.31)

Table 4: Most Informative Features, Odds Ratios. Features prefixed with “constituent:” correspond to constituent predictions, while those prefixed with “metadata” correspond to post metadata features. All other features are unigram words.

Training Course	Test Course	Kappa
Stats. in Med. (2013)	Stats. in Med. (2014)	0.629
Stat. Learning	Stats. 216	0.590
Economics 1	Stats. in Med. (2013)	0.267
Stats. in Med. (2013)	Women’s Health	0.175

Table 5: Nature of Confusion Across Domains. Training and testing on similar courses typically resulted in high performance.

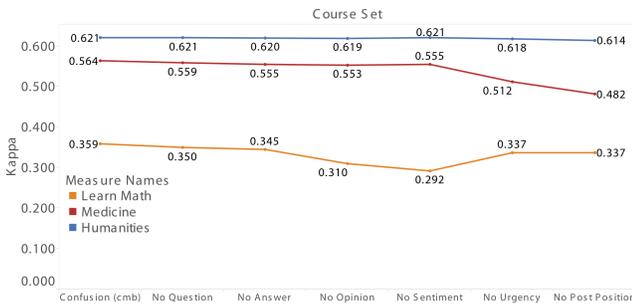


Figure 3: Ablative Analysis, Kappas. No Question is the combined classifier without the question constituent; No Answer is No Question without the answer constituent; and so on.

informative feature in the *Managing Emergencies* course.)

## 6. PHASE II: RECOMMENDING CLIPS

### 6.1 The Recommendation Algorithm

In this section, we describe how YouEDU recommends instructional material for a forum post that has been labelled as *confused* by Phase I. Every course can be thought of as a collection of several video lectures. Each video lecture on average is about 12-14 minutes long. We focus on the problem of identifying a ranked list of snippets,  $S$ , for each *confused* post. Each snippet  $s_i$  in  $S$  is a tuple  $(video\_id, seek\_minute)$  where  $video\_id$  is an identifier for the recommended video and  $seek\_minute$  is the time in the video to which the learner must seek and start playing the video. We would not necessarily need to recommend an  $end\_minute$  in a deployed setting (learners could choose when to stop watching).

Phase II of YouEDU is divided into an offline indexing phase and an online retrieval phase. We define a *bin* as a time-indexed section of a video. Each bin  $b_i$  contains the transcribed text content of the video at a minute-long time interval  $i$ . We define  $binscore(w, b)$  of a word  $w$  and bin  $b$  as the number of times word  $w$  appears in bin  $b$ . We formulate video recommendation to learners as a classical information retrieval problem. In classical IR, the goal is to retrieve the top documents that match a user’s query. In our case, the query corresponds to a confused post, and the document corresponds to a bin. We want to retrieve a ranked list of

bins that addresses the content of the confused post.

#### 6.1.1 Offline—Indexing Pipeline

In the indexing pipeline, we first divide each video into bins. We then use a part-of-speech tagger [5] to pre-process each bin. Nouns and noun-phrases tend to produce keywords that typically express what the content is about [13]. Hence, we represent a bin as a triplet  $(video\_id, start\_min, noun\_phrase\_list)$  where  $noun\_phrase\_list$  is a collection of only the nouns and noun-phrases in the bin.

We scan through each of the pre-processed bins and build an index from each word to the corresponding bin that the word appears in. This index would enable us to retrieve the list of bins  $B_w$  that corresponds to time epochs in the entire course when the word  $w$  was discussed. We also maintain a data structure that keeps track of  $binscore(w, b)$  for every word and bin. The constructed index and data structures are serialized to disk and are used by the retrieval phase.

#### 6.1.2 Online—Retrieval and Ranking:

In the online phase, we take as input confused posts, processing each with a part-of-speech tagger. Similar to the technique we used for bins, we represent each post as a list of its constituent nouns and noun-phrases. Scanning through each of the words in the pre-processed post, we add bin  $b$  to the candidate set of retrieved bins if at least one term in the pre-processed post was mentioned in  $b$ . Since we have the index constructed offline, we can use it to prune candidates from a large number of available videos (and hence, bins) in the course.

We convert each post and bin into a  $V$  dimensional vector, where  $V$  is the size of the vocabulary computed over all words used in all lectures of the course. In this vector, the value on the dimension corresponding to word  $w_i$  is  $binscore(w_i, bin)$ . We define  $simscore(P, B)$  as the cosine similarity of the post and the bin.

$$simscore(P, B) = \frac{P \cdot B}{\sqrt{\sum_{i=1}^V P_i^2} \sqrt{\sum_{i=1}^V B_i^2}} \quad (1)$$

For each candidate bin  $C_i$  in the list of candidates  $C$ , we compute  $simscore(C_i, post)$ . We rank all bins in  $C$  by their  $simscore$  values and return the ranking.

## 6.2 Evaluation

We evaluated our ranking system on the 2013 run of the *Statistics in Medicine* MOOC, offered at Stanford University, which had 24,943 learners. We chose a random sample

of queries from our MOOCPosts dataset for that course. We ran each of those posts through Phase I of YouEDU and chose 20 random posts from the posts that were labeled as confused. For each of those confused posts our algorithm produced a list of six ranked video recommendations (that is, six bins, or one-minute snippets). We then randomized the order within each group of six, obscuring the algorithm's ranking decisions. Four domain experts in statistics at Stanford independently evaluated the relevance of each snippet to its respective post; the ratings of one expert were unfortunately lost due to technical difficulties. This process induced a human-generated ranking, which we then compared to the algorithm's rank order. The rating scale given to the raters is described below:

2: **Relevant.** The recommended snippet precisely address the learner's confusion.

1: **Somewhat relevant.** The recommended snippet is somewhat useful in addressing the learner's confusion.

0: **Not Relevant:** The recommended snippet does not address the learner's confusion.

### 6.2.1 Metrics

We used two metrics to evaluate the relevancy of our recommendations: NDCG and k-precision.

*Normalized Discounted Cumulative Gain (NDCG):* NDCG measures ranking quality as the sum of the relevance scores (gains) of each recommendation. However, the gain is discounted proportional to how far down the document is in the ranking. The underlying intuition is that the gain due to a relevant document (say, relevance score of 2) that appears as the last result should be penalized more than it would be if it appeared as the first result. Hence, the DCG metric applies a logarithmic discounting function that progressively reduces a document's gain as its position in the ranked list increases [15]. The base  $b$  of the logarithm determines how sharp the applied discount is.

If  $rel_i$  is the gain associated with the document at position  $i$ , the DCG at a position  $i$  is defined recursively as

$$DCG(i) = \begin{cases} rel_i & i < b \\ DCG(i-1) + \frac{rel_i}{\log_b i} & otherwise \end{cases} \quad (2)$$

Since we want a smooth discounting function, we set  $b$  to 2. We use a graded relevance scale of 0, 1 and 2, corresponding to the types listed above, and computed the DCG for the ranked recommendations we obtained for each confused post. The ideal value of DCG (IDCG) is defined as the DCG based on the ideal ranking as judged by the raters. To obtain the IDCG, we sort the rankings given by the raters in decreasing order of relevance scores and compute the DCG of the sorted ranking. This corresponds to the maximum theoretically possible DCG in any ranking of the recommendations for that post. We normalize the DCG for our ranking by the IDCG to get the Normalized DCG (NDCG):

$$NDCG(i) = \frac{DCG(i)}{IDCG(i)} \quad (3)$$

If there are  $n$  recommended documents, then we report  $NDCG(n)$  as  $NDCG$ , the overall rating for the ranking.

Rater	NDCG	k-precision k=1	k=2	k=3
Rater1	0.66	0.66	0.61	0.62
Rater2	0.90	1.0	0.97	0.97
Rater3	0.82	0.55	0.52	0.52
Avg	0.79	0.74	0.70	0.70

Table 6: NDCG and k-Precision for recommendations

*Precision at top  $k$ :* We define the precision of a ranking  $R$  with  $n$  recommendations as the fraction of the recommendations that are relevant. The precision at  $k$  of a ranking  $R$  is defined as the precision of  $R$  restricted to its first  $k$  recommendations.

### 6.2.2 Results

Our results across the raters are summarized in Table 6. Our average precision at  $k=1$  is 0.74. This intuitively means that on 74% of cases, the first video that we suggest to a learner (as a recommendation for his or her confused post) is a relevant video. The values at  $k=2$  and  $k=3$ , at 0.70, are encouraging as well. Our NDCG numbers are high, indicating that we perform relatively well compared to the IDCG.

## 7. FUTURE WORK

The work we presented here is a first step; many opportunities for future work remain. We are actively investigating whether we can strengthen our snippet ranking further by considering which video portions learners re-visited several times. This analysis catalogs the number of views that occurred for each second of each instructional video in a course.

Another thrust of future work will use the question and answer classifiers to connect learners to each other. The challenge to meet in this work is to identify learner expertise by their answer posts, and to encourage their participation in answering questions related to their expertise. As in YouEDU, auxiliary data, such as successful homework completion, will support this line of investigation.

A third ongoing project in our group is the development of user interfaces for both instructors and learners. Using our classifiers, we have been experimenting with interactive visualizations of our classifiers' results. The hope is, for example, to have instructors see major forum-borne evidence of confusion in a single view, and to act in response through that same interface.

Video recommendations are not the only source of help for confused learners. Many online courses are repeated during multiple quarters. It should therefore be possible for our system to search forum posts of past course runs for answers to questions in current posts. Also, not all confusion is resolvable through videos. For example, difficulty in operating the video player is unlikely to have been covered in the course videos. Identifying such posts is an additional challenge.

## 8. CONCLUSION

We presented our two phase workflow that in its first phase identifies confusion-expressing forum posts in very large on-line classes. In a second phase, the workflow recommends excerpts from instructional course videos to the confused authors of these posts. Our approach utilizes new datasets of human tagged forum posts, data from learner interactions

with online learning platforms, and video closed caption files that are produced in concert with the videos for hearing-impaired learners. Evaluations of our classifiers and recommendations show that both phases of YouEDU perform well, and provide insight into the manifestations of confusion.

As novel online teaching methods are developed, the same underlying challenges will need to be met: keeping learners engaged, allowing them to feel like members of a community, and maximizing instructor effectiveness in the difficult environment of large public classes. Teaching online to very large numbers of learners from diverse backgrounds is formidable. But the potential benefits to underserved populations should encourage the investigative effort required for further research efforts.

## 9. ACKNOWLEDGMENTS

We sincerely thank Alex Kindl, Petr Johanes, MJ Cho, and Kesler Tannen for slogging through the snippet evaluations.

## 10. REFERENCES

- [1] How to access the Stanford online learning data. <http://vp01.stanford.edu/research>, 2012+.
- [2] A. Agrawal and A. Paepcke. The Stanford MOOCPosts Dataset. <http://datastage.stanford.edu/StanfordMoocPosts/>, December 2014.
- [3] P. N. Bennett, S. T. Dumais, and E. Horvitz. The combination of text classifiers using reliability indicators. *Information Retrieval*, 8(1):67–100, 2005.
- [4] H. H. Binali, C. Wu, and V. Potdar. A new significant area: Emotion detection in e-learning using opinion mining techniques. In *Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on*, pages 259–264. IEEE, 2009.
- [5] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.
- [6] C. Boulis and M. Ostendorf. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. Technical report, University of Washington, 2005.
- [7] S. Chaturvedi, D. Goldwasser, and H. Daumé III. Predicting instructor's intervention in MOOC forums.
- [8] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 1960.
- [9] M. Freeman and A. Bamford. Student choice of anonymity for learner identity in online learning discussion forums. *International Journal on E-learning*, 3(3):45–53, 2004.
- [10] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference, L@S '14*, pages 41–50, New York, NY, USA, 2014. ACM.
- [11] A. F. Hayes and K. Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.
- [12] F. M. Hollands and D. Tirthali. MOOCs: Expectations and reality, May 2014.
- [13] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics, 2003.
- [14] Information Retrieval Group at University of Glasgow. Stop word list. [http://ir.dcs.gla.ac.uk/resources/linguistic\\_utils/stop\\_words](http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words). Accessed: 2015-02-05.
- [15] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- [16] Z. Liu, J. Ocumpaugh, and R. S. Baker. Sequences of frustration and confusion, and learning. In *Proc. Int. Conf. Ed. Data Mining*, pages 114–120, 2013.
- [17] A. McGuire. Building a sense of community in MOOCs. <http://campustechnology.com/articles/2013/09/03/building-a-sense-of-community-in-moocs.aspx>, 2013. Accessed: 2015-02-01.
- [18] A. Y. Ng. Feature selection,  $L_1$  vs.  $L_2$  regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04*, pages 78–, New York, NY, USA, 2004. ACM.
- [19] G. Shani and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning.
- [20] D. Song, H. Lin, and Z. Yang. Opinion mining in e-learning system. In *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on*, pages 788–792. IEEE, 2007.
- [21] K. Stephens-Martinez, M. A. Hearst, and A. Fox. Monitoring MOOCs: Which information sources do instructors value? In *Proceedings of the First ACM Conference on Learning @ Scale Conference, L@S '14*, pages 79–88, New York, NY, USA, 2014. ACM.
- [22] J. H. Tomkin and D. Charlevoix. Do professors matter?: Using an a/b test to evaluate the impact of instructor involvement on MOOC student outcomes. In *Proceedings of the First ACM Conference on Learning @ Scale Conference, L@S '14*, pages 71–78, New York, NY, USA, 2014. ACM.
- [23] H. D. Wactlar, T. Kanade, M. A. Smith, and S. M. Stevens. Intelligent access to digital video: Informedia project. *Computer*, 29(5):46–52, May 1996.
- [24] M. Wen, D. Yang, and C. P. Rosé. Sentiment analysis in MOOC discussion forums: What does it tell us? *Proceedings of Educational Data Mining*, 2014.
- [25] N. Wilson. Learning from confusion: Questions and change in reading logs. *English Journal*, pages 62–69, 1989.
- [26] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.
- [27] D. Yang, M. Wen, I. Howley, R. Kraut, and C. Rosé. Exploring the effect of confusion in discussion forums of massive open online courses. In *Proceedings of the Second ACM Conference on Learning @ Scale Conference, L@S '15*, New York, NY, USA, 2015. ACM.

# Seeing the Instructor in Two Video Styles: Preferences and Patterns

Suma Bhat  
University of Illinois  
Urbana-Champaign, USA  
spbhat2@illinois.edu

Phakpoom  
Chinprutthiwong  
University of Illinois  
Urbana-Champaign, USA  
chinpru2@illinois.edu

Michelle Perry  
University of Illinois  
Urbana-Champaign, USA  
mperry@illinois.edu

## ABSTRACT

Instructional content designers of online learning platforms are concerned about optimal video design guidelines that ensure course effectiveness, while keeping video production time and costs at reasonable levels. In order to address the concern, we use clickstream data from one Coursera course to analyze the engagement, motivational and navigational patterns of learners upon being presented with lecture videos incorporating the instructor video in two styles - first, where the instructor seamlessly interacts with the content and second, where the instructor appears in a window in a portion of the presentation window.

Our main empirical finding is that the video style where the instructor seamlessly interacts with the content is by far the most preferred choice of the learners in general and certificate-earners and auditors in particular. Moreover, learners who chose this video style, on average, watched a larger proportion of the lectures, engaged with the lectures for a longer duration and preferred to view the lectures in streamed mode (as opposed to downloading them), when compared to their colleagues who chose the other video style. We posit that the important difference between the two video modes was the integrated view of a ‘real’ instructor in close proximity to the content, that increased learner motivation, which in turn affected the watching times and the proportion of lectures watched. The results lend further credibility to the previously suggested hypothesis that positive affect arising out of improved social cues of the instructor influences learner motivation leading to their increased engagement with the course and its broader applicability to learning at scale scenarios.

## 1. INTRODUCTION

Lecture videos constitute the primary source of course content in the massively open online courses (MOOCs) offered by platforms such as Coursera and EdX. Not surprisingly they are also the most-used course component (compared to

quiz submissions and discussion forum participation)[4, 12, 17]. Owing to the asynchronous and virtual nature of teaching and learning in these environments, lecture videos comprise the only channel through which learners have access to their instructors, an important factor affecting student motivation, satisfaction, and learning [19].

The important role of lecture videos as the primary content-bearers of a course results in instructional content designers rightly concerned about optimal video design guidelines that ensure course effectiveness; of having video lectures that maximize student learning outcomes while keeping video production time and costs at reasonable levels [9].

A recent study addresses some aspects of these concerns by comparing learner engagement patterns with video lectures across courses in the context of MOOCs [9]. The outcome of the study was a set of broad recommendations answering the concerns at a broad level. In particular, one of the take-away messages was to include the instructor’s head in the presentation at opportune times by means of a picture-in-picture view of the instructor. From the perspective of this past work, our current study is a more focused version of [9]. Using the case of a Coursera course that *concurrently* made its video lectures available in two modes (the modes differ in ways in which they present a view of the instructor), the current study is unique in that it seeks to refine the recommendations made in [9]. We do this by observing how learners interact with the course in a MOOC-sized community. The central component of the current study is an empirical analysis of the course logs to highlight the differences and similarities between the motivational, navigational and engagement tendencies of the users who interact with the two available lecture modes. The uniqueness of the study is that the same set of lectures is available in two modes, which permits us to see if there are navigational behaviors and engagement patterns that are supported by specific video types.

Our empirical findings in this study are summarized below: When comparing users who watched the lectures in only one video mode,

1. We observe that learner group preferences of one mode over the other differ considerably with a ratio of 10:1.
2. Learner group preferences of the video mode for viewing lectures directly translate to differences in the pro-

portion of available lectures watched, engagement times with the videos (via differences in watch times) and in the manner in which videos are watched (streamed vs. downloaded) between the two groups.

3. Certificate earners and auditors (learners who primarily engage with a course by only watching videos) were more likely to choose one video mode over the other.

In addition, analyzing users who watched video lectures in both modes (switching twice - from one mode to the other and back to the mode first used), we notice that the disparity in preference persists (as noted above in the case of users who watched only one video mode), although the within-user differences in engagement times and the proportion of lectures watched were not statistically significant.

While many factors could be at play here, and while proposing the need for further studies to confirm our hypothesis, we posit that the video mode preferred by the majority of learners who use only one mode has the following advantage; it offers an integrated, rather than separated, access to the instructor's eye-gaze (whether the instructor is looking at the student or the content) and gestures in close proximity to the lecture content that results in a better learning experience for the learners via the availability of more realistic social cues.

## 2. RELATED WORK

MOOCs are criticized for their high attrition rates and are alluded to as a learning environment where a majority of students are passive lurkers who do not actively engage with the course. The low levels of engagement and completion could, in part, be attributed to the demand of the MOOC environment. MOOCs require students to be autonomous learners, who can remain motivated despite low levels of instructor presence in the course, the feeling of isolation and the unclear sense of purpose in an asynchronous learning environment. Unfortunately, aside from a handful of interactions in online discussion forums, the pre-recorded videos are the only chances for an instructor to create a sense of presence in a MOOC environment.

Prior analyses of MOOCs (e.g. [4]) have found that students spent the majority of their time watching lecture videos and that many students are auditors whose course interaction is limited primarily to watching video lectures [12]. It then follows that the design of effective videos is a critical component not only for learning effectiveness but also for the success of the course in terms of making the material accessible not just to certificate earners but also to auditors.

The design of effective video lectures, however, is informed by studies in psychology, cognitive science and online learning. Recent findings suggest that a richer instructor-student interaction in an online course is afforded by video-based sessions when compared to courses with only audio narration [3]. In addition, studies on online learning reveal that learners need to have a sense of relatedness to their instructors and that this sense is often communicated through information that is superfluous to the learning objectives [19, 5]. For instance, the presence of a humanoid pedagogical agent, be it in the form of an avatar or a cartoon figure, in a computer

aided learning environment can improve a student's learning experience [6].

While the importance of non-verbal modalities of interaction (via gestures and eye-gaze) in human-human communication has long been recognized [18, 1], only recently are non-verbal modalities being harnessed in virtual communication scenarios (e.g., access to the course instructor in a window at the corner of the presentation screen in a video lecture). It is likely that increasing access to non-verbal communication can improve the instructor's sense of presence in an online-only learning environment such as a MOOC, and thus improve students' learning and their desire to stay engaged in their learning.

Clark and Mayer [6] emphasize the effectiveness of bringing instructor non-verbal modalities to the presentation because they encourage deeper engagement with the lecture content and trigger social responses in the learner [16, 7]. However, empirical evidence on its effect on learning outcomes is largely inconclusive [14, 15].

The effect of the instructor's face in visual attention, information retention and learner affect has been explored in studies such as [11, 2]. In [11] it was found that including an instructor's face in a presentation resulted in positive affective response in learners which in turn influenced the time devoted to learning. However, access to the instructor's face had no specific effect on attention or retention. In [2], an analysis of the perceptions of students being presented with two modes of video lectures incorporating the instructor's face in the presentation is available. Results suggested that having access to the instructor's gestures were potentially related to increased user satisfaction. Both these studies were not conducted in MOOC-scale environments and had a small subject pool ([11] had  $n=22$ , and [2] had  $n=60$ ).

In [9] the results of a retrospective study based on course logs of MOOCs showed the effect of different video lectures produced in different styles on the engagement patterns of learners. Based on a large dataset, results indicated that video lectures that involved a talking head were more engaging to the students than lectures without a talking head. The recommendation based on these results was to include the instructor's head in the presentation at opportune times by means of a picture-in-picture view of the instructor.

This study is set with a similar goal such as that of [9] - that of understanding learners' navigational and engagement patterns with different modes of video presentations. The different modes are chosen in a way that afford access to the instructor as recommended in [9]. This permits us to see if there are navigational behaviors and engagement patterns that are supported by specific video types.

Three factors set this study apart from prior related studies. First, we compare two modes of lecture videos with access to the instructor in the *same* course. Second, the two video modes are available to the learners over a reasonable duration (three weeks/22 lectures) thus permitting the analysis over a longer duration compared to studies [11] and [2]. Third, the setting is a realistic learning at scale setting where students rely solely on video instruction.

### 3. METHOD

We conducted a *retrospective study* of the engagement, motivational and navigational patterns of learners as a response to video lectures presented in two styles. The learners were enrolled in the Coursera course on programming massively parallel processors offered from January to March 2014.

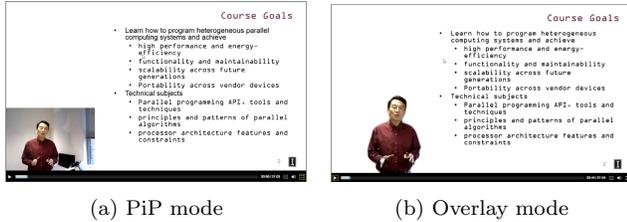


Figure 1: Screenshots of the two video modes of the lectures

#### 3.1 Video Styles

Today's advancement in video capture technology allows for ways of improving an instructor's presence in the online classroom by including the instructor's face in the presentation at substantial reductions in video production costs. The video lectures for the course were available in two modes: the *picture-in-picture* mode and the *overlay* mode both produced in non-studio settings by the instructor and recorded simultaneously. The audio quality for both modes was excellent and similar.

**Picture-in-picture mode:** Presentation creation technologies can embed a video of the instructor inside a presentation, with the instructor appearing inside a window alongside the content window. In this course, the instructor window appears in the lower left corner of the presentation. We will refer to this video style as the *PiP* mode (see Figure 1a for a screenshot of this mode). The size of the instructor's video is limited by the constraints of window placement in the presentation screen.

**Overlay mode:** New screen capture tools are able to capture only the instructor's video without the background and overlay the video of the instructor into a presentation such as PowerPoint slides much like the green screen technology used in weather forecasts. As a result of this overlay and the screen capture technology, the instructor is able to interact with the content seamlessly by pointing at relevant sections via gestures. In addition, the instructor appears in a much closer proximity to the content window, and in a larger relative proportion compared to the instructor appearing in a window alongside the content window (PiP mode above). We will refer to this video style as the *overlay* mode (refer to Figure 1b for a screenshot). Notice how the instructor appears beside the content on the left.

The first 22 lectures, which constituted the material of the first three weeks of the course, were offered in these two modes. Both modes were available in the video lectures page on the course wiki during the entire duration of the course and were available for streamed view as well as for download. The average duration of the videos was 19.23 min. The file size of a lecture in overlay mode was about 1.2 times that of its corresponding PiP version. When the course began the course syllabus had a note about the availability of the

lectures in two modes for the first three weeks and that the students were free to choose the format of their choice.

Because this was a retrospective study and not a controlled study, rather than assigning users to watch a given mode, we observed how students used the resources and interacted with them. The users<sup>1</sup> were classified into three groups based on the lecture modes they viewed (a user who clicked to view at least one lecture was counted in the group). There were users who viewed the lectures of the first 3 weeks only in the PiP mode (we call this group the **PiP** group,  $N = 899$ ), those who viewed them only in the overlay mode (we call this group the **Overlay** group,  $N = 5740$ ) and those who viewed them in both modes (the **Both** group,  $N = 3791$ ). We compare the groups with respect to the analysis variables described below.

#### 3.2 Analysis Variables

We created the following sets of analysis variables to reflect aspects of engagement, motivation and navigation.

**Engagement:** Because our analysis was based on the course logs, a true measurement of learner engagement is impossible. We approximate engagement via two proxy measures:

**Video watching time (wtime):** This is the total length of time that a student spends viewing video lectures (lectures 1 to 22) and we use it as the main index of engagement. This measure is limited in scope because it only provides information for streamed lecture views. Moreover, it has no indication whether the engagement with the video is an active one or a passive one (as in playing it in the background).

**Discussion forum visits following a lecture view (dfvisit):** We use a visit to the discussion forum (either to begin a thread, comment on an existing post or view a related post) immediately following a lecture (within 30 minutes) as an index of engagement. This reflects the intent of the learner to be open to aspects of the lecture beyond what is available in the video lecture.

**Motivation:** A limitation of this retrospective study was that access to learners' motivation (by interviewing a sample of learners, for instance) was unavailable. As a proxy to measuring motivation, we consider the following two indices:

**Certificate-earner proportion (certprop):** The fraction of users who went on to earn a certificate.

**Coverage (cov):** The fraction of lectures (and quizzes) that the learner viewed (and submitted) is our second measure of motivation. Again, an important limitation of this measure is that it only represents the fraction of lectures viewed in the streamed mode and gives no indication about those viewed after downloading<sup>2</sup>.

**Navigation:** We analyzed the navigation behavior of the

<sup>1</sup>We only took into account users who did not explicitly drop the course.

<sup>2</sup>Analysis of this variable by limiting it to users who only watched a video streaming would have been a possibility but for the fact that the sample for PiP was very small ( $< 30$ ).

students by observing their interaction with the course components. The measures we use are:

**Streaming index (SI):** In [12] streaming index was used as a measure of video consumption and is defined as the proportion of overall lecture consumption that occurs online on the platform (streamed), as opposed to off-line (downloaded),

$$\text{Streaming Index(SI)} = \frac{\text{streamed lecture consumption}}{\text{total lecture consumption}}.$$

Here we use it as a measure of video access.

**Discussion forum activity (dfview and dfpost):** The discussion forum constitutes a highly under-utilized resource in a MOOC platform and activities associated with it can be considered to be an important index of interaction with the course. Even though this measure involves a minority of course participants, we compared the number of views and posts by the users in the two groups to see if users of a video group show a tendency to participate more in discussion forums.

**Back-jump proportion (bjprop):** As used in [10], we first define a learning sequence as an ordered sequence of learning activities and its length as the number of activities in the sequence. An example of a learning sequence of length two in one session would be a lecture view followed by a quiz attempt. For our study, we consider the learning sequences of the users involving the first 22 lectures and the associated quizzes limiting the learning activities to lecture views, quiz attempts and quiz submissions.

A back-jump is a backward navigation in a learning sequence. The count of back-jumps indicates the number of times a student navigated backwards in the learning sequence and is suggestive of a departure from a linear learning sequence. In our case, this would be from a lecture to a lecture release earlier (lecture 4 to lecture 2) or from a quiz to a previous lecture (such as quiz 3 to lecture 2.3). Back-jump proportion is the number of back-jumps divided by the length of the learning sequence of the student. In [10], this measure served as an index of non-linear navigation through the course material to differentiate field-dependent learners (those who follow a sequential learning path as laid out by the content creators) from field-independent learners (those who resort to a non-linear fashion of exploring the learning environment) [8, 13], which we use in our study as well.

Other measures of comparison such as that of performance (in terms of quiz scores and assignment scores) could have been used here, but the course managed them in a server whose logs were not available in the Coursera data set.

#### 4. EMPIRICAL OBSERVATIONS

The groups **PiP** and **Overlay** (as described in Section 3.1) are first compared with respect to the analysis variables just described and the resulting observations are summarized. Following that we analyze the users in the **Both** group.

We chose a course-week (as listed in the course wiki) as a unit and counted the number of video views during that week. In Figure 2 we see the number of unique views by the users in each of the groups during the first 3 weeks. Each

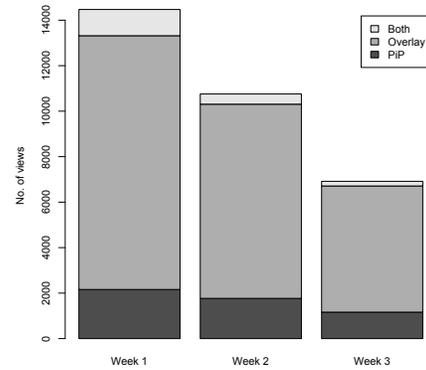


Figure 2: The number of video views in each group (Overlay, PiP and Both) over the first three weeks of the course.

bar includes the number of unique views of all lectures by a particular group during that week. What is apparent from the figure is that, over the three weeks when the lectures were available in two modes, a majority of views occurred in the Overlay mode. In addition, it is of interest to note that even in the third week there was a non-trivial number of users who watch both the modes. These views could be attributed both to the late entrants to the course and to those who switched modes in that week.

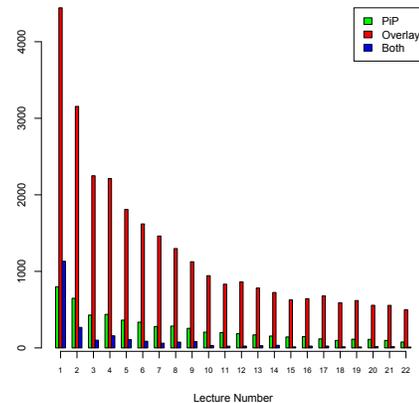


Figure 3: No. of views of each lecture over the duration of the course.

Another perspective of the views of each group is available in Figure 3 which shows the number of unique views of the 22 lectures by users in each group. Here again we notice that the *Overlay* mode was preferred by the vast majority of users compared to the *PiP* mode. It is also interesting to note from Figure 3 that the number of users who viewed the lectures in both modes is quite significant (even larger than the number of views in the *PiP* mode) for lecture 1 and then drops drastically for the lectures that follow. This could be interpreted to mean that users decide on their preferred mode as early as the first lecture. (In both these plots, the decrease in the number views is indicative of learner attrition

through the duration of the course.)

#### 4.1 Analysis Variables Compared

We filtered out all users whose total watching time lasted less than 110s (approximating individual sessions lasting on an average shorter than 5s which could have been a result of users who paused immediately after beginning to watch a video or navigated to another page). This resulted in groups of size 385 (**PiP**), 3725 (**Overlay**) and 3791 (**Both**) respectively. Below we summarize the results upon comparing the analysis variables between the first two groups.

A majority of the analysis variables considered here have highly skewed distributions thus deviating from the assumptions of normality. Under these circumstances, we resort to the Mann-Whitney U test to compare the two distributions. The null hypothesis tested here is not that the medians (or means) are equal but that the two groups come from the same underlying distribution. That is to say, we are testing for equality of location and shape of the distributions, not for equality of any one aspect of the distribution. Although the distributions were skewed we tabulate the mean of the variable for the two groups for the purpose of representation (see Table 1. The final column of the table indicates the p-value of the Mann-Whitney test. Statistically significant differences between groups are indicated in bold-face.

**The Overlay and the PiP group:** From Table 1, we observe that the underlying distributions for watch time, coverage, and streaming index differs significantly between the two groups. The **Overlay** group had a larger mean watch time compared to the **PiP** group (median watch times=33.65 min. and 21.55 min. respectively). In addition, streaming is the dominant way of accessing videos for both the groups. Streamed videos constituted an average 77% of the video usage for the **Overlay** group as opposed to 60% for the **PiP** group (respective medians 93% and 66%).

Measure	Overlay	PiP	p-value
<b>Watch time (min)</b>	<b>83.82</b>	<b>63.32</b>	< 0.01
Disc. forum visit	0.29	0.24	0.23
Certificate prop. (%)	8.48	6.75	0.24
<b>Coverage</b>	<b>0.24</b>	<b>0.18</b>	< 0.01
<b>SI</b>	<b>0.77</b>	<b>0.60</b>	< 0.01
Forum post	0.36	0.43	0.80
Forum view	11.86	17.22	0.59
Back-jump prop.	0.09	0.09	0.92

Table 1: Comparison of the measures for the two groups.

The 95% confidence interval of the two medians for wtime were (26.64, 38.75) for *PiP* and (49.77, 55.46) for *Overlay*. For SI the 95% confidence interval of the two medians were (0.8332, 0.8333) for *Overlay* and (0.564, 0.649) for *PiP*. Because the two confidence intervals for the medians of each group were non-overlapping, we infer that the corresponding distributions are different (also indicated by the Mann-Whitney U test).

This situation lends itself to two possible interpretations. Either more videos were watched streaming (with the same number of downloaded videos), or more *Overlay* videos were streamed compared to *PiP* with fewer *Overlay* videos down-

loaded. Both the interpretations imply that the streamed view was the primary way in which videos in *Overlay* mode were accessed.

As for coverage, we found that users in the *Overlay* group viewed a larger proportion of available lectures compared to their colleagues in the *PiP* group. Taken together with the lower coverage for *PiP*, its lower watch time is then justified since a smaller proportion of video views were streamed.

Although we noticed an apparent difference in the proportion of certificate earners between the two groups, a two-sample Z-test indicates that the difference in proportion was not statistically significant ( $p=0.24$ ).

**Certificate Earners:** We next restricted the analyses to the certificate-earners of the course, knowing that these were the most committed users in a course. The results limited to the certificate earners (N=316 for *Overlay* and 26 for *PiP*) are summarized in Table 2.

Measure	Overlay	PiP	p-value
watch time (min)	233.35	194.57	0.23
Disc. forum visit	1.53	1.69	0.84
<b>Coverage</b>	<b>0.70</b>	<b>0.58</b>	< 0.01
<b>Streaming Index</b>	<b>0.70</b>	<b>0.56</b>	0.02
Forum post	2.25	3.23	0.18
Forum view	76.44	113.08	0.08
Back-jump prop.	0.09	0.05	0.12

Table 2: Comparison of the measures for certificate earners.

We first computed the posterior probability of a certificate earner choosing one video mode over the other. Using empirical counts, we have the priors of the three groups: the probability of choosing the *Overlay* mode is 47%, that of choosing *PiP* is 5% and that of choosing *Both* is 48%. We also have the likelihoods: the probability that the student is a certificate-earner given that the student chose *Overlay* is 8.5%, the probability that the student is a certificate earner given that the student chose *PiP* is 6.8% (both from Table 1) and the probability that the student is a certificate-earner given that the student chose *Both* is 10.4% (empirically obtained).

Using this information, we calculated the probability that a certificate-earner chooses *Overlay* to be 0.43, that he/she chooses *PiP* is 0.04 and that he/she chooses *Both* is 0.53. This suggests that that a certificate earner is most likely to try both before settling for one mode. However, among the two modes, the more likely choice would be the *Overlay* mode.

Limiting the comparative analysis to the certificate earners of the two groups, from Table 2 we notice that the trends observed in the overall comparison are also largely applicable here with the exception of watch time. A surprising observation here is that despite the differences in the distributions for coverage and streaming index, differences in the distributions of the video watching times were not statistically significant. A likely explanation is that the certificate earners in the *PiP* group revisited portions of the same video, resulting in longer watch times compared to their *Overlay*

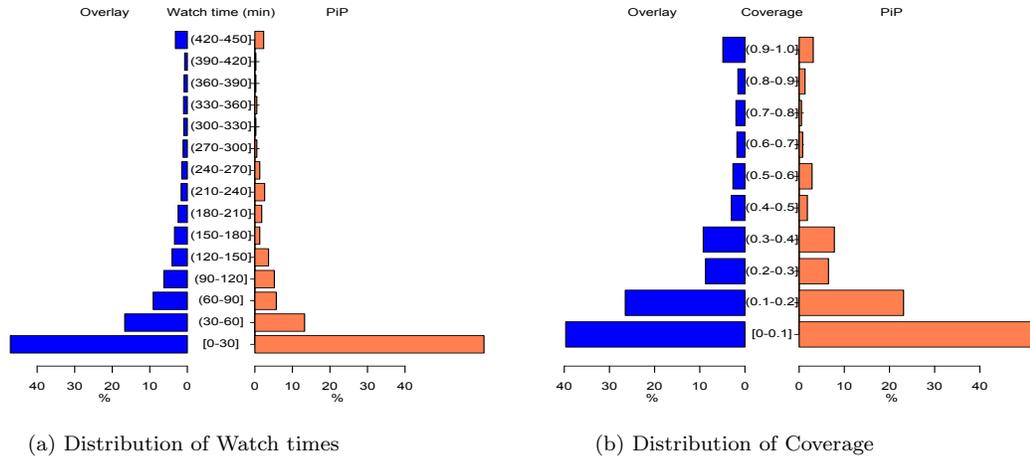


Figure 4: Histograms of Watch time (left) and Coverage (right) for the two groups compared. Each plot shows the density corresponding to each bin in the y-axis.

colleagues.

What is new here is that certificate earners in the *Overlay* group show apparently different non-linear navigational patterns compared to their *PiP* counterparts as evidenced by the difference in means. However, the distribution of back-jump fractions is not statistically significant ( $p=0.12$ ) possibly owing to the relatively small sample size of the PiP certificate earners ( $n=26$ ).

**Auditors:** From [12] we know that auditors (defined in that study as learners who did assessments infrequently if at all and engaged instead by watching video lectures) are nearly as engaged and motivated in the course as certificate earners in terms of using lecture materials in MOOCs and show similarly high levels of overall learning experience to certificate earners. Here, we investigate the extent to which users in the two groups had engagement levels similar to that of certificate earners.

We identified the auditors by clustering the users using k-means in the Overlay and the PiP groups by three factors into 3 classes (certificate earners, auditors, and lurkers):

- coverage (answering the question ‘How many lecture units were watched?’);
- streaming index (answering the question ‘How were the lectures watched?’);
- watch time (answering the question ‘For how long were the lectures watched?’).

We observed that the certificate users fell into a predominant group, which also included a set of non-certificate users ‘similar’ to the certificate users; these users behaved like the certificate users with respect to the 3 factors considered here. We refer to these users as auditors since they used resources much like the certificate users, except for the fact that they did not earn a certificate. We noticed that 3.5% of Overlay users were auditors in this sense and nearly 6% of users

in the PiP were auditors. The difference in proportion of auditors was statistically significant ( $p=0.012$ ), suggesting that PiP had a larger proportion of auditors compared to Overlay.

We then calculated the likelihood of an auditor choosing a specific viewing mode using empirical counts and note that the probability that an auditor chooses Overlay was 0.86 much greater than the probability that an auditor chose PiP, which was 0.14.

## 4.2 The Both group

While a comparison between the Overlay and the PiP groups served as a type of between-subjects analysis, a within-subjects type of analysis is afforded by analyzing the Both group. Although users watched both video modes in this group, to get a more reliable picture of engagement patterns and video mode choices, we included only those users who watched at least half of all the available lectures. With this set-up we assume that the users had sufficient exposure to the mode in which they began watching lectures before switching to the other mode. In addition, they had sufficient opportunities to experience the second mode and revert back to the original mode if they chose to do so.

Users in this group watched lectures in both modes and could be divided into three groups: 1) those who viewed a set of lectures in one mode and then switched to the other mode and remained in that second mode for the rest of the lectures, 2) those who switched twice eventually returning to watch the remaining lectures in the original mode in which they began, and 3) those who showed no apparent preference for one mode over another. For the purpose of our analysis, we focus on the second of these three groups because the sample size of the first group was too small ( $< 30$ ) to draw meaningful inferences and we had no meaningful analyses to conduct with the third group.

With this restriction on the users, we were left with 271 users (34% of the users in Both), of which 241 (89%) watched

	OPO	POP	p-value
<b>Coverage</b>	0.71	0.61	< 0.01
<b>Streaming Index</b>	0.80	0.57	< 0.01
Watch time (min)	291.69	260.85	0.10
Disc. forum visit	1.83	1.61	0.56
Back-jump prop.	5.6	4.6	0.15
<b>Certificate prop.</b>	0.37	0.63	<0.01

Table 3: Comparison of the mean values of the measures for the users in the *Both* group.

most of the lectures in the overlay mode and the remaining 30 watch most of the lectures in the PiP mode. It is clear that the majority of users in this group began watching the lectures in the overlay mode, switched to the PiP mode, and reverted to watching in the overlay mode. We represent this majority group as OPO and the other group as POP. For each user in the POP and OPO groups, we computed the measures of coverage, streaming index and watching time over the lectures watched in a given mode, yielding a measure for each video mode watched. We summarize these measures in Table 3.

We observe from Table 3 that the distributions of coverage and streaming index for the Overlay mode and PiP mode differ substantially and that the difference is statistically significant. We infer that a larger proportion of lectures were watched by the users following an OPO pattern compared to a POP pattern and that the videos in Overlay mode were streamed, while the videos in PiP mode were mostly downloaded. We notice that the distributions of watch times were not different between the OPO and POP. This implies that when the users had a chance to watch both the modes, their engagement patterns with their ‘preferred’ mode was similar.

Unlike in the case of the groups that watched only one mode, a comparison of the proportion of certificate earners between the two Both groups shows that a larger proportion of POP were certificate earners and that the difference in proportion was statistically significant via a two-sample Z-test ( $p < 0.01$ ).

## 5. INTERPRETATION OF RESULTS

The present study suggests that learners showed a strong preference for the Overlay mode over the PiP mode. Comparing the user groups that viewed the lectures in only one mode, we saw that the two groups differed significantly in their watching times, choice of video access and proportion of lecture materials viewed. The preference of Overlay was also exhibited by the users that watched both modes. This suggests that the Overlay mode was preferred and we hypothesize that these videos appeared more engaging. Taken in light of the results of studies such as [7], the findings here could be interpreted to mean that this was the result of a positive affective response of the learners to social cues in the learning environment (here the videos). It is likely that the overlay mode offered several affordances over the PiP mode – integrated rather than separated access to the instructor’s eye-gaze and gestures, the instructor’s proximity to the slides, and the larger size of the instructor – which

could have yielded differences in social cues available via the video modes.

This primary social cue that was different between the two video modes, we hypothesize, was the integrated view of a real instructor and this is likely to have increased learner motivation, which then affected the amount of time learners spent watching a lecture and the proportion of lectures they watched. Aside from this hypothesis on the difference in the availability of social cues, in the absence of watching actual behaviors of the learners affording a more fine-grained characterization of their watching patterns (such as the actual time users spent watching the video or the amount of time they spent looking at the instructor’s face) and a qualitative analysis via interviewing users for their opinions about the videos, the true implications of the difference on the video watching/consuming patterns cannot be determined. Another set of experiments to quantify the differences more specifically in terms of the perceptions of the students via qualitative and quantitative measures is currently underway and the results will be a valuable extension to the results of this study.

Based on empirical estimates of likelihood and priors, both certificate earners and auditors, two groups most engaged with the lectures, showed a higher chance of choosing the Overlay mode suggesting the possibility of this mode being conducive to the viewing characteristics of these learners. The higher chance of a certificate earner choosing the overlay mode over the PiP could be interpreted to mean that improved access to instructor’s presence is important to even the most motivated of users of a course in a MOOC environment.

## 6. LIMITATIONS AND FUTURE WORK

A primary limitation of this study is the lack of a qualitative analysis of user affect and satisfaction with the video mode of their choice. In the absence of the qualitative dimension to our study, most of the quantitative analysis were done based on proxy measures of motivation and navigational intent. Moreover, the measures chosen for the quantitative comparison were approximations based on the course logs with their inherent limitations. A more controlled study encompassing both qualitative aspects and more representative measures of engagement and navigation would shed more light on design guidelines for video lectures.

Our primary measure of engagement, video watching time, only measured the overall interaction with videos without regard to the finer engagement patterns such as the number of pauses and restarts, segments revisited, and playback rate changes that characterize a video view session. Incorporating these details as part of engagement patterns will offer a more refined view of patterns of engagement that are supported by different video presentation styles.

Other aspects for future work in this context would be exploring the preferences based on differences in demographic backgrounds of learners<sup>3</sup>. This would offer key insights about the preferences of a global audience that MOOCs aspire to

<sup>3</sup>Although learner IP address information was available, their potential of being considered as personally identifiable information precluded their inclusion in the analyses.

serve. Another important direction for future work is to explore if the same preferences and outcomes would arise regardless of the demographics the course topic attracts and the immediate functionality of seeing the instructor clearly (i.e content/topic specificity of the course).

## 7. CONCLUSION

Recognizing the important role that lecture videos play as primary content-bearers of a course in MOOCs, instructional designers are justified in their concerns about the kinds of video presentations that lead to best learning outcomes, keeping video production costs at reasonable levels. In this study we compared two video modes that offered the same set of lectures for a significant duration of a course in programming parallel processors. We found that a significantly large proportion of learners preferred one mode over the other. We hypothesize that the modes primarily differed in their ability to make the instructor's gaze and gestures more directly accessible to learners and that the mode that offered more access to instructor's gestures and eye-gaze was probably the preferred mode by the vast majority of learners. We also hypothesize that these users, possibly owing to the resulting positive affect created by improving the instructor's social presence, showed more engagement with the videos (via larger watch times), preferred the streamed mode of viewing videos (indicating immediacy in user response) and covered a larger proportion of lectures. The results also support the possibility that certificate earners (the most motivated of learners) and auditors (learners who primarily engage with a course by only watching videos) showed a higher chance of choosing the video mode offering better access to instructor's gaze and gestures, suggesting that the mode is perhaps conducive to the viewing characteristics of these learners.

## 8. REFERENCES

- [1] M. Argyle. *Bodily communication*. Routledge, 2013.
- [2] S. Bhat and G. Herman. Student perceptions of differences in visual communication mode for an online course in engineering. In *Frontiers in Education Conference, 2013 IEEE*, pages 1471–1473. IEEE, 2013.
- [3] J. Borup, R. E. West, and C. R. Graham. Improving online social presence through asynchronous video. *The Internet and Higher Education*, 15(3):195–203, 2012.
- [4] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. Seaton. Studying learning in the worldwide classroom: Research into edx's first mooc. *Research & Practice in Assessment*, 8:13–25, 2013.
- [5] M. C. Carlisle. Using you tube to enhance student class preparation in an introductory java course. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 470–474. ACM, 2010.
- [6] R. C. Clark and R. E. Mayer. *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons, 2011.
- [7] G. Cui, B. Lockee, and C. Meng. Building modern online social presence: A review of social presence theory and its instructional design implications for future trends. *Education and information technologies*, 18(4):661–685, 2013.
- [8] N. Ford and S. Y. Chen. Matching/mismatching revisited: An empirical study of learning and teaching styles. *British Journal of Educational Technology*, 32(1):5–22, 2001.
- [9] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.
- [10] P. J. Guo and K. Reinecke. Demographic differences in how students navigate through moocs. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 21–30. ACM, 2014.
- [11] R. F. Kizilcec, K. Papadopoulos, and L. Sritanyaratana. Showing face in video instruction: effects on information retention, visual attention, and affect. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2095–2102. ACM, 2014.
- [12] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 170–179. ACM, 2013.
- [13] J. O. Liegle and T. N. Janicki. The effect of learning styles on the navigation needs of web-based learners. *Computers in Human Behavior*, 22(5):885–898, 2006.
- [14] R. E. Mayer. 14 principles based on social cues in multimedia learning: Personalization, voice, image, and embodiment principles. *The Cambridge Handbook of Multimedia Learning*, page 345, 2014.
- [15] R. E. Mayer and C. S. DaPra. An embodiment effect in computer-based learning with animated pedagogical agents. *Journal of Experimental Psychology: Applied*, 18(3):239, 2012.
- [16] B. Reeves and C. Nass. The media equation: How people respond to computers, television, and new media like real people and places. 2010.
- [17] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? *Communications of the ACM*, 57(4):58–65, 2014.
- [18] R. Vertegaal, G. van der Veer, and H. Vons. Effects of gaze on multiparty mediated communication. In *Graphics Interface*, pages 95–102, 2000.
- [19] A. Wise, J. Chang, T. Duffy, and R. d. Valle. The effects of teacher social presence on student satisfaction, engagement, and learning. *Journal of Educational Computing Research*, 31(3):247 – 271, 2004.

# Using Partial Credit and Response History to Model User Knowledge

Eric G. Van Inwegen

Seth A. Adjei

Yan Wang

Neil T. Heffernan

100 Institute Rd

Worcester, MA, 01609-2280

+1-508-831-5569

{egvaninwegen, saadjei, ywang14, nth} @wpi.edu

## ABSTRACT

User modelling algorithms such as Performance Factors Analysis and Knowledge Tracing seek to determine a student's knowledge state by analyzing (among other features) right and wrong answers. Anyone who has ever graded an assignment by hand knows that some answers are "more wrong" than others; i.e. they display less of an understanding of the skill(s) involved. This investigation seeks to understand the effects of progression through wrong answers to right answers in a way to determine how the "level" of wrongness affects future performance. The key findings are that A.) where in a series of opportunities a student reaches the goal impacts future performance, as does B.) the "level" of previous wrongness, even two questions before the current opportunity.

---

Right students are all alike;  
every wrong student is wrong in his or her own way.  
(with apologies to Ms. Karenina and Mr. Tolstoy)

## 1. INTRODUCTION

The use of algorithms to estimate student knowledge based on performance on intelligent tutoring systems (ITS) has been around for two decades. Two of the more well-known methods are knowledge tracing (KT) [6] and performance factors analysis (PFA) [11]. Both models use a student's right or wrong answers and develop a model to estimate the chance that a student has "learned" a particular skill. KT uses Bayes nets to determine four parameters per skill; PFA uses logistic regression to determine three parameters per skill. Although the order of correctness is incorporated into the models, both use only correctness as their input. Other pieces of information that may be collected by the ITS are neglected in these models.

ITS may collect any number of additional pieces of information about a student, their actions, their exact answers, etc. For example, Baker et. al. use over 20 features to make their predictions [2]. Some even make use of biometrics through additional sensors. (See Cavalio and D'Mello's review of several methods [3]. The goal of many of these algorithms is to try to

make a computer tutor that is at least as responsive, observant, and effective as a human tutor would be. Incorporating more data about a student's affect can be seen as an attempt to give a computer access to the information that a human tutor would notice. However, the more detailed that a model becomes, the more computationally time-consuming it becomes. Also, as the number of inputs increases, fewer ITS's can make use of it (as a complex set of inputs may not be collected on all systems). One feature that might be incorporated into these algorithms is the use of the number of attempts and hints a student uses to answer a problem to classify more conditions than binary right and wrong and to look for the effect of how long it takes a student to achieve a particular classification.

Human teachers often employ the idea of partial credit, both as a motivational tool, and as a more accurate measure of knowledge (when compared to the binary correctness). Any teacher who has graded papers knows that some wrong answers (and workflow) demonstrate a nearly full understanding of a skill, while other wrong answers demonstrate a near-total lack of understanding. The idea of using dynamic testing (that is, a testing medium that gives hints to and tracks the number of attempts made by students) has been around since at least the 1980's. Bryant, Brown and Campione [5] compared traditional testing (binary correctness) to dynamic testing (tracking how many hints students needed to be successful). Others (e.g. Grigorenko and Sternberg) reviewed this kind of dynamic testing (among other methods) [8] and concluded that dynamic testing provides a more accurate measure [12]

Unfortunately, some ITS's can only determine the "worthiness" of a wrong answer if all wrong answers are somehow programmed in. Some ITS's do make use of pre-programmed wrong answers, but partial credit may or may not be given. Efforts before this one have been made to use partial credit to measure student knowledge [16]. E.g., in ASSISTments<sup>1</sup>, wrong answers may be programmed to give a student a particular message, but A.) students are still marked completely wrong (and given no credit) and B.) all of these wrong answer messages must be programmed into the problems (which is incredibly time-intensive).

A more common method of assigning partial credit in ITS's is to give partial credit based on the number of attempts it takes a student to get the right answer [1] and/or the number of hints a student uses [9]. This is much faster to program, and does not require looking at all possible wrong answer to determine which ones show a limited understanding of the skill (as opposed to no understanding of the skill). The basic argument would be that a student who is "only slightly wrong" might figure out her mistake

---

<sup>1</sup> ASSISTments is an online learning system primarily for math, based out of Worcester Polytechnic Institute.

after only one wrong attempt, while a student who is “very wrong” might need several hints and several attempts before he can get the problem right. We are not analyzing specific wrong answers in this treatment; we using a student’s partial credit history to modify the probability of that student getting the next question correct.

In this paper, we are analyzing a dataset from ASSISTments from the years 2012-2013. (The dataset contains ~ 500K student-problem instances; the content is mainly middle-school mathematics.) We analyze the student entries for patterns of attempts, hints use, and a simplistic order of actions to determine “bins” of students. We are also able to analyze the data to seek patterns of moving through bins (that is, as a single student uses more or less assistance on subsequent problems), and when in a particular opportunity count a bin (or sequence of bins) is encountered. We build off of our earlier work presented at the Learning Analytics & Knowledge Conference, 2015.

## 1.1 Background

In our previous work [13], we built off of other works that looked at attempt use, hint use (Assistance Model – AM – [15]), and simple sequence of action (Sequence of Action model – SOA – [7 and 17]), and modified and combined these models to make our own. We looked at the combination of number of attempts used to get the right answer, hint use, whether the “bottom-out hint” (BOH) was used, and a simplistic order of actions. In our model, the values for each parameter were:

- attempt use: 1, 2, 3, 4, (5+)
- hint use: 0, 1, 2, (3+)
- first action: hint or attempt
- BOH: used or not used

This gave us 35 different combinations. By analyzing the similarities of actions and future performance - defined as the average next problem correctness (NPC) and found by using pivot tables on 80% of the dataset, the 35 bins were combined into only 16. This gave us the “Fine-Grain Action” model (FGA). Table 1 shows the bins and re-grouped bins, and the NPC values.

**Table 1a: The Fine-Grain-Action model  
1<sup>st</sup> action = attempt**

	1 att.	2 att.	3 att.	4 att.	5 + att.
0 hint	0.8156 Bin 1	0.7380 Bin 2	0.6771 Bin 3	0.6380 Bin 4	0.6211 Bin 5
1 hint	-----	0.7012 Group A		0.6321 Group C	
2 hint	-----	0.5812 Group E			
3+ hint	-----				
BOH	0.5099 Group G				

**Table 1b: The Fine-Grain-Action model  
1<sup>st</sup> action = hint**

	1 att.	2 att.	3 att.	4 att.	5 + att.
0 hint	-----	-----	-----	-----	-----
1 hint	0.7083 Bin 6	0.6192 Group B		0.5702 Group D	
2 hint	0.5250 Bin 11	0.4688 Group F			
3+ hint	0.4118 Bin 16				
BOH	0.3396 Group H				

## 1.2 Research Questions

Extending from our previous analysis, we have three questions we want to address here:

- 1.) What is the significance of the bins?
  - a) What is the statistical significance of the different bins? E.g. are bins “x” and “y” (arbitrary names) reliably different?
  - b) Can the bins be re-grouped into larger groups without loss of predictive power? (E.g. Why 16? Why not 35 or 3?)
- 2.) Can the sequence of students moving through “Super Bins” be used to make more accurate predictions? (E.g. Is there a difference in expected outcome when comparing a student who moves from Super Bin 3 to 1 vs. 5 to 1?)
- 3.) Should all wrong answers be treated equally? Can we use reasonably simple and replicable methods to identify what student actions demonstrate different levels of understanding of the material?
  - a) Is there an impact of bin sequence and / or opportunity count on predicted outcome?

## 2. METHODS

### 2.1 Creating the “SuperBins” (Method 1)

A quick glance at the next problem correctness (NPC) values in Table 1 shows that some bins are very nearly equivalent. When displayed in the above format, local values vary enough to warrant the bins. However, when put in order by bin values (which are just the mean NPC for instances falling into that category), we can now run a simple t-test (two tailed) analysis to compare one bin to the one that comes immediately after. This gives us Table 2.

**Table 2: The bins from the FGA reordered and showing the p-value that compares one bin to the one immediately below.**

Bin	NPC	stdev	n	p-value	Ordinal
1	0.8156	0.3878	215,870	< 0.0001	1st
2	0.7380	0.4397	22,229	0.0055	2nd
6	0.7083	0.4545	1,958	0.5827	3rd
A	0.7012	0.4577	3,414	0.0162	4th
3	0.6771	0.4676	5,616	0.0009	5th
4	0.6380	0.4806	2,326	0.7168	6th
C	0.6321	0.4822	1,408	0.4941	7th
5	0.6211	0.4851	2,518	0.9416	8th
B	0.6192	0.4856	407	0.1339	9th
E	0.5812	0.4934	4,011	0.8154	10th
D	0.5702	0.4950	114	0.3782	11th
11	0.5250	0.4994	541	0.4851	12th
G	0.5099	0.4999	40,652	0.0781	13th
F	0.4688	0.4990	465	0.1252	14th
16	0.4118	0.4922	289	0.0141	15th
H	0.3396	0.4736	13,989	-----	16th

In Table 2, the p-value analysis comparing the bin of that line to the one below it allows us to identify natural break points and groups. Bins are regrouped according to these break points. That

is, bins are grouped together as long as two bins fail to be statistically different. This gives us five “SuperBins” (Table 3).

It may seem somewhat arbitrary to keep bins 16 and H separate (with a p-value of 0.0141), while grouping A and 3 together (with a p-value of 0.0162). We could argue that we used a deciding value of 0.015, but that would be an arbitrary value. The real reason for keeping 16 and H separate is that the action of using the bottom out hint (and using a hint as the first action) seems to be different than any other combination of actions and should be kept separate. Throughout the rest of this analysis, we will see that the results of keeping this bin separate as its own SuperBin gives us more predictive ability.

This gives us a useful and relevant way to regroup bins that are not reliably different. One can easily make the argument against the 16 bins in FGA that, if two bins are not statistically different, why have them? By combining statistically similar bins, there is more meaning (in prediction) to assigning a particular value for the next problem correctness, even if the recombination “smooths over” the different ways that a student could arrive at a particular prediction.

**Table 3: The five “SuperBins” with their predictive values, and relevant statistics. The colors are used consistently throughout the paper for clarity sake.**

SuperBin	NPC	stdev	n	p-value
1	0.8156	0.3878	215,870	<< 0.0001
2	0.7380	0.4398	22,229	<< 0.0001
3	0.6902	0.4624	11,015	<< 0.0001
4	0.5297	0.4991	52,731	<< 0.0001
5	0.3396	0.4736	13,989	----

If we use the colors to remake a condensed Table 1, we can see that the SuperBins are locally consistent within the FGA. This is significant in that it suggests that, although many of the 16 bins from FGA may be statistically similar, these similarities (and differences) occur logically throughout the chart. (See Table 4.)

**Table 4: FGA color coded according to SuperBins.**

Hints	1 att.	2 att.	3 att.	4 att.	5+ att.
0	Bin 1, 0.816	Bin 2, 0.738	Bin 3, 0.677	Bin 4, 0.638	Bin 5, 0.621
1	Bin 6, 0.708	Grp A 0.701	Grp B 0.619	Grp C 0.632	Grp D 0.570
2	Bin 11 0.525	Grp E 0.581		Grp F 0.469	
3+	Bin 16 0.412	Grp E		Grp F	
BOH	attempt 1st		Grp G 0.510		
	hint 1st		Grp H, 0.340		

It is also worth noting that, although the bin numbers that went into the SuperBins may seem random, there is a pattern. SuperBin 1 consists of students who get a problem right. SB2 is populated by only students who made only one wrong attempt (and used no hints) before getting the answer right on their own. SB3 comes from three bins that represent only a small number of attempts / hint use. SB4, which incorporates the bulk of the FGA bins, is anything left, except for using the bottom-out hint, with the first action being hint use. We can now use these SuperBins as the identifier of “wrongness”.

**Table 5: Meaning (in terms of attempt and hint use) and interpretation of “wrongness” of the five SuperBins**

SuperBin	Meaning	“Wrongness”
1	Student got it right	Right
2	Student made one wrong attempt, and then got it right.	Barely wrong
3	Student used a few attempts, and 0 or 1 hint.	Partially wrong
4	Student used many attempts and/or hints.	Significantly wrong
5	Student could not start without a hint, and needed the answer.	Completely wrong

In ASSISTments, a *must* get the right answer before moving onto the next question, no matter how many attempts they make or hints they use. Clearly, a student who makes one wrong attempt and then gets the answer right with no hints demonstrates that their thinking was “less wrong” than a student who makes a series of attempts and uses many hints before getting to the correct answer. SuperBins give us a working definition of “wrongness”.

## 2.2 Impact of previous bin; 2 SuperBin (2SB) combinations (Method 2)

Looking at the sequence of students “moving” through SuperBins can help us to better understand how a student’s knowledge on a skill is changing. As we look at a student’s performance on one skill, progression through SuperBins would indicate that the student’s knowledge is improving; most humans would call this “learning.” Likewise, a student who gets an answer right, and then regresses could have “slipped” (to use KT terminology loosely).

The first (and simplest) method to look at the impact of previous SuperBins on future success is to look at two-bin combinations. That is, after the first problem, we will look at not just the SuperBin a student falls into on opportunity n, but also the SuperBin they were in on opportunity (n-1). This gives 25 different combinations. Our naming convention is (current).(previous). Thus, 2.1 is a student who is in SuperBin 2 (used one wrong attempt before getting a problem right on the second try) and was in SuperBin 1 (got the problem right on the first attempt). To use knowledge tracing language, 2.1 could represent a “slip”. Two-SuperBin code 1.2 is a student who was in SuperBin 2 and has improved to SuperBin 1. Two-SuperBin codes run from [(1.1-1.5) - (5.1-5.5)].

Table 6 (next page) illustrates the impact of the previous question’s “wrongness” on the outcome after the current question. For instance, if we compare the values of the 1.x family, we should not be surprised that the 1.1 (two correct in a row) has the highest probability of success on the next problem. However, the four other two-bin combinations (1.2-1.5) all have (statistically significantly) different predictions for the next problem. That is, how wrong a student was on the previous question can be an indicator for how likely they are to get a question right, even after they have gotten one right.

Perhaps the best demonstration of the importance of using a partial credit metric (of some sort) is to compare the predicted outcomes for 2.2 and 5.5. In both cases, the students would be marked wrong on two consecutive problems. However, a student who manages to make a mistake and then correct themselves with no

aid (twice) is (un-surprisingly) much more likely to get the next problem correct than one who needs the answer given to them (and won't even start without a hint). A student in 2.2 has a nearly 70% chance of success on the next problem, while a student in 5.5 has a mere 16.7% chance! Without looking at partial credit, they would be marked equally wrong.

**Table 6: Two SuperBin Combinations. Code 1.x refers to students who are currently in SuperBin 1 and who were in SuperBin x on the last problem. "Families" (1.x, 2.x, etc.) are color coded according to current SuperBin. Codes without decimal (bolded) are values from Table 3. The p-values compare a 2SB to the one below it.**

2SB	NPC	n	p-value
<b>1</b>	<b>0.816</b>	<b>215,870</b>	
1.1	0.840	121,317	< 0.0001
1.2	0.806	15,440	< 0.0001
1.3	0.775	7,085	< 0.0001
1.4	0.703	26,109	< 0.0001
1.5	0.655	4,317	-----
<b>2</b>	<b>0.738</b>	<b>22,229</b>	
2.1	0.783	11,421	< 0.0001
2.2	0.699	2,137	0.5322
2.3	0.688	1,016	< 0.0001
2.4	0.608	2,850	0.4391
2.5	0.587	373	-----
<b>3</b>	<b>0.690</b>	<b>11,015</b>	
3.1	0.733	4,799	< 0.0001
3.2	0.637	796	0.3058
3.3	0.611	674	0.3582
3.4	0.590	1,433	0.5120
3.5	0.567	233	-----
<b>4</b>	<b>0.530</b>	<b>52,731</b>	
4.1	0.617	19,044	< 0.0001
4.2	0.551	2,321	0.5579
4.3	0.561	1,336	< 0.0001
4.4	0.434	15,452	< 0.0001
4.5	0.380	3,263	-----
<b>5</b>	<b>0.340</b>	<b>13,989</b>	
5.1	0.540	2,155	0.8180
5.2	0.548	228	0.2658
5.3	0.491	165	< 0.0001
5.4	0.332	3,165	< 0.0001
5.5	0.167	4,429	-----

### 2.3 Impact of opportunity count on 2SB combination predictions (Method 3)

The data set we are analyzing has been limited to only up to opportunity counts of 20. (This was done to speed the analyses.) Even with 25 two-SuperBin combinations, there was enough information in the data set to run a linear regression on the effect of when a two-SuperBin combination was reached. E.g. there is a difference between students who reach 1.1 (two right in a row) on opportunity 2 versus opportunity 20.

To create this model, pivot tables in excel were used to find the average next problem correctness (NPC) on two-SuperBin combinations that fall on particular opportunities. Although not

all two-SuperBin combinations were achieved on all opportunities, there was enough information to run a linear regression. This, of course, gives an intercept and slope. The model was applied using the regression, not by using the actual calculated values.

### 2.4 Impact of 3 SuperBin (3-SB) combinations (Method 4)

Just as the state of the previous SuperBin could have an effect on future performance, it is conceivable that the SuperBin two opportunities back could have an effect. Consider the following two hypothetical students and their first three SuperBins:

**Table 7: Two hypothetical students and their SuperBin values on three questions.**

Student	Q1	Q2	Q3	Q4
Alice	SB 2	SB 1	SB 1	?
Barney	SB 5	SB 1	SB 1	?

Intuitively, we would expect Alice to have a higher probability of success on question 4 than Barney. Alice almost got the first question right, while Barney needed to use the bottom out hint (and used a hint as his first action). Although they both got questions 2 and 3 correct, their performances on question 1 are drastically different. To user models such as KT and PFA, however, they were both equally "wrong" on question 1.

To identify a 3-SuperBin combination, we will use the two-SuperBin code and add a decimal, we would have (current).(previous)(n-2) or [1.11-5.55]; this gives 125 three-SuperBin combinations. In the example above, after question 3 (and as the model predicts their correctness on question 4), Alice would be in 1.12, while Barney is in 1.15.

We are now looking at 125 combinations; some of these combinations have too few instances to have a prediction value that is reliable. 47 out of 125 3-SB combinations have fewer than 100 instances; eight combinations have 10 or fewer instances. Instead of using 125 different values (many of which would be unreliable), we will use a linear regression to approximate values for the impact of the (n-2) SuperBin. However, it is a slightly complex process.

In order to have "smooth" regressions, some assumptions are made:

- 1.) The effects can be modelled linearly. E.g., for the regression to the (n-1) SuperBin prediction = intercept + slope\*SuperBin (n-1).
- 2.) The effect of the (n-2) SuperBin value will be similar in pattern to the effect of SuperBin (n-1), but reduced in effect. (In other words, we would expect that 1.1x to follow the basic pattern of 1.x, but with a smaller change in values)
- 3.) Even though many of the three-SuperBin combinations are unreliable due to small numbers of instances, the average slope of a "family" could be used to deduce the effect size that is applied to the pattern found in assumption 2.

To create the model, five regression lines (one each for 1.x, 2.x, 3.x, 4.x, and 5.x) were created by simply using the average next problem correctness as the y-values and the decimal (previous SuperBin) as the x-values.

Next, twenty-five regressions were run for 1.1x - 5.5x. Although many of the three-SuperBin combinations were too small to be reliable, we used the average slope from a "family" (e.g. 1.3x) to adjust the effect from the two-SuperBin combination regressions. E.g., the regression lines for 1.1x - 1.5x were found and averaged.

To approximate the slopes of 1.1x-1.5x, the slopes of 1.x - 5.x were used, but multiplied by the ratio of the average (1.1x-1.5x) to the average (1.x - 5.x). Since the intercepts from (1.x-5.x) might not have the same meaning when compared to (1.1x - 1.5x), the intercepts from the three-bin regressions were left as is. Table 8 (below) shows the 2-SuperBin regressions (found using the values in Table 6), followed by the actual regression values for one of the 3-SuperBin families, and the idealized slopes.

## 2.5 First Possible Opportunity Count

Lastly, when fitting our methods (many of which would have to be some combination of the above four versions), we decided to separate SuperBins and combinations by the first available opportunity count, and all others. In our numbering scheme, we used a “dummy code” of 09 to designate that we are looking at the average of NPC for only the first available opportunity count. See next section for examples which may help.

## 2.6 Method Examples

We now arrive at the methods by which our model is applied. To see the differences between the methods, it may be useful to look at the same hypothetical sequence of SuperBins for two imaginary students and compare the different methods. (See Table 9, next page.) In all methods below, we compare “Chuck” and “Denise” and the parameters that would be used to predict their success. It’s important to note that method 1 identifies the SuperBin into which each student is placed on questions 1-4; this does not change throughout the methods.

The simplest method uses only the average NPC for all SuperBins, and pays no attention to opportunity count or SuperBin combinations. This is Method 1. This can be thought of as a simplified FGA.

**Table 8: demonstration of idealization of regression to third bin using second bin regression values.**

2 SB “family”	m	b
1.x	-0.047	0.898
2.x	-0.048	0.818
3.x	-0.038	0.741
4.x	-0.059	0.686
5.x	-0.096	0.704
3 SB “family” actual	m actual	b actual
1.1x	-0.032	0.913
1.2x	-0.031	0.845
1.3x	-0.025	0.089
1.4x	-0.034	0.778
1.5x	-0.018	0.695
3 SB “family” idealized	m idealized	b actual
1.1x	-0.023	0.913
1.2x	-0.023	0.845
1.3x	-0.018	0.089
1.4x	-0.029	0.778
1.5x	-0.047	0.695

The prediction for (e.g.) question 5 is based solely on the SuperBin value for question 4. SuperBin values are modified by a multinomial logistic regression based on skill. This gives a total number of parameters as 5 + 1/skill.

In method 2, the prediction of NPC for question 1 is based on the average value for the SuperBin, but only including values from the first opportunities. (The “dummy code” of 09 is used to indicate first opportunity only.) All questions from then on use the value for the two-bin combinations. This gives a total number of parameters of 30 + 1/skill. (Five for SBx.09, and 25 for 1.1-5.5, plus the regression to skill)

In method 3, the prediction of NPC from question 1 is based on SuperBin at first opportunity, while all others are based on the regression to opportunity count values. This gives a total number of parameters of 55 + 1/skill. (Five for SB x.09, and 50 for the intercept and slope of the 25 different two-SuperBin combinations, plus the regression to skill)

In method 4, the prediction of NPC for question 1 and 2 are based on SuperBin x.09 and 2-SuperBin combinations x.y09. For question 3 and on, the prediction is based on the linear regression to the SuperBin of (n-2). This gives a total number of parameters of 65 + 1/skill. (Five for SB x.09, 25 for 2SB combo x.y09, 25 for the intercepts, five for the slope of 1.x-5.x, and five for the slope modification parameter, plus the regression to skill). The slope and intercept in the regressions in method 4 are not the same as those in method 3.

A demonstration of the application of all four methods can be found in Table 9 below. Method 1, being simply the single SuperBin prediction identifies a “score” or “condition” for the hypothetical students. The other methods start with this information.

**Table 9: Hypothetical application of four different methods; it is important to note that the methods are different, but the results of “Chuck” and “Denise” are not. “X.09” (or “X.Y09”) is a code meaning prediction values are derived from the first available bin only. E.g. “1.09” uses only the scores from SuperBin 1 and the first opportunity.**

Method 1: SuperBin Only (“SB_1”)					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB1	SB2	SB1	SB1	...
Denise	SB5	SB3	SB1	SB2	...
Method 2: Two-SuperBin combinations (“SB_2”)					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB 1.09	2SB (2.1)	2SB (1.2)	2SB (1.1)	...
Denise	SB 5.09	2SB (3.5)	2SB (1.3)	2SB (2.1)	...
Method 3: Two-SuperBin combinations, with opportunity regression (“SB_3”)					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB 1.09	$b(2.1) + m(2.1)*2$	$b(1.2) + m(1.2)*3$	$b(1.1) + m(1.1)*4$	...
Denise	SB 5.09	$b(3.5) + m(3.5)*2$	$b(1.3) + m(1.3)*3$	$b(2.1) + m(2.1)*4$	...
Method 4: Two-SuperBin combinations, with third bin regression (“SB_4”)					
Student	Q1	Q2	Q3	Q4	Q5
Chuck	SB 1.09	2SB (2.109)	$b'(1.2) + m'(1.2)*1$	$b'(1.1) + m'(1.1)*2$	...
Denise	SB 5.09	2SB (3.509)	$b'(1.3) + m'(1.3)*5$	$b'(2.1) + m'(2.1)*3$	...

### 3. RESULTS

In order to better show methods, many of the tables that would be considered “results” are found throughout the paper. We hope this does not inconvenience the reader too much at this time.

Tables 1 through 5 show that a statistical analysis of student actions (based on next problem correctness) can simplify a complex table, while still retaining meaningful groupings of student actions. In our last paper, we argued that not only should hint use and attempt count be used in the model, but a simple action-order analysis should be included. We can point out that the regrouping process does not contradict this conclusion. Had group A not been split from group B, the model might not have fared so well.

Table 6 demonstrates that there is an effect of the previous SuperBin that will modify the prediction of the current SuperBin. For example, we can see that students in SuperBin 1 who were just in SuperBin 5 have almost a 20% (absolute) less chance of success on the next problem when compared to a student who was in SuperBin 1 twice running. This may not be too surprising, as SuperBin 1 represents getting the answer right. However, there is still a roughly 15% (absolute) difference in expected outcomes between 2SB 1.2 and 1.5. Both of these represent a student who got a problem wrong, and then got the next right. Algorithms such as KT or PFA would treat these conditions as identical.

When analyzing the 2SB combinations, the pattern is amazingly clear: the impact of wrongness does not disappear after one question, and the different levels have different (and predictable) impacts. Being in SuperBin 5 on the previous problem gives a student a worse outcome than 4; 4 is worse than 3, etc. There are only a few deviations from this pattern throughout Table 6. The p-value analysis indicates that the differences are reliable most of the time; that is, the patterns appear to be reliable, although a larger dataset is needed to state that definitively across all patterns.

One interpretation of the pattern of effect from the previous SuperBin would be that students in SuperBin 5 have more to learn than those in SuperBin 4, and that even getting the next question right is not a clear sign of having learned the knowledge component. The summary table (Table 5) gives another interpretation on this: the students in SuperBin 5 needed a hint before they even got started, and then needed the answer to finish. Clearly, these students are nowhere in the same state of learning as a student who makes one mistake and fixes their answer on their own (SB2).

This differentiation of “wrongness” demonstrates the power of looking at non-binary correctness. Perhaps the most dramatic observation is that a student who is wrong twice, but corrects themselves each time (2SB combination 2.2) is very different from a student who cannot start without a hint and cannot get to the correct answer on their own (2SB combination 5.5). To treat these two states as the same (wrong twice running) is to give up on information that can help differentiate a student who is nearly 70% likely to be correct on the next problem, versus one who as a paltry 16.7% chance (2.2 vs 5.5).

With these new predictions, we can compare predictions to other models. In Table 10, we compare the scores from RMSE, AUC, and R-squared. This shows that not only is the “SuperBin” method as valid as the FGA model (tying in two out of three metrics), taking opportunity regression (method 3) and 3-SB regression both improve on the basic SuperBins idea (method 1).

One table that a reader might be missing is one detailing the relation of 2SB to opportunity. Rather than add an eleventh table, we will summarize as: the  $R^2$ -values for the regressions ranged from 0.832 to 0.001; some are clearly not reliable. However, given the results in Table 10, we think that accounting for opportunity count by linear regression to the 2SB combinations is a worthwhile first approximation.

**Table 10: Analysis of various knowledge models. Baseline predicts the average value of the training set. For AUC, 1.0 is ideal; 0.5 is no better than random. RMSE: 0.00 is ideal; 0.5 is no better than random.  $R^2$ : 1.00 is ideal, 0.0 is no better than random.**

Method	AUC	RMSE	Rsqr
Baseline (predict mean)	0.500	0.446	0.000
PFA [11]	0.653	0.426	0.058
KT [6, 4, 10]	0.710	0.413	0.115
SOA [7, 17]	0.708	0.426	0.087
AM [15]	0.714	0.422	0.103
FGA [13]	0.715	0.400	0.128
SB method 1	0.715	0.411	0.128
SB method 3	0.726	0.407	0.142
SB method 4	0.727	0.406	0.145
Avg (methods 3 & 4)	0.728	0.406	0.145

### 4. CONCLUSIONS

The regrouping of 16 bins of the FGA into 5 “Super Bins” does not adversely affect the predictive power of the model (in two out of three metrics). In fact, by having fewer bins, we are able to look at history in a way we would not have, had we kept the 16 bins of the Fine-Grain Action model. This gives us a chance to improve on the FGA.

We can conclude that not all wrong answers<sup>2</sup> are equal, and that there is value to be gleaned from analyzing different wrong answers. The impact of “how wrong” an answer is has an effect even up to two answers later. That is, your “wrongness” two questions back can be used to make a better prediction for your next problem. (It is possible that wrongness further back could be used, but it would require a dataset that is larger by orders of magnitude.)

Not only is the combination of “wrongness” useful in making predictions, so too is the opportunity on which a student achieves a combination. That is, a student who gets the first two questions right is (usually) more likely to get the third right than a student who gets the 11<sup>th</sup> and 12<sup>th</sup> questions right is to get the 13<sup>th</sup> correct.

It is perhaps not too surprising that this method is able to outperform established models such as PFA and KT. (And we will freely admit that the previous statement is limited only to this one dataset; more research is needed to definitively make this statement.) PFA and KT use only the information in binary correctness. A new model that outperforms existing models by using additional information does not negate the previous models; it merely shows that this information is worth incorporating into models of user knowledge.

<sup>2</sup> Or, more precisely, combinations of student actions that are treated as wrong answers; actual analysis of wrong answers is left to another paper.

## 4.1 Answers to the Research Questions

1.) The bins from FGA were useful, but needed to be regrouped. Regrouping by next problem correctness (and t-test analysis) kept local and logical groupings that yield meaningful descriptions of wrongness.

2.) The level of wrongness that a student demonstrates has an effect on more than just the current question. This effect is clear and reliable on the next problem and may impact the following.

3.) Not all wrong answers are identical. Knowledge estimation models such as KT and PFA leave out “levels” of wrongness that can be used to make a more accurate prediction of student success.

The paraphrased Anna Karenina quote at the start of the paper summarizes both our hypothesis and our findings: A careful analysis of wrong answers will help improve knowledge estimation models.

## 4.2 Novel Contributions

This paper seeks to show that there is information to be gained by treating different kinds of wrong answers as different. Presented herein is a statistical method of differentiating student actions into groups of actions that represent meaningful differences in performance. Use of these groups in a knowledge modelling algorithm can improve the results of the predictions, without needing continuous values (as in [14]).

## 4.3 Future Work

Although all of the linear regressions can be considered first-order approximations, the idealization of the third bins may be perhaps only a zeroth-order. As more data becomes available, we may be able to bypass the idealization and simply use 125 different parameters that are statistically reliable. Beyond improving the results of this model, the incorporation of other models that seek to use information from incorrect answers should bolster the performance of the model(s).

## 5. ACKNOWLEDGEMENTS

Special thanks also go out to the ASSISTments team for all the work they do in harvesting the data from the system. We also acknowledge and thank funding for ASSISTments from the NSF (1316736, 1252297, 1109483, 1031398, 0742503, and 1440753), the U.S. Dept. of Ed. GAANN (P200A120238), ONR’s “STEM Grand Challenges,” and IES (R305A120125, R305C100024).

## 6. REFERENCES

- [1] Attali, Y., & Powers, D. (2010). Immediate feedback and opportunity to revise answers to open-ended questions. *Educational and Psychological Measurement*, 70(1), 22-35.
- [2] Baker, R. S., Goldstein, A. B., & Heffernan, N. T. (2010). Detecting the moment of learning. *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.
- [3] Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *Affective Computing, IEEE Transactions on*, 1(1), 18-37.

- [4] Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006). A bayes net toolkit for student modeling in intelligent tutoring systems. *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.
- [5] Campione, J. C., Brown, A. L., & Bryant, N. R. (1985). Individual differences in learning and memory. In R. J. Sternberg (Ed.). *Human abilities: An information-processing approach*, New York: W. H. Freeman. pp. 103-126.
- [6] Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4), 253-278.
- [7] Duong, H. D., Zhu, L., Wang, Y., & Heffernan, N. T. (2013). A Prediction Model Uses the Sequence of Attempts and Hints to Better Predict Knowledge: Better to Attempt the Problem First, Rather Than Ask for a Hint. *Proceedings of the 6<sup>th</sup> International Conference on Educational Data Mining*. 2013.
- [8] Grigorenko, E. L., & Sternberg, R. J. (1998). Dynamic Testing. *Psychological Bulletin*, 124, 75-111.
- [9] Hawkins, W., Heffernan, N., Wang, Y., & Baker, R. S. Extending the Assistance Model: Analyzing the Use of Assistance over Time.
- [10] Murphy, K.: Bayes Net Toolbox for Matlab. < <https://code.google.com/p/bnt/> > Accessed 4 September, 2014
- [11] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Online Submission*.
- [12] Sternberg, R.J., & Grigorenko, E.L. (2002). *Dynamic testing: The nature and measurement of learning potential*. Cambridge, England: Cambridge University Press.
- [13] Van Inwegen, E. G., Adjei, S., Wang, Y., & Heffernan, N. T. An Analysis of the Impact of Action Order on Future Performance: the Fine-Grain Action Model, LAK2015, in publication
- [14] Wang, Y., & Heffernan, N. (2013). Extending knowledge tracing to allow partial credit: using continuous versus binary nodes. *Artificial Intelligence in Education*. Springer Berlin Heidelberg.
- [15] Wang, Y., & Heffernan, N. T. (2011). The " Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *FLAIRS Conference*.
- [16] Wang, Y., Heffernan, N. T., & Beck, J. E. (2010). Representing Student Performance with Partial Credit. *EDM*.
- [17] Zhu, L., Wang, Y., & Heffernan, N. T. The Sequence of Action Model: Leveraging the Sequence of Attempts and Hints.

# Translating Head Motion into Attention - Towards Processing of Student's Body-Language

Mirko Raca  
CHILI Laboratory  
École polytechnique fédérale  
de Lausanne  
RLC D1 740, CH-1015  
Lausanne  
mirko.raca@epfl.ch

Łukasz Kidziński  
CHILI Laboratory  
École polytechnique fédérale  
de Lausanne  
RLC D1 740, CH-1015  
Lausanne  
lukasz.kidzinski@epfl.ch

Pierre Dillenbourg  
CHILI Laboratory  
École polytechnique fédérale  
de Lausanne  
RLC D1 740, CH-1015  
Lausanne  
pierre.dillenbourg@epfl.ch

## ABSTRACT

Evidence has shown that student's attention is a crucial factor for engagement and learning gain. Although it can be accurately assessed ad-hoc by an experienced teacher, continuous contact with all students in a large class is difficult to maintain and requires training for novice practitioners. We continue our previous work on investigating unobtrusive measures of body-language in order to predict student's attention during the class, and provide teachers with a support system to help them to "scale-up" to a large class.

Our work here is focused on head-motion, by which we aim to mimic large-scale gaze tracking. By using new computer vision techniques we are able to extract head poses of all students in the video-stream from the class. After defining several measures about head motion, we checked their significance and attempted to demonstrate their value by fitting a mixture model and training support vector machines (SVM) classifiers. We show that drops in attention are reflected in a decreased intensity of head movement. We were also able to reach 61.86% correct classifications of student attention on a 3-point scale.

## Keywords

computer vision, head movement, attention, classroom

## 1. INTRODUCTION

One of the early studies of attention in classrooms showed that only 46% of students pay attention during the class [4]. Later studies raised that estimation to a more optimistic but still insufficient 67% [20]. This means that in practice the teachers are lecturing half-empty classrooms, even if all chairs are occupied. How can we help the teachers learn to recognize which chairs are empty?

Processing of social cues comes natural in human-to-human communication, but still remains an object of much research and few technical applications. The ambiguity of the medium limits our attempts, but in the scenarios where body language becomes the dominant form of expression, we are inclined to dig further into the matter. One such scenario is the classroom. We argue that computer vision (CV) technologies, in combination with machine learning approaches give us tools to scale-up teacher's attention to every student in the classroom, regardless of the class size. This would provide the teachers with a timely opportunity to address lower attentive class areas and draw students into the lecture, encouraging teacher's reflection in action.

Behaviour of people in large groups is unpredictable to an observer in most situations. The overwhelming amount of information forces us to focus on few individuals who we deem as the representatives of the group, and mental effort and training are required to re-divide the attention equally among many subjects [7]. In case of a lecture, teachers are active participants, splitting their attention between personal actions, material presentation and orchestration of the whole process [8].

In this work we started from the success of eye-tracking in predicting focus and tried to generalize it to students' head movement in the classroom. Birmingham et al [3] illustrate the social aspect of gaze – given an image, people first analyse the gaze, then the head and finally the posture of the people in the image to collect information about where to focus their attention. Langton [13] showed that we combine the input from head and eyes into a single stimulus. These two observations together gave us the ground to consider head orientation as *i*) informative to other humans, and thus potentially also for our algorithms; *ii*) an approximation of human gaze on larger scales of motion.

In this paper we present our process for extracting head motion and pose features from videos of classroom audience, and our initial set of analysis of the features' quality. We will try to answer if there is a general connection between head motion and attention level? What are the features of head motion that we can use in predicting attention? How do these features change with attention levels? And finally, can we use these features to predict students attention levels?

## 2. RELATED WORK

The umbrella of affective computing [15] has been growing in the last 15 years, and expanding the domains of its application. The emerging sub-field of Social Signal Processing (SSP) [24, 25] made a major point of emphasizing that encoding human social and cultural information might raise the performance of the machine algorithms aimed at understanding behaviour (e.g. analysing large sport gathering [6]).

In case of human attention, it is attributed with the ability to modulate or enhance the selected information source according to the state and goals of the perceiver, and that the “perceiver becomes an active seeker and processor of information, able to intelligently interact with their environment” [5] and can be highly relevant in a learning environment [14]. Roda et al [19] already tried to incorporate the attention indication as one of the inputs in human-computer interaction, but early attempts in the classroom were not formulated as a technology which can be wide-spread, due to their complexity [1].

Detecting and displaying the gaze direction, as one of the key indicators of focus of attention, was shown to be both useful in making the interaction feel more natural [23], and indicative of the material comprehension [21] in on-line environments. Lacking the possibility of capturing gaze in a real-life scenario, Ba et al [2] demonstrated that we can estimate the VFOA (visual focus of attention) in meetings successfully based on the head pose. In the similar scenario Stiefelhagen et al [22] showed that head orientation contributes 68.9% in the overall gaze direction (where is the attention directed) and achieved 88.7% accuracy at determining the focus of attention. This gives us the indication that head motion has potential as a focus indicator, but it does not come without problems. Deeper exploration of head motion depicts it as an ambiguous indicator. Heylen’s overview [10] shows that head-signals are either very contextual-dependant or are complementary signal to the main information channel (usually – talking).

Our conclusion from the literature overview is that head motion has the potential as a low-resolution measurement which we can passively acquire to determine the attention level and/or direction of another person. To fully decode it we need contextual information which will be unavailable in our approach of passive/unobtrusive data collection [16]. The features we hope to find need to be positioned in the middle between measurable and context-dependant.

## 3. METHOD

Training and validation of our head detector/pose estimator pipeline was detailed in our previous work [17]. We will give a quick overview of the experiment setup and detection pipeline, and focus on the steps and problems we encountered in the later stages of data extraction.

### 3.1 Experiment design

We collected a total of 6 recorded sessions with 2 classes (demographic information shown in Table 1). Each classroom was observed with several cameras positioned above teacher’s head around the blackboard area of the classroom (camera view of the classroom is shown in Figure 1). The

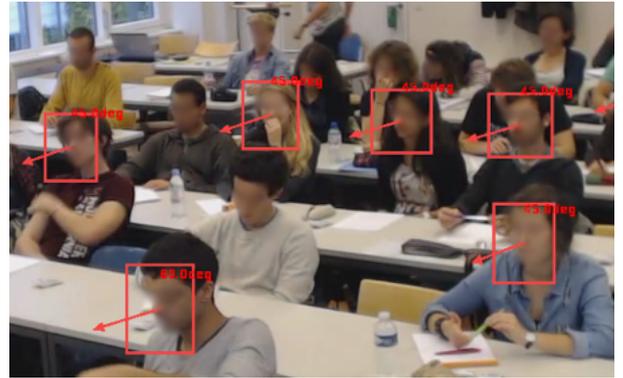


Figure 1: Examples of gaze detections, showing the classroom during the lecture.

cameras were synchronized and each student visible in the video was annotated with a unique ID (maintained over all recorded sessions) and a rectangular area of the video which the student occupies. Given that the angle of the face detected is relative to the camera viewpoint, we introduced angle offsets for each student. If a student was visible from several cameras, best quality recording was used.

Class	Size	F.ratio	Mean attend.	Sess	Cams
1	62	35.48%	39.34( $\sigma = 1.15$ )	3	5
2	43	34.88%	27.5( $\sigma = 6.55$ )	3	4

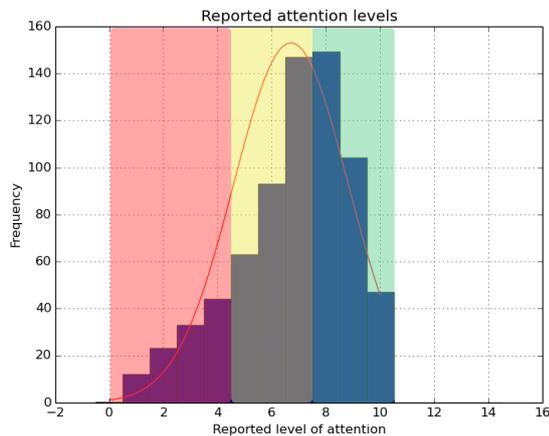
Table 1: Statistics of the two captured classes, showing the number of students, percentage of female students, attendance, number of sessions recorded and number of cameras used.

Similar to attention probing used in earlier experiments [4] we asked students to fill out the questionnaire about their attention during the class. At four different times the classes were interrupted and students recorded their attention on a Likert scale from 1–10 (details of the questionnaire design are presented in [17]). The distribution of all collected answers is shown in Figure 2. From each of the 6 processed classes we recorded 4 measurements of attention per student, associated to the time period before our interruption, duration of 7-10 minutes. In order to turn the problem into a classification one, we labelled the values of the students’ responses as *low* (reported attention 1–4), *medium* (5–7) or *high attention* (8–10), based on our observations of attention distribution (regions marked in Fig.2).

### 3.2 Video analysis

The head-pose detection and pose estimation was built on top of the part-based model for head detection published by Zhu et al [26] which was re-trained for lower resolution images and different head poses on the AFLW dataset [12]. We trained a geometrical head-pose estimator (focusing on horizontal angle or “*pan*” of the head) by using the dlib library [11]. The precision of the estimators was checked on the Pointing’04 dataset [9]. Each detection consists of the assumed rectangle of face area, estimated angle of the face (“*pan*”) and score (detector confidence).

The major problem for reaching the meaningful measure-



**Figure 2: Histogram of all reported levels of attention with the used limits to designate the *low* (red zone, <5), *medium* (yellow 5-7) and *high* (green, 8-10) levels of attention.**

ments was the instability of the detector/estimator output. The measurements were very noisy since the feature extraction step was not formulated as a tracker, which would provide temporal consistency. The second problem came from the setup itself — given the location of the cameras (around the black-board, visible in Figure 1), the subjects sit closely together. This causes a considerable amount of *i*) inter-personal occlusions and *ii*) gaps in detection and *iii*) miss-assignment of detection instances (visualized in Figure 3a).

Simple attempts to pick the best-scoring detection within the region did not yield a stable output, given that on most occasions the head of the neighbouring student would wander into the region and take over as the best detection. Fitting prior distributions (2D Gaussians) for expected head locations also did not improve the assignment, as students usually create 2 or 3 mixtures of points (depending on their sitting poses), which is indistinguishable from the case when two people occupy the given space.

Finally we settled for the formulation with labelled GMM (Gaussian Mixture Model). By taking sparsely sampled detections over time (one frame every 2 seconds) and accumulating all the detections, we depicted the overall probability of detecting faces in different positions of the camera view. The “labelled” part consists of manually specifying the relevance of each mixture in the probability, by either labelling the mixture as a specific person or miss-detection. With this we could filter-out all the irrelevant detections for a specific person by only considering detections which were assigned to one of the person-related clusters in the GMM (Figure 3b).

To improve the precision of the GMM fits, before training the model we eliminated the outlier points by thresholding the minimal number of neighbours a point needs to have in order for it to be further considered. This is possible due to the fact that the people remain in distinct positions for long periods of time, causing dense groupings of detections. The threshold was dynamically determined for each video,

by eliminating the 0.5% of points with lowest number of neighbours. The major role of the GMM filtering step was to eliminate false positives, as the clusters could not always be mapped one-to-one to an individual. Additional constraints during the GMM training phase could solve this problem.

After filtering out the miss-detections, temporal consistency was ensured by using a simplified Kalman filter approach – the next detection is expected to be in the close proximity of the previous detection. If no detections were observed within a specified radius from the previous detection, the radius is increased for the next processed frame and no detection is reported, simulating the increase in uncertainty. The major differences from the Kalman filter is the absence of motion model (the face is expected to remain at the same place) and the lack of probability propagation. This enabled us to use only the real detections and not estimates, which is relevant in order to model the heads in a bow-down position. The region growing was preferred over moving Gaussian in order to put a hard limit on the detections which can be considered.

After each processed person in the video, to make sure that the detection would not be used two times, we removed the detection after it has been assigned to a person. This turns the algorithm into a greedy approach, and making the order in which the persons are processed important. We chose to process the persons from front-to-back given that each person sitting closer to the cameras is more likely to be correctly detected. After extracting detection tracks for each person, values of the detection rectangle position and gaze angle are smoothed with a “sliding window” approach.

### 3.3 Features extracted

The input features used in our predictions were largely based on the information extracted from the cameras, but not exclusively. All features used are shown in Table 3.3. As we noted before, the time and spatial arrangement also plays significant role in the attention estimation [18], so we included the information about the distance of the student from the teacher (distance and row fields), and time of the sample within the class (period).

We tried to model the eye contact in the class with the percentage of time that we detected the student’s face in the video. Initial assumption is that this would allow us to measure the time the student spent looking down just by noting how long was the head absent. The noise in the measurement originates from the false negatives of the detector, which is dominantly influence by the distance from the camera. Even though we resorted to using zoom-lenses for the distant people in the class (which makes the measurements comparable even on the capture level to the people in the front rows), there still was a significant correlation between the row in which the student sat and percentage of time detected ( $r = -0.1867$ ,  $p = 0.009$ ), although it was weaker than the correlation with the Cartesian distance from the teacher ( $r = -0.2137$ ,  $p = 0.002$ ) which encodes width as well as depth of the classroom.

“Head travel” records the total accumulated head travel in the horizontal plane. We ignored the potential head-travel in the periods when we did not detect the face of the stu-

dent. In order to neutralize the potential influences of person’s rhythm and distance from camera, we also included a normalized version of the measure, by using all the measurements of a single person to determine the mean and scaled it with the variance of those measurements. Samples with a single measurement were excluded.

We modelled the focus of the student with 3 connected measures of stillness – number of still periods, mean duration of the still period and percentage of time spent still. Stillness was defined as periods during which the head changes are less than  $10^\circ$ , and where the head’s angle does not move away from the initial angle more than  $10^\circ$  (in order to prevent slow drifting to be classified as stillness). “Stillness periods” are defined as non-overlapping periods of minimum duration of 5 seconds, in which the stillness condition is true. From there we get the first two measures by counting the number of such periods and their mean duration. Percentage of time spent still is the ratio of time classified as being still over the duration of the attention period.

All measurements were considered per attention period and per person in order to associate the features to the labels acquired from the questionnaire. In case of regressions/ correlation tests, we also tested the correlation of the measures after the logit transformation, by first bounding the value scopes (finding minimum and maximum values for all measurements and scaling them to the 0.1 – 0.9 interval) and applying the  $\log_e\left(\frac{p}{1-p}\right)$ .

## 4. RESULTS AND DISCUSSION

### 4.1 Features

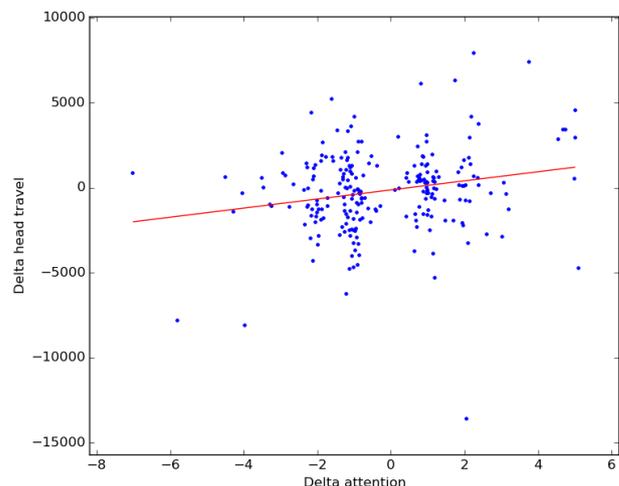
First significance tests showed the correlation between the pure attention level with the percent of time the person was detected (Pearson’s  $r = 0.1158$ ,  $p = 0.01$ , 577 samples). This can be explained with the idea that engaged students will maintain more contact with the activities in the classroom. Apart from being more visible, students head travel did not show significant difference on the overall scale. We expected this as the measurement itself can be easily affected by noisy measurements, even though we did take steps in smoothing the data.

Head travel became significant when testing its potential to measure the change in behaviour. After eliminating the individual differences with normalization of head travel, we found that positive changes in attention were reflected in increase in head travel (Pearson’s  $r = 0.21$ ,  $p < 0.01$ , 236 samples), as shown in Figure 4.

Of the measures of stillness, only “percentage of time spent still” recorded a significant, but very weak correlation (Pearson’s  $r = 0.09$ ,  $p = 0.02$ ). After comparing it with the “percentage of time detected” we found a very high and significant correlation between the two measures ( $r = 0.91$ ,  $p < 0.01$ ), which does not allow for great significance of the measure. We kept the measures for further testing.

### 4.2 Models

Next step in demonstrating the usefulness of the features was to try to predict the attention levels based on their combinations. After initial attempts with linear regression



**Figure 4: Change in normalized head travel correlated to the change in attention. Red line represents the linear fit. Pearson’s  $r = 0.21$ ,  $p < 0.01$ . Number of samples 236. Noise added for the visualization after the linear fit.**

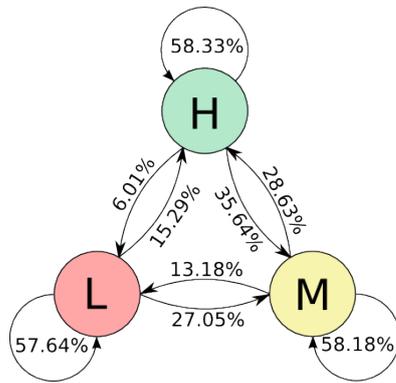
which were not successful, we switched to the mixture model. Our mixed model for *logit attention* (**A**) with *period* (**P**), *row* (**R**), *number of still periods* (**N**) and *head travel normalized* (**H**) takes form

$$L(A) = 1.061 - 0.060P - 0.128R + 0.012N - 0.035H.$$

Although its predictive power ( $R_{random}^2 = 0.54$  and  $R_{fixed}^2 = 0.05$ ) is limited, significance encourages further investigation of more advance supervised learning methods.

With that in mind, we tried an exhaustive search of all feature combinations and SVM parameters to achieve the best prediction of the three categories of “labelled attention” – *low* (100 samples), *medium* (270 samples), *high* (246 samples). Training of the classifiers was repeated in several rounds (500 iterations) with random drawing of training and testing samples, while making sure that the ratio of samples for each output category is maintained (roughly 16%, 44% and 40%). Our training procedure was based on the 80–20 split — 80% of the data used for training, and 20% data for testing the prediction of the trained classifier. To evaluate SVM parameters during the training we additionally split the 80% used for training into another 80–20 split. This gives us the final data configuration — 64–16–20 split, where 64% of the data was used for training, 16% for evaluating the SVM parameters during the training, 20% for the final evaluation of the trained classifier.

For each combination of features we iterated over the SVM parameters with sampling step of 0.1 (kernel type considered – *linear*, *polynomial*, *rbf*, and their relevant parameters). On the top scoring feature combinations we applied gradual refinement of the parameter sampling step (step size was reduced down in sequence 0.1, 0.01, 0.001 around the best scoring parameter values from the previous round). Four best scoring classifiers are given in Table 3, with the best result of 61.86% correct classifications (Cohen’s kappa 0.30)



**Figure 5: Transition probabilities between the three attention levels (*low, medium, high*).**

on the independent test set.

Our concern was that the main informative source would rely on the *Detection percentage* or *Percentage still*, the two being highly correlated. This did happen in the early training attempts, but the features are not represented in the final set of classifiers (*Detection percentage* is used in the 10th best classifier). All of the best classifiers included a similar mix of features – head motion representatives, and some indications of distance and time of the class. *Normalized head-travel measurements* and *Mean duration of still periods* appears to be the most salient feature (both used in 3 of the 4 detectors).

Even though we saw no significant correlation of attention with class period in the feature analysis, we also tested the “attention labelled” for Markov property and got highly informative transitions probabilities shown in Figure 4.2. The trend of remaining in the same state with lower possibilities of transition to neighbouring, although not directly relevant to the attention level definitely puts additional constraints on the predictions. In order integrate this knowledge into our model, the next step was to connect our SVM predictions (observational model) and temporal consistency (transition probabilities) into a Hidden Markov Model, but due to time constraints we are unable to report the results in

this publication.

## 5. CONCLUSION

The goal of this study was not only to answer questions about the link between student’s movement and attention, but also to investigate to what extent can we approximate these variables by current techniques, without manual annotation. We defined a number of head metrics that can be extracted from a video of the audience attending a class. Considering measures that are “global” in nature (not relying on specific events such as gesturing, nodding etc.) we have shown that the change in head motion usage correlates with the change in reported level of attention. We also experimentally confirmed that higher percentage of head detection mirrors higher time spent in contact with the classroom events, indicating higher attentiveness.

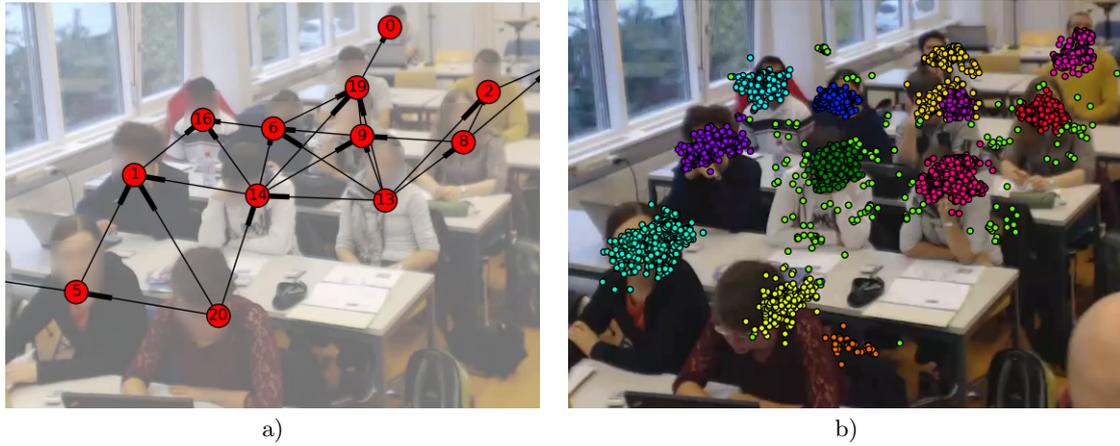
For classification tasks, we found that head measurements alone were not enough to give us definitive answers about the person’s attention. Each of the high-scoring classifiers used other contextual cues which related person’s actions to the temporal or spacial domain (e.g. class period, distance). Also, in this report we did not explore social-level cues – how the students actions are contrasted against their immediate environment or general classroom population. We have expectations that these features will provide further contextual information, which will raise the precision of predictions.

Apart from the “global” measurements, we are also looking to explore discrete gestures which can be detected with the system (e.g. nodding, yawning, turning), of which only “bowing the head down” was used at this stage, encoded within the “percentage of time detected”. The problem that we perceive is that the noise of the measurements was evident in the current setup, and that relying on the features which are more sensitive will depend on further improvements in the computer vision algorithms.

Our current conclusion is that the technology shows promise and that future investigations will bring higher accuracy and new tools to the classrooms. Our future work will try to work in parallel on finding more meaningful measures, and coordinate with the teachers to determine the best way to present the found information back to the teaching process.

## 6. REFERENCES

- [1] I. Arroyo, D. G. Cooper, W. Burleson, B. P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *AIED*, volume 200, pages 17–24, 2009.
- [2] S. O. Ba and J.-M. Odobez. Recognizing visual focus of attention from head pose in natural meetings. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):16–33, 2009.
- [3] E. Birmingham, W. F. Bischof, and A. Kingstone. Social attention and real-world scenes: The roles of action, competition and social content. *The Quarterly Journal of Experimental Psychology*, 61(7):986–998, 2008.
- [4] P. Cameron and D. Giuntoli. Consciousness sampling in the college classroom or is anybody listening?. *Intellect*, 101(2343):63–4, 1972.
- [5] M. M. Chun and J. M. Wolfe. Chapter nine visual attention. *Blackwell Handbook of Sensation and Perception*, pages 272–311, 2001.
- [6] D. Conigliaro, F. Setti, C. Bassetti, R. Ferrario, and M. Cristani. Attento: Attention observed for automated spectator crowd analysis. In *Human Behavior Understanding*, pages 102–111. Springer, 2013.
- [7] J. A. Daly and A. Suite. Classroom seating choice and teacher perceptions of students. *The Journal of Experimental Educational*, pages 64–69, 1981.
- [8] P. Dillenbourg, G. Zufferey, H. Alavi, P. Jermann, S. Do-Lenhand, Q. Bonnard, S. Cuendet, and F. Kaplan. Classroom orchestration: The third circle of usability. In *International Conference on Computer Supported Collaborative Learning Proceedings*, pages 510–517. 9th International Conference on Computer Supported Collaborative Learning, 2011.
- [9] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial structures. In *FG Net Workshop on Visual Observation of Deictic Gestures*, pages 1–9. FGnet (IST–2000–26434) Cambridge, UK, 2004.
- [10] D. Heylen. Challenges ahead: head movements and other social acts during conversations. In L. Halle, P. Wallis, S. Woods, S. Marsella, C. Pelachaud, and D. Heylen, editors, *Joint Symposium on Virtual Social Agents*, pages 45–52. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 2005. Imported from HMI.
- [11] D. E. King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [12] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.
- [13] S. R. Langton. The mutual influence of gaze and head orientation in the analysis of social attention direction. *The Quarterly Journal of Experimental Psychology: Section A*, 53(3):825–845, 2000.
- [14] S. I. Lindquist and J. P. McLean. Daydreaming and its correlates in an educational environment. *Learning and Individual Differences*, 21(2):158–167, 2011.
- [15] R. W. Picard. *Affective computing*. MIT press, 2000.
- [16] M. Raca and P. Dillenbourg. System for assessing classroom attention. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 265–269. ACM, 2013.
- [17] M. Raca and P. Dillenbourg. Holistic analysis of the classroom. In *Proceedings of the 2014 ACM workshop on Multimodal Learning Analytics Workshop and Grand Challenge*, pages 13–20. ACM, 2014.
- [18] M. Raca, R. Tormey, and P. Dillenbourg. Sleepers’ lag-study on motion and attention. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, pages 36–43. ACM, 2014.
- [19] C. Roda and J. Thomas. Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, 22(4):557–587, 2006.
- [20] J. R. Schoen. Use of consciousness sampling to study teaching methods. *The Journal of Educational Research*, 63(9):387–390, 1970.
- [21] K. Sharma, P. Jermann, and P. Dillenbourg. “with-me-ness”: A gaze-measure for students’ attention in moocs. In *International Conference Of The Learning Sciences*, number eplf-conf-201918, 2014.
- [22] R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. In *CHI’02 Extended Abstracts on Human Factors in Computing Systems*, pages 858–859. ACM, 2002.
- [23] R. Vertegaal. The gaze groupware system: mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 294–301. ACM, 1999.
- [24] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759, 2009.
- [25] A. Vinciarelli, M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D’Errico, and M. Schröder. Bridging the gap between social animal and unsocial machine: A survey of social signal processing. *Affective Computing, IEEE Transactions on*, 3(1):69–87, 2012.
- [26] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.



**Figure 3: Processing of detections.** *a)* Overlaps between subjects areas. Each graph edge shows neighbouring students areas and potential for miss-assignment of detections. *b)* All detections over the duration of the class, coloured depending on the cluster to which they were assigned.

Feature name	Description	Valid samples
Period	Period of the class (1–4), associated with the attention	776
Distance	Distance from the teacher on a Cartesian plane of the classroom	776
Row	Student’s row in the classroom	776
Detection percentage	Percentage of the recorded time that the student was detected	668
Head travel	Accumulated changes (deltas) of the head horizontal rotations over time.	496
Head travel (norm.)	Head travel normalized over the measurements of the specific person in the class.	482
Number of still periods	Number of periods (of minimal duration of 5 seconds) during which the head movement can be considered still	668
Mean still period duration	Mean duration of the still period (as defined in the previous row)	618
Still time percentage	Percentage of time within the attention period during which the head was still.	668
Attention	Reported level of attention (1–10)	715
Attention labelled	Attention reports mapped to categories <i>low</i> , <i>medium</i> , <i>high</i>	715

**Table 2: Features used in the analysis.**

Kernel	Features	Score	Cohen’s kappa
RBF( $c=1.31$ , $g=0.0211$ )	Distance, Head travel norm., Num. still periods	61.86%	0.30
RBF( $c=1.21$ , $g=0.11$ )	Period, Row, Head travel norm., Mean duration still	61.72%	0.32
RBF( $c=1.11$ , $g=0.061$ )	Head travel norm., Mean duration still	60.42%	0.28
RBF( $c=1.4$ , $g=0.04$ )	Period, Distance, Row, Mean duration still	59.23%	0.30

**Table 3: Classifier scores for predicting “attention labelled”.** Score given represent the prediction score on the 20% test sample. Parameters of the kernels are abbreviated as  $c$  - penalty for the error term;  $g$  - gamma.

# Using Visual Analytics Tool for Improving Data Comprehension

Jan Géryk  
KD Lab Faculty of Informatics  
Masaryk University  
Brno, Czech Republic  
geryk@fi.muni.cz

## ABSTRACT

The efficacy of animated data visualizations in comparison with static data visualizations is still inconclusive. Some researches resulted that the failure to find out the benefits of animations may relate to the way how they are constructed and perceived. In this paper, we present visual analytics (VA) tool which makes use of enhanced animated data visualization methods. The time is an important variable that needs to be modeled in VA. VA methods like Motion Charts show changes over time by presenting animations in two-dimensional space and by changing element appearances. The tool is primarily designed for exploratory analysis of academic analytics and supports various interactive visualization methods which enhance the Motion Charts concept. We evaluate the usefulness and the general applicability of the designed tool with a controlled experiment to assess the efficacy of the described methods. To interpret the experiment results, we utilized one-way repeated measures ANOVA.

## Keywords

Animation; motion charts; visual analytics; academic analytics; experiment.

## 1. INTRODUCTION

Higher education institutions have a strong interest in improving the quality and the efficacy of the education. In [1], hundreds of higher education executives were surveyed on their analytic needs. Authors resulted that the advanced analytics should support better decision-making, studying enrollment trends, and measuring student retention. They also pointed out that management commitment and staff skills are more important in deploying academic analytics (AA) than the technology. In [2], authors concluded that the increasing accountability requirements of educational institutions represent a key for unlocking the potentials of AA in order to effectively enhance student retention and increase graduation levels. The authors also resulted that AA facilitate creation of actionable intelligence to enhance learning and student success, however, it is highly dependent on the quality of the accountability. The authors utilized AA for developing several predictive models of student enrollment and retention, and for identifying students being at the risk. They also highlighted three critical success factors—executives committed to decision-making based on the evidence, staff members with adequate data analysis skills and the flexible and effective technology platform. However, the authors also warned that more elaborated accountability can raise several privacy issues, faculty executive's involvement, and data administration.

The principal goals can be achieved by using educational data mining methods, as emphasized in [3]. The application of data

mining (DM) techniques in higher education systems have some specific requirements not present in other areas, as pointed out in [4]. Common DM methods were developed independently of visualization techniques. However, some key ideas influenced the research in the DM field. It resulted into the recent research topic called visual analytics (VA). Google Analytics, released in 2005, made a real progress in web-based interactive analytics. In 2007, Hans Rosling presented a TED talk demonstrating the power of animations to show the story in data. In 2009, Tim O'Reilly emphasized that data analysis, visualizations, and other techniques for searching patterns in data are going to be an increasingly valuable skill set [5]. While some researches resulted that animations appeared better than static visualizations in enhancing learning, an elaborate examination of the studies revealed a lack of equivalence between animated and static visualizations in content [6]. Also, the failure to ascertain the benefits of animations in learning may also relate to the way how they are constructed, perceived, and conceptualized [7].

Visualizations are common methods used to gain a qualitative understanding of data prior to any computational analysis. By displaying animated presentations of the data and providing analysts with interactive tools for manipulating the data, visualizations allow human pattern recognition skills to contribute to the analytic process. The most commonly used statistical visualization methods (e.g. line plots, or scatter plots) generally focus on univariate or bivariate data. The methods are usually used for tasks ranging from the exploration to the confirmation of models, including the presentation of the results. However, fewer methods are available for visualizing data with more than two dimensions (e.g. motion charts or parallel coordinates), as the logical mapping of the data dimension to the screen dimension cannot be directly applied. Data exploration and interactive visualizations of multivariate data without significant dimensionality reduction remains a challenge. Animations represent a promising approach to facilitate better perception of changing values. In [6], authors pointed out that animations help to keep the viewer's attention. Visualizations and animations can also facilitate the learning process [8].

We develop visualization methods for multivariate data analyses that are adapted for academic settings. In this paper, we show the importance of data visualizations for successful understanding of complex and large data. In the next section, we examine characteristics of changes using Motion Charts (MC). Subsequently, we present several papers successfully utilizing MC for data visualization and analysis. This is followed by the elaborate description of our VA tool. Further, we conducted an empirical study with 22 participants on their data comprehension to compare the efficacy of static and animated data visualizations. We then

discuss the implications of our experiment results. Finally, we draw the conclusion from the experiment and outline future work.

## 2. EXAMINE CHARACTERISTICS OF CHANGE

Although a snapshot of the data can be beneficial, presenting changes over time provides a more sophisticated perspective. The efficacy of animated transitions for common statistic data visualizations such as bar charts and scatter plots was examined in [9]. The authors extended the theoretical model of data visualizations and introduced the taxonomy of transition types. Subsequently, they proposed design principles for creating effective transitions and illustrated the application of these principles in a dynamic visual system. Finally, they conducted two controlled experiments to assess the efficacy of various transition types, finding that animated transitions can significantly improve the visual perception. The visualization challenge posed by each of these experiments was to keep the viewer's attention during transitions. The survey resulted that viewers found animations more helpful and engaging. Unlike transition animations, which primarily help users to stay in the context, trend animations convey the meaning. While a transition animation moves from a still view to a new still view, a trend animation moves continuously between states. One early use of animations in visualization was for an algorithm animation. Kehoe et al. [10] describe a study that demonstrated that animations could help and noted that it improved the motivation of making a difficult topic more approachable. The study suggested that using animations for trend understanding could be valuable.

Animations allow knowledge discovery in complex data and make it easier to see meaningful characteristics of changes over time. To reduce the cognitive load and improve tracking accuracy, the target states of all transitioning elements should be predictable after viewing a fraction of the animation. The proper use of the acceleration should also improve the spatial and temporal predictability. A perceptual study in [11] provides evidence that animations and divergence motions are easier to understand than rotations. Animations with unpredictable motion paths or multiple simultaneously changing elements result in the increased cognitive load. Contrarily, simple transitions reduce confusion and improve clarity. In [12], authors concluded that animation stages should be long enough for accurate change tracking as well as to decrease the number of errors. However, too slow animations can disproportionately prolong the analytic phase and subsequently reduce the engagement.

Generally, effective analyses depend on the consistent and high-quality data. In [9], authors concluded that the correctly designed animations significantly improve the visual perception at both the syntactic and the semantic level. Visualizations are often engaging and attractive, but a naive approach can confuse analysts. Visualizations are just representations of the data which may or may not represent the reality. As Few pointed out in [13], computers cannot make sense of the data, only people can. The perception of animations can also be problematic because of severe issues with timing and the overall complexity that can occur during transitions as pointed out in [14]. Misleading results can be obtained if animations violate the underlying data semantics.

MC is a dynamic and interactive visualization method that enables analysts to display complex and quantitative data in an intelligible way. The dynamic refers to the animation of rich multidimensional

data changing over time. The interactive refers to dynamic interactive features which allow analysts to explore, interpret, and analyze information concealed in complex data, as presented in [15]. MC displays changes of element appearances over time by showing animations in a two-dimensional space. An element is basically a two-dimensional shape representing one object from the dataset. The variable mapping is one of the most important parts of the exploratory data analysis and no optimal method for mapping the data to variables is available. Naturally, the data mapping have a significant impact on the data comprehension and analysts should be free to choose variable mapping according to their intentions. Both the data characteristics and the investigative hypothesis influence the variable mapping.

## 3. APPLICATIONS OF MOTION CHARTS

Visualization tools represent an effective way how to make statistical data understandable to analysts, as showed in [16]. MC methods proved to be useful for data presentation and the approach was verified that can be successfully employed to show a story in data [17] or support decision making [18]. In [19], authors utilized MC for both the interpretation of results for better comprehension and the analysis when detecting topics of tweets. Several web-based data analysis tools allowing analysts to interactively explore associations, patterns, and trends in data with temporal characteristics are available. In [20], authors presented a visualization of energy statistics using an existing web-based data analysis tools, including IBM's Many Eyes, and Google Motion Charts. In [15], authors presented a Java-based infrastructure, named SOCR Motion Charts, designed for exploratory analysis of multivariate data. SOCR is developed as a Java applet using object-oriented programming language. The authors successfully validated this visualization paradigm using several publicly available datasets containing housing prices or consumer price index.

A pair of online assessments designed to measure students' computational thinking skills were presented in [21]. The assessments represent a part of a larger project that brings computational thinking into high school STEM classrooms. Each assessment included interactive tools that highlight the power of computation in the practice of the scientific and mathematical inquiry. The computational tools including Google Motion Charts used in the assessments enabled students to analyze data with dynamic visualizations and explore concepts with computational models.

Successful visualizations of language changes using the diachronic corpus data were presented in [22]. In two case studies, authors illustrated recent changes in American English. In the first study, they visualized changes in a diachronic analysis of nouns and verbs. In the second study, they showed structural changes in the behavior of complement-taking predicates. They emphasized that MC are useful for the analysis of multivariate data over time and concluded that viewing the resulting data points in separate time slices offers a proper representation of the complex linguistic changes.

In [23], authors incorporated examples using recent business and economic data series and illustrated how MC can tell dynamic stories. They utilized a database of Bureau of Labor Statistics which publishes data on inflation, prices, employment, and many other labor related subjects. For the first analysis, they utilized the data about Current Employment Statistics and presented differences between the perception of common static tables and graphs, and the

dynamic nature of MC. They concluded that the static presentation style serves well the purpose of relaying accurate and non-biased quantitative data to analysts. Subsequently, they utilized the same data, but imported them to Google Docs. By loading the Motion Charts Gadget within the spreadsheet, they generated MC and visualized several areas of Labor Statistics. They emphasized that the benefit of MC lays in displaying complex multidimensional data changing over time on a single plane with the dynamic and interactive features. Users are then allowed to easily explore, interpret, and analyze the information in the data. They concluded that MC is an excellent and interesting way how to present valuable information that may be otherwise lost in the data.

The report on the implementation of AA in a new medical school can be found in [24]. Authors pointed out that analytics address two challenges in the curriculum: providing the evidence of the appropriate curriculum coverage and assessing the student engagement during the clinical placement. The paper describes tools and approaches applied on the data gained from their web-based clinical log system. The authors utilized common data visualization methods and examined their potentials to generate important questions. They also examined the value of a flexible approach to select the tools, the need for relevant skills, and the importance of keeping the viewer's attention. Subsequently, they utilized more sophisticated visualization methods, namely MC and Tree map. Using MC, they mapped several important variables including entry date, frequency of entries, clinical problems, the level of involvement, and the level of confidence. The authors appreciated the benefits of comparison of the variation of the frequency of entries, the confidence, and the level of involvement between students. The authors concluded that AA analysis using visualizations have already been a critical enabler of educational excellence, but there is undoubtedly further potential.

A beneficial feature for better visual perception of changes in time-series analysis is presented in [25]. Initially, the author highlighted the need for effective ways to examine quantitative data that changes over time and also noted that according to several studies, more than 70 percent of all business charts display time-series information. Then, the author emphasized both the benefits and the drawbacks of common data visualization methods, namely line plots and bar charts. Subsequently, the author described issues with the time-series analysis and presented capabilities of MC. The author pointed out that patterns of changes over time can take many meaningful forms and introduced a new feature, called visual trails, specially designed for MC. The feature allows seeing the full path for each variable from one point in time to another. It can be used for overcoming visual perception limitations of MC and allows analysts to examine degree of change, shape, velocity, and direction of change. Finally, the author conducted the experiment as an evaluation of the proposed improvement.

#### 4. THE EDAIME TOOL

The preliminary version of the EDAIME tool was presented in [26]. We also described the results originated from the analysis of AA data. We utilized the data stored in the Information System of Masaryk University. The motivation to develop an enhanced version of MC was to improve its expression capabilities, as well as to facilitate analysts to depict each student or study as a central object of their interest. Moreover, the implementation enhances the number of animations that express the students' behavior during their studies more precisely. We validated usefulness of the

developed methods with a case study where we successfully utilized the capabilities of the tool for the purpose of confirming our hypothesis concerning student retention. Although, we concluded that the methods proved to be useful for analytic purposes, more adjustments are needed.

Two main challenges are addressed by the presented VA tool. EDAIME enables visualization of multivariate data and the qualitative exploration of data with temporal characteristics. The technical advantages over other implementations of MC are its flexibility and the ability to manage many animations simultaneously. The Force Layout component of D3<sup>1</sup> provides the most of the functionality behind the animations and collision detection utilized in the interactive visualization methods. Technical aspects of enhanced MC methods are elaborately described in [27]. Investigated data can be imported directly using the tool. In cases where datasets have missing values at the beginning or the end, the missing values are extrapolated from nearby data. In other cases, gaps are filled with interpolated values. For the purposes of the MC analysis, it is not important that the data are not entirely accurate.

In two figures below, two examples of our enhanced MC methods can be seen. We already utilized the methods to verify a hypothesis concerning student retention. Figure 1 depicts a snapshot of the method captured in the second semester. Each element represents a field of study and consists of a pie chart. It allows analysts to investigate another data dimension easily. Each pie chart animates a relationship between finished and unfinished studies where the green sector quantifies the complete ones, and the red sector quantifies the others. Figure 2 represents a snapshot of the second method utilized for the same dataset also captured in the second semester. The large clusters of elements represent the particular field of study consisting of small elements that represent individual students. Therefore, the size of the cluster of elements corresponds to the number of students enrolled in the particular field of study. The size of the small elements determines the number of credits gained by students in the particular semester of the study. Besides the study progress, the animations are also utilized to express the study termination, the change of the mode of study and the change of the field of study. During the animation process, dropout students turn red and fall down the chart in the semester when they left the study. The stroke-width of the elements represents states of the study and the element color represents attributes of the study.

When animations are used for exploratory analysis of unfamiliar data, analysts do not know what elements are important and play the animation hoping that something emerges. Analysts may determine areas that look promising and replay the animation several times focusing on each of the potentially interesting areas in depth. This can become an issue, perhaps making trend animations slower and more error prone for analyses. If there is a lot of variability in the data, there will be a lot of random motions, making hard to perceive trends. If there are too many elements, a clutter and counter-trends can easily intricate an observation. In the next section, we describe several user interface features that may solve some of these issues. Naturally, all methods using animations have several limitations, but appropriately designed user interface features can considerably aid visual inspection of data.

---

<sup>1</sup> <http://d3js.org/>

## 4.1 User Interface Features

The EDAIME tool offers several beneficial configurable interactive features for a more convenient analytic process. User interface features are highly customizable and allow analysts to arrange the display and variable mapping according to his or her needs. Available features include a mouse-over data display, color and plot size representation, traces, animated time plot, variable animation speed, changing of axis series, changing of axis scaling, distortion, and the support of statistical methods.

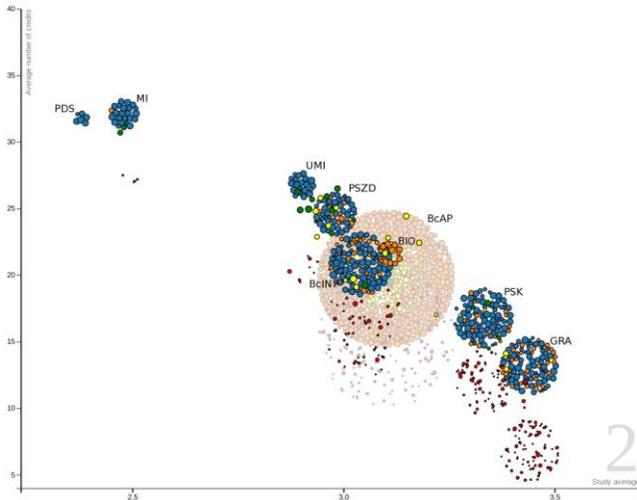


Figure 1. EDAIME snapshot: clusters of students.

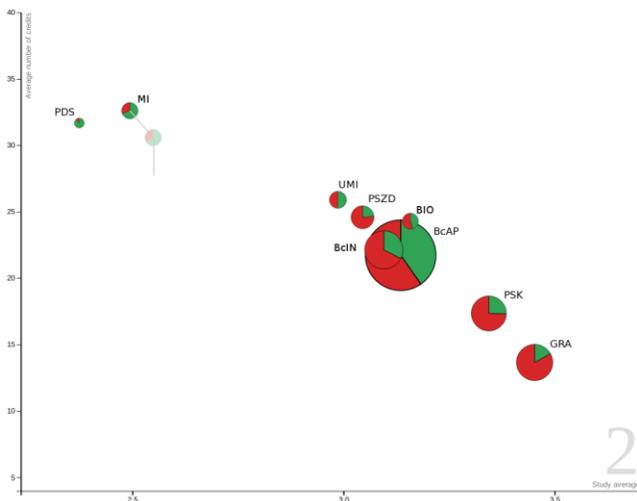


Figure 2. EDAIME snapshot: additional dimension using pie charts.

The focus-plus-context technique allows to interactively exploring objects of interest in detail while preserving the surrounding context. More precisely, if an analyst zooms in for detail, the chart area is too big to full overview. Contrarily, if an analyst zooms out the screen to see the overall chart area, the tiny but potentially important characteristics can disappear. Generally, distortions are particularly beneficial to overcome the aforementioned issues. The circular distortion magnifies the area around the mouse pointer, while leaving the chart area unaffected for the context. This

distortion is useful especially to distinguish individual elements in a cluster. However, the area near the circumference of the elements is then compressed. Therefore, it is not suitable for representing quantitative values. However, a function which magnifies the details continuously in order to avoid such local errors exists. It applies the distortion to each dimension separately which results in Cartesian distortion. If elements overlap each other during the animation, it will be more difficult to track their paths. Using the jitter feature, a better visual perception of data can be obtained by adding small random quantities to all elements' values before displaying them. As mentioned earlier, it is not important that the data are not entirely accurate for the purposes of a trend analysis.

Regardless of the power of a human brain, a memory is limited. It is difficult to reconstruct the past events from a memory, to recapture the sequence of events and details of each moment. The tool provides analysts with the ability to select particular elements and show a trace for each of the selected elements as it progresses. This is particularly useful in verifying apparent anomalies noticed during an animation. The traces show elements at each location and sizes for each time point. The traces are then connected with edges to help clarify their sequences. Analysts can observe any interesting element while the previous states are still fresh in their memory. Anomalies emerge and can be examined even without animations, so analyses may be faster and less error prone. Points that move continuously through a range of values appear as clear trends. One key challenge must be addressed in the design of this view. The trend line direction must be made visually expressive, because there is no animation to indicate the direction. We solved this problem by using element transparency, fading from mostly transparent in the earliest elements to mostly opaque in the latest elements in the sequence. In order to perceive the flow direction even for smaller elements we employed the same approach with lines connecting the elements. In addition, it was necessary to render larger bubbles first to avoid occluding smaller bubbles. As described in [25], traces are particularly useful to reveal the nature of change and can help to examine the magnitude, shape, velocity, and direction of changes.

The support of statistical methods is also useful for examining the nature of change. The statistics provide simple summaries that form the basis of the initial description of the data and also serve as a part of a more extensive analysis. We implemented several measures that are commonly used to describe a dataset, i.e. measures of central tendency or measures of variability. The measures may be beneficial when identifying meaningful data characteristics of changes over time. We utilized both the univariate and the bivariate statistical methods. Input parameters for statistical methods consist of investigated MC variables. When an animation is running, each statistical measure is computed for every element on the background. Any combination of measure and variable can be selected using the user interface. The list of univariate measures includes coefficient of variation, skewness, mean, variance, standard deviation, median absolute deviation, median, geometric mean, and interquartile range. The mouse-click event on any element will extract an interactive HTML table on the right side of the chart area. The table consists of the measure computed for every element sorted in the descending order of the specified variable. If analysts select a row, the corresponding element will be highlighted. More precisely, the other elements are either transparent or hidden. Bivariate measures can be applied to any pair of variables. The list of bivariate measures includes sample covariance, sample correlation, and paired t-test.

The layout of the EDAIME user interface is presented in Figure 3. Using control, analysts can pause and advance the animation or change the speed. The Play, Pause, and Restart buttons are situated in the upper right corner next to the chart area. Above the buttons, the time slider is situated. Analysts can grab the time slider control to adjust the playback speed. Traces control is situated beneath the control buttons and it allows selecting elements of the interest to show their traces. This makes the selected elements more distinguishable and solves clutter issues.

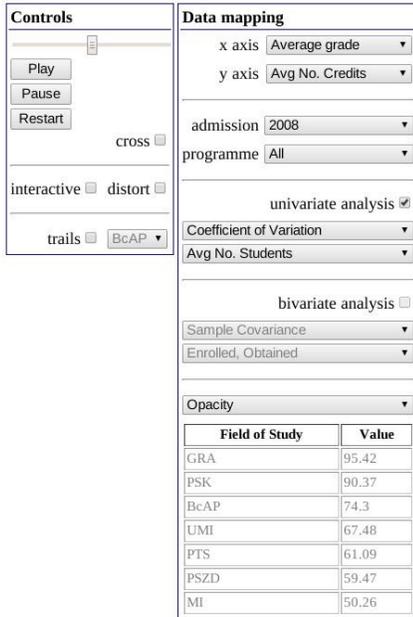


Figure 3. The EDAIME user interface layout.

## 5. EXPERIMENTATION

Any quantitative research of AA also requires a preliminary exploratory data analysis. Though useful, MC involves several drawbacks in comparison with common data visualization methods. Thus, empirical data is needed to evaluate its actual usability and efficacy.

In this section, we describe the experiment for the purpose of evaluating the efficacy of the enhanced MC methods implemented in EDAIME. We present the results including a detailed discussion. Twenty-two subjects (9 females, 13 males) with the average age of 31.6 (SD = 6.8) participated in our experiment. The participants ranged from 24 to 46 years of age. All participants came from professions requiring the use of data visualizations, including college students, analysts, and administrators. The experiment was conducted using standard desktop PCs. All subjects performed the experiment on an Intel Core i3 PC with 4 GB of RAM running Windows 7 or Fedora Core 20. Each PC had a 24" LCD screen running at the resolution of 1920 x 1080. We prefer Chrome as a web browser as it excellently supports HTML5 and CSS3 standards.

We performed a study to validate the usefulness and the general applicability of the enhanced version of MC in comparison with common data visualization methods when employed to analyze study related data. The experiment used a 4 (visualization) x 2 (size) within-subjects design. The visualizations varied between the static

and the animated methods. The static methods were represented by line plots (LP) and scatter plots (SP) which were generated for each semester. The animated methods were represented by the standard MC with the basic user interface (BMC) and the enhanced MC with advanced user interface features (EMC) described in the previous section. The size of datasets varied between small and large ones with the threshold of 500 elements. For the experiment, we utilized study related data about students admitted to bachelor studies of the Faculty of Informatics Masaryk University between the years of 2006 and 2012.

### 5.1 Hypotheses and Tasks

We designed the experiment to address the following three hypotheses:

- H1. BMC methods will be less effective than static methods when used for small datasets, and more effective when used for large datasets. In other words, the participants will be (a) faster and (b) make fewer errors when analyzing large datasets using BMC methods.
- H2. EMC will be more effective than the other methods for all datasets. In other words, the participants will be (a) faster and (b) make fewer errors when using EMC methods for all dataset sizes.
- H3. The participants will be more effective with small datasets than with large datasets. In other words, the participants will be (a) faster and (b) make fewer errors when analyzing small datasets.

In each trial, the participants completed 16 tasks, each with 1 to 5 required answers. Each task had students' IDs as the answer. Several questions have more correct answers than requested. The participants were asked to proceed as quickly and accurately as possible. In order to reduce learning effects, the participants were told to make use of as many practice trials as they needed. We also instructed them to practice until they had reached the desired performance level. Moreover, the participants had access to the tool several days before the experiment.

Sample of tasks:

- Select 4 students whose rate of enrolled credits was faster than their rate of obtained credits.
- Which student had the most significant decrease of the average grade?
- Select 5 students with the significant increase of the number of credits.
- Select 3 students whose average grade increased first and decreased later.
- Which student had the most significant increase in the number enrolled credits?

The participants selected answers by selecting student IDs in legend box located in the upper right from the chart area. In order to complete the task, two buttons can be used—either "OK" button to confirm the participant's choice or "Skip Question" button to proceed to the next task without saving the answer. There was no time limit during the experiment. For each task, the order of the datasets was fixed with the smaller ones first. This also allowed the participants to build their skills as they proceeded.

## 5.2 Study Method

The experiment used a 4 (visualization) x 2 (size) within-subjects design. Each experiment block was preceded with a training session in which we showed the subjects the correct answers after they confirmed it to allow participants to get familiarized with the settings and UI. It was followed by 16 tasks (8 small dataset tasks and 8 large dataset tasks in this order). After that, the subjects completed survey with questions specific for the visualization. Each block lasted about 2 hours. The subjects were screened to ensure that they were not color-blind and understood common data visualization methods. We also attempted to balance gender. The study results are divided into three sections: accuracy, completion time, and subjective preferences. To test for significant effects, we conducted repeated measures analysis of variance (ANOVA). Only significant results are reported. Post-hoc analyses were performed by using the Bonferroni technique.

## 5.3 Accuracy

Since some of the tasks required multiple answers, accuracy was calculated as a percentage of the correct answers. Thus, when a subject selected only three correct answers from five, we calculated the answer as 60 % accurate rather than an incorrect answer. The analysis revealed several significant accuracy results at the .05 level. The type of visualization had a statistically significant effect on the accuracy for large datasets ( $F(1.930, 40.535) = 25.655, p < 0.001$ ). Figure 4 illustrates graph of the mean accuracy of visualizations for large datasets including error bars that show the 95% confidence interval. Pair-wise comparison of the visualizations found significant differences showing that both animated methods were significantly more accurate than the static methods. EMC was more accurate than LP ( $p = 0.001$ ). EMC was also more accurate than the BMC ( $p < 0.001$ ). LP were more accurate than SP ( $p = 0.016$ ). For small datasets, visualizations were not statistically distinguishable, except for SP which had lower accuracy than other methods. Also, the subjects were more accurate with small datasets ( $F(1, 21) = 38.679, p < 0.001$ ) as can be seen in Figure 5.

## 5.4 Task Completion Time

An answer was considered to be incorrect if none of the correct answers was provided. In terms of time to task completion, we also observed a statistically significant effect ( $F(1.764, 37.044) = 43.875, p < 0.001$ ). Post-hoc tests revealed that BMC was the slowest for both dataset sizes. For large datasets, the LP was faster than the EMC ( $p < 0.001$ ). EMC and SP were not statistically distinguishable. The mean time for LP was 76.36 seconds compared to 85.95 seconds for the EMC—about 13% slower, 88.59 seconds for the SP—about 16% slower, and 91.64 seconds for the BMC—about 20% slower. For small datasets, static methods were significantly faster than animated. Pair-wise comparison of the visualizations found significant differences between all of them except for EMC and SP. LP were the fastest for all datasets. EMC was slower than the LP ( $p < 0.001$ ) and faster than the BMC ( $p < 0.017$ ). The mean time for BMC was 70.18 seconds compared to 67.6 seconds for the SP—about 3% faster, 66.55 seconds for the EMC—about 6% faster, and 61.36 seconds for the LP—about 14% faster.

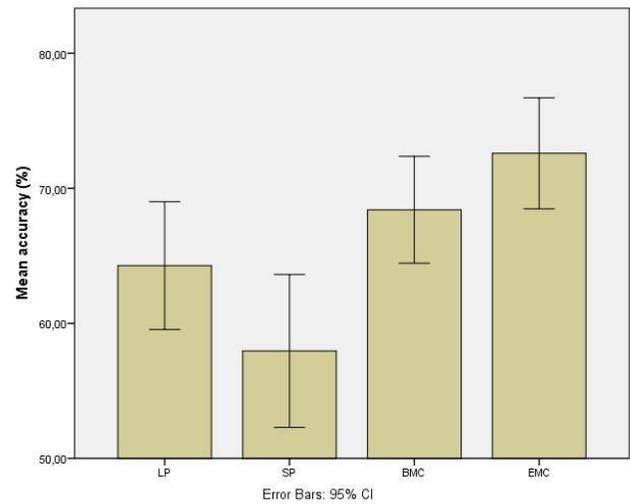


Figure 4. Mean accuracy of answers per visualization method.

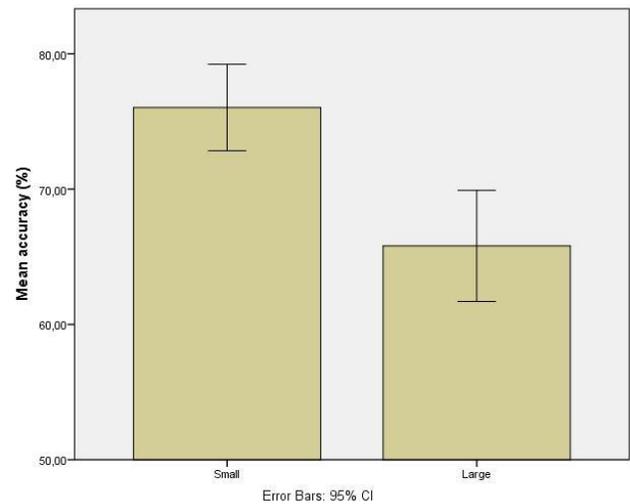


Figure 5. Mean accuracy of answers per dataset size.

## 5.5 Subjective Preferences

For each experiment block, the subjects completed a survey where the subjects assessed their preferences regarding analyses. The subjects rated the static and animated methods on a ten-point Likert scale (1 = strongly disagree, 10 = strongly agree). Using RM-ANOVA, we revealed statistically significant effects ( $F(1.696, 35.611) = 80.1332, p < 0.001$ ). Post-hoc analysis found that EMC was significantly more helpful than other methods, more precisely BMC ( $p < 0.001$ ) and LP ( $p < 0.001$ ). The obtained results are presented in Table 1, indicating the resulted mean values of the preferences for each question.

The significant differences indicate that animated methods were judged to be more helpful than the static methods. The subjects significantly preferred the LP to use for small datasets. However, animated methods were judged to be more beneficial than static methods for large datasets ( $p < 0.001$ ). The results also showed that

animated methods were more entertaining and interesting than the static methods ( $p < 0.001$ ).

**Table 1. The resulted mean values of the preferences.**

	LP	SP	BMC	EMC
The visualization was helpful in answering the questions.	5.41	4.27	6.86	7.55
I found this visualization entertaining and interesting.	5.36	5.14	7.14	8.05
I prefer visualization for small datasets.	6.70	4.41	5.59	5.82
I prefer visualization for large datasets.	5.90	5.18	7.41	8.32

## 6. DISCUSSION

Our first hypothesis (H1) was that BMC would outperform both the static methods for large datasets and will be less effective when used for small dataset. This hypothesis was confirmed only partially. BMC methods were more accurate than the static methods, but contrary to the hypothesis, the static methods proved to achieve better speed than the BMC for the both dataset sizes. Moreover, the methods were not statistically distinguishable in terms of accuracy for small datasets. The second hypothesis (H2) expected that EMC will be more effective than the other methods for all dataset sizes. The hypothesis was only partially confirmed as well. EMC was the most accurate method for all datasets. Contrary to the hypothesis, LP was the fastest method for all datasets. We also hypothesized that the accuracy will be higher for smaller datasets (H3). The hypothesis H3.a was supported, because the subjects were faster with small datasets. The mean time for large datasets was 85.64 seconds and for small datasets was 66.42 seconds. The hypothesis H3.b was also supported, because the subjects committed fewer errors with small datasets when compared with large datasets. Generally, the accuracy is the issue for static visualizations when large datasets were employed.

The EDAIME tool facilitates users to utilize the enhanced MC methods with advanced interactive features. After the experiment, multiple subjects reported that they make use of advanced user interface features and spent a lot of time exploring the data during the practice trials. In the final discussion, the several subjects reported that the animations were entertaining and interesting. Contrarily, several subjects reported that for large datasets as the number of elements rose they experienced increasing difficulty to identify and remember the element of their interest that they were following and without user interface features it would be hard to handle it. The overall accuracy was quite low in the study with average about 70%. However, only three questions were skipped.

The study supports the intuition that using animations in analysis requires convenient interactive tools to support effective use. The study suggests that EMC leads to fewer errors. Also, the subjects found MC methods to be more entertaining and exciting. They slightly preferred it to the static method. The evidence from the study indicates that the animations were more effective at building the subjects' comprehension of large datasets. However, the simplicity of static methods was more effective for small datasets. These observations are consistent with the verbal reports in which

the subjects refused to abandon the static visual methods generally. This finding illustrates that interest in animations does not preclude the subjects' appreciation of common methods. Overall, the participants would prefer to utilize both types of visual methods. Results supported the thoughts that MC does not represent a replacement of common statistic data visualizations but a powerful addition.

## 7. CONCLUSION AND FUTURE WORK

Commonly used static methods have principal limitations in terms of the volume and the complexity of the processed data. Animations are substantially transparent techniques that can present a good overview of the complex and large data. MC presents multiple elements and dimensions of the data on a single two-dimensional plane. The main contribution lies in enabling critical questions about data relationships and characteristics.

In the EDAIME tool, we enhanced the MC concept and expanded it to be more suitable for AA analyses. We also developed an intuitive, yet powerful, user interface that provides analysts with instantaneous control of MC properties and data configuration, along with several customization options to increase the efficacy of the exploration process. The tool provides a smart, convenient, and visually appealing way to identify potential correlations between different variables. We validate the usefulness and the general applicability of the designed tool with the experiment to assess the efficacy of the described methods in comparison with visual static methods.

The study suggests that animated methods lead to fewer errors for the large datasets. Also, the subjects find MC to be more entertaining and interesting. The entertainment value probably contributes to the efficacy of the animation, because it serves to hold the subjects' attention. This fact can be useful for the purpose of designing methods in learning settings. The more entertaining a method is, the easier it is to concentrate on the process and the more information can be acquired. The study also indicates that we need to appropriately adjust analytic tools when we begin to process time-varying, high-dimensional data. Especially, we need to focus on user interface features.

The current limitations of the tool are predominantly originated in the use of HTML5 standard, because there are still serious performance problems in several web browsers. Thus, only a certain number (generally less than 1000) of data points may be effectively visualized using animations. Features enabling effective data manipulation are essential. The additional representation of the data using enhanced MC methods gives analysts more possibilities in exploring the data.

We plan to create the synergy of EDAIME animated methods with common DM methods to follow the VA principle more precisely. We already implemented a standalone EDAIME method utilizing decision tree algorithm providing visual representation. We prefer decision trees because of their clarity and simplicity to comprehend.

## 8. ACKNOWLEDGMENTS

We thank all colleagues of IS MU development team and Knowledge Discovery Lab for their assistance. This work has been partially supported by Faculty of Informatics, Masaryk University.

## 9. REFERENCES

- [1] Goldstein, P. J. 2005. Academic analytics: The uses of management information and technology in higher education. EDUCAUSE. Retrieved from <https://net.educause.edu/ir/library/pdf/ers0508/rs/ers0508w.pdf>.
- [2] Campbell, J. P., DeBlois, P. B., and Oblinger, D. G. 2007. Academic analytics: A new tool for a new era. *EDUCAUSE Review*, 40-57.
- [3] Romero, C. and Ventura, S. 2013. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12-27. DOI: <http://dx.doi.org/10.1002/widm.1075>.
- [4] Delavari, N., Phon-Amnuaisuk, S., and Beikzadeh, M. R. 2008. Data Mining Application in Higher Learning Institutions. *Informatics in Education*, 31-54.
- [5] O'Reilly, T. and Battelle, J. 2009. Web squared: Web 2.0 five years on. DOI: <http://dx.doi.org/10.4304/jait.2.4.204-216>.
- [6] Tversky, B., Morrison, J. B., and Betrancourt, M. 2002. Animation: Can It Facilitate?. *International Journal Human-Computer Studies*, 247-262. DOI: <http://dx.doi.org/10.1006/ijhc.2002.1017>.
- [7] Margaret, S., Chan, J., and Black, B. 2005. When can animation improve learning?. Some implications on human computer interaction and learning. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 933-938.
- [8] Le, D.-T. 2013. Bringing Data to Life into an Introductory Statistics Course with Gapminder. *Teaching Statistics*, 114-122. DOI: <http://dx.doi.org/10.1111/test.12015>.
- [9] Heer, J. and Robertson, G. 2007. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 1240-1247. DOI: <http://dx.doi.org/10.1109/TVCG.2007.70539>.
- [10] Kehoe, C., Stasko, J., and Taylor, A. 2001. Rethinking the Evaluation of Algorithm Animations as Learning Aids: An Observational Study. *International Journal of Human-Computer Studies*, 265-284. DOI: <http://dx.doi.org/10.1006/ijhc.2000.0409>.
- [11] Bertamini, M. and Proffitt, D. R. 2000. Hierarchical motion organization in random dot configurations. *Journal of Experimental Psychology: Human Perception and Performance*, 1371-86.
- [12] Robertson, G., Cameron, K., Czerwinski, M., and Robbins, D. 2002. Animated Visualization of Multiple Intersecting Hierarchies. *Information Visualization*, 50-65.
- [13] Few, S. 2009. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press. DOI: <http://dx.doi.org/10.1080/10543401003641225>.
- [14] Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., and Ramos, G. 2006. Phosphor: explaining transitions in the user interface using afterglow effects. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. DOI: <http://dx.doi.org/10.1145/1166253.1166280>.
- [15] Al-Aziz, J., Christou, N., and Dinov, I. D. 2010. SOCR Motion Charts: an efficient, open-source, interactive and dynamic applet for visualizing longitudinal multivariate data. *Journal of Statistics Education*, 18(3), 1-29.
- [16] Grossenbacher, A. 2008. The globalisation of statistical content. *Statistical Journal of the IAOS. Journal of the International Association for Official Statistics*, 133-144.
- [17] Baldwin, J. and Damian, D. 2013. Tool usage within a globally distributed software development course and implications for teaching. *Collaborative Teaching of Globally Distributed Software Development*, 15-19. DOI: [10.1109/CTGSD.2013.6635240](http://dx.doi.org/10.1109/CTGSD.2013.6635240).
- [18] Sultan, T., Khedr, A., Nasr, M., and Abdou, R. 2013. A Proposed Integrated Approach for BI and GIS in Health Sector to Support Decision Makers. *Editorial Preface*.
- [19] Yoon, S., Elhadad, N., and Bakken, S. 2013. A Practical Approach for Content Mining of Tweets. *American journal of preventive medicine*, 122-129. DOI: <http://dx.doi.org/10.1016/j.amepre.2013.02.025>.
- [20] Vermeylen, J. 2008. *Visualizing Energy Data Using Web-Based Applications*. American Geophysical Union.
- [21] Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., and Wilensky, U. 2014. Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms. *Intelligent Technologies for Interactive Entertainment*, 22-25. DOI: [http://dx.doi.org/10.1007/978-3-319-08189-2\\_3](http://dx.doi.org/10.1007/978-3-319-08189-2_3).
- [22] Hilpert, M. 2011. Dynamic visualizations of language change: Motion Charts on the basis of bivariate and multivariate data from diachronic corpora. *International Journal of Corpus Linguistics*, 435-461. DOI: <http://dx.doi.org/10.1075/ijcl.16.4.01hil>.
- [23] Battista, V. and Cheng, E. 2011. Motion Charts: Telling Stories with Statistics. *JSM Proceedings, Statistical Computing Section*, 4473-4483.
- [24] Olmos, M. and Corrin, L. 2012. Academic analytics in a medical curriculum: enabling educational excellence. *Australasian Journal of Educational Technology*, 1-15.
- [25] Few, S. 2007. *Visualizing Change: An Innovation in Time-Series Analysis*. In *Visual Business Intelligence Newsletter*, White paper SAS.
- [26] Geryk, J. and Popelinsky, L. 2014. Analysis of Student Retention and Drop-out using Visual Analytics. In *Proceedings of the 7th International Conference on Educational Data Mining*. International Educational Data Mining Society, 331-332.
- [27] Geryk, J. and Popelinsky, L. 2014. Visual Analytics for Increasing Efficiency of Higher Education Institutions. In *Proceedings of the 6th Workshop on Applications of Knowledge-Based Technologies in Business*, 117-127. DOI: <http://dx.doi.org/10.1007/978-3-319-11460-6>.

# Data-driven Proficiency Profiling

Behrooz Mostafavi  
Department of Computer  
Science  
North Carolina State  
University  
Raleigh, NC 27695  
bzmstaf@ncsu.edu

Zhongxiu Liu  
Department of Computer  
Science  
North Carolina State  
University  
Raleigh, NC 27695  
zliu24@ncsu.edu

Tiffany Barnes  
Department of Computer  
Science  
North Carolina State  
University  
Raleigh, NC 27695  
tmbarnes@ncsu.edu

## ABSTRACT

Deep Thought is a logic tutor where students practice constructing deductive logic proofs. Within Deep Thought is a data-driven mastery learning system (DDML), which calculates student proficiency based on rule scores weighted by expert-decided weights in order to assign problem sets of appropriate difficulty. In this study, we designed and tested a data-driven proficiency profiler (DDPP) method in order to calculate student proficiency without expert involvement. The DDPP determines student proficiency by comparing relevant student rule scores to previous students who behaved similarly in the tutor and successfully completed it. This method was compared to the original DDML method, proficiency based on average rule scores, and proficiency based on minimum rule scores. Our testing has shown that while the DDPP has the potential to accurately calculate student proficiency, more data is required to improve it.

## Keywords

Data-driven, Tutoring system, Student classification

## 1. INTRODUCTION

Data-driven methods, methods where each step and calculation is based on analyzing a set of historical data, have been used to great effect to improve individualized computer instruction. They have been used in intelligent tutoring systems to accurately predict student behavior and improve learning outcomes. In contrast to individualized tutoring systems based on developing complex and context specific models of behavior, data-driven systems reduce the need for expert involvement to design the system, and can potentially adapt to new users without refinement of a behavioral model. This is because data-driven systems analyze previous student data in order to model student behavior and determine the best course of outcome in the tutor. Therefore, developing a data-driven intelligent tutoring system is based on gathering data, and developing the methods the system uses to analyze and react to student behavior.

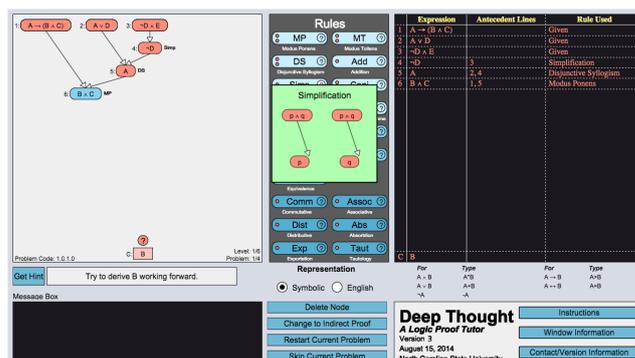
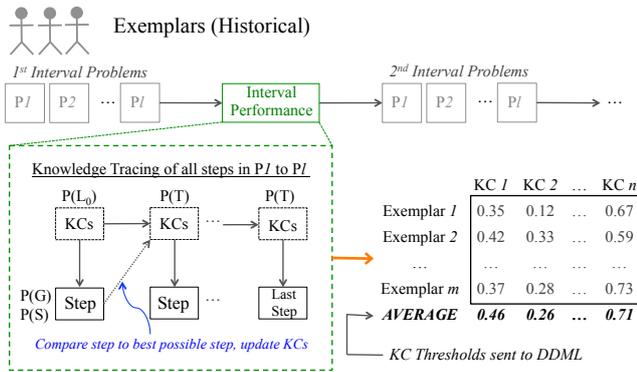


Figure 1: The Deep Thought DT3 logic tutor. Students apply logic rules (axioms) to premises to derive new statements until the conclusion (at the bottom) is justified. The right window displays the proof in standard list format.

We have been incrementally augmenting the Deep Thought logic tutor (Fig. 1) with data-driven methods for formative feedback and problem selection to improve student learning and reduce tutor dropout. Our long term goal is to create an intelligent tutor for logic proof construction that is fully data-driven and can adapt to students learning logic with varying curricular requirements without the need for further expert input. To this end, the next step in our work is to replace the expert-authored assessment parameters built into our problem selection system with a data-driven proficiency calculation that approximates the original system's performance.

Deep Thought utilizes a data-driven mastery learning system (DDML) consisting of 6 strictly ordered levels of proof problems. Each level is split into a higher proficiency track with a lower number of complex problems, and a lower proficiency track with a greater number of simpler problems. The first level of problems are the same for all students, and are used to estimate their initial proficiency. Proficiency is calculated using the knowledge tracing of all rule-application actions taken in the tutor. These action scores are compared to the average score thresholds of corresponding problems solved by past *exemplars* – students who have successfully completed the entire tutor, and have therefore demonstrated sufficient proficiency in the subject matter (Fig. 2).



**Figure 2: The DDML’s threshold builder. Knowledge components (KCs) for each exemplar are updated using action steps from an interval set of tutor problems. The KC score averages at each interval are used as thresholds in the DDML system.**

The difference between each action score minus its threshold is weighted by the expert-decided *priorities* of those actions within the level (Eq. 1). The sign of the resulting score determines placement in either the higher (+) or lower (−) proficiency track. On each subsequent level the system will first estimate a student’s proficiency and then assign them to the higher or lower proficiency track based upon their prior performance. This system was shown to increase student completion and reduce tutor dropout over unordered and hint-based versions of Deep Thought [10].

Level  $l$  End Proficiency =

$$\text{sign} \left[ \sum_{i=rule_0}^{rule_n} (scoreSign_{l,i} \times rulePriority_{l,i}) \right] \quad (1)$$

Since the current DDML system uses expert-decided priorities for each of the rule application actions when calculating a student’s proficiency, any new problems or levels added to the system will require expert involvement to determine which rules were prioritized in each new or altered level. This paper describes a study to develop a data-driven method of determining student proficiency that can replace the current expert-decided rule priorities in Deep Thought. This Data-driven Proficiency Profiler (DDPP) uses the clustering of exemplar scores at each level interval for each rule, weighted by primary component importance, to classify exemplars into *types* of student progress through the tutor. New students using the tutor will be assigned to a proficiency track based on comparison to existing types.

The DDPP method is compared alongside proficiency calculations using the minimum rule scores and average rule scores of exemplars, also weighted by primary component importance, to see how these methods compare to each other and to the expert authored system. We hypothesize that the DDPP will perform more accurately than the minimum or average methods of student proficiency classification. This would allow Deep Thought to be used in other classrooms where the pedagogical method and problem-solving ability of the class may be disparate from the current exemplar data

from Deep Thought.

Our results show that proficiency calculation using average rule scores performs more accurately than proficiency calculated based on minimum rule scores. In addition, the DDPP method performs more accurately than the average method in some parts of the tutor, while it is less accurate in other parts. Unfortunately, the DDPP system does not yet reach the accuracy of the original system overall in calculating student proficiency. We conclude that more data is required in order for the DDPP to properly approximate the accuracy of the original system’s proficiency calculation.

## 2. RELATED WORK

### 2.1 Data-driven Tutoring

An early example of a data-driven intelligent tutor is the Cognitive Algebra Tutor[12]. Here the authors introduce an algebra tutor which models student behavior based on the cognitive theory ACT-R and student data gathered from several previous studies. The Cognitive Algebra Tutor was several years and studies into development at this time, and the result is an example of a mostly-realized data-driven tutor. The tutor as it stood improved student performance, and the authors noted that although it over-predicted student performance, it would be improved the more data was collected. However, this system still took a long time and a great deal of expert involvement to design and improve. Conversely, developing a data-driven method of student assessment would reduce this time and effort, since it would be based on analyzing previous data rather than developing and improving on a cognitive model.

Later analyses on the potential benefits, and recommendations, for using data-driven methods to develop intelligent tutoring systems have focused on improving the modelling of student behavior rather than using data to improve on student assessment. Koedinger et al[7] give a very detailed overview on developing data-driven intelligent tutoring systems, and techniques for incorporating data in a useful way. They discuss optimizing the cognitive model using learning factors analysis; fitting statistical models to individual students; modeling student mood and engagement by modeling off-task behaviors, careless errors, and mood; and improving how the tutor selects actions for the student via MDP or POMDP. In a later work[8] the authors compare and contrast current data-driven methods for intelligent tutoring and discuss the potential for these methods to improve MOOCs. They go over the success cases for using data to improve tutors and coursework, in particular cognitive task analysis.

There have been several recent studies that demonstrate the potential for data driven methods to result in tutors that more accurately assess student performance and react to student behavior. Lee and Brunskill[9] examined the benefits and drawbacks to basing model parameters on existing data from individual students in comparison to data from an entire population, specifically as it pertained to the number of practice opportunities a student would require (estimated) to master a skill. The authors estimated that using individualized parameters would reduce the number of practice opportunities a student would need to master a skill. Gonzalez et al.[4] demonstrated a data-driven model which au-

tomatically generated a cognitive and learning model based on previous student data in order to discover what skills students learn at any given time, and when they use skills they have learned. The resulting model predicted student behavior without the aid of previous domain knowledge and performed comparably to a published model.

Data-driven intelligent tutors not only have the potential to more accurately predict student behavior, but interpret why it occurs. For instance, Elmadani et al. [2] proposed using data-driven techniques to detect student errors that occur due to genuine misunderstanding of the concepts (misconception detection). They processed their data using FP-Growth in order to build a set of frequent itemsets which represented the possible misconceptions students could make. The authors were able to detect several misconceptions based on the resulting itemsets of student actions. Fancsali[3] used data-driven methods to detect behaviors that usually detract from a student's experience with an ITS (off-task behavior, gaming the system, etc).

## 2.2 Cluster-based Classification

Cluster-based classification has several advantages when applied to data-driven tutoring. New educational technologies may reveal unexpected learning behaviors, which may not yet be incorporated in expert-decided classification processes. For example, Kizilec et al. [5] clustered MOOC learners into different engagement trajectories, and revealed several trajectories that are not acknowledged by MOOC designers. In addition, experts classify using their perception of the average students' performance[11] [13]. This perception may be different from the actual participant group. Cluster-based classification methods, however, are able to classify and update classifications based on actual student behaviors.

Moreover, previous studies have shown that personalized tutoring based on cluster-based classification not only helps learning, but improves users' experience. Klasnja-Milicevic et al. [6] gave students different recommendations on learning content based on their classified learning styles. As a result students who used hybrid recommendation features completed more learning sessions successfully, and perceived the tutor as more convenient. Despotovic-Zratic et al. [1] adapted different course-levels, learning materials, and content in Moodle, an e-learning platform, for students in different clusters. Results showed that students with adapted course design had better learning gain, and a more positive attitude towards the course.

However, the majority of previous work clustered students solely on their overall performance statistics. In contrast, our method clusters students based on their application of specific knowledge components throughout the tutor.

## 3. METHODS

The Data-driven Proficiency Profiler (DDPP) is a system which calculates student proficiency at the end of each level in Deep Thought based on how a given student performs in comparison to exemplars who employed similar problem solving strategies (see Fig. 3), with rule scores weighted as determined through principal component analysis. Based on how similar exemplars were assigned in subsequent lev-

els, the DDPP can determine the best proficiency level for a new student. In contrast to the DDML system previously employed, this proficiency calculation and rule weighting is entirely data-driven, with no expert involvement. We hypothesize that the DDPP based calculation will perform more accurately when compared to average and minimum methods.

### 3.1 Data-driven Problem Profiler

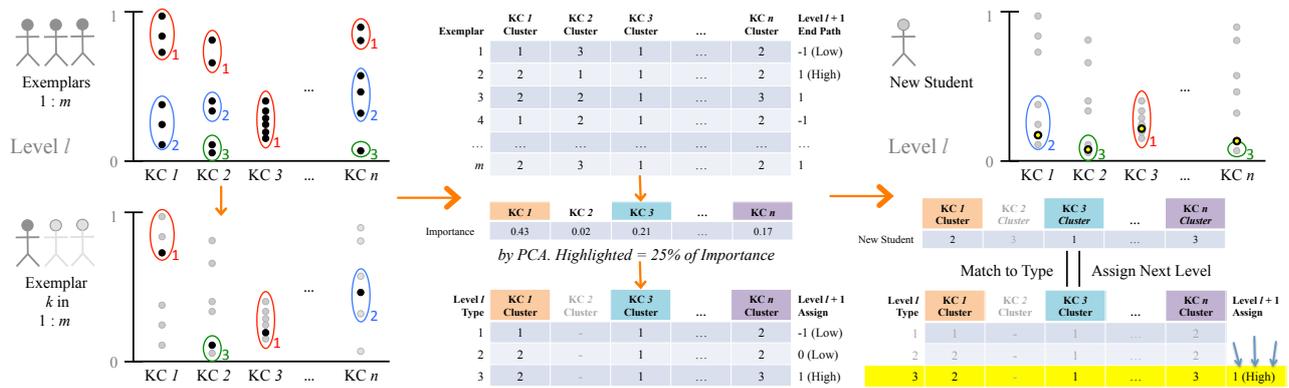
We first determined similar problem solving strategies among the exemplars by clustering the exemplars' rule scores (*KCs*) based on hierarchical clustering. For the initial single-point distance measure we used Euclidean squared distance, while for the hierarchical clustering algorithm we used cluster centroids to determine the distance between individual clusters. As a result each exemplar is assigned to a set of  $n$  clusters (where  $n$  is equal to the number of *KCs*), as shown in the table in Fig. 3.

Expert weighting was replaced by principal component analysis (PCA) of the frequency of the rules used for each exemplar for each level, accounting for 95% variance of the results. PCA is typically used to reduce the dimensionality of a data set by determining the most influential factors in the data set. The influence of a given factor is based on how much that factor contributes to the variability in the data. We use PCA analysis on the Deep Thought data set to determine which rules were most important to success in the tutor at each level. Rules which account for 25% of importance and higher are considered most important for completing a level. This percentage was determined through testing, and is the percentage that maximized accuracy. For each rule, its PCA importance value is the new weight for that rule score. Unlike expert authored weights, these rule score weights are based on each rule's importance as determined by the data.

When a new student uses the tutor, the student's rule scores are calculated throughout the level. At the end of each level, the DDPP looks at each student's individual rule score and assigns it to a cluster for that rule. The DDPP then finds which clusters the scores for the most important rules fall into for that level (based on the same PCA based weighting), and then classifies that student into a type based on the set of clusters the student matches (see Fig. 3, right). Finally the system assigns the student to a proficiency track based on data from the matching type of exemplars, and how those exemplars were placed in the next level. The more exemplars we have of a given type, the stronger the prediction we can make for a new student. In the event that a new student doesn't match an existing type in the exemplar data, their proficiency is calculated using the average scores. Average scores are used as a default because, as shown in the results, for most levels it is a better prediction approximation than using the minimum scores.

### 3.2 DDPP Advantages

In the original system, the student proficiency was determined based on one set of rule thresholds and a set of expert authored weights. However as a result, the system didn't take varying student problem-solving strategies into account. The data is based on students who completed the tutor, who have therefore shown the level of mastery required to successfully complete Deep Thought. However the



**Figure 3: The Data-Driven Proficiency Profiler.** (Left) At each level interval, exemplar KC scores are clustered, and exemplars are assigned a cluster for each KC Score. (Center) KCs that make up 25% of importance in the current level are used to assign exemplars to types. (Right) New student scores are assigned to clusters, and compared to existing types to determine next level path.

scores are averaged over all the students at the end of each level. By taking the average of these student scores at this point, we’re still assuming only one successful problem solving strategy for completing each level in the tutor. However while most strategies might be the same for earlier levels, there may be a variety of strategies in later levels that can still result in successful completion.

The DDPP method accounts for that possible variety in problem solving methods. In using an unsupervised clustering method, we’re able to account for different clusters while not knowing how many clusters there are for each rule. By clustering the scores, we’re essentially looking for different strategies that utilize particular rules and determining these strategies based on the student data. Once we determine which strategy a new student is utilizing, we can look to the data again to see how exemplars who employed a similar strategy were placed in the tutor and how they performed, thus determining the best way for the tutor to react to that particular student. Using PCA based weights allows us to weight rule scores based on rule importance as determined by previous students who completed the tutor, rather than expert determination.

### 3.3 Evaluation

Testing was performed on data collected from two courses using Deep Thought with the DDML system. The first was a Philosophy deductive logic course ( $n = 47$ ) using Deep Thought as a regular assignment over the course of a 15-week semester. The second was a Computer Science discrete mathematics course ( $n = 84$ ), using Deep Thought as a two week assignment during the course’s 4-week logic curriculum. From the students in these data, 26 of the Philosophy students (55%) and 50 of the Computer Science students (60%) completed the tutor, and were used as exemplars for the compared methods. By completing all levels in Deep Thought, these students have demonstrated sufficient mastery of the skills needed for introductory proof problem-solving.

By using data from both Computer Science and Philosophy based teaching methods for propositional logic, we expand

the range of problem solving strategies analyzed and exemplar types determined. This allows us to test the tutor’s performance across different classroom conditions, and determine whether the methods for proficiency path placement are effective for students in different disciplines that use different teaching methods.

The DDML system used the average of exemplar rule scores, weighted by expert-authored end of level rule priorities, to calculate student proficiency. In total there were 19 individual rule actions in Deep Thought on which students were evaluated. Based on the results of this calculation, the DDML system determines whether to send a student on the higher or lower proficiency path in the next level. The system also allowed for the possibility of students switching proficiency paths in situations where the student cannot complete the level on the path they were originally assigned. Because students can switch paths in the middle of a level, we can determine if they finish the current level on the same path they were assigned. If the student did not finish the level on the same proficiency path, it is an indication that the DDML system may have initially assigned the student to the wrong proficiency path. Therefore we can calculate the accuracy of the original system by determining how often students who completed the entire tutor changed proficiency paths throughout. Given  $S_{sameTrack}$  as the number of students who finished a level on the same proficiency track, and  $S_{total}$  as the total number of students who completed the level, the path prediction accuracy for each level (*LevelAccuracy*) is calculated as follows:

$$LevelAccuracy = \frac{S_{sameTrack}}{S_{total}} \quad (2)$$

The *LevelAccuracy* for each level is added together to determine the path prediction accuracy. This calculation tells us, for students who completed the entire tutor, how well the original system predicted the paths for them to continue on. This serves as a basis of comparison between the DDPP and the original DDML system.

### 3.3.1 Minimum & Average

The average rule scores are the set of average scores for each rule in each level. Minimum scores are the smallest scores in the exemplar data set for each rule in each level. This calculation is based on the assumption that if a student scores at least at this minimum for a given rule in that level, the student should be able to perform as well as an exemplar throughout the tutor. The difference between the current DDML system and average score or minimum score based proficiency calculation is that the DDML weighted scores with expert-decided rule priorities, while average or minimum weighted average or minimum scores with PCA-determined weights. Calculating proficiency based on average and minimum scores offers insight into how introducing PCA to students' performance baseline changes the prediction accuracy.

## 4. RESULTS

The prediction accuracy of the minimum, average, and DDPP methods were calculated for the 76 exemplars from the Philosophy and Computer Science data sets. Ten-fold cross validation was used to train and test the methods across the combined data. We focus on the results of the path prediction accuracy described in section 3.3 as a basis of comparison between the original system, the DDPP, proficiency based on average scores, and proficiency based on base minimum scores. These results are in tables 1, 2, and 3.

### 4.1 Path Prediction Accuracy

Table 1 shows the path prediction accuracy of the DDML system, the DDPP system, average score assessment, and minimum score assessment across all the students in the Philosophy and CS courses. The original system accuracy was very high, ranging from 75% at the end of level 3 to 88.2% at the end of level 1. The DDPP was somewhat accurate, ranging from 61.8% path prediction accuracy at the end of level 4 to 67.1% path prediction accuracy at the end of level 2. While these accuracies are not nearly as high as in the original system, they are very good considering that, unlike the original system, path prediction in the DDPP is entirely data-driven. It should also be noted that the DDPP was more consistent in its accuracy, only varying by at most 5% between levels (in comparison to the original DDML system, which ranged in accuracy by 9.3%).

**Table 1: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for both Philosophy and CS students at the end of each level**

	Original	DDPP	Average	Minimum
Lvl 1	88.2%	65.8%	65.8%	35.5%
Lvl 2	85.5%	67.1%	73.7%	18.4%
Lvl 3	75.0%	63.2%	60.5%	69.7%
Lvl 4	78.9%	61.8%	64.5%	40.8%
Lvl 5	78.9%	64.5%	59.2%	59.2%

Overall the original system predicted paths more accurately than the DDPP, average, or minimum methods across all levels. The minimum method was least accurate across all levels. In comparison to the average method, the DDPP was more accurate than the average method at the end of

levels 3 and 5. The DDPP was equally as accurate as the average method at the end of level 1, and less accurate at the end of levels 2 and 4. However, some of the lower accuracy was likely due to the distribution of exemplars across the two courses. Recall that the CS students made up a higher proportion of the analyzed exemplars than the Philosophy students. Analyzing the path prediction accuracy by the individual course reveals more detail on the path prediction accuracy.

### 4.2 Philosophy & CS Accuracy

In the case of the Philosophy students, where proportionally fewer of the students were selected as exemplars, the DDPP system was more accurate than the original system on every set of levels except for the end of level 5 (see Table 2). In comparison to the average calculation method, the DDPP was only more accurate at the end of level 3. At the end of levels 1 and 5, the DDPP was as accurate as the average method, and at the end of levels 2 and 4 the DDPP was less accurate.

**Table 2: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for Philosophy students**

	Original	DDPP	Average	Minimum
Lvl 1	76.9%	80.8%	80.8%	23.1%
Lvl 2	65.4%	69.2%	76.9%	19.2%
Lvl 3	50.0%	84.6%	80.8%	38.5%
Lvl 4	65.4%	69.2%	76.9%	30.8%
Lvl 5	53.8%	46.2%	46.2%	26.9%

In the CS course, where proportionally more of the students were selected as exemplars, not only was the original system far more accurate than it was for the entire set of students overall, but the DDPP path accuracy was much worse in some places. However, in comparison to the average method, the DDPP method was only less accurate in level 2. In all other levels the DDPP was either more accurate than the average method (levels 3 and 5) or equally as accurate (levels 1 and 4).

**Table 3: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for Computer Science students**

	Original	DDPP	Average	Minimum
Lvl 1	94.0%	58.0%	58.0%	42.0%
Lvl 2	96.0%	66.0%	72.0%	18.0%
Lvl 3	88.0%	52.0%	50.0%	86.0%
Lvl 4	86.0%	58.0%	58.0%	46.0%
Lvl 5	92.0%	74.0%	66.0%	76.0%

### 4.3 Discussion

In the original DDML method, the weight of each rule was determined by domain experts. Our results show that when replacing the original weights by weights determined through principal component analysis in the average score method, the prediction accuracy increases for all levels in the philosophy class, but decreases for all levels in the computer

science class. This may be because the experts were computer science students and teachers, who prioritized rules with the performance of computer science students in mind. When the real participants were philosophy students, Principal Component Analysis outperformed experts because it prioritized rule based on the performance of the real participants. It's possible that expert involvement may be constrained by the expert's background, whereas a data-driven approach is more flexible when adapting to the diversity of participants.

When comparing the path prediction accuracy of the original method to the DDPP, our result shows that the DDPP calculated student proficiency with more accuracy in the case of the Philosophy students, but less accuracy overall or in the case of the Computer Science students. It is likely that these results are a product of the limited, uncontrolled nature of the dataset. Only 76 exemplars were chosen overall, and of those exemplars a disproportionate number of them were selected from the computer science course. We noticed in the data that the students in the Computer Science course had KC weights that were vastly different than the expert weights. This means the students in the Computer Science course were showing some unorthodox problem solving strategies, particularly in the earlier levels. With enough data and more students with varying strategies, the DDPP could more accurately assign other students who employ different proof solving strategies. However for this limited dataset, it is possible that there were not enough students employing the same unorthodox strategies that a type could be determined.

**Table 4: The average number of types found per level during training (exemplars), and the number of students typed during testing (new students). There were a total of 76 students in the data set.**

Level	1	2	3	4	5
Avg. Types Found (Train)	14	13	21	17	26
# Types Matched (Test)	0	4	2	2	10

Table 4 shows the average number of types found in the training dataset, and the number of students matched to a type during testing. While there were several types found in the training step, far fewer students could be matched to a type in the testing step. This would explain the lower accuracy in the DDPP system, as well as why it performed similarly to the average method; it is likely that many of the students in the test set could not be classified into a type, which would result in the DDPP using the calculation based on average scores to determine student proficiency.

That said, the DDPP is still very accurate considering that, in all aspects of proficiency calculation, it is completely data-driven. Its accuracy when applied to the students in the Philosophy class in particular shows the potential for this system to be useful in different classroom conditions. The clustering step at each level produced between 14 and 26 possible types of exemplars to compare students to, compared to what would have been 76 individual students in the original system. This results in a system of proficiency calculation that, given more data, has the potential to calculate

student proficiency just as accurately and more efficiently as the original.

## 5. CONCLUSIONS & FUTURE WORK

We have presented a fully data-driven student proficiency calculator, the Data-driven Proficiency Profiler (DDPP). The DDPP clusters exemplar student data into types, attempts to classify new students into one of the exemplar types, and calculate proficiency based on exemplars who employed similar problem strategies. We hypothesized that the DDPP would be more accurate than proficiency calculated using average scores or minimum scores. Instead, our results showed that the DDPP performed about as well as the average method overall, and did not approximate the accuracy of the original system. However our data set was very limited, and the high accuracy the DDPP achieved for the Philosophy students shows this system has potential once more data can be acquired.

In the future, we would like to be able to test this system with more data. The more students use the system, the greater the data set we will be able to use and the more conclusions we will be able to draw on the qualities of the DDPP system. In particular we will analyze in greater detail the types found on each level and the differences between each type in terms of problem solving strategy. We can also determine the importance, in depth, of certain rules to each level and the problems within it based on student problem solving strategies. Our final step is to implement the DDPP into Deep Thought and use it to direct students through the levels. Implementing the DDPP into Deep Thought will allow us to test whether, ultimately, the DDPP is an accurate, data-driven proficiency calculation.

## 6. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grants 1432156 and 0845997.

## 7. REFERENCES

- [1] M. Despotovic-Zrakic, A. Markovic, Z. Bogdanovic, D. Barac, and S. Krco. Providing adaptivity in moodle lms courses. *Educational Technology Society*, 15(1):326–338, 2012.
- [2] M. Elmadani, M. Mathews, and A. Mitrovic. Data-driven misconception discovery in constraint-based intelligent tutoring systems. In *Proceedings of the 20th International Conference on Computers in Education (ICCE)*, pages 26–20, 2012.
- [3] S. E. Fancsali. Causal Discovery with Models: Behavior, Affect, and Learning in Cognitive Tutor Algebra. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, pages 28–35, 2014.
- [4] J. P. Gonzalez-Brenes and J. Mostow. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*, pages 236–239, 2013.
- [5] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In

*Proceedings of the 3rd international conference on learning analytics and knowledge*, pages 170–179, 2013.

- [6] A. Klasnja-Milicevic, B. Vesin, M. Ivanovic, and Z. Budimac. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers Education*, 56(3):885–899, 2011.
- [7] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.
- [8] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Data-driven Learner Modeling to Understand and Improve Online Learning: MOOCs and technology to advance learning and learning research (Ubiquity symposium). In *Ubiquity 2014*. 2014.
- [9] J. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012)*, pages 118–125, 2012.
- [10] B. Mostafavi, M. Eagle, and T. Barnes. Towards Data-driven Mastery Learning. In *To appear in Proc. Learning, Analytics, and Knowledge (LAK 2015)*.
- [11] E. V. Perez, L. M. R. Santos, M. J. V. Perez, J. P. de Castro Fernandez, and R. G. Martin. Automatic classification of question difficulty level: Teachers’ estimation vs. students’ perception. In *Proceedings of the IEEE Frontiers in Education Conference*, pages 1–5, 2012.
- [12] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic BulletinReview*, 14(2):249–255, 2007.
- [13] G. van de Watering and J. van der Rijt. Teachers and students perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items. *Educational Research Review*, 1(2):133–147, 2006.

# Interaction Network Estimation: Predicting Problem-Solving Diversity in Interactive Environments.

Michael Eagle, Drew Hicks, and Tiffany Barnes  
North Carolina State University, Department of Computer Science  
890 Oval Drive, Campus Box 8206  
Raleigh, NC 27695-8206  
{mjeagle, aghicks3, tmbarnes}@ncsu.edu

## ABSTRACT

Intelligent tutoring systems and computer aided learning environments aimed at developing problem solving produce large amounts of transactional data which make it a challenge for both researchers and educators to understand how students work within the environment. Researchers have modeled student-tutor interactions using complex networks in order to automatically derive next step hints. However, there are no clear thresholds for the amount of student data required before the hints can be produced. We introduce a novel method of estimating the size of the unobserved interaction network from a sample by leveraging Good-Turing frequency estimation. We use this estimation to predict size, growth, and overlap of interaction networks using a small sample of student data. Our estimate is accurate in as few as 10-30 students and is a good predictor for the growth of the observed state space for the full network, as well as the subset of the network which is usable for automatic hint generation. These methods provide researchers with metrics to evaluate different state representations, student populations, and general applicability of interaction networks on new datasets.

## 1. INTRODUCTION

Data-driven methods to provide automatic hints have the potential to substantially reduce the cost associated with developing tutors with personalized feedback. Modeling the student-tutor interactions as a complex network provides a platform for researchers to automatically generate next step hints. An *Interaction Network* is a complex network representation of all observed student and tutor interactions for a given problem in a game or tutoring system. In addition to their usefulness for automatically generating hints, interaction networks can provide an overview of student problem-solving approaches for a given problem.

Data-driven approaches cannot reliably produce feedback until sufficient data has been collected, a problem often referred to as the Cold Start problem. The precise amount of

data needed varies by problem and environment. However, some properties of Interaction Networks allow us to estimate how much data is needed. Eagle et al. explored the structure of these student interaction networks and argued that networks could be interpreted as an empirical sample of student problem solving [5]. Students employing similar problem-solving approaches will explore overlapping areas of the Interaction Network. The more similar a group of students is, the smaller the overall explored area of the interaction network will ultimately be. Since we expect different populations of students to have different interaction networks, and different domains to require varying amounts of student data before feedback can be given, good metrics for the current and predicted quality of Interaction Networks are important.

In this work, we adapt Good-Turing frequency estimation to interaction level data to predict the size, growth, and “hintability” of interaction networks. Good-Turing frequency estimation estimates the probability of encountering an object of a hitherto unseen type, given the current number and frequency of observed objects [8]. It was originally developed by Alan Turing and his assistant I. J. Good for use in cryptography efforts during World War II. In our context, network states (vertices) are the object types, and the student interactions (edges) leading to those states are observations.

We present several metrics, derived from Good-Turing frequency estimation. Our hypotheses are that these metrics: **H1:** Predict the probability that a student interaction will result in a state which was not previously observed **H2:** Describe the proportion of the network that has been observed for a population **H3:** Predict the expected size and growth of an interaction network when additional student data is added **H4:** Provide a quantitative comparison of different state representations for their ability to represent greater proportions of the network **H5:** Are useful for comparing different populations of users in how they explore the problem space

Additionally, we use the metrics to explore the subset of the interaction network that is useful for providing automatically generated hints. This provides us with estimates of the size, growth, and coverage of automatically generated hints. We find that our metrics quickly become accurate after collecting a sample of about 10 students. This has value as a metric to compare the quality of the interaction networks,

and will aid future researchers in determining an adequate state representation. We also show how two experimental groups, despite having the same amount of network coverage, have substantially different numbers of unique states. This supports previous work, suggesting that different populations of students produce different interaction networks [5], which has broad implications for generating hints as well as using the networks to evaluate student behavior.

## 1.1 Previous Work

Creation of adaptive educational programs is costly. This is, in part, because developing content for intelligent tutors requires multiple areas of expertise. Content experts and pedagogical experts must work with tutor developers to identify the skills students are applying and the associated feedback to deliver [13]. In order to address the difficulty in authoring intelligent tutoring content, Barnes and Stamper built an approach called the Hint Factory to use student data to build a Markov Decision Process (MDP) of student problem-solving approaches to serve as a domain model for automatic hint generation [18]. Hint Factory has been applied in tutoring systems and educational games across several domains [7, 14, 6], and been shown to increase student retention in tutors [19].

Early work with the Hint Factory method used a Markov Decision Process constructed from students' problem-solving attempts. Eagle and Barnes further developed this structure into a complex network representation of student interactions with the system, called an *Interaction Network* [5]. Complex networks are graphs or networks which contain non-trivial topological features unlikely to appear in simple or random networks. The Interaction Network representation can be used as a visualization of student work within tutors. The effectiveness of Interaction Networks as visualizations was shown by Johnson et al. who created a visualization tool *InVis* to aid instructors in analyzing student-tutor data [11].

Other approaches to automated generation of feedback have attempted to condense similar solutions in order to address sparse data sets. One such approach converts solutions into a canonical form by strictly ordering the dependencies of statements in a program [15]. Another approach compares *linkage graphs* modelling how a program creates and modifies variables, with nested states created when a loop or branch appears in the code [10]. In the Andes physics tutor, students may ask for hints about how to proceed. Similarly to Hint Factory-based approaches, a solution graph representing possible correct solutions to the problem was used. However their solution space was explored procedurally rather than being derived from student data, and they used plan recognition to decide which of the problem derivations the student is working towards [20].

Interaction networks are scale-free networks. This is a property of complex networks whose degree distribution is heavy-tailed, often a power law distribution. In practice, this means that a few vertices have degree that is much larger than the average, while many vertices have degree somewhat lower than average [5]. Eagle et al. argued that students with similar problem solving ability and preferences would travel into similar parts of the network, resulting in

some states being more important to the problem than others [5]. Using these "hub" states, sub-regions of the network corresponding to high-level approaches to the problem were derived. These sub-regions captured problem-solving differences between two experimental groups [4].

## 2. METHODS AND MATERIALS

For the purposes of this work, we are using datasets from three different environments to build our interaction networks. Summaries of these datasets are found in Table 1. The first dataset is from the Deep Thought tutor, used in previous work by Stamper et al. [19]. This dataset was collected for a between groups experiment investigating the use of data-driven hints, so we split the dataset into two groups, DT1-C, the control group from that experiment, and DT1-H, the group that received hints. We selected this dataset to explore and evaluate H5.

The second dataset comes from the game BOTS. Here, we have the same students and interactions represented in two different ways: First, using *codestates* (the programs users wrote) and second using *worldstates* (the output of those programs). The advantages and disadvantages of these state representations were explored in previous work by Peddycord and Hicks [14]. We split this dataset into two groups as well (BOTS-C and BOTS-W) one for each state representation used. We selected this dataset for evaluation of H4.

Our third and largest dataset comes from an updated version of the Deep Thought tutor, called Deep Thought 3. Unlike with the other datasets, Deep Thought 3 features an AI problem selection component [12]. This means that not all students will have had access to all problems. In addition, there is a larger number of problems in this dataset. We selected this dataset, as the larger number of problems effectively splits student data across multiple networks. H1-H3 are relevant towards measuring the quality of networks produced for new problems.

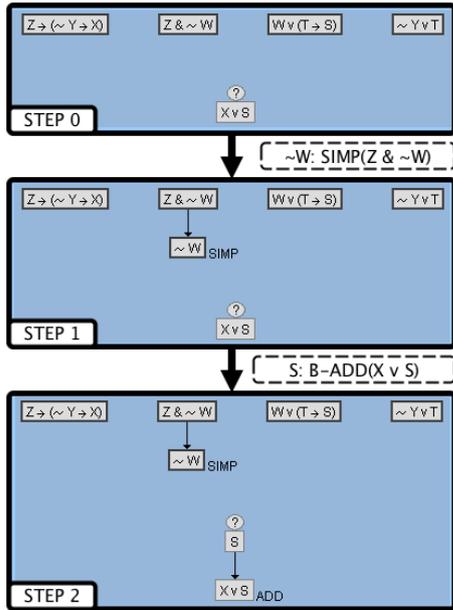
**Table 1: Dataset summary: the total number of students in the dataset, the number of distinct problems, and the average number of students represented in each network.**

Dataset	Total N	Num Problems	Mean Net N
DT1-H	203	11	83.73
DT1-C	203	11	63.82
DT3	341	59	78.41
BOTS-C	125	12	99.75
BOTS-W	125	12	99.75

### 2.1 Constructing an Interaction Network

An *Interaction Network* is a complex network representation of all observed student and tutor interactions for a given problem in a game or tutoring system. To construct an Interaction Network for a problem, we collect the set of all solution attempts for that problem. Each solution attempt is defined by a unique user identifier, as well as an ordered sequence of interactions, where an interaction is defined as {initial state, action, resulting state}, from the start of the

problem until the user solves the problem or exits the system. The information contained in a *state* is sufficient to precisely recreate the tutor's interface at each step. Similarly, an *action* is any user interaction which changes the state, and is defined as {action name, pre-conditions, post-conditions}. In Deep Thought, for example, an action would be the logical axiom applied, the statements it was applied to, and the resulting derived statement. Figure 1 displays two Deep Thought interactions. The first interaction works forward from STEP0 to STEP1 with action *SIMP* (simplification) applied to  $(Z \wedge \neg W)$  to derive  $\neg W$ . The second interaction works backward from STEP1 to STEP2 with action *B-ADD* (backwards addition) applied to  $(X \vee S)$  to derive the new, unjustified statement *S*.



**Figure 1: Example of state to state transitions within the Deep Thought (DT1) propositional logic tutoring system.**

Once the data is collected, we use a *state matching function* to combine similar states. In Deep Thought, we combine states that consist of all the same logic statements, regardless of the order in which those statements were derived. This way, the resulting state for a step STEP0, STEP1, or STEP2 in Figure 1 is the set of justified and unjustified statements in each screenshot, regardless of the order that each statement was derived. In BOTS, two state matching functions were used: one which combined states based on the code in students' programs, and another which instead used the output of those programs. Similarly, we use an action matching function to combine actions which result in similar states, while preserving the frequency of each observed interaction.

## 2.2 Providing Hints

Stamper and Barnes' Hint Factory approach generates a next step Hint Policy by modeling student-tutor interactions as a Markov Decision Process [18]. This has been adapted to work with interaction networks by using a Value Itera-

tion algorithm on the states [5]. We generate a graph of all student interactions, combining identical states using a state matching function. Then, we calculate a fitness value for each state. We assign a positive value (100) to each goal state, that is a state configuration representing a solution to the problem. We assign an error cost (-5) for error states. We also assign a small cost to performing any action, which biases hint-selection towards shorter solutions. We then calculate fitness values  $V(s)$  for each state  $s$ , where  $R(s)$  is the initial fitness value for the state,  $\gamma$  is a discount factor, and  $P(s, s')$  is the observed frequency with which users in state  $s$  take an action resulting in state  $s'$ . After this, we use value iteration [2] to repeatedly assign each state a value based on its neighbors and action costs, weighted by frequency.

After applying this algorithm, we can provide a hint to guide the user toward the goal by selecting the child state with the best value. We can do this for any observed state, provided that a previous user has successfully solved the problem after visiting that state. In the original work with Hint Factory on the Deep Thought tutor, the algorithm was permitted to backtrack to an earlier state if it failed to find a hint from the current state. However, not all environments allow the user to backtrack and there are risks of the backtracking hints to provide irrelevant information. Because of this inconsistency across domains, we did not permit backtracking for the purposes of the comparisons in this paper.

We define a state,  $S$  to be *Hintable* if  $S$  lies on a path which ends at a goal state. We define the *Hintable* network to be the subset of the interaction network containing only *Hintable* states and edges between hintable states; That is, the induced subgraph on the set of *Hintable* states.

## 2.3 Cold Start Problem

Barnes and Stamper [1] approached the question of how much data is needed to get a certain amount of overlap in student solution attempts by incrementally adding student attempts and measuring the step overlap over a large series of trials. This was done with the goal of producing automatically generated hints, and solution attempts that did not reach the goal were excluded. Peddycord et al. [14] used a similar technique to evaluate differences in overlap between two different interaction network state representations.

The "Cold Start problem" is an issue that arises in all data-driven systems. For early users of the system, predictions made are inaccurate or incomplete [17, 16]. If there are insufficient data to compare to (not enough user ratings, or not enough student attempts) then the quality of the recommendations suffers and in some cases no recommendation can be provided. The term is commonly used in the field of collaborative filtering and recommender systems, but it can be used to describe three related issues, the "new user," the "new item," and the "new community" [3] Cold Start problems. The "new user" problem refers to the difficulty of making recommendations to a user who has performed no actions. The "new item" problem refers to the difficulty of suggesting users visit a newly added, unobserved state. The new community Cold Start problem refers to situations where not enough observations exist to make recommendations for new users. The "new community" definition corresponds most closely to the difficulty of generating hints for

an entirely new problem in an intelligent tutoring system or educational game.

To measure our ability to address this problem, we add all interactions from a single student, one at a time, to the interaction network. This is in order to simulate the growth of the network. We repeat this process for each student, measuring the performance of our model each time. We measured the proportion of currently observed states to total observed states for the entire data set, as well as for the subset of states from which a goal is reachable. To control for ordering effects, we repeated this trial 1000 times using a different random ordering of students each time, and aggregated the results.

## 2.4 Good-Turing Network Estimation

We present a new method for estimating the size of the unobserved portion of a partially constructed Interaction Network. Our estimator makes use of Good-Turing frequency estimation [8]. Good-Turing frequency estimation estimates the probability of encountering an object of a hitherto unseen type, given the current number and frequency of observed objects. It was originally developed by Alan Turing and his assistant I. J. Good for use in cryptography efforts during World War II. Gale and Sampson revisited and simplified the implementation [8]. In its original context, given a sample text from a vocabulary, the Good-Turing Estimator will predict the probability that a new word selected from that vocabulary will be one not previously observed.

The Good-Turing method of estimation uses the frequency distribution, the “frequency of frequencies,” from the sample text in order to estimate the probability that a new word will be of a given frequency. Based on this distribution, the probability of observing a new word in an additional sample is estimated with the observed proportion of words with frequency one. This estimate of unobserved words is used to adjust the probabilities of encountering words of frequencies greater than one.

We adapt the Good-Turing Estimator to interaction networks by using the states with an observed frequency of one to estimate the proportion of “frequency zero” states. Interaction networks represent the observed interactions and therefore we also use this value to estimate the probability that a new interaction will transition into a new state. We use  $P_0$  as the expected probability of the next observation being an unseen state.  $P_0$  is estimated by:

$$P_0 = \frac{N_1}{N} \quad (1)$$

Where  $N_1$  is the total number of frequency 1 states, and  $N$  is the total number of interaction observations. Since  $N_1$  is the largest group of states, the observed value of  $N_1$  is a reasonable estimate of  $P_1$ .  $P_0$  can then be used to smooth the estimation proportions of the other states. The proportion of states with observed frequency  $r$  is found by:

$$P_r = \frac{(r+1)S(N_{r+1})}{N} \quad (2)$$

where  $S()$  is a smoothing function that adjusts the value for large values of  $r$  [8].

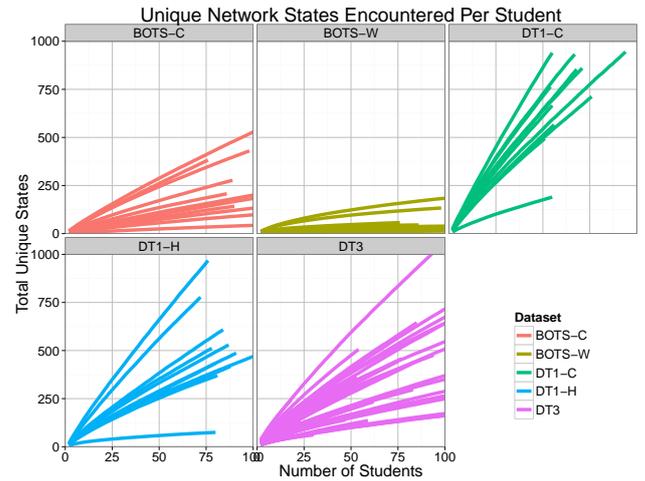


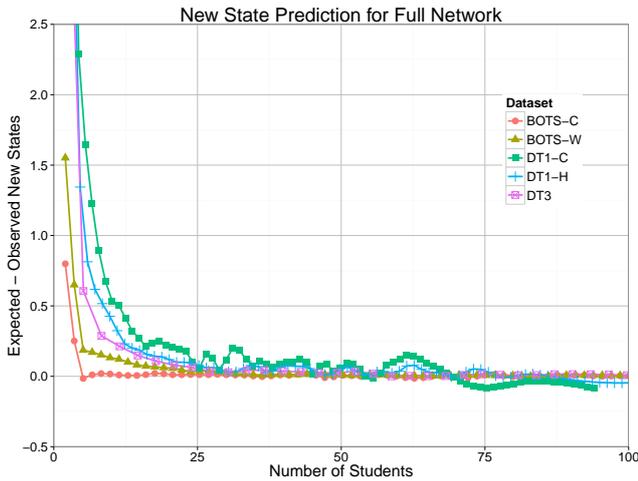
Figure 2: The growth of new states as new students are added for each problem, for each dataset.

Our version of  $P_0$  is the probability of encountering a new state (a state that currently has a frequency of zero,) on a new interaction. We also interpret this as the proportion of the network missing from the sample. We will refer to an interaction with a unobserved state as having *fallen off* of the interaction network. We will use the complement of  $P_0$  as the estimate of *network coverage*,  $I_C$ , the probability that a new interaction will remain on the network:  $I_C = 1 - P_0$ .

The *state space* of the environment is the set of all possible state configurations. For both the BOTS game and the Deep Thought tutor the potential state space is infinite. For example, in the Deep Thought tutor a student can always use the addition rule to add new propositions to the state. However, as argued in Eagle et. al. [5], the actions that reasonable humans perform is only a small subset of the theoretical state space; the actions can also be different for different populations of humans. We will refer to this subset as the *Reasonable State Space*, with *unreasonable* being loosely defined as actions that we would not expect a human to take. An interaction network is an empirical sample of the problem solving behavior from a particular population, and is a subset of the state space of all possible *reasonable* behaviors. Therefore, our metrics  $P_0$  and  $I_C$  are estimates of how well the observed interaction network represents the reasonable state space.

## 3. RESULTS

In order to evaluate the performance of the unobserved network estimator,  $P_0$ , and the network coverage estimator,  $I_C$ , for each problem in each of our 5 datasets we randomly added students from the sample, one at a time until all student data had been included. At each step,  $T$ , we recorded the values of our estimators using only the data that had been encountered up until then. This simulates a real world use-case, where additional students are added over time. We repeated this process 1000 times and averaged the results. Figure 2 shows the growth of unique states as students are added for the interaction networks generated by each problem (line) in each of the five datasets.



**Figure 3:** The average absolute error between the estimated number of new states and the observed new states over the number of students for all problems in each of the four datasets.  $P_0$  accurately predicts the observed values after roughly 10 students, rarely being off by more than one after that.

### 3.1 H1: Prediction of New States

In order to evaluate  $P_0$  for the prediction of new states (states that are frequency = 0 on time  $T_i$ , but will be frequency = 1 on  $T_{i+1}$ ). At each  $T$  we add an additional student and compare the expected number of frequency 1 states,  $E_{S1}$ , vs. the observed number,  $O_{S1}$ . Across all five datasets, Figure 3 shows the differences between the expected and observed number of new states. The  $P_0 \times Interactions$  prediction for new states follows closely with the observed number, the estimates increase in accuracy rapidly over the first ten students and are rarely off by more than a fraction of a state afterwards. Figure 4 shows the results of running this process on only the hintable portion of the interaction network for each data set.

### 3.2 H2: Network Coverage

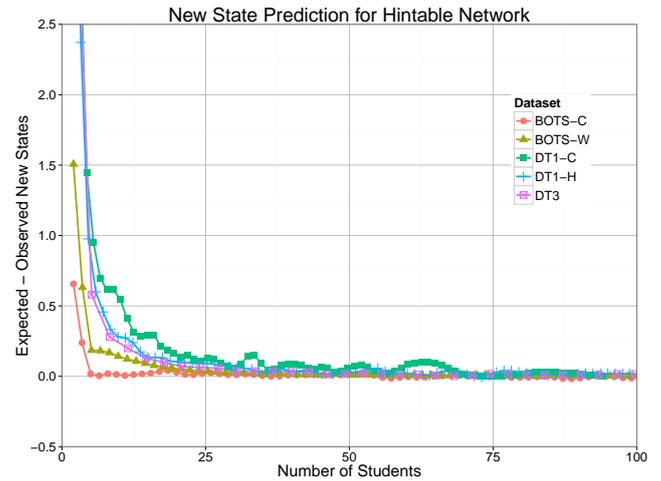
We have defined network coverage  $I_C$  as the proportion of interactions which lie within the previously observed network. Another interpretation is that  $I_C$  is the probability of an interaction resulting in a state that has been previously observed. This value is the complement of  $P_0$ . Figure 5 and 7 display the results of network coverage and its growth as additional students are added.

### 3.3 H3: Predicting Future Network Size

In order to further evaluate the use of  $P_0$  and  $I_C$  we calculated a prediction for the final size of the network, given the number of students in each dataset, at each time stamp. The equation for this prediction is:

$$|V(IN)| = (NewSample * P_0) + U_T. \quad (3)$$

Where  $|V(IN)|$  is the number of unique vertices (states) in the final network,  $NewSample$  is the number of new interactions added,  $P_0$  is the estimation of new states added, and  $U_T$  is the number of unique states observed at time  $T$ . The results are averaged across all problems for each dataset and



**Figure 4:** For the hintable states, the average difference between the estimated number of new states and the observed new states over the number of students for all problems in each of the four datasets.  $P_0$  accurately predicts the observed values after roughly 10 students, rarely being off by more than one after that.

are presented in figures 8 and 9. This prediction rapidly improves and after roughly 20% of the sample is added, can accurately predict the final number of unique states for the network. This combined with the accuracy of  $P_0$  reveals the short term and long term accuracy for the estimator.

### 3.4 H4: Comparing State Matching Functions

The network coverage metric,  $I_C$ , allows an easy method of estimating the differences in state matching functions and student network overlap. We can use  $I_C$  with two potential matching functions, and get an estimate of the remaining network, to quickly compare different potential state representations as well as to find a state generalization that will allow for a desired amount of network coverage.

The estimate based on the above methods has proven useful for comparing State Matching functions to help determine which produces more relevant hints. Figure 6 shows the BOTS interface, with the user's program (codestate) and the game world (worldstate) both illustrated. In previous work investigating the Cold Start problem on the BOTS data set, we measured "coverage" in terms of how much of the newly added test data was already present in the training set [9, 14]. Compare this analysis to Figure 5 which shows the estimated probability that a student's next action will result in an observed state,  $I_C$ . After 100 students, the probability that a student will generate a new *codestate* is still quite high,  $P_0 > .25$ . In comparison, after the same number of students, the probability of generating a new *worldstate* is extremely low,  $P_0 < .02$ . This result supports both our intuition and our results from the previous work, that students will continue to generate new *codestates*, but that these different *codestates* will collapse to previously observed *worldstates*.

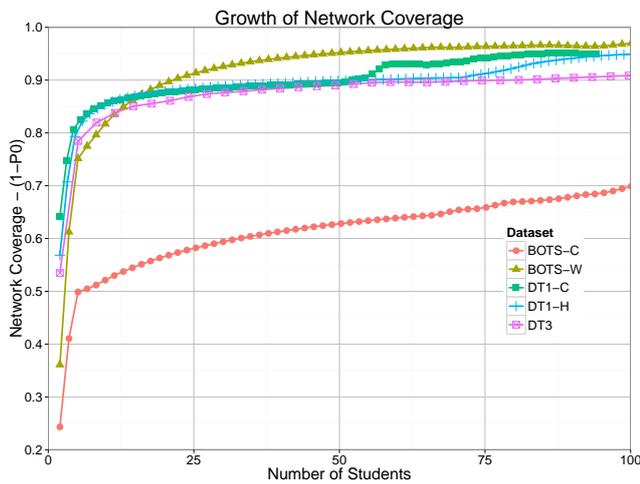


Figure 5: The estimated network coverage  $I_C$  for each of the 5 datasets, note the poor coverage for the BOTS-C dataset. The BOTS-W state is more general and has the much higher coverage.

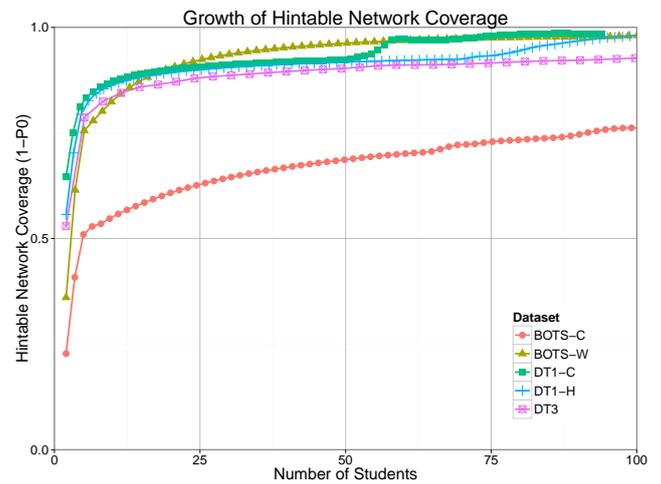


Figure 7: For the hintable network: the estimated network coverage  $I_C$  for each of the 5 datasets. Even the lowest performing hint network BOTS-C reaches roughly 70% coverage by 100 students.

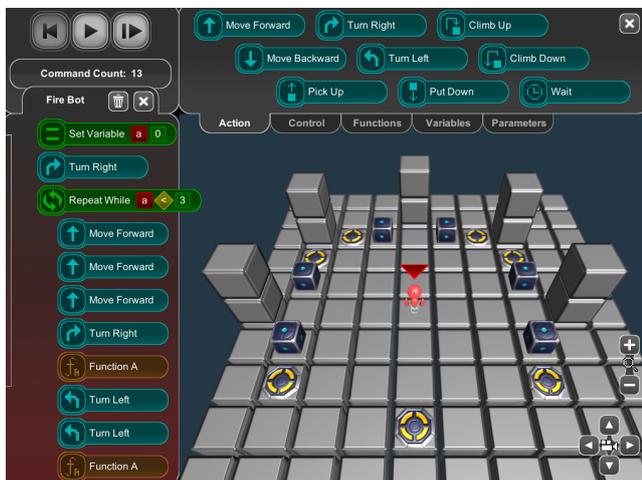


Figure 6: An image of the main gameplay interface for BOTS. The left hand side of the screen shows the user’s program, used to derive code states. The right-hand side shows the game world, where the program output determines the world states.

### 3.5 H5: Comparing Populations

Samples from different populations have different resulting interaction networks. The size of the represented network can tell us about the similarity of student approaches in the sample. If students are more alike in the types of actions they perform, fewer students will be needed to achieve a similar amount of overlap. We can also see that adding students from a dissimilar population will not always increase estimated network coverage ( $I_C$ ), and can potentially decrease it. This has implications about the importance of building hints for one population and applying it for another. In other work we have already shown that different groups are likely to visit different parts of the networks [4]. Here we expand on that analysis by showing that the two

Table 2: Different populations have different spread in problem exploration.

Group	$P_0$	States	Interactions	$F_1$
Hint	0.09	514.61	2709.84	250.09
Control	0.10	720.12	3904.92	340.00

groups, while having the same amount of network coverage, have a different number of unique states. Table 2 shows the results between the Hint group, which received hints on a subset of the problems, and the Control group which never received hints. This corresponds with results from Eagle et al. [4] in which they uncovered significant differences in the student overall approaches. This result adds to that an estimation of how complete each network was, revealing that additional data was not likely to change the result. It also shows some evidence for a *trail blazing effect*. When provided hints, students collectively explore a smaller area of the state space.

### 3.6 Estimating the effect of filtering

Visualizations must struggle with an “information to ink” ratio. There is a trade-off between displaying full information and overwhelming the viewer, and displaying only the most frequent states and potentially misleading the viewer by eliminating information. *InVis*, a visualization tool for exploring Interaction Networks allowed users to filter by frequency[11]. We can use the Good-Turing Estimation to calculate the amount of information removed by filtering frequency of a certain degree.  $P_0$  is the proportion of the network missing,  $I_{C>r} = I_C - P_1 - \dots - P_r + P_0$ , where  $r$  is a threshold value for removing low frequency states, and  $P_1 - \dots - P_r$  is the sum of  $P_1$  through  $P_r$ . This should be a useful metric for visualizations for measuring the amount of network that is hidden by filtering. It is also useful to show that sometimes a large number of graphical elements can be removed, with only a small amount of interaction information lost.

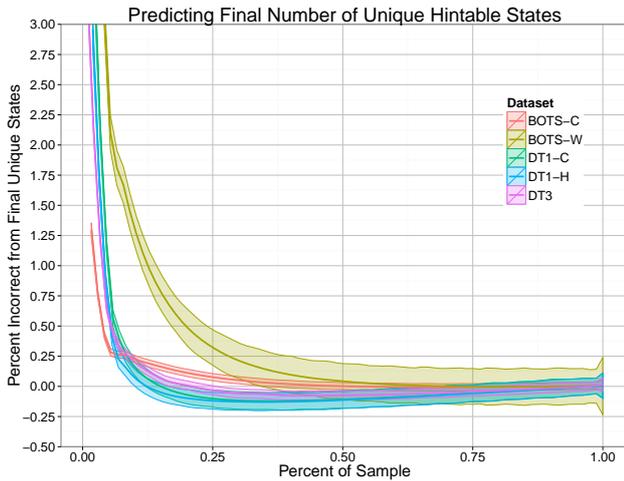


Figure 8: Prediction of total final number of states, as observed number of states increases. Note that for small  $t$ , the estimate is very high (up to 300% over prediction), but becomes fairly accurate after roughly 20% of the sample is measured.

#### 4. DISCUSSION

Good-Turing Estimation works well in the contexts of interaction networks. We were able to provide an easily calculable estimate of the proportion of the network not yet observed  $P_0$ . This value alone is a useful high level metric for the percentage of times a student interaction results in a previously unobserved state. The  $P_0$  score for the hintable network is likewise an estimate of the probability that a student will “fall off” of the network from which we can provide feedback. Our network coverage metric  $I_C$  allows a quick and easy to calculate method of comparing different state representations, as well as quantifying the difference. We believe that this metric can replace the commonly used cold start method of evaluating the “hintability” of a network.  $I_C$  is also valuable to quickly gauge the applicability of a new domain to interaction networks. The majority of the calculations can be performed on the transactional data. The growth trends for our five datasets were often clear after only ten students.

Our network estimators also have implications given our previous theories on the network being a sample created from biased (non-random) walks on the problem-space, as the more homogeneous the biased walkers are, the faster the network will represent the population and the fewer additional states will be explored. We revisited our previous results [4], and found that students with access to hints explored less overall unique states. This implies that the students were more similar to each other in terms of the types of actions and states they visited within the problem. Overall, this result supports the idea that different populations of students will have different interaction networks. The implications of this for generating hints are great. Building hints on one population might not work as well in another, and adding interventions or hints can dramatically reduce the number of states visited by the students. Future work should explore the possibility of having multiple network representations

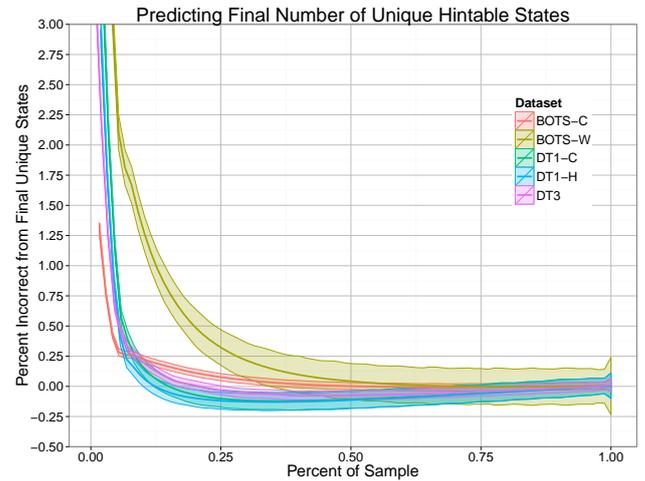


Figure 9: Prediction of total final number of goal states, as observed number of states increases. Note that for small  $t$ , the estimate is very high, but becomes an underestimate as  $t$  increases.  $P_0$  can predict the number of additional hintable states that can be added for a additional sample of data.

and choosing to match the student with the one closely resembling them.

As you can see in figure 8, our estimator starts out drastically overestimating the number of unobserved states in the network. As we collect data, this eventually becomes a slight underestimate, eventually converging on the correct number of states. One explanation for why this might be the case is the method by which undiscovered states are added to the network. By using this model for our estimator, we are making an assumption that states are selected independently of one another. At the beginning, when data is sparse, this assumption is not particularly harmful, since undiscovered states are relatively common. However, as our dataset becomes richer, we underestimate the probability of adding an unobserved state because we do not take into account the effect of “trail-blazing” which increases the probability of adding additional unobserved states after the first. Eagle and Barnes found that interaction networks had properties of scale-free networks. [5]. In particular, their degree distributions follow a power law, with a few vertices having much higher degree than the average for the network. It is likely that taking into account the scale-free and hierarchical nature of the networks will provide methods to improve on our estimators.

#### 5. CONCLUSIONS AND FUTURE WORK

We have adapted Good-Turing frequency estimation for use with networks built from student-tutor interactions. We found that the estimator for the missing proportion of the network  $P_0$  was accurate in predicting the number of new states discovered with new data. We also found that we could accurately measure network coverage with  $I_C$  for both the regular network, as well as the network of hintable states. This provides us with a metric to compare different state representations as well as determine the suitability of inter-

action network methods to different tutoring environments. We were also able to use these metrics to provide accurate predictions for the size of networks expected given more data samples, which will be useful for predicting the amount of additional data needed to provide a desired amount of hintable network coverage. Finally, we used the estimate of network coverage to compare different student populations to show that the addition of hints in one environment had an effect on the number of states explored by students.

Future work will include expanding on these *global* measures of the network and exploring *local* measures of coverage. Rather than compute coverage for the entire network we can use methods such as approach map regioning [4] to find meaningful sub-networks and calculate the metrics for those. The region level values of  $P_0$  can estimate the “riskiness” of certain approaches to the problem. The  $I_C$  metric can direct attention to parts of the network that are not well explored, perhaps allowing additional hints to be obtained by starting advanced users in those areas.

## 6. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. #0845997, #1432156, #1015456, #0900860 and #1252376.

## 7. REFERENCES

- [1] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS 2008)*, pages 373–382, 2008.
- [2] R. Bellman. A markovian decision process. Technical report, DTIC Document, 1957.
- [3] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26(0):225 – 238, 2012.
- [4] M. Eagle and T. Barnes. Exploring differences in problem solving with data-driven approach maps. *Proceedings of the Seventh International Conference on Educational Data Mining*, 2014.
- [5] M. Eagle, D. Hicks, P. III, and T. Barnes. Exploring networks of problem-solving interactions. *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK 15)*, 2015.
- [6] M. Eagle, M. Johnson, T. Barnes, and A. K. Boyce. Exploring player behavior with visual analytics. In *FDG*, pages 380–383, 2013.
- [7] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, L. Chen, and D. Cosejo. I learn from you, you learn from me: How to make ilist learn from students. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 491–498, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.
- [8] W. A. Gale and G. Sampson. Good-turing frequency estimation without tears\*. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.
- [9] A. Hicks, B. Peddycord III, and T. Barnes. Building games to learn from their players: Generating hints in a serious game. In *Intelligent Tutoring Systems*, pages 312–317. Springer, 2014.
- [10] W. Jin, T. Barnes, J. Stamper, M. J. Eagle, M. W. Johnson, and L. Lehmann. Program representation for automatic hint generation for a data-driven novice programming tutor. In *Intelligent Tutoring Systems*, pages 304–309. Springer, 2012.
- [11] M. W. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks.
- [12] B. Mostafavi, M. Eagle, and T. Barnes. Towards data-driven mastery learning. *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge (LAK 15)*, 2015.
- [13] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.
- [14] B. Peddycord III, A. Hicks, and T. Barnes. Generating hints for programming problems using intermediate output.
- [15] K. Rivers and K. R. Koedinger. Automating hint generation with solution space path construction. In *Intelligent Tutoring Systems*, pages 329–339. Springer, 2014.
- [16] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
- [17] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.
- [18] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197–201, 2008.
- [19] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education (IJAIED)*, 22(1):3–18, 2013.
- [20] K. Vanlehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204, 2005.

# Why do the rich get richer? A structural equation model to test how spatial skills affect learning with representations

Martina A. Rau  
Department of Educational Psychology  
University of Wisconsin—Madison  
1025 W. Johnson St  
Madison, WI 53706  
+1-608-262-0833  
marau@wisc.edu

## ABSTRACT

Spatial skills predict students' success in STEM domains. This paper aims to better understand the difficulties of students with low spatial skills in using interactive graphical representations. I present a mediation analysis with test and log data from 117 students who worked with an intelligent tutoring system for chemistry. The analysis is based on (1) a knowledge component model that describes knowledge students acquire as they solve problems with graphical representations, (2) a search for features that describe students' interactions with the representations and that are predictive of students' learning gains, and (3) a structural equation model that tests whether these features statistically mediate the effect of spatial skills on students' learning gains. Results show that only students' ability to plan representations before they construct them mediates the effect of spatial skills on learning gains. This finding suggests that these students may need more support *before* they construct representations.

## Keywords

Spatial skills, intelligent tutoring systems, interactive representations, STEM learning.

## 1. INTRODUCTION

Students' spatial skills predict learning success in STEM domains [1, 2]: students with low spatial skills tend to show lower achievements in STEM domains and they are less likely to pursue careers in these domains. Spatial skills are important for STEM learning because many concepts in STEM domains are inherently visuo-spatial. For example, astronomers have to visualize the solar system, engineers have to visualize interactions among components of a machine, and chemists have to visualize movements of atoms and electrons. To make these concepts accessible to students, instructional materials in STEM domains tend to heavily rely on the use of graphical representations [5, 6]. Graphical representations are external representations that use visuo-spatial features to depict domain-relevant concepts (as opposed to text or symbols). As a consequence, students have to make sense of visuo-spatial relationships depicted by graphical representations to understand abstract concepts in STEM domains [7].

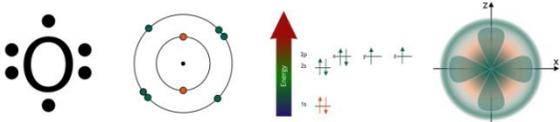


Figure 1. Graphical representations of an oxygen atom: Lewis structure, Bohr model, energy diagram, orbital diagram.

Consider, for example, a student who is learning about atomic structure. Figure 1 shows the graphical representations that instructional materials typically use to illustrate atomic structure [8]. Lewis structures (left) show paired and unpaired valence electrons, Bohr models (center-left) show all electrons in atomic shells, energy diagrams (center-right) depict electrons in orbitals with their energy level, and orbital diagrams (right) show the spatial arrangement of non-empty orbitals. To understand atomic structure, students have to integrate the information depicted in these graphical representations into a visuo-spatial mental model of how electrons are arranged relative to the atom's nucleus, and how they move according to probabilistic laws.

Integrating such information into a mental model of the domain-relevant concepts requires students to hold the relative location of the depicted objects in working memory and to mentally rotate these objects [9]. The cognitive load imposed by this task is arguably higher for students with low spatial skills than for students with high spatial skills [1]. As a consequence, students with low spatial skills may fail at this task, which might jeopardize their learning success [1, 5, 9]. On the flip side, students with high spatial skills are more successful at integrating visuo-spatial information into mental models, and—consequently—are likely to show higher learning gains. Thus, the rich (in spatial skills) get richer (in content knowledge).

Educational technologies such as intelligent tutoring systems (ITSs) hold particular promise for breaking the “the-rich-get-richer” rule and for creating an “everyone-gets-richer” rule, because they can address the needs of students with low spatial skills in several ways. First, ITSs can provide interactive tools that students can use to construct representations while receiving assistance and feedback. Such support for learning with interactive graphical representations can enhance learning outcomes [10], in particular for students with low spatial skills [11]. Second, ITSs have the capability to provide individualized support that adapts to student characteristics [12]. Adapting instructional support to the individual student's spatial skills has been shown to improve their spatial skills [13] as well as their learning of content knowledge [14].

However, before we can design ITSs that tailor support for using interactive representations to the needs of students with low spatial skills, we first have to understand what makes this learning task difficult for these students. This paper presents a first step towards this goal. Specifically, this paper investigates the following two questions: (1) Which aspects of problem solving with interactive graphical representations are more difficult for students with low spatial skills than for students with high spatial skills? (2) Which of these difficulties explain why students with

Atoms and Electrons

Let's make the Bohr model for oxygen!

- Oxygen is in row  of the periodic table. The atomic number shows that it has  electrons and is in A-group .
- The first shell is full because it has  electrons. Therefore, oxygen has a second shell with the remaining  electrons.
- Oxygen's row in the periodic table corresponds to its number of shells . Its A-group number corresponds to its number of valence electrons .
- Show the Bohr model for oxygen in the area to the left.
- In oxygen, the second shell is the valence shell. The Bohr model shows that the valence electrons are in the shell farthest from the nucleus.
- The Bohr model shows that oxygen has  unpaired electrons in its valence shell, so  of its electrons will form bonds.

Hint: No, this is not correct. The Bohr model shows all of the electrons, not only the valence electrons.

Periodic Table

Identify properties of the atom

Plan features of the representation

Construct representations with an interactive tool

Make inferences about the atom

Figure 2. Example screen shot of a tutor problem: students construct a Bohr model of oxygen.

low spatial skills have lower learning outcomes in chemistry? To address these questions, I conducted a mediation analysis that tested which aspects of students' problem-solving performance account for the effect of spatial skills on learning outcomes. The mediation analysis was carried out with a data set obtained from an experiment with an ITS for chemistry learning in which students had to use interactive tools to construct graphical representations of atoms.

## 2. CHEM TUTOR

The data set used in this paper was obtained from an experiment with Chem Tutor: an ITS for undergraduate chemistry [15]. The goal of Chem Tutor is to enhance learning by helping students understand graphical representations of abstract concepts [16]. Chem Tutor targets foundational concepts of introductory undergraduate courses, such as atomic structure and bonding. The design of Chem Tutor is based on surveys with undergraduate chemistry students and instructors, interviews and eye-tracking studies with undergraduate and graduate students, and extensive pilot testing in the lab and the field [15]. Chem Tutor was built with Cognitive Tutor Authoring Tools [17], which facilitates rapid iterations of prototyping and pilot-testing involved in such user-centered design approaches.

In the present experiment, students worked with the atoms and electrons unit of Chem Tutor. This unit features interactive tools that students use to construct a variety of graphical representations of atoms: Lewis structures, Bohr models, energy diagrams, and orbital diagrams (see Figure 1). The tutor problems are structured as follows. First, students are prompted to think about the properties of the atom. They can use the periodic table to look up information about the atom (e.g., oxygen has eight electrons). Second, students are prompted to plan what the given representation will look like (e.g., the Bohr model of oxygen should show two shells). Third, students use an interactive tool to construct the representation of the given atom. Students receive error-specific feedback on their interactions (e.g., "The Bohr model shows all of the electrons, not only the valence electrons"). Students have to construct a correct graphical representation before they can continue. Fourth, students are prompted to make inferences from the given graphical representation about the atom (e.g., the number of valence electrons allow to approximate the number of bonds the

atom forms). Figure 2 shows an example tutor problem in which students construct the Bohr model of an oxygen atom. The interface of the problems builds up step-by-step, as shown in Figure 3.

Atoms and Electrons

Let's make the energy diagram for oxygen!

- Oxygen is in row  of the periodic table. The atomic number shows that it has  electrons and is in A-group .

Identify properties of the atom

Atoms and Electrons

Let's make the energy diagram for oxygen!

- Oxygen is in row  of the periodic table. The atomic number shows that it has  electrons and is in A-group .
- As for the first-shell electrons, oxygen's orbital contains  electrons. In the second shell, oxygen's 2s orbital has  electrons and its 2p orbitals have  electrons in total.

Plan features of the representation

Atoms and Electrons

Let's make the energy diagram for oxygen!

- Oxygen is in row  of the periodic table. The atomic number shows that it has  electrons and is in A-group .
- As for the first-shell electrons, oxygen's orbital contains  electrons. In the second shell, oxygen's 2s orbital has  electrons and its 2p orbitals have  electrons in total.
- Show the energy diagram for oxygen in the area to the left.

Construct representations with an interactive tool

Atoms and Electrons

Let's make the energy diagram for oxygen!

- Oxygen is in row  of the periodic table. The atomic number shows that it has  electrons and is in A-group .
- As for the first-shell electrons, oxygen's orbital contains  electrons. In the second shell, oxygen's 2s orbital has  electrons and its 2p orbitals have  electrons in total.
- Show the energy diagram for oxygen in the area to the left.
- Looking at the energy diagram, the energy level of the 2s is  the energy level of the 2p orbital.
- The exact exact number of bonds oxygen will form cannot be determined from this energy diagram because .

Make inferences about the atom

Figure 3. Sequence of screen shots showing how the interface updates step by step as students construct an Energy diagram.

### 3. EXPERIMENT

The experiment investigated whether Chem Tutor helps undergraduate students learn chemistry. For a detailed description of the experiment, refer to [18].

#### 3.1 Participants

117 undergraduate students from a university in the mid-western United States participated in the experiment. 79% of the students were enrolled in general chemistry for non-science majors. According to the instructor of this course, these students had no experience with the graphical representations used in the Chem Tutor unit, with the exception of the common Lewis structure. 13.4% of the students were enrolled in general chemistry for science majors, 2.5% were enrolled in advanced general chemistry. According to the instructors of these courses, these students had experience with all graphical representations used in the Chem Tutor unit. The remaining 5% of the students were not currently enrolled in a chemistry course.

#### 3.2 Assessments

Students' chemistry knowledge was assessed three times: before they started working with Chem Tutor (pretest), after they completed half of the tutor problems (intermediate posttest), and after they completed all tutor problems (final posttest). Three isomorphic test forms were used: they asked structurally identical questions but used different problems (e.g., with different atoms). The order in which students received the test forms was counterbalanced. The tests assessed reproduction and transfer of the chemistry content covered in Chem Tutor. Reproduction items used a format similar to the Chem Tutor problems. Transfer items asked students to apply the knowledge Chem Tutor covered in ways they had not been asked to do in the Chem Tutor problems. The tests included items with and without representations. In addition, spatial skills were assessed with the Vandenberg & Kuse mental rotation ability test [19]. This test presents students with a drawing of an object and asks them to identify which of four other drawings show the same object. This task requires spatial skills because students have to mentally rotate the given object to align it with the comparison objects. This test was chosen because it has been used in prior research on the impact of students' spatial skills on STEM learning [1, 2, 4, 5, 7].

#### 3.3 Procedure

The experiment took place in the laboratory and involved two sessions of about 90 minutes each. Sessions were scheduled no more than three days apart. In session 1, students first completed the mental rotation test and the chemistry pretest. They then received an introduction into using Chem Tutor. Next, they worked through half of the problems in Chem Tutor's atoms and electrons unit. At the end of session 1, students took the intermediate chemistry posttest. In session 2, students worked through the remainder of the tutor problems. At the end of session 2, they took the final chemistry posttest. All students worked on the tutor problems at their own pace and were able to finish the assigned tutor problems in the available time.

#### 3.4 Results

Results from the analysis of the test data show that there were significant learning gains on the chemistry knowledge test,  $F(2,230) = 6.18, p < .01$ . A regression of students' spatial skills on learning gains (i.e., performance on the posttest, controlling for pretest performance) showed that spatial skills were a significant predictor of learning gains ( $\beta = .34, p < .01$ ), such that students with high spatial skills showed higher learning gains than students with low spatial skills.

### 4. OPEN QUESTIONS

The finding that students with lower spatial skills had lower learning gains as the result of an intervention that relies on graphical representations is not surprising: it aligns with prior research on the role of spatial skills in STEM learning [1, 4, 5, 9]. It is conceivable that working with interactive graphical representations requires students to make sense of how abstract properties of atoms can be translated into visuo-spatial elements of graphical representations. It is well documented that this is more difficult for students with lower spatial skills [1, 4, 5, 9].

A first question that remains thus far unanswered, however, is how these difficulties affect how students interact with tutor problems. There are several aspects of the problems in Chem Tutor that may be more difficult for students with low spatial skills. First, these students may struggle with the first part of the tutor problems: identifying properties of atoms. Students with low spatial skills may have trouble retrieving facts that describe properties of atoms because they cannot imagine what an atom looks like. They might also struggle in using resources such as the periodic table to retrieve this information. Second, students with low spatial skills may struggle with the planning part of the tutor problems, because this step requires them to think about how properties of an atom can be visualized. Third, it is possible that these students struggle more when constructing graphical representations because they have to translate text-based information into visuo-spatial elements of the graphical representations. Finally, it is possible that these students struggle more in using representations to make inferences about the atom because this requires them to imagine how the visualized properties determine dynamic behavior of electrons (e.g., electron movement) and of atoms (e.g., tendency to form bonds).

A second question that remains open is how these difficulties relate to learning gains. While it is possible that all of the aspects just described are more difficult for students with low spatial skills, some difficulties may play a larger role than others in explaining why these students show lower learning gains. Understanding which difficulties account for the fact that students with lower spatial skills show lower learning gains will enable us to provide more appropriate support for these students.

### 5. FEATURE SELECTION

To investigate why spatial skills predict students' learning gains as they work with interactive graphical representations, I used a structural equation model to conduct a mediation analysis. Structural equation models provide a unified framework to test mediation hypotheses, estimate total effects, and separate direct from indirect effects. The first step in constructing a structural equation model is to determine candidate mediator variables to be included in the model. To do so, I first investigated how best to represent the knowledge students acquire as they are working on the tutor problems by comparing different knowledge component models. Second, I used the knowledge component model to generate a number of features that describe student performance during problem solving. Third, I searched for features that are predictive of learning outcome, using linear regressions.

#### 5.1 Knowledge component model

First, I constructed a knowledge component model that adequately describes knowledge students acquire when working with interactive representations to learn about atomic structure. Knowledge components are "acquired units of cognitive function or structure that can be inferred from performance on a set of related tasks" [19]. I contrasted the following knowledge component models:

1. A *single-step baseline model* that treats all problem-solving step as one skill;
2. A *step-type model* that does not distinguish between the graphical representation used in the given problem but distinguishes between step types (i.e., providing information about atoms, planning the graphical representation of the atom, constructing graphical representations, and making inferences about the atom; see Figures 2 and 3);
3. A *representation-construct model* that distinguishes between the graphical representation used in the given problem (i.e., Lewis structure, Bohr model, energy diagram, and orbital diagram; see Figure 1) for the step in which students are asked to construct the graphical representation, but that does not distinguish between graphical representations for the remaining step types;
4. A *step-type / representation model* that distinguishes between the graphical representation used in the given problem for each step types except for providing information about atoms.

Each model was evaluated as to how well it predicts student behavior during problem solving. Following standard practice in ITS research [19, 20], I considered each step in a given tutor problem as a learning opportunity for the particular knowledge component involved in the step. Student behavior was assessed based on whether a student solved the step correctly (i.e., without hints and without errors). To evaluate model fit, I used the Additive Factors Model (AFM) in the PSLC DataShop [20]. As a metric for model fit, I used 3-fold item-stratified cross validation [21]. Table 1 shows the root mean squared errors (RMSEs) for each knowledge component model. The *step-type / representation* model had the best model fit. Hence, this knowledge component model was used as a basis to generate features that describe students' learning about atomic structure with interactive graphical representations.

**Table 1. RMSEs for knowledge component models.**

Knowledge component model	Knowledge components	Item-stratified RMSE (lower is better)
Single-step baseline model	1	0.464794
Step-type model	4	0.375733
Representation-construct model	7	0.372553
Step-type / representation model	13	0.363908

## 5.2 Feature generation

Based on the step-type / representation model, I generated features that describe how students interact with the tutor problems. Students' problem-solving behaviors can be described based on the outcome (proportion of incorrect first attempts, proportion of hint requests at the first attempt, proportion of total incorrect attempts, proportion of total hint requests) and based on durations (time spent per step in total, time spent on steps with first correct attempt / steps with at least one incorrect attempt, time spent before first attempt, time spent before first attempt if it was a correct / incorrect attempt). Additionally, when students use an interactive tool (e.g., to construct representations) they can make a large variety of errors. Thus, the number of different error types when constructing representations is another measure of interest. To generate features, I computed these metrics for each knowledge component, yielding a total of 134 features (i.e., four outcome-based and six duration-based set of metrics for each of the 13

KCs, plus number of mistake types for constructing each of the four representations).

## 5.3 Search for predictive features

Since it is impractical to include all 134 features in a structural equation model, it was necessary to narrow down the number of features to consider. The most interesting features when investigating the role of spatial skills on learning outcomes are those features that are predictive of students' learning outcomes. To find predictive features, I conducted linear regressions on each set of features (i.e., proportion of correct steps, time spent on correct steps, etc.), computed for the given KCs. It was necessary to conduct separate regressions for each set of feature because the feature sets are not independent of one another. For example, the total incorrect attempts subsume the first incorrect attempts. Learning outcomes on the final posttest was the dependent variable in each linear regression model. Pretest performance was included as a predictor in all regression models. Regressions were conducted using 10-fold cross-validation. I used the results from the regression analyses to determine what characterizes predictive features. To do so, I compared the standardized coefficients and significance of features based on the metric they used and based on the KC they described. Table 2 shows the results for the regression analyses.

The goal of the selection procedure was to identify a set of predictive features that are independent of one another. Overall, features based on *knowledge components* related to planning, constructing, and making inferences were predictive of learning outcomes. However, features based on retrieving information about atoms were not predictive of learning outcomes. Thus, atoms steps were excluded from further analysis. Among the *outcome-based features*, those using proportion of incorrect first attempts and those using proportion of total incorrect attempts were equally predictive of learning outcomes. However, when excluding atoms steps, the features based on proportion of incorrect total attempts were slightly more predictive than those based on incorrect first attempts. Thus, features based in incorrect total attempts were selected for further analysis. Features based on proportion of hint requests at first attempt and proportion of total hint requests had low predictive value because hint use was generally low. Thus, these features were excluded. Features describing error types while constructing representations had high predictive value. Thus, these features were selected for further analysis. Among the *duration-based features*, those based on time spent on steps with at least one incorrect attempt as a metric were selected because they were more predictive than the other duration-based features.

Based on these findings, the following variables were selected for the structural equation model:

- Average duration of planning steps with at least one incorrect attempt (plan\_timeError)
- Average duration of representation-construction steps with at least one incorrect attempt (repr\_timeError)
- Average duration of inference steps with at least one incorrect attempt (infer\_timeError)
- Proportion of total incorrect attempts on planning steps (plan\_incorrect)
- Proportion of total incorrect attempts on representation-construction steps (repr\_incorrect)
- Proportion of total incorrect attempts on inference steps (infer\_incorrect)
- Number of error types on representation-construction steps (repr\_errorTypes)

**Table 2. Standardized coefficients for mediators in regression models, using color gradients to illustrate the strength of association with performance on the final posttest.**

predictor	outcome-based features			duration-based features					
	total incorrects	incorrect 1st attempt	error-Types	total step duration	correct step duration	error step duration	before 1st attempt	before 1st correct	before 1st error
pretest	0.275	0.281	0.307	0.364	0.356	0.258	0.334	0.372	0.305
atom	-0.002	-0.027		0.013	0.009	-0.076	0.006	-0.007	-0.054
planning-Bohr	0.112	0.082		-0.137	0.018	-0.039	0.068	0.024	0.016
planning-Energy	-0.393	-0.112		-0.163	-0.001	0.230	0.075	0.036	0.025
planning-Lewis	-0.116	-0.114		-0.025	-0.006	-0.048	-0.118	-0.093	-0.046
planning-Orbital	0.018	0.112		-0.004	0.112	-0.118	-0.066	0.07	-0.071
construct-Bohr	-0.028	0.230	-0.201	-0.080	-0.053	-0.050	0.031	-0.103	0.062
construct-Energy	-0.030	-0.174	-0.093	0.269	-0.144	0.003	0.087	-0.086	-0.155
construct-Lewis	0.203	-0.053	-0.169	-0.109	0.025	-0.113	-0.077	0.029	-0.158
construct-Orbital	-0.028	-0.119	0.139	0.056	0.045	-0.166	-0.211	0.064	-0.202
inference-Bohr	-0.030	0.011		-0.080	-0.138	-0.114	-0.017	-0.046	0.059
inference-Energy	-0.121	-0.064		0.269	0.196	0.091	0.121	0.064	0.116
inference-Lewis	0.071	0.040		0.169	0.013	-0.093	-0.023	0.025	0.053
inference-Orbital	-0.140	-0.147		-0.107	-0.044	-0.106	0.075	0.039	0.010
<i>Average of absolute values</i>	0.112	0.112	0.182	0.132	0.083	0.108	0.094	0.076	0.095

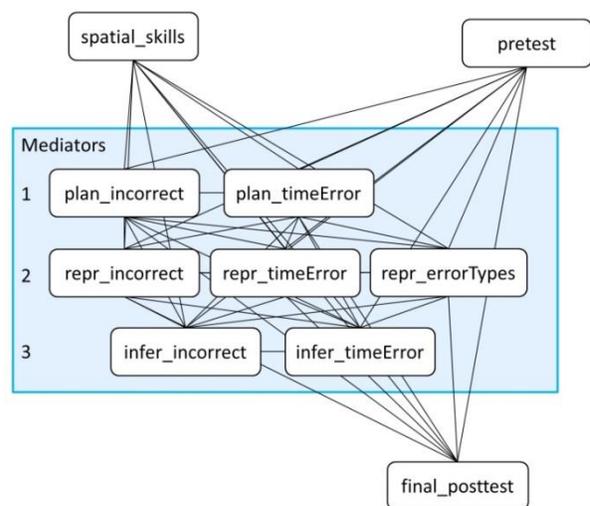
## 6. STRUCTURAL EQUATION MODEL

The goal of the structural equation model was to investigate why students with low spatial skills show lower learning gains. The structural equation model allows testing whether students' problem-solving behaviors statistically mediate the effect of spatial skills on learning gains. To carry out this analysis, I considered the variables that I identified as predictive of students' learning outcomes as potential mediators of the effect of spatial skills on learning outcomes at the final posttest, controlling for pretest.

### 6.1 Model Search

Since there are many models that might describe the nature of the effect of spatial skills on learning outcomes, I conducted a model search. Because a factor analysis indicated that the chemistry content pretest and the mental rotation ability test load onto separate factors that correlate weakly, I assumed that pretest and spatial skills are independent. I assumed that pretest is prior to the mediators and to the final posttest, that spatial skills are prior to the mediators and to the final posttest, and that mediators are prior to the final posttest. For the mediators, I assumed that planning is prior to constructing representations, which is prior to making inferences. Even under these constraints, there are at least  $2^{49}$  distinct models that are consistent with these assumptions. Figure 4 shows the fully saturated model that would be compatible with these assumptions. A fully saturated model contains all possible edges (or "effects") compatible with the assumptions. Therefore,

Figure 4 illustrates the search space of models: the search was conducted among models that had all, none, or a subset of the edges in the fully saturated model.



**Figure 2. Fully saturated model consistent with the assumptions. Mediators are highlighted in blue and organized by tiers (1 = planning; 2 = representation-construction, 3 = inference).**

To search for models that are theoretically plausible and consistent with the data, I used the Tetrad V program's<sup>1</sup> GES algorithm along with background knowledge constraining the space of models searched [22] to those that are theoretically tenable and compatible with my assumptions [23]. In the model search, each edge shown in Figure 4 is evaluated as to whether including it yields a better model fit than not, and whether it is a statistically reliable effect. As Figure 4 illustrates, there are many distinct models consistent with the background knowledge and that are plausible tests for the mediation hypothesis. Yet, it is important to know which of these models fits the data best, because parameter estimates and the statistical inferences we make about them are conditional on the model being true. Parameter estimates of models that do not fit the data well are scientifically unreliable. Thus, searching for the model that is most consistent with the data ensures that the parameters of the model can be trusted.

To conduct the model search at a technical level, I represented the qualitative causal structure of each model by a Directed Acyclic Graph (DAG). If two DAGs entail the same set of constraints on the observed covariance matrix,<sup>2</sup> then they are empirically indistinguishable. If the constraints considered are independence and conditional independence, which exhaust the constraints entailed by DAGs among multivariate normal varieties, then the equivalence class is called a *pattern* [23, 24]. The GES algorithm is asymptotically reliable,<sup>3</sup> and outputs the *pattern* with the best BIC score.<sup>4</sup> The pattern identifies features of the causal structure that are distinguishable from the data and background knowledge, as well as those that are not. The algorithm's limits lie primarily in its background assumptions involving the non-existence of unmeasured common causes and the parametric assumption that causal dependencies can be modeled with linear functions. The outcome of the model search is a structural equation model model that (1) is theoretically plausible, (2) fits the data well, and (3) contains only edges that describe statistically reliable effects.

## 6.2 Results

Figure 5 shows a model found by GES, with unstandardized parameter estimates. Table 2 shows standardized parameter estimates. Each edge is evaluated as to whether it is a reliable effect using *t*-tests, assuming an alpha-level of .05. A Bonferroni correction of the *p*-values is not necessary in a structural equation model because the significance tests are not independent. Table 2 shows the results from these tests. Altogether, the model fits the data well<sup>5</sup> ( $\chi^2 = 32.77$ ,  $df = 27$ ,  $p = .21$ ).

<sup>1</sup> Tetrad, freely available at [www.phil.cmu.edu/projects/tetrad](http://www.phil.cmu.edu/projects/tetrad), contains a causal model simulator, estimator, and over 20 model search algorithms, many of which are described and proved asymptotically reliable in [24].

<sup>2</sup> An example of a testable constraint is a vanishing partial correlation, e.g.,  $\rho_{XY.Z} = 0$ .

<sup>3</sup> Provided the generating model satisfies the parametric assumptions of the algorithm, the probability that the output equivalence class contains the generating model converges to 1 in the limit as the data grows without bound. In simulation studies, the algorithm is quite accurate on small to moderate samples.

<sup>4</sup> All the DAGs represented by a pattern will have the same BIC score, so a pattern's BIC score is computed by taking an arbitrary DAG in its class and computing its BIC score.

<sup>5</sup> The usual logic of hypothesis testing is inverted in path analysis: a *low* *p*-value means the model can be rejected.

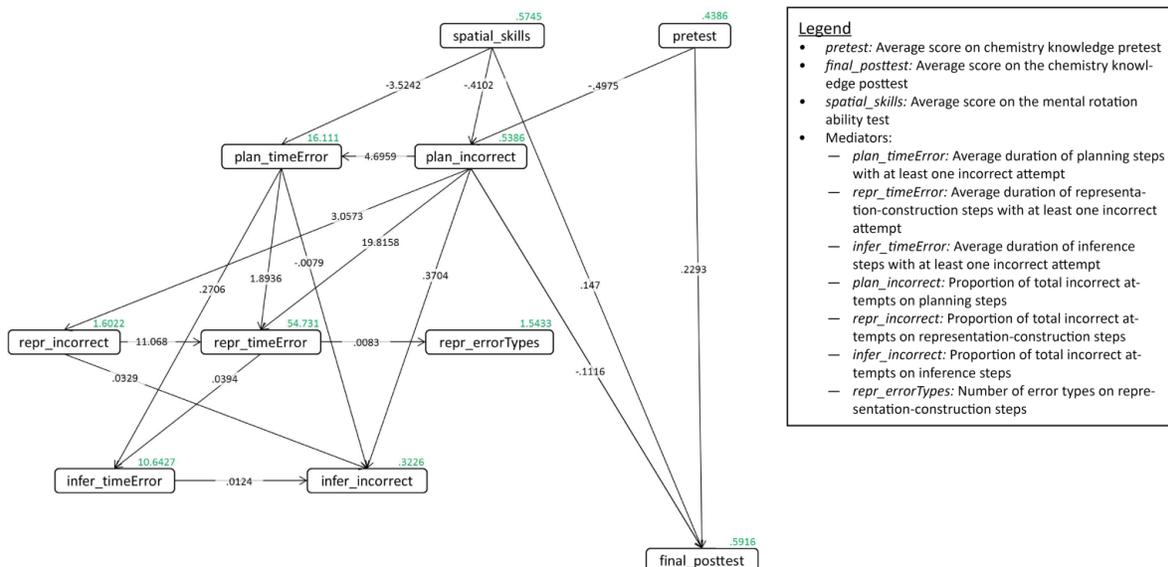
**Table 3. Parameter estimates (PE) for all edges and result of *t*-tests assessing whether the PE is significantly different from 0.**

Edge from...	to...	PE	<i>t</i>	<i>p</i>
infer_timeError	infer_incorrect	.0124	3.2999	.0013
plan_incorrect	final_posttest	-.1116	-2.4706	.0150
plan_incorrect	infer_incorrect	.3704	6.3202	< .001
plan_incorrect	plan_timeError	4.6959	3.7759	< .001
plan_incorrect	repr_incorrect	3.0573	9.5622	< .001
plan_incorrect	repr_timeError	19.8158	2.8253	.0056
plan_timeError	infer_incorrect	-.0079	-2.2244	.0281
plan_timeError	infer_timeError	.2706	3.1104	.0024
plan_timeError	repr_timeError	1.8936	4.7591	< .001
pretest_content	final_posttest	0.2293	2.9336	.0040
pretest_content	plan_incorrect	-.4975	-3.1908	.0018
repr_incorrect	infer_incorrect	.0329	2.6633	.0088
repr_incorrect	repr_timeError	11.068	7.529	< .001
repr_timeError	infer_timeError	.0394	3.1303	.0022
repr_timeError	repr_errorTypes	.0083	8.6891	< .001
spatial_skills	final_posttest	.147	1.9078	.0589
spatial_skills	plan_incorrect	-.4102	-2.6326	.0096
spatial_skills	plan_timeError	-3.5242	-1.639	.1039

The final model shows that spatial skills have a direct positive effect on students' learning outcomes at the final posttest. Furthermore, spatial skills predict students' problem-solving behaviors while they are planning the graphical representation, which, in turn, has an effect on outcome-based and duration-based measures of problem-solving behaviors while they construct the graphical representation and while they make inferences from graphical representations about domain-relevant concepts. Only the proportion of incorrect attempts on planning steps mediates the effect of spatial skills on learning outcomes: *plan\_incorrect* is the only variable that mediates the effect of *spatial\_skills* on *final\_posttest*. The edge from *spatial\_skills* to *plan\_incorrect* shows that a student with a perfect score on the spatial skills test makes .4102 fewer incorrect attempts per step than a student with the lowest possible score on the spatial skills test. The edge from *plan\_incorrect* to *final\_posttest* means that a student who makes one incorrect attempt per step scores 11.16% lower on the final posttest than a student who makes no incorrect attempts (controlling for pretest performance). In sum, the mediated effect of *spatial\_skills* to *final\_posttest* through *plan\_incorrect* is  $.4102 * .1116 = .0458$ . Incorrect attempts while planning representations only partially mediate the effect of spatial skills on learning outcomes, because there is a direct effect of .147 from *spatial\_skills* to *final\_posttest*. Yet, making more incorrect attempts while planning graphical representations explains a considerable portion (about 25%) of the effect of spatial skills on learning outcomes.

## 7. CONCLUSIONS

The goal of the mediation analysis was to investigate (1) which aspects about working with interactive representations are harder for students with low than with high spatial skills and (2) which of these aspects explain why students with low spatial skills show lower learning gains than students with high spatial skills. With respect to the first question, results show that spatial skills have an effect on all aspects of students' problem-solving behaviors,



**Figure 3. Final structural equation model with unstandardized parameter estimates. Green values show means.**

except for looking up information about the atoms: planning, constructing, and making inferences from graphical representations. Spatial skills affect outcome-based measures of performance as well as duration-based measures of performance. Yet, the structural equation model shows that planning has a central role: students' ability to plan graphical representations has an impact on all further problem-solving behaviors as students construct graphical representations and make inferences about domain-relevant concepts based on the graphical information. With respect to the second question, results show that planning is the only aspect that mediates the effect of spatial skills on learning gains. The difficulties that students with low spatial skills have in constructing representations and in making inferences may merely be symptomatic—they do not explain why these students show lower learning gains. Only the fact that students with low spatial skills tend to struggle more in planning representations explains why they benefit less from interactive representations.

Why might students' ability to plan graphical representations be so strongly affected by their spatial skills? Planning a representation requires students to describe what the representation should look like, based on the properties of the atom. This task requires them to mentally picture visuo-spatial features based on text-based information about the atom's properties. This takes more cognitive effort for students who struggle with such visuo-spatial tasks. Hence, these students are at risk of cognitive overload during planning, which jeopardizes learning. Perhaps difficulties in planning are amplified by the fact that the interactive representation tool is not visible during the planning step (see Figure 3).

Why might the ability to plan representations determine students' learning gains? Learning with graphical representations means that students have to visualize new information externally while integrating this information with their internal mental models of the domain-relevant concepts [26]. Planning might play a central role because it helps students organize their initial mental model of the domain-relevant concepts. Having a well-organized initial mental model might facilitate integration of new information into this model: learning occurs as students expand and repair their mental models throughout the learning intervention, for instance by self-explaining how the new information relates to their initial mental models [27].

In summary, the findings from the mediation analysis shed light into the broader theoretical question of how spatial ability affects learning outcomes in STEM. Spatial skills seem to be important because students' benefit from interactive representations depends on their ability to mentally visualize abstract concepts *before* they use an external representation to visualize the concept. Mental visualization may play a key role in students' learning of abstract concepts because it allows students to integrate new information into their mental models. These findings also yield new hypotheses about the practical question of how best to support students with low spatial skills. These students might benefit from receiving additional assistance in planning graphical representations. They might benefit from seeing the interactive representation tool during the planning steps, so that they can more easily visualize the representation. They may also benefit from receiving examples of successful planning. It would be interesting to investigate whether such support increases learning gains for students with low spatial skills. In light of the interpretation that planning is so important because it helps students organize their initial mental models, it would be interesting to conduct a think-aloud study to assess whether, indeed, helping students plan representations facilitates mental model integration.

Several limitations of the present analysis need to be discussed. First, performance on planning steps only partially mediates the effect of spatial skills on learning outcomes. Thus, there might be other mediators that we did not assess. Further research is needed to investigate other aspects of problem solving that explain why students with low spatial skills tend to show lower learning gains. Second, the data is correlational: it is impossible to randomly assign students to having "low" or "high" spatial skills. As in any correlational data set, there may be other unknown factors that affect the effects of interest. Third, the structural equation model assumes linear relations between the variables in the model. This assumption is reasonable but not infallible. Finally, the analysis is based on a sample of 117 students. Even though that is sizable compared to many ITS studies, model search reliability increases with sample size, but decreases with model complexity. Hence, it is impossible to put confidence bounds on finite samples [21].

To conclude, the mediation analysis presented in this paper yields new insights into why students with lower spatial skills struggle in

learning with interactive graphical representations. It seems that planning representations is a crucial aspect of learning success. This finding yields new hypotheses about what types of interventions these students may benefit from. Even though the present paper merely presents a first step towards better understanding the mechanisms that underlie the “the-rich-get-richer” rule in STEM domains, it may help us address the unfortunate fact that students with low spatial skills tend to show lower achievements in STEM domains and they are less likely to pursue careers in these domains. In other words, this paper is a first step towards creating an “everyone-gets-richer” rule for STEM learning.

## 8. ACKNOWLEDGMENTS

This work was supported by the UW-Madison Graduate School and WCER. We thank Teri Larson, Ned Sibert, Stephen Block, Amanda Evenstone, and Jocelyn Kuhn for their help with recruitment, and Sally Wu and the RAs in the Learning, Representations, & Technology Lab for their help in conducting the experiment.

## 9. REFERENCES

- [1] Uttal, D.H., Meadow, N.G., Tipton, E., Hand, L.L., Alden, A.R., Warren, C., Newcombe, N.S.: The malleability of spatial skills: a meta-analysis of training studies. *Psychological Bulletin* 139, 352-402 (2013)
- [2] Wai, J., Lubinski, D., Benbow, C.P.: Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology* 101, 817-835 (2009)
- [3] Barnea, N., Dori, Y.J.: High-school chemistry students' performance and gender differences in a computerized molecular modeling learning environment. *Journal of Science Education and Technology* 8, 257-271 (1999)
- [4] Stieff, M.: Sex differences in the mental rotation of chemistry representations. *Journal of Chemical Education* 90, 165-170 (2013)
- [5] Stieff, M.: Mental rotation and diagrammatic reasoning in science. *Learning and Instruction* 17, 219-234 (2007)
- [6] Kozma, R., Russell, J.: Students becoming chemists: Developing representation competence. In: Gilbert, J. (ed.) *Visualization in science education*, pp. 121-145. Springer, Dordrecht, Netherlands (2005)
- [7] Stieff, M., Hegarty, M., Deslongchamps, G.: Identifying representational competence with multi-representational displays. *Cognition and Instruction* 29, 123-145 (2011)
- [8] Griffiths, A.K., Preston, K.R.: Grade-12 students' misconceptions relating to fundamental characteristics of atoms and molecules. *Journal of Research in Science Teaching* 29, 611-628 (1992)
- [9] Hegarty, M., Waller, D.A.: Individual differences in spatial abilities. In: Shah, P., Miyake, A. (eds.) *The Cambridge handbook of visuospatial thinking*, pp. 121-169. Cambridge University Press, New York, NY (2005)
- [10] Clements, D.H.: 'Concrete' Manipulatives, Concrete Ideas. *Contemporary Issues in Early Childhood* 1, 45-60 (1999)
- [11] Ai-Lim Lee, E., Wong, K.W.: Learning with desktop virtual reality: Low spatial ability learners are more positively affected. *Computers & Education* ahead of print (2014)
- [12] VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist* 46, 197-221 (2011)
- [13] Tuckey, H., Selvaratnam, M., Bradley, J.: Identification and rectification of student difficulties concerning three-dimensional structures, rotation, and reflection. *Journal of Chemical Education* 68, 460-464 (1991)
- [14] Davidowitz, B., Chittleborough, G.: Linking the macroscopic and sub-microscopic levels: Diagrams. In: Gilbert, J.K., Treagust, D.F. (eds.) *Multiple representations in chemical education*, pp. 169-191. Springer, Dordrecht, Netherlands (2009)
- [15] Rau, M.A., Michaelis, J.E., Fay, N.: Connection making between multiple graphical representations: A multi-methods approach for domain-specific grounding of an intelligent tutoring system for chemistry. *Computers and Education* 82, (2015)
- [16] Ainsworth, S.: DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction* 16, 183-198 (2006)
- [17] Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education* 19, 105-154 (2009)
- [18] Rau, M.A., Wu, S.P.W.: ITS support for conceptual and perceptual processes in learning with multiple graphical representations. submitted to AIED 2015 (under review)
- [19] Peters, M., Laeng, B., Latham, K., Jackson, M., Zaiyouna, R., Richardson, C.: A Redrawn Vandenberg & Kuse Mental Rotations Test: Different Versions and Factors that affect Performance. *Brain and Cognition* 28, 39-58 (1995)
- [20] Koedinger, K.R., Corbett, A.T., Perfetti, C.: The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science* 36, 757-798 (2012)
- [21] Koedinger, K.R., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the EDM community: The PSLC Data-Shop. In: Romero, C. (ed.) *Handbook of educational data mining*, pp. 10-12. CRC Press, Boca Raton, FL (2010)
- [22] Stamper, J.C., Koedinger, K.R., McLaughlin, E.A.: A Comparison of Model Selection Metrics in DataShop. In: D'Mello, S.K., Calvo, R.A., Olney, A. (eds.) *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*, pp. 284-287. International Educational Data Mining Society (2013)
- [23] Chickering, D.M.: Optimal Structure Identification with Greedy Search. *Journal of Machine Learning Research* 3, 507-554 (2002)
- [24] Spirtes, P., Glymour, C., Scheines, R.: *Causation, Prediction, and Search*. MIT Press (2000)
- [25] Pearl, J.: *Causality: Models, Reasoning, and Inference*. Cambridge University Press (2000)
- [26] Bodner, G.M., Domin, D.S.: Mental models: The role of representations in problem solving in chemistry. *University Chemistry Education* 4, 24-30 (2000)
- [27] Wylie, R. and Chi, M.T., 2014. The Self-Explanation Principle in Multimedia Learning. In *The Cambridge Handbook of Multimedia Learning*, R.E. Mayer Ed. Cambridge University Press, New York, NY, 413-43