

# Short Papers

# Investigating Swarm Intelligence for Performance Prediction\*

Mohammad Majid al-Rifaie<sup>†</sup>  
Goldsmiths, University of London  
Department of Computing  
London SE14 6NW, UK  
m.majid@gold.ac.uk

Matthew Yee-King  
Goldsmiths, Uni of London  
Department of Computing  
London SE14 6NW, UK  
m.yee-king@gold.ac.uk

Mark d’Inverno  
Goldsmiths, Uni of London  
Department of Computing  
London SE14 6NW, UK  
dinverno@gold.ac.uk

## ABSTRACT

This paper proposes a new technique for analysing the behaviour of students on an online course. This work considers a range of social learning behaviours supported in our recently designed and implemented collaborative learning system which supports students giving and receiving feedback on each other’s developing work and practice. The course was delivered to several thousand students on Coursera during which students were directed onto our social learning environment to take part in group work and assessment activities. This work introduces a swarm intelligence technique, Stochastic Diffusion Search (SDS), and shows how it can be adapted and applied to our data in order to perform classification tasks. The novelty of the approach is not only in using this technique, but also applying it to data linked to *social behaviour* (i.e. how students interact with each other) which differentiates the work apart from many clickstream analysis studies. This paper investigates what combined activity is the best predictor of success or failure in the course. The aim is to argue that the results obtained using the proposed approach indicate the promising potential of predicting students performance through applying swarm intelligence technique to social behaviours. This work has a number of potential benefits including designing better social learning systems, designing more effective social learning and assessment exercises, and encouraging disengaged students. In addition, this work is an important step in addressing our long term goal of evidencing how critical student learning takes place as they give and receive feedback to and from each other on work in progress.

## Keywords

Social learning, swarm intelligence, education system modelling, MOOC

## 1. INTRODUCTION

\*(Does NOT produce the permission block, copyright information nor page numbering). For use with ACM\_PROC\_ARTICLE-SP.CLS. Supported by ACM.

<sup>†</sup>Corresponding author

Increasingly researchers are focusing on the significance of social learning and investigating its impact within the various online learning environments. Acknowledging the importance of collaboration and ‘teamwork’, as an embedded element in the Massive Open Online Courses (MOOCs), this method of learning is desirable for many employers who rely on highly collaborative and online-based works. Our programme of work is concerned with designing a novel learning technology, online courses and assessments, which provide us with a range of data we can use to understand how learning takes place through online social interaction. Our pedagogy is influenced by our home institution’s ‘art-school’ pedagogy across practice-based subjects (such as art, music and design) where students learn by sharing ‘work in progress’ within tutor groups and giving and receiving feedback to each other. The aim of this work is to use learning analytics to build strong arguments for the adoption of social learning pedagogies supported by innovative technology. Therefore this paper focuses on extracting information from *social learning activity logs*, not the full range of more traditional courseware access and activity logs. The objective is to gain a better understanding if these activities have any measurable relation to learning, and if so which are the most important activities and in which combinations. The analysis presented here is a first step in that direction, where the attempt is to predict if students will pass or fail a course, using only low level user interface telemetry data gathered from our social learning platform. Given the undeniable significance of data classification in different and diverse scientific domains (e.g. computer science, psychology, medicine), various techniques have been proposed over the years. Nature-inspired metaheuristic algorithms are among one of the categories which aimed at providing solutions to this problem.

In this paper a novel method in addressing data classification in the context of educational data is used where a swarm intelligence algorithm is adapted for this purpose. A recent review [2] details the extensive applications of this algorithm in the last two decades in various fields (e.g. discrete and global optimisation, pattern recognition, resource allocation, medical imaging, etc).

The research questions which drive this paper are as follows:

1. How can the proposed swarm intelligence technique (SDS) be applied to educational data?
2. What kinds of social learning activities, and what combinations of social learning activities are the best predictors?
3. Does social interaction data contain strong predictive potential of student success?
4. How does an SDS analysis of social learning data help us in designing and delivering learning activities, in improving social/group learning activities, and in building better social learning systems?

In this paper, first Stochastic Diffusion Search (SDS) algorithm is explained, detailing its behaviour and highlighting one of its main features (i.e. partial function evaluation). Then, an introduction is given to the classification problem in general followed by a brief section on the nature of the educational dataset used in this paper and the features available from the dataset. After elaborating on the data in the datasets in the context of the work, the swarm intelligence algorithm used is adapted for the purpose of the experiments conducted in this paper and the results are reported. A discussion on the behaviour of the proposed algorithm is presented showing its potential in using all the available features as well as identifying the most significant features. Finally, the paper is concluded with the summary of the research reported in the paper along with directions for future research.

## 2. RELATED WORK

With the increasing use of online learning platforms, a large number of researchers have been working on predicting grades from students performance over the course of the studies. This topic of research is of importance because, for example, only in the United States several hundred thousand students drop out of high school every year and perhaps interventions can provide the means to reduce the number of those falling behind in their studies [1, 7]. With the growing interest in MOOCs as alternative or adjunct learning platforms, behaviour prediction has attracted the attention of many educational data analyst, such as Brady et al. [15] who used higher granularity temporal information for their analytics work; in another work, Macfadyen et al. [8] explain the concept of “an early warning system” for educator, aiming to provide the means for the educators to intervene with an appropriate set of actions to improve the performance of the weaker students; a similar work was presented by Rogers et al. [11] which aims to identify students at risk of failure. The predictive power of demographics versus activity patterns in MOOCs are discussed by Brooks et al. [3] focusing on whether it is possible to find a link between performance and demographics. Other researchers, such as Coleman et al. [4] or Elbadrawy et al. [6], have also been exploring whether it is feasible to identify behavioural patterns for prediction. In addition to attempting to improve students performance, Yang et al. [14] have been focusing on the concept of dropouts which is a critical challenge for online courses. Considering the above recent work, it is evident that extracting useful knowledge from education data should ultimately be incorporated in the design of the online systems. In a recent work by researchers from Harvard

University and MIT, Whitehill et al. [13] emphasised on the importance of intervention and especially automatic intervention in MOOCs in order to take measures to reduce the number of students quitting; they claim that their proposed system might encourage students to return into the course. In another work, by Rollinson and Brunskill, [12] emphasis has been put on the importance of coupling predictive models with an alternative student model and policy (which constitute the core of the Intelligent Tutoring Systems), focusing again on the importance of using predictive models along with other tools. Having mentioned the above research, it is important to state that arguably one of the important features in MOOCs is enabling learners to discuss their work with their peers and receive feedback. In a recent research, Olsen et al. [9] direct the prediction power towards collaborative learning environment; in their work, they argue that by adding collaborative learning features they were able to enhance their understanding on the impact of collaborative learning. Tightly related to the mentioned work, the importance of social centrality in the context of MOOCs is discussed by Dowell et al. [5] where they adopt an approach, which uses language and discourse as a tool to explore the association with the existing and established measures related to learning (i.e. traditional academic performance and social centrality). While this work does not endorse or reject the impact of social learning, it clearly shows an increasing interest in exploring the impact of collaborative learning.

## 3. STOCHASTIC DIFFUSION SEARCH

Stochastic Diffusion Search (SDS) [2] which was first proposed in 1989 is a probabilistic approach for solving best-fit pattern recognition and matching problems. SDS, as a multi-agent population-based global search and optimisation algorithm, is a distributed mode of computation utilising interaction between simple agents. Its computational roots stem from Geoff Hinton’s interest in 3D object classification and mapping and its applications span from continuous optimisation to medical imaging. The SDS algorithm commences a search or optimisation by initialising its population and then iterating through two phases: the test and diffusion phases. In the test phase, SDS checks whether the agent hypothesis is successful or not by performing a hypothesis evaluation which returns a boolean value. Once the activity (i.e their status as being ‘true’ or ‘false’) of all the agents are determined, successful hypotheses diffuse across the population and in this way information on potentially good solutions spreads throughout the entire population of agents. In other words, each agent recruits another agent for interaction and potential communication of hypothesis. The spreading of information occurs during the diffusion phase.

In standard SDS (which is used in this paper), *passive recruitment mode* is employed. In this mode, if the agent is inactive, a second agent is randomly selected for diffusion; if the second agent is active, its hypothesis is communicated (*diffused*) to the inactive one. Otherwise there is no flow of information between agents; instead a completely new hypothesis is generated for the first inactive agent at random. Therefore, recruitment is not the responsibility of the active agents. In this work, activity of each agent is determined when its fitness is compared against a random agent (which is different from the selecting one); if the selecting agent has a better fitness (smaller value in minimisation problems)

Table 1: The list of features logged, along with examples of the total figures for a single student. The last column represents the grade correlation of each individual figure.

	Description	Example	Corr
F1	Play video	199	0.41
F2	Delete a reply	16	0.12
F3	Open item in search result list	0	0.15
F4	Report problem with media	22	0.48
F5	Load media	7580	0.41
F6	Report problem with reply	24	0.26
F7	Delete an annotation	0	0.19
F8	Save after edit	0	0.15
F9	View my files	954	0.40
F10	View set of shared files	8865	0.41
F11	Save after edit	0	0.11
F12	Delete video	0	0.18
F13	Periodically log and comment when video is playing	1928	0.30
F14	Play region and view thread	1313	0.53
F15	Save user profile	32	0.23
	Course final grade	100	1.00

than the randomly selected agent, it will be flagged as active, otherwise inactive. A higher rate of inactivity boosts exploration, whereas a lower rate biases the performance towards exploitation.

#### 4. CASE STUDY AND DATASET

The analysis presented in this paper is based on a dataset gathered during a seven week creative programming course on Coursera which ran in Summer 2014. The course presented students with a series of worked example programs written using Processing [10] that were either musical, graphical or game based. It was assessed using weekly quizzes and three, biweekly peer assessments. The peer assessments required the students to select one of the tutor-supplied worked examples and extend it in some way of their choosing. They then had to create a five minute screencast video wherein they explained the changes they had made from the example code and demonstrated the running program. This video was uploaded to our social learning system and then a link to this was submitted to the main MOOC LMS. Our system allowed them to place comments along the timeline of the video and to view a range of suggested content from other students, such as highly commented and un-commented videos. Our system collects detailed logs of certain interface elements that the user clicked on or moused over, including a user id and a timestamp. The data set used in this paper consists of these clickstream logs plus final grades achieved on the course. There were a total of 993 students who created logs on our system and gained a final grade on the Coursera platform. The dataset spanned a period of about seven weeks. Each student's log data and final grade was converted into a feature vector containing totals for all of the observed log types taken over the entire time period of the study. Table 1 shows an example of such a vector. The research began by attempting to correlate individual elements of the vector to *final\_grade* but individual correlations were statistically insignificant to predict grades so instead a multivariate classification approach is attempted, the results of which form the remainder of this paper. The main aim was to label students as pass ( $\geq 50$ ) or fail ( $< 50$ ).

#### 5. APPLY THE SDS ALGORITHM

Here the process through which the SDS algorithm was adapted to perform the classification tasks is detailed and the steps taken during the *test* and *diffusion* phases are explained. In order to apply this swarm intelligence algorithm to the dataset the following are considered:

- **Search space** is the entire dataset
- **SDS hypothesis** refers to a student record
- **Student attributes:** Each student record has fifteen attributes or features (i.e. *play*, *report\_media*, *region\_block*, etc; see Table 1).
- **Micro-features:** The fifteen features of each student record are considered the micro-features of the hypothesis. Therefore each SDS hypothesis has fifteen micro-features referring to the attributes of the student.

Next, the phases used in SDS algorithm are highlighted and each phase is described briefly in the context of the dataset presented.

During the *initialisation phase*, one student is chosen randomly from the dataset and is set as a model. Then each agent is randomly associated with a student record from the search space. During the *test phase*, each agent (which is already allocated to a student) randomly picks one of the fifteen micro-features and compares its value against that of the model. If the difference between the two corresponding micro-features is within a specific threshold,  $\tau_d$  (where  $\tau$  is the threshold and  $d$  is the dimension) the agent becomes active, otherwise inactive. The process in the *diffusion phase* is the same as the one detailed in the algorithm description: each inactive agent picks an agent randomly from the population; if the randomly selected agent is active, the inactive agent adopts the hypothesis of the active agent (i.e. they refer to the same student as their hypothesis), otherwise the inactive agent picks a random student from the dataset.

**Categories, Classes and Termination** The agents iterate through the test and diffusion phases again until all agents are active. At this stage, the students referred to by all the active agents are assigned to a category. Additionally, the number of active agents on each student is logged. Once a category is determined, the process is repeated from the initialisation phase where agents are initialised throughout the search space and the first student which has not yet been assigned to any categories is set as the new model. Then the algorithm iterates through the test and diffusion phases until all students are allocated to a category. Finally, categories form the classes, and when there exist students that belong to more than one class, they will be allocated to the one which has attracted a larger number of active agents. The only tunable parameters for SDS is the swarm size,  $N$  which is empirically set to  $N = 10,000$ . Threshold,  $\bar{\tau}$ , which is the acceptable distance between the model and other samples for each dimension,  $d$ , is calculated using the following formula:

$$\bar{\tau}_d = \sum_{i=1}^c \left| \frac{\text{MAX}(\bar{I}_{id}^t) - \text{MIN}(\bar{I}_{id}^t)}{c} \right| \quad d = [1, 2, \dots, 15] \quad (1)$$

where  $c$  is the number of student types or classes in the dataset (i.e. pass and fail);  $\bar{I}_{id}^t$  represents the value of  $i^{\text{th}}$  student with type  $t$  and dimension  $d$ . There are 2 student types ( $c = 2$ ) and the dimensionality of the problem is 15 (see Table 1). Therefore the difference between the minimum and maximum values in each band (e.g. pass and fail) is calculated, then the sum of the differences in each dimension is averaged and used to calculate the threshold. Using the formula above the threshold  $\bar{\tau}$  is calculated using the

Table 2: Weekly breakdown of and fail/pass rate

	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7
Active students	245	974	629	683	488	528	265
Ratio	25%	98%	63%	69%	49%	53%	27%
% of fails	28%	39%	28%	16%	10%	5%	2%
% of passes	72%	61%	72%	84%	90%	95%	98%

training dataset. Using the threshold vector presented, if the randomly picked model falls on the first class (e.g. the fail class), it is likely that the active agents have a bigger presence in this class. It is worth noting that while in some iterations there is a high presence of active agents for some students, in some other iterations there is a high number of inactive agents on the same students. The reason why a student record could make an agent active in one iteration and inactive in another can be explained through SDS’s random micro-features selection: each record consists of fifteen micro-features (the same as the number of attributes for each student), therefore if an agent picks one of the micro-features that are within the threshold, the agent becomes active, but if it randomly picks one of the other micro-features, the agent becomes inactive. Deducing from this, it is evident that having more micro-features within the range of the model results in more agents becoming (and staying) active, and as a result forming a stable category.

## 6. EXPERIMENTS AND RESULTS

In this section, the results of several experiments are reported along with a discussion on the relevance of the experiments to the research questions. The total number of students who used the online learning platform and obtained a final grade was 993. The number of active students each week and the fail/pass rate of students are detailed in Table 2, and the SDS algorithm is used as the classifier.

### 6.1 Experiment I: Weekly data analysis

The logged actions of all students who have participated in the previous and current weeks are cumulated and fed into the system for analysis.

One of the important elements in the cumulative data is the distribution of fail and pass in each of the training and test datasets. Fig. 1 shows this distribution in the test dataset. Note that the training datasets will have the same distribution as the test dataset. As illustrated in the figure, other than the first week, in the rest of the week, the cumulative data shows 39% and 61% of the data belonging to the fail and pass categories respectively. The classifier is trained and the prediction accuracy of the classifier is evaluated on the test datasets.

Table 3 and Fig. 2 show the weekly prediction-accuracy on the test datasets. As expected, and due to the presence of more data as students progress to the next weeks, there is a gradual increase in the prediction accuracy of the swarm

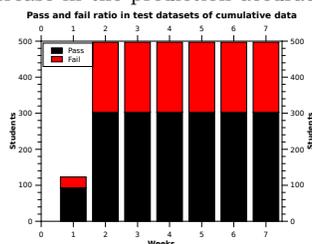


Figure 1: Pass / fail ratio in test datasets of cumulative data

Table 3: Weekly accuracy percentages

	Mean	Median	StDev	Min	Max
Week1	38.40	39	3.67	32	46
Week2	46.97	47	1.59	45	53
Week3	59.93	60	2.83	49	64
Week4	72.07	74	6.44	54	80
Week5	74.37	78	8.83	47	83
Week6	82.30	84.50	5.67	59	87
Week7	80.67	84	9.47	50	88

intelligence classifier. Looking at the maximum value in Table 3, the prediction accuracy rises to 88% on week 7. The notable increase in the accuracy starts in week 4 (i.e. with median accuracy of 74% and the maximum accuracy of 80%, allowing the teachers to have a rough estimate about the students who are likely to pass or fail. The results reported in this paper are based on 30 runs for each experiment.

### 6.2 Experiment II: Analysis of feature vector

As highlighted before, one of the main purposes of analysing the presented data is identifying weaker students as early as possible and therefore finding ways of improving their performance. However, there are many features collected from the online learning platform and identifying the “more relevant” features from the entire feature vector (of size 15) is of importance. Therefore, each of the features, have been singled out and used both for training the swarm intelligence classifier as well as the evaluation phase. The summary of the solo performance of these features are reported in Fig. 3 and Table 4. For instance, feature 13 (F13 or ‘playing’) in all weeks (except week 1, 2 and 3) is the most influential feature and has returned the highest prediction accuracy. While the grade correlation of this feature is only 0.41, this finding highlights the role of watching videos in the learning process. Knowing what the feature represents, its value is evident and the algorithm proved capable of identifying this important feature. Identifying the most influential features would entail that the analysis could be focused on the  $n$  most important features, instead of stretching the computational power to consider all the input features for predication analysis. The results in this section demonstrate that there could exist some individual features which would provide stronger prediction power when used individually than along with the other features.

### 6.3 Experiment III: Feature combinations

As shown in Table 4, in order to identify the important features, the three most influential features in each week are labelled 1-3 in brackets. The impact of each feature is calculated by giving the weights of 6 to the most influential feature (shown as (1) in the table), and 3 and 1 to the second two influential features (shown as (2) and (3) in the table). The impact of each feature is then calculated using the aforementioned weights. The six most important features are listed below in the order of importance:

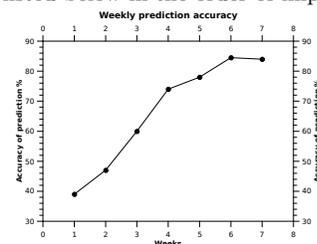


Figure 2: Prediction accuracy of the weekly cumulative data

Table 4: Analysing the impact of individual features (1-15). Prediction accuracies are shown in percentages. The three most influential features in each week are labelled 1-3. The impact of each feature is calculated by giving the weights of 6 to the most influential feature (shown as (1)), and 3 and 1 to the second two influential features (i.e (2) and (3)). The impact of each feature is calculated using the weights.

	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Impact
<b>F1</b>	32	39	49	74(2)	76(2)	84(1)	83(2)	15
<b>F2</b>	32	39	39	39	39	39	39	
<b>F3</b>	34	45	42	44	45	46	47	
<b>F4</b>	32	39	39	54	34	41	74	
<b>F5</b>	45(3)	59	65(1)	71(3)	75(3)	73(3)	74	9
<b>F6</b>	32	39	39	41	39	39	39	
<b>F7</b>	32	39	39	39	39	39	39	
<b>F8</b>	32	39	39	39	39	46	45	
<b>F9</b>	50(2)	61(2)	57	68	70	78(2)	77	9
<b>F10</b>	58(1)	62(1)	65(1)	69	71	72	73	18
<b>F11</b>	32	39	39	39	39	39	39	
<b>F12</b>	32	39	39	39	39	39	39	
<b>F13</b>	32	39	58(3)	82(1)	83(1)	84(1)	85(1)	25
<b>F14</b>	38	52(3)	60(2)	71(3)	74	78(2)	82(3)	9
<b>F15</b>	32	39	40	40	40	40	40	

1. F13: Periodically log when video is playing
2. F10: View set of shared files
3. F01: Play video
4. F05: Load media
5. F09: View my files
6. F14: Play region and view thread.

The top six features include a combination of *individual* learning activities (e.g. playing a video to watch, as well as viewing the files saved by the student themselves) and *social* learning activities (e.g. periodically making notes and logging information while watching a video, which could be uploaded by the student themselves or their classmates, knowing that the logged items are visible to the rest of the students) all contributing to the learning process. Investigating the above list, one of the interesting observations is that the social learning activity (of interacting with the posted video) has had the largest score (i.e. 25 as shown in Table 4) and is identified as the most important feature.

In the first part of this experiment, the six highest impact features shown before are selected as input to the system and results are demonstrated in Table 5. While the results are comparable to the previous experiment when all the features were used, the outcome exhibits a slight reduction in the prediction accuracy which could be due to some of the conflicting nature of the features (e.g. combining features which are as diverse as having the impact of 25 and 9). Please note that this hypothesis should be treated with caution as a more in-depth analysis is required to verify this thought. In the second experiment of this section (and in an attempt to explore the previous hypothesis), only two of most significant features (which are the social learning features) are used; the two features used are F13 (periodically log when video is playing) and F10 (view set of shared files). As shown in Table 6, the results demonstrate the highest prediction

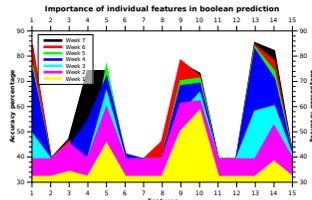


Figure 3: Impact of using individual features. Layers in this diagram represents accuracy of features in each week.

accuracy found on this dataset from week 4 of the term. The median prediction accuracy for week 4 is 83% which is 10% and 9% higher than when six most important features and all features are used respectively (see Tables 3 and 5). Comparing the prediction accuracy reported in Tables 3, 5 and 6 shows that while using the two most important social features, does not improve the prediction accuracy at the very early stages of the term (week 1, 2 and 3), it does enable a stronger prediction from the middle week (week 4) onwards. While this may or may not be extendible to other case studies, this finding highlights the usefulness of investigating the positive or negative nature of social features in online learning environment.

## 6.4 Discussion

Here, the key research questions raised in Section 1 are discussed next and various aspects of the findings are analysed. As stated in the first research question, this paper applies the Stochastic Diffusion Search (SDS) to classify educational data. The potential and strength of the this algorithm is demonstrated in the results and the flexibility of the algorithm to deal with various feature vector is also highlighted. Given SDS's existing 'partial function evaluation' feature (i.e. each micro-feature, or attribute, is used independently of the others in the test phase), and the resulting low computational cost of comparing samples, this algorithm is likely to be particularly useful when applied to problems with huge dimensionality, which is usually the case in educational data analysis. In this context, the link between cheap computational cost and scalability is the subject of an ongoing research. To address the second research question, three experiments are run (see Fig. 4). Neither of the three experiments (using all features, 6 best features, and 2 best social features) are able to provide a reliable prediction in the first three weeks (e.g. less than 60%) of this seven-week course analysed in this paper; it is worth noting that in the first three weeks, when the social features are solely used in the analysis, the algorithm exhibits the worst outcome, possibly due to the lack or reduced social interactions among the students in the very first a few weeks. However, looking at the performance of the algorithm in weeks 4-7, it can be seen that while using all features or the six most significant features are not causing a huge difference in week 4, the gap widens from week 5-7, showing that the use of all features could prove better than the top six features. On the other hand, having picked the two top features (which are inherently social in nature and involve interactions with other students), the algorithm outperforms the other configurations and provides the prediction accuracy as high as 83% in week 4, and up to nearly 90% in week 7. To address the third research question, the role of social features reflecting the social learning activities are investigated. These features are shown to have played a significant role and as highlighted in the fourth research question, identifying the link between the *social* learning activities and the *student success* in this dataset could give insight to course developers and educators with regards to designing and delivering

Table 5: Combining the most influential six features.

	Mean	Median	StDev	Min	Max
<b>Week 1</b>	45.2	45.5	4.41	32	52
<b>Week 2</b>	52.5	52	2.21	48	57
<b>Week 3</b>	59.57	60	2.75	46	63
<b>Week 4</b>	72.67	74	6.22	62	82
<b>Week 5</b>	72.67	75	7.84	57	83
<b>Week 6</b>	78.43	82	8.03	55	86
<b>Week 7</b>	79.77	80.5	4.85	68	87

Table 6: Combining two of the most influential features.

	Mean	Median	StDev	Min	Max
Week 1	32	32	0	32	32
Week 2	39	39	0	39	39
Week 3	54.37	54	1.03	52	57
Week 4	81.4	83	4.00	66	84
Week 5	81.77	82.5	2.42	75	85
Week 6	87.4	88	1.00	85	89
Week 7	87.8	88	0.76	86	89

course activities. Having established a link between social learning and student success, the results highlight the possibility of providing a more surgical feedback (based on the important features verses all features) to the students who are picked as likely to fail by the system. This study has also shown the importance of the social features used which could be of help when providing feedback to students.

## 7. CONCLUSIONS

The paper demonstrates the ability of the proposed swarm intelligence classifier in dealing with the existing educational data. The simplicity of this algorithm with one tunable parameter (i.e. agent size) makes it an attractive technique to use. One of the key contribution of the paper is to provide evidence that the data collected on our social learning platform (delivered to several thousand students on Coursera), which records the way in which students share, view and comment on each other's work, is related to performance. Specifically, whilst predicting the final fail/pass of students might be difficult on the first few weeks, the prediction accuracy rises to 83% in week 4 and as high as 89% on week 7. Given two of the social features are demonstrated to have played an important role in the prediction accuracy of the algorithm, as the work progresses, the authors will start to look at questions such as what social behaviours are the best predictors of performance? When can such predictions be made? What kinds of social behaviour impact upon the predicted grades of students? Is it possible to help design interventions for students and tutors to help each other? Finally, after several years of building a system through participatory design and concentrating on the user experience, we are now in a position to use a data driven approach to build systems to support communities of learners.

## 8. REFERENCES

- [1] E. Aguiar, H. Lakkaraju, N. Bhanpuri, D. Miller, B. Yuhas, and K. L. Addison. Who, when, and why: a machine learning approach to prioritizing students at risk of not graduating high school on time. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 93–102, 2015.
- [2] M. M. al-Rifaie and M. Bishop. Stochastic diffusion search review. In *Paladyn, Journal of Behavioral Robotics*, volume 4(3), pages 155–173. Paladyn, Journal of Behavioral Robotics, 2013.

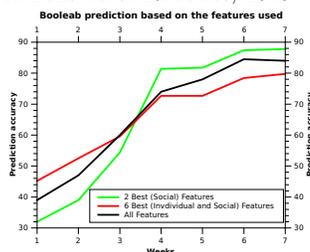


Figure 4: Impact of using various features in the accuracy of the prediction

- [3] C. Brooks, C. Thompson, and S. Teasley. Who you are or what you do: Comparing the predictive power of demographics vs. activity patterns in massive open online courses (moocs). In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 245–248. ACM, 2015.
- [4] C. a. Coleman, D. T. Seaton, and I. Chuang. Probabilistic Use Cases: Discovering Behavioral Patterns for Predicting Certification. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale - L@S '15*, pages 141–148, 2015.
- [5] N. M. Dowell, O. Skrypnyk, S. Joksimović, A. Graesser, S. Dawson, D. Gašević, P. de Vries, T. Hennis, and V. Kovanović. Modeling learners's social centrality and performance through language and discourse. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.
- [6] A. Elbadrawy, R. S. Studham, and G. Karypis. Collaborative multi-regression models for predicting students' performance in course activities. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge - LAK '15*, pages 103–107, 2015.
- [7] Y. Gong and J. E. Beck. Towards detecting wheel-spinning: Future failure in mastery learning. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 67–74. ACM, 2015.
- [8] L. P. Macfadyen and S. Dawson. Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers and Education*, 54(2):588–599, 2010.
- [9] J. K. Olsen, V. Alevan, and N. Rummel. Predicting student performance in a collaborative learning environment. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.
- [10] C. Reas and B. Fry. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, aug 2007.
- [11] T. Rogers, C. Colvin, and B. Chiera. Modest analytics: using the index method to identify students at risk of failure. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, pages 118–122. ACM, 2014.
- [12] J. Rollinson and E. Brunskill. From predictive models to instructional policies. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.
- [13] J. Whitehill, J. J. Williams, G. Lopez, C. A. Coleman, and J. Reich. Beyond prediction: First steps toward automatic intervention in mooc student stopout. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.
- [14] D. Yang, T. Sinha, D. Adamson, and C. P. Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-Driven Education Workshop*, 2013.
- [15] C. Ye, J. S. Kinnebrew, G. Biswas, B. J. Evans, D. H. Fisher, G. Narasimham, and K. A. Brady. Behavior prediction in moocs using higher granularity temporal information. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 335–338. ACM, 2015.

# Predicting Student Progress from Peer-Assessment Data

Michael Mogessie Ashenafi  
University of Trento  
via Sommarive 9, 38123  
Trento, Italy  
+390461285251  
michael.mogessie@unitn.it

Marco Ronchetti  
University of Trento  
via Sommarive 9, 38123  
Trento, Italy  
+390461282033  
marco.ronchetti@unitn.it

Giuseppe Riccardi  
University of Trento  
via Sommarive 9, 38123  
Trento, Italy  
+390461282087  
giuseppe.riccardi@unitn.it

## ABSTRACT

Predicting overall student performance and monitoring progress have attracted more attention in the past five years than before. Demographic data, high school grades and test result constitute much of the data used for building prediction models. This study demonstrates how data from a peer-assessment environment can be used to build student progress prediction models. The possibility for automating tasks, coupled with minimal teacher intervention, make peer-assessment an efficient platform for gathering student activity data in a continuous manner. The performances of the prediction models are comparable with those trained using other educational data. Considering the fact that the student performance data do not include any teacher assessments, the results are more than encouraging and shall convince the reader that peer-assessment has yet another advantage to offer in the realm of automated student progress monitoring and supervision.

## Keywords

Progress prediction; peer-assessment; learning analytics.

## 1. INTRODUCTION

Common examples of traditional student assessment methods are end-of-course examinations that constitute a very high proportion of final scores and other standardised and high stakes tests.

There are, however, other student-centric, yet less practiced, forms of assessment. Formative assessment is a fitting example [7]. It is designed with the goal of helping students meet specified learning goals through continuous discussion, gauging and reporting of their performance.

Peer-assessment is another form of assessment, which may be designed with summative or formative goals. It is a form of assessment where students evaluate the academic products of their peers [15].

Automated peer-assessment provides a rich platform for gathering data that can be used to monitor student progress. In such context, another dimension of peer-assessment emerges – its potential to serve as a foundation for building prediction models on top of.

In this study, we demonstrate how this potential can be exploited by building linear regression models for predicting students' weekly progress and overall performance for two undergraduate-level computer science courses that utilised an automated peer-assessment.

The rest of this paper is organised as follows. The next section discusses recent advances in student performance prediction. Section 3 presents a brief overview of the web-based peer-assessment platform using which the data was collected. Section 4 discusses details of the data and the features that were selected to

build the prediction models. Section 5 provides two interpretations of student progress and details how these interpretations determine which data shall be used for building the models. Section 6 introduces the reader to how the prediction models are trained and provides details of the prediction performance evaluation metrics reported. Section 7 discusses the first interpretation of progress prediction and demonstrate the respective prediction models. Section 8 builds upon the second interpretation and follows the same procedure as section 7. Section 9 provides a short discussion and conclusion of the study.

## 2. PREVIOUS WORK IN PREDICTING STUDENT PERFORMANCE

Earlier studies in student performance prediction investigated the correlation between high school grades and student demographic data and success in college education as evidenced by successful completion of studies [1, 6].

Unsurprisingly, many of these studies were conducted by scholars in the social sciences and involved the use of common correlation investigation methods such as linear and logistic regression. The large majority of recent studies have, however, been conducted in the computer science discipline. These studies use data from courses administered as part of either computer science or engineering programmes at the undergraduate level. Of these, many focus on predicting performance of freshman and second year students enrolled in introductory level courses.

A generic approach to student performance prediction is to predict overall outcome such as passing or failing a course or even forecasting successful completion of college as marked by graduation [9, 13, 14]. A further step in such an approach may include predicting the classification of the degree or achievement [8].

More fine-grained and sophisticated approaches involve predicting actual scores for tests and assignments as well as final scores and grades for an entire course.

Due to the varying nature of the courses and classes in which such experiments are conducted and advanced machine learning techniques that are readily available as parts of scientific software packages, the number distinct, yet comparable, studies in performance prediction has been growing steadily. Another factor, the proliferation of MOOCs, has fuelled this growth with the immense amount of student activity data generated by these platforms.

Examples of studies that utilise information from students' activities in online learning and assessment platforms in predicting performance include [2, 10, 11].

Apart from predicting end-of-course or end-of-programme performance, prediction models may be used to provide continuous predictions that help monitor student progress. When used in this manner, such prediction models could serve as instruments for early detection of at-risk students. Information provided by these models could then serve the formative needs of both students and teachers. Studies that demonstrate how prediction models can be used to provide continuous predictions and may serve as tools of early intervention include [5, 10].

The most common algorithms in recent literature that are used for making performance predictions are Linear Regression, Neural Networks, Support Vector Machines, Naïve Bayes Classifier, and Decision Trees.

Studies that follow less common approaches include those that use smartphone data to investigate the correlation between students' social and study behaviour and academic performance [16] and those that perform Sentiment Analysis of discussion form posts in MOOCs [4].

Two studies that present algorithms developed for the sole purpose of student performance prediction are [12] and [17].

### 3. THE PEER-ASSESSMENT PLATFORM

In 2012, an experimental web-based peer-assessment system was introduced into a number of undergraduate level courses at an Italian university. Using this peer-assessment system, students completed three sets of tasks during each week of the course. The weekly cycle started with students using the online platform to submit questions about topics that were recently discussed in class. These questions were then reviewed by the teacher, who would select a subset and assign them to students, asking them to provide answers. The assignment of the questions to students was automatically randomised by the system, which guaranteed anonymity of both students who asked the questions and those who answered them. Once this task was completed, the teacher would assign students the last task of the cycle, in which they would rate the answers provided by their peers and evaluate the questions in terms of their perceived difficulty, relevance and interestingness.

Eight cycles of peer-assessment were carried out in two undergraduate-level computer science courses, IG1 and PR2. Participation in peer-assessment activities was not mandatory. However, an effort to engage students in these tasks was made by awarding students with bonus points at the end of the course in accordance with their level of participation and the total number of peer-assigned marks they had earned for their answers. The design and development of the peer-assessment platform and the theoretical motivations for it are discussed in [3].

### 4. THE DATA

Because participation in peer-assessment tasks was not mandatory, there was an apparent decline in the number of participants towards the end of both courses. In order to minimise noise in the resulting prediction models, only peer-assessment activity data of those students who completed at least a third of the total number of tasks and for whom final grades were available were selected for building the models. This led to the inclusion of 115 student records for IG1 and 114 for PR2.

In a previous study [2], a linear regression model for predicting final scores of students using the same data was discussed. Experiments in that study revealed that predicting the range within which a score would fall was more accurate than predicting actual scores. Indeed, this is tantamount to predicting grades. During the

experiments in that study, although attempts were made to build classification models that predicted grades in a multiclass classification manner, the results were found to be much better when actual scores were predicted using linear regression and those scores were mapped to grades according to mappings which were specified beforehand. Hence, the authors decided to apply those techniques in this study as well.

Grades are arguably the ideal approach to judging the performance levels of students because they usually span a wider range of scores, within which a student's scores are likely to fall if the student sits the same exam in relatively quick successions. Consequently, scores predicted by the linear regression models were transformed into grades.

The parameters used to build the linear regression models are:

**Tasks Assigned (TA)** – The number of tasks that were assigned to the student

**Tasks Completed (TC)** – The number of tasks that the student completed

**Questions Asked (QAS)** – The number of 'Ask a Question' tasks the student completed

**Questions Answered (QAN)** – The number of 'Answer a Question' tasks the student completed

**Votes Cast (VC)** – The number of 'Rate Answers' tasks the student completed

**Questions picked for answering (QP)** – The number of the student's questions that were selected by the teacher to be used in 'Answer a Question' tasks

**Votes Earned (VE)** – The number of votes the student earned for their answers

**Votes Earned Total Difficulty (VED)** – The sum of the products of the votes earned for an answer and the difficulty level of the question, as rated by students themselves, for all answers submitted by the student

**Votes Earned Total Relevance (VER)** – The sum of the products of the votes earned for an answer and the relevance level of the question, as rated by students themselves, for all answers submitted by the student

**Votes Earned Total Interestingness (VEI)** – The sum of the products of the votes earned for an answer and the interestingness level of the question, as rated by students themselves, for all answers submitted by the student

**Selected Q total difficulty (SQD)** – The sum of the difficulty levels of the student's questions, as rated by students themselves, which were selected to be used in subsequent tasks

**Selected Q total relevance (SQR)** – The sum of the relevance levels of the student's questions, as rated by students themselves, which were selected to be used in subsequent tasks

**Selected Q total interestingness (SQI)** – The sum of the interestingness levels of the student's questions, as rated by students themselves, which were selected to be used in subsequent tasks

Details of the linear regression model, possible justifications for its prediction errors and experiments comparing its performance to baseline predictors are provided in [2].

## 5. TWO INTERPRETATIONS OF PROGRESS PREDICTION

Monitoring student progress using prediction models requires making predictions using evolving student data at several intervals. Continuous peer-assessment data are the ideal candidate for building such prediction models.

Through years of experience, teachers are usually able to make educated guesses about how student are likely to perform at end-of-course exams by studying their activities throughout the course. Prediction models that use data from previous editions of the same course adopt and formalise such experience with greater efficacy.

Indeed, prediction models can be used not only to make one-off predictions of student performance at the end of a course, but also at several intervals throughout the course. While continuous predictions focus on determining student progress by evaluating performance at different stages, one-off predictions put more importance on whether a student would finally pass a course on not.

This study focuses on the former, making continuous predictions to measure student progress and provides two interpretations of student *progress*.

One interpretation compares a student's standing at any point in the course to the standings of students at the same point but from previous editions of the course. For instance, in a previous edition of a course, if student performance data at every week of the course were collected and if these data were complemented with end-of-course grades, in subsequent editions of the course, a student's performance at any week would be compared to the performances of students at that specific week in the previous edition of the course and the respective grade for the student's level of performance could be predicted. In favour of brevity, this interpretation of progress will be referred to as *Progress Type A*.

The other interpretation focuses on evaluating how far a student is from achieving goals that they are expected to achieve at the end of a course. In a fairly simplified manner, this evaluation may be made by comparing the expected final grade of student at any point during the course to what is deemed to be a desirable outcome at the end of the course. For instance, predicting a student's end-of-course grade in the second week of an eight-week course and comparing that predicted grade to what is considered to be a favourable grade at the end of the course, which is usually in the range A+ to B-, can provide information about how far the student is from achieving goals that are set out at the beginning of the course. In favour of brevity, this interpretation of progress will be referred to as *Progress Type B*.

## 6. TRAINING AND MEASURING THE PERFORMANCE OF THE PREDICTION MODELS

Peer-assessment data collected during the course were divided into weekly data according to the three sets of tasks completed every week. The final score of each student for the course was then converted into one of four letter grades.

The data for each week incorporate the data from all previous weeks. In this manner, the prediction model for any one week is built using more performance data than its predecessors. Naturally, the data used to build the model for the first week would be modest and the data for the final week model would be complete. In general, the performances of models from consecutive weeks were expected to be better.

A common metric used in measuring the performance of linear regression prediction models is the Root Mean Squared Error (RMSE). While RMSE provides information about the average error of the model in making predictions, the conversion of numerical scores to letter grades enables using more informative performance evaluation metrics.

The conversion of numerical scores to letter grades transforms this prediction into a classification problem, with grades treated as class labels. Although multiclass classification algorithms were not applied due to their relatively low performance for this specific task, transformation of predicted scores into grades permitted the application of any of the classification performance evaluation metrics. Therefore, performance is reported in terms of precision, recall, F1, False Positive Rates (FPR) and True Negative Rates (TNR).

When evaluating student performance prediction models, the two questions that are more critical than others are:

- How many of the students the model predicted not to be at-risk were actually at-risk and eventually performed poorly (False Positives) and
- How many of the students that the model predicted to be at-risk of failing were indeed at-risk (True Negatives).

A prediction model with a high FPR largely fails to identify students who are at risk of failing. Conversely, a model with a high TNR identifies the majority of at-risk students. The ideal prediction model would have a very low FPR and, consequently, a very high TNR.

The prediction models are evaluated at two levels. The first level is their performance in making exact prediction of grades. The second is their performance in making a prediction that is within a one grade-point range of the actual grade.

For the purpose of this study, the performance metrics are defined as follows.

Grade – Any of the letters A, B, C, D – A and B denote high performance levels and C and D, otherwise. Although C is usually a pass grade, it is generally not favourable and considered to be a low grade.

Positive – A prediction that is either A or B

Negative – A prediction that is either C or D

True – A prediction that is either the exact outcome or falls within a one grade-point range of the actual outcome

False – A prediction that is not True

Any combination of positive or negative predictions with true or false predictions yields one of the following counts – True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Important statistics that use these counts are Precision (P), Recall (R) and, inherently, F1 scores.

It should be noted that FPR and TNR provide two interpretations of the same outcome and that they are inversely proportional. Indeed,  $FPR = 1 - TNR$ .

## 7. MODELLING PROGRESS TYPE A

This type of progress monitoring compares a student's current progress at any week during the course to the progresses of past

students at the same week of the course. The question that such an approach aims to answer is: 'Compared to how other students were doing at this stage in the past, how well is this student doing now?' 'How well' the student is doing is predicted as follows. First, a linear regression model is built using data collected from the first week to the week of interest. This data comes from a previous edition of the course and the predicted variable is the final score or grade, which is already available. Then, the student's performance at the week in question, represented using the parameters in section 4, is fed to the model to make a prediction. Such weekly information shall provide insight into whether the student is likely to fall behind other students or not.

The prediction errors for the course PR2 gradually decreased for successive weeks, as expected. For IG1, however, early decreases were followed by increases and a slight decrease in the final week. The average RMSE for PR2 for the eight models was 3.4 while it was 3.6 for IG1. The scores predicted were in the range 18 to 30. Figure 1 shows the weekly prediction errors for each course.

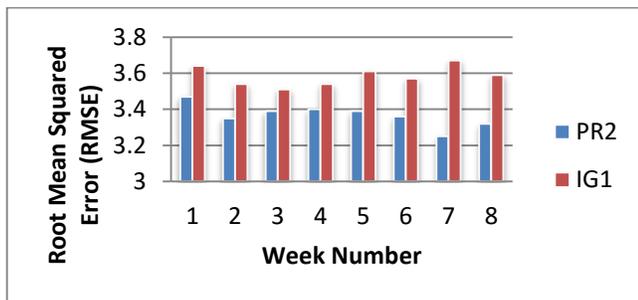


Figure 1. Prediction Errors for the models of each course over eight weeks

Low performance levels were recorded for exact grade prediction of the models for both courses. Specifically, High false positive rates persisted throughout the eight-week period.

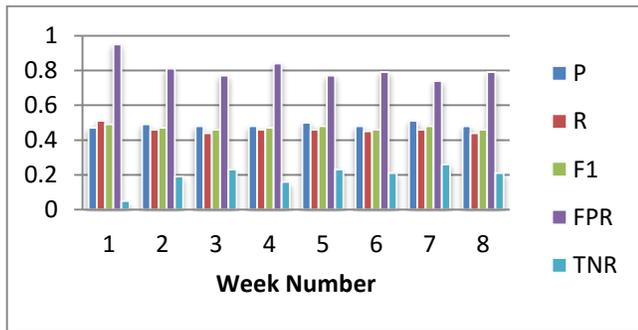


Figure 2. Exact grade prediction performance for PR2

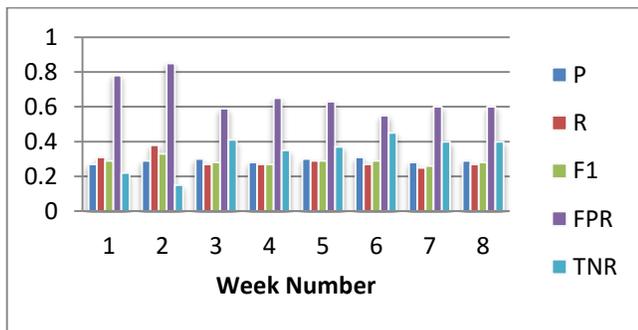


Figure 3. Exact grade prediction performance for IG1

As expected, performance levels of the models for both courses significantly increased for within one grade-point predictions. Low FPR and, consequently, high TNR were recorded even in the first week and performance increased gradually for both courses over the eight-week period.

The models that made within-one-grade-point predictions performed well from the very first week of the course. Although predictions are not made on exact grades, the wider range helps lower the rate of false positives and increase true positives. The same consideration may lead to an increase in false negatives, and hence, a decrease in true positives. However, the high precision and recall values for these models attest that this is not so in this case.

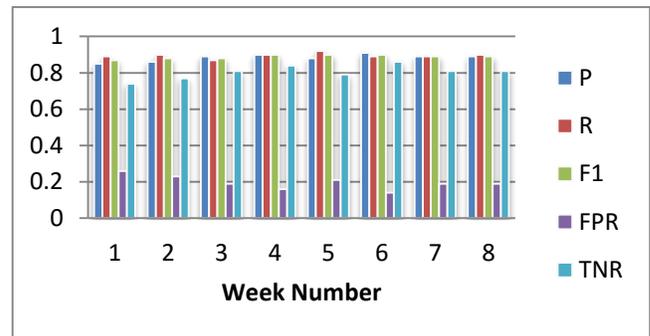


Figure 4. Within-one-grade-point prediction performance for PR2

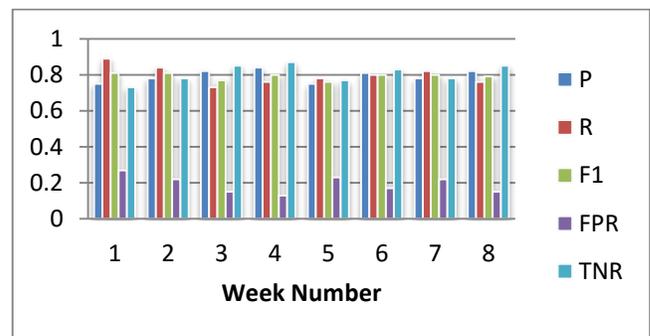
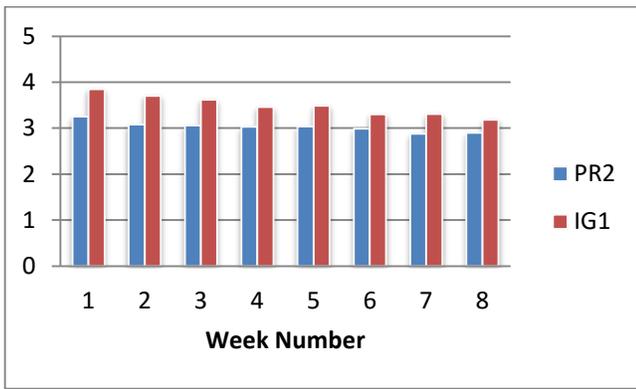


Figure 5. Within-one-grade-point prediction performance for IG1

## 8. MODELLING PROGRESS TYPE B

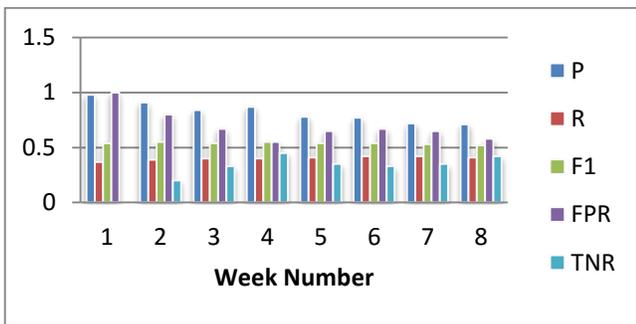
The focus of this type of measuring progress can be informally described as *measuring the gap* between a student's performance *now* and what it is expected to be *at the end* of the course. Modelling this type of progress only requires building a single linear regression model using the entire data from previous editions of the same course. Then, a student's performance data at any week, which is represented by an instance of the values for the parameters discussed in section 4, is fed to the linear regression equation to compute the expected score of the student. This score is then transformed to a grade. Such weekly information would help keep track of a student's progress towards closing this gap and achieving the desired goals.

The prediction errors of this model for the eight weeks are reported in Figure 6. The prediction errors for both courses were significantly lower than those for Progress Type A, with the model for PR2 having an average RMSE of 3.0 and the model for IG1 scoring a higher average RMSE of 3.5. Moreover, prediction errors for both courses consistently decreased throughout the eight weeks.

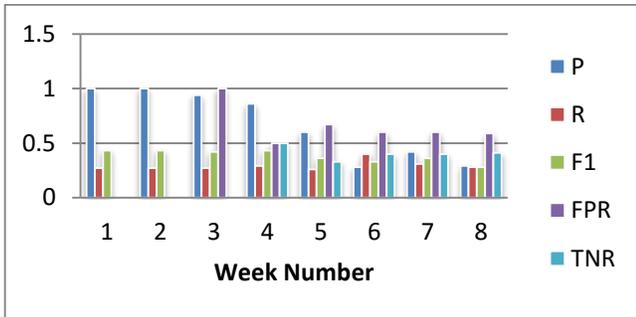


**Figure 6. Prediction errors of the model of the two courses over an eight-week period**

Exact grade prediction performance, although better than that of Progress Type A, was still low for both courses.



**Figure 7. Exact grade prediction performance for PR2**

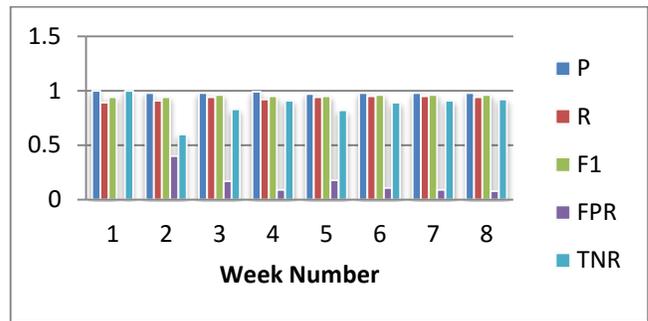


**Figure 8. Exact grade prediction performance for IG1**

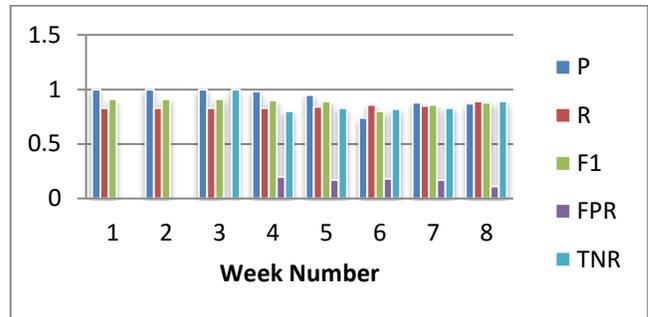
Similar to the models of Progress Type A, this model had very high levels of performance in predicting grades that fell within one grade-point of the actual grades. Prediction performance was very high in the first week and consistently increased, albeit by small amounts, throughout the remaining weeks for both courses.

Missing FPR and TNR values for both courses in the beginning weeks imply that predictions of the model were distributed over TP and FN values. However, high precision values during those weeks indicate that FN values were very low.

Overall, the model for Progress Type B outperformed the models that from Progress Type B, for both courses.



**Figure 9. Within-one-grade-point prediction performance for PR2**



**Figure 10. Within-one-grade-point prediction performance for IG1**

## 9. DISCUSSION AND CONCLUSION

From peer-assessment tasks that were conducted over an eight-week period in two courses, data were used to build several prediction models according to two distinct interpretation of performance prediction. While the first interpretation focused on comparing the performance of a student at any week during the course to those of past students' performance levels obtained in the same week, the second focused on measuring how far a student is from achieving the desired level of performance at the end of a course.

The approach of using data from previous editions of the same course may raise doubts as to whether different editions of the same course are necessarily comparable. However, the extents to which the prediction models discussed here performed should convince the reader that this is indeed possible. Performance of the models is in fact expected to improve with increase in the number of previous editions of the course used as input for making predictions. Indeed, the long-term consistency in the number of below-average, average and above average students over many editions of a course is how many teachers usually measure the overall difficulty level of questions that they include in exams.

Although exact grade predictions did not produce satisfactory levels of performances for either approach, high levels of performance were obtained for both interpretations of student progress when making within-one-grade-point predictions. This signifies the promising potential of carefully designed peer-assessment and the prediction models built using data generated from it as tools of early intervention.

While the statement that a student's performance at the end of a course can be fairly predicted as early as the first weeks of the course from their peer-assessment activity may be construed as simplistic, it is worth noting that the experiments were carried out

in two computer science courses and that the results suggest otherwise.

While a comparison between the performances of the models for the two courses may be made, the reasons behind one model outperforming the other may be latent at this stage and require detailed investigation. Hence, the authors decided to defer making such comparisons until a later stage.

## 10. REFERENCES

- [1] Al-Hammadi, A. S., and Milne, R. H. (2004). A neuro-fuzzy classification approach to the assessment of student performance. In *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on* (Vol. 2, pp. 837–841 vol.2). <http://doi.org/10.1109/FUZZY.2004.1375511>
- [2] Ashenafi, M. M., Riccardi, G., and Ronchetti, M. (2015). Predicting students' final exam scores from their course activities. In *Frontiers in Education Conference (FIE), 2015 IEEE* (pp. 1–9). <http://doi.org/10.1109/FIE.2015.7344081>
- [3] Ashenafi, M.M., Riccardi, G., & Ronchetti, M. (2014, June). A Web-Based Peer Interaction Framework for Improved Assessment and Supervision of Students. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (Vol. 2014, No. 1, pp. 1371-1380).
- [4] Chaplot, D. S., Rhim, E., and Kim, J. (2015). Predicting student attrition in moocs using sentiment analysis and neural networks. In *Proceedings of AIED 2015 Fourth Workshop on Intelligent Support for Learning in Groups*.
- [5] Coleman, C. A., Seaton, D. T., and Chuang, I. (2015). Probabilistic Use Cases: Discovering Behavioral Patterns for Predicting Certification. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale* (pp. 141–148). New York, NY, USA: ACM. <http://doi.org/10.1145/2724660.2724662>
- [6] Evans, G. E., and Simkin, M. G. (1989). What best predicts computer proficiency?. *Communications of the ACM*, 32(11), 1322-1327. <http://doi.org/10.1145/68814.68817>
- [7] Harlen, W., & James, M. (1997). Assessment and Learning: differences and relationships between formative and summative assessment. *Assessment in Education*, 4(3), 365-379. <http://doi.org/10.1080/0969594970040304>
- [8] Jiang, S., Williams, A., Schenke, K., Warschauer, M., and O'dowd, D. (2014). Predicting MOOC performance with week 1 behavior. In *Educational Data Mining 2014*.
- [9] Karamouzis, S. T., and Vrettos, A. (2009). Sensitivity Analysis of Neural Network Parameters for Identifying the Factors for College Student Success. In *Computer Science and Information Engineering, 2009 WRI World Congress on* (Vol. 5, pp. 671–675). <http://doi.org/10.1109/CSIE.2009.592>
- [10] Koprinska, I., Stretton, J., and Yacef, K. (2015). Predicting Student Performance from Multiple Data Sources. In C. Conati, N. Heffernan, A. Mitrovic, and M. F. Verdejo (Eds.), *Artificial Intelligence in Education SE - 90* (Vol. 9112, pp. 678–681). Springer International Publishing. [http://doi.org/10.1007/978-3-319-19773-9\\_90](http://doi.org/10.1007/978-3-319-19773-9_90)
- [11] Manhães, L. M. B., da Cruz, S. M. S., and Zimbrão, G. (2014). WAVE: An Architecture for Predicting Dropout in Undergraduate Courses Using EDM. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (pp. 243–247). New York, NY, USA: ACM. <http://doi.org/10.1145/2554850.2555135>
- [12] Meier, Y., Xu, J., Atan, O., and van der Schaar, M. (2015). Predicting Grades. *Signal Processing, IEEE Transactions on*, PP (99), 1. <http://doi.org/10.1109/TSP.2015.2496278>
- [13] Nghe, N. T., Janecek, P., and Haddawy, P. (2007). A comparative analysis of techniques for predicting academic performance. In *Frontiers in Education Conference - Global Engineering: Knowledge without Borders, Opportunities without Passports, 2007. FIE '07. 37th Annual* (pp. T2G–7–T2G–12). <http://doi.org/10.1109/FIE.2007.4417993>
- [14] Plagge, M. (2013). Using Artificial Neural Networks to Predict First-year Traditional Students Second Year Retention Rates. In *Proceedings of the 51st ACM Southeast Conference* (pp. 17:1–17:5). New York, NY, USA: ACM. <http://doi.org/10.1145/2498328.2500061>
- [15] Topping, K.J. (1998). Peer Assessment Between Students in Colleges and Universities. *Review of educational Research*, 68(3), 249-276. <http://doi.org/10.3102/00346543068003249>
- [16] Wang, R., Harari, G., Hao, P., Zhou, X., and Campbell, A. T. (2015). SmartGPA: How Smartphones Can Assess and Predict Academic Performance of College Students. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 295–306). New York, NY, USA: ACM. <http://doi.org/10.1145/2750858.2804251>
- [17] Watson, C., Li, F. W. B., and Godwin, J. L. (2013). Predicting Performance in an Introductory Programming Course by Logging and Analyzing Student Programming Behavior. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on* (pp. 319–323). <http://doi.org/10.1109/ICALT.2013.99>

# Topic-wise Classification of MOOC Discussions: A Visual Analytics Approach

Thushari Atapattu, Katrina Falkner, Hamid Tarmazdi  
School of Computer Science  
University of Adelaide  
Adelaide, Australia  
{firstname.lastname}@adelaide.edu.au

## ABSTRACT

With a goal of better understanding the online discourse within the Massive Open Online Course (MOOC) context, this paper presents an open source visualisation dashboard developed to identify and classify emergent discussion topics (or themes). As an extension to the authors' previous work in identifying key topics from MOOC discussion contents, this work visualises lecture-related discussions as a graph of relationships between topics and threads. We demonstrate the visualisation using three popular MOOCs offered during 2013. This work facilitates the course staff to locate and navigate the most influential topic clusters as well as the discussions that require intervention by connecting the topics with the corresponding weekly lectures. Further, we demonstrate how our interactive visualisation can be used to explore correlation between discussion topics and other variables such as views, posts, votes, and instructor intervention.

## Keywords

Visualisation, learning analytics, topic model, MOOC, online discourse, discussion forum.

## 1. INTRODUCTION

Within the educational context, visualisation of learning analytics, often known as 'visual analytics', provides insights for many end users including teachers, learners, researchers, educational platform developers, and institutions. According to Thomas and Cook [1], visual analytics focuses on analytical reasoning facilitated by interactive visualisation interfaces. In the educational context, visual analytics support teachers in identifying at-risk students, analysing students' engagement and performance of the course, social collaborations, and developing analytics on the students' online discourse. Visualisation dashboards also support self-evaluation for students in reflecting on their own learning process, setting goals and monitoring progress to achieve these goals.

Visual analytics are often useful in large to massive classrooms such as Massive Open Online Courses (MOOCs), facilitating the understanding of interesting patterns in large volume of students' data, which is challenging to observe using statistical analysis. Visualising the patterns of student engagement (e.g. lecture/forum view), behavior, social interactions and their relationship with final grade/performance has being a focus of many studies [2-4].

Even though the *system-generated* analytics on students' engagement and behavior are important to identify patterns that positively correlate with the successful learning outcomes or attrition, it is likely that these can generate some inconsistencies. For instance, a download of a lecture does not necessarily imply student engagement. Similarly, it is uncertain whether an up-

vote of a forum post means the learner has an interest in the content or, alternatively, that they have problems associated with the topic discussed in the post. Therefore, the analysis of *learner-generated* online discourse (i.e. content) facilitates the interpretation of learners' cognitive processes as well as situating learner behavior in context. According to Mercer [5], the sociocultural perspective highlights "the possibility that educational success and failure may be explained by the quality of educational dialogue, rather than simply in terms of the capability of individual students or the skill of their teachers". This includes identification of individual's understanding of – and interest in – particular course content, and their level of expertise and activity in seeking assistance to rectify conflicts, provide opinions and interact with instructors and peers through dialogs [6, 7]. Existing research focuses on visualising discussion participation and social interactions [8, 9], however, analysis and the visualisation of discussion content (i.e. written discourse) is lacking. Furthermore, there is no support from existing MOOC models to effectively organise and visualise these data. In a preliminary work, Chen [10] and Speck et al. [11] focus on identifying and visualising topic models from online discussion platforms.

Due to the overwhelming abundance of information generated within MOOCs, it is challenging for the learners and the course staff to effectively locate and navigate information. Therefore, topic analysis from MOOC discussions is important in identifying main themes from students' discussions, supporting forum facilitators to become aware of the key themes and the amount of discussions in each theme. We have previously developed a framework for discourse analysis in the MOOC context that identifies latent discussion topics [12]. Our work connects lecture-related discussion topics with the corresponding weekly lectures, allowing course staff to visualise the discussions as clusters of lectures. We have experimented with our framework using three MOOCs and obtained promising results [12].

This paper focuses on developing an open source dashboard to visualise topics extracted from MOOC discussion contents. Our topic visualisation dashboard expects to answer two main questions important to the educators: *What are the emergent topics?*, and *What topics need more attention?*. Further, we also explore the topic distribution using additional variables such as views, votes, replies, and the degree of instructor intervention and answer the questions including '*what is the relationship between topics and views?*', '*what is the relationship between topics and votes?*', and '*what is the relationship between topics and instructor replies?*'. These questions have emerged from the authors' involvement in several MOOC courses and environments to explore key course management issues and pedagogical decisions. To answer these questions, we conducted

a statistical analysis using 3 popular MOOCs – *Machine Learning, Statistics* and *Psychology* and compared the results using the proposed visualisation dashboard.

## 2. BACKGROUND

Visual analytics within the educational context often facilitate educators in understanding large amount of learners’ data to make inferences. Learners’ data can be categorised as *system-generated* and *learner-generated*. System-generated data (also known as clickstream data) are generally analysed and visualised to predict the performance (e.g. CourseSignals [4]). Social Networks Adapting Pedagogical Practice (SNAPP) [8] visualises the evolution of social interactions among participants of online discussion forums.

Within the MOOC context, Coffrin et al. [2] visualises patterns of engagement and performance based on student types (e.g. auditor, active, qualified). Xu et al. [13] utilises visual analytics to explore the correlation between student behavior and student success. In a preliminary work, they analysed five MOOCs using a commercial visualisation software called *Tableau* and reported that there are multiple ways to be successful in a course (e.g. submitting quizzes, lecture views). While there is considerable, as highlighted above, contributing to the development of visual analytics capacity to better understand system-generated educational data, visualisation systems to understand learner-generated data (e.g. online discourse) is lacking.

ForumDash, a preliminary work by Speck et al. [11], focuses on visualising which students are contributing, struggling, or distracted in order to facilitate instructors in targeting their efforts effectively. Using three visualisation tools, ForumDash attempts to provide insights for teachers on which students contribute to most discussions (i.e. Thought-leaders), identify topic clusters to determine the popular topics, and through a ‘contribution score visualisation’, students’ are capable of monitoring how much they are contributing to discussion forums compared to their peers. KISSME (The Knowledge, Interaction and Semantic Student Model Explorer) is a visualisation framework to analyse online discourse with the aim of understanding the nature of interactions among learners including contributions and relationships using LSA and social network analysis [14]. Chen [10] conducts a preliminary study on visualising topic models from online discussion platforms. Another existing tool of interest that takes elements of topic identification and social network analysis is ‘Cohere’ [15]. The authors use argument-mapping techniques to analyse the discussion posts based on some dimensions such as whether the post is an idea, question, or opinion, in measuring the learner’s performance and attention. Topic Facet Model (TFM) incorporates forum posts (mainly questions) about Java from StackOverflow for topic analysis and visualisation [16].

Thus, our motivation for developing this research occurs due to a lack of an established research to produce ‘labeled’ topic models to analyse overwhelming abundance of MOOC discussion contents and visualisations.

## 3. TOPIC VISUALISATION DASHBOARD

The overview of topic analysis and visualisation is shown in the Figure 1. The process of topic analysis is briefly discussed in

Section 3.1 and the full description can be found in the authors’ previous works [12] (full analysis of this work is under review).

### 3.1 Topic Analysis

Our previous work focuses on identifying topic clusters from *lecture-related* MOOC discussion contents. For this, we have used a state of the art topic modeling technique called Latent Dirichlet Allocation (LDA) [17]. LDA is an unsupervised learning approach focusing on discovering hidden thematic structures in large text corpora. One of the issues associated with existing topic models is its inability to label the topics, limiting their usage in end-user applications such as visualisations. It is challenging to label discussion topics due to a lack of a reference source. As a solution, we proposed an automated topic labeling approach by generating candidate topic labels from course lectures. A Naïve Bayes classifier was trained to classify discussion topic into a week or set of weeks, and document summarisation techniques were applied to obtain the most suitable labels for each topic cluster. Our approach facilitates classifying and labeling the discussion threads using course lectures.

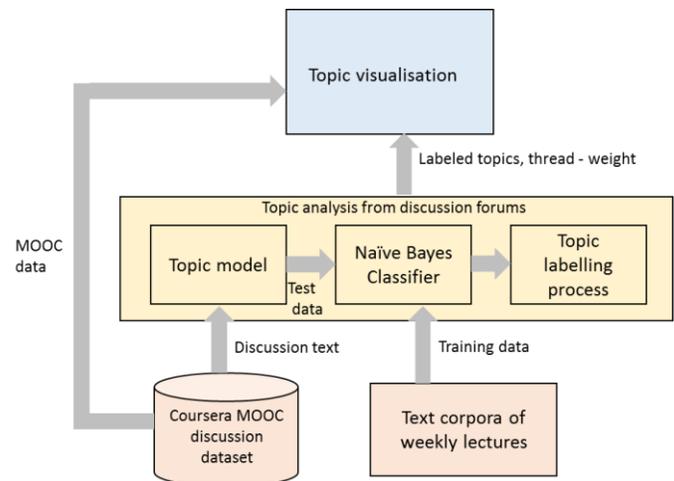


Figure 1: Overview of topic analysis and visualisation

We conducted experiments to evaluate our topic analysis approach using Machine Learning (ML), Statistics (STAT) and Psychology (PSY) MOOCs offered during 2013. In each course, we analysed approximately 5448, 2530 and 9384 number of posts and obtained 40, 25 and 40 strong topics for human annotation, respectively. Three human experts from each MOOC were recruited to label the topics manually and their mean inter-rater agreement (Kappa) was obtained as 0.75 (SD=0.09), 0.77 (SD=0.07) and 0.69 (SD=0.07) for ML, STAT and PSY respectively. We calculated the effectiveness of automated topic labeling process and obtained F-measure of 0.702, 0.75 and 0.69 for ML, STAT and PSY, respectively, demonstrating that the human-machine agreement is similar or slightly lower than inter-rater agreement. Our classifiers also performed well with a macroaveraged F-measure of 0.946, 0.926 and 0.896 for ML, STAT and PSY courses respectively. We also calculated Mean Average Precision (MAP) to evaluate the ranked retrieval results of machine and obtained 0.806 (ML), 0.869 (STAT) and 0.774 (PSY). The promising results obtained from three MOOCs demonstrate that the proposed approach is effective for topic analysis of discussion contents.





interest towards emergent topics by viewing them more often. Similarly, less popular topics are viewed infrequently. Figure 5 depicts the visualisation correspond to this statistical analysis using the Machine Learning course.

According to the Figure 5, most discussed topics are illustrated by the size of the topic node while the most viewed topics are depicted using ‘higher resolution blue’ as shown in the color slider. The thread nodes are labeled using the number of views. Therefore, it is observable that the mostly discussed topics are similar to the mostly viewed topics in the Machine Learning course and vice versa. For instance, ‘gradient descent for linear regression’ and ‘normal equation noninvertibility’ are mostly discussed topics (determined by the size of the topic node) and they are also viewed more than thousand times. This kind of visualisation in classifying discussions according to topics will prioritise which posts to view and interact with based on specific requirements, resulting in a significant saving of time for both learners and teachers, particularly when reviewing massive amounts of data.

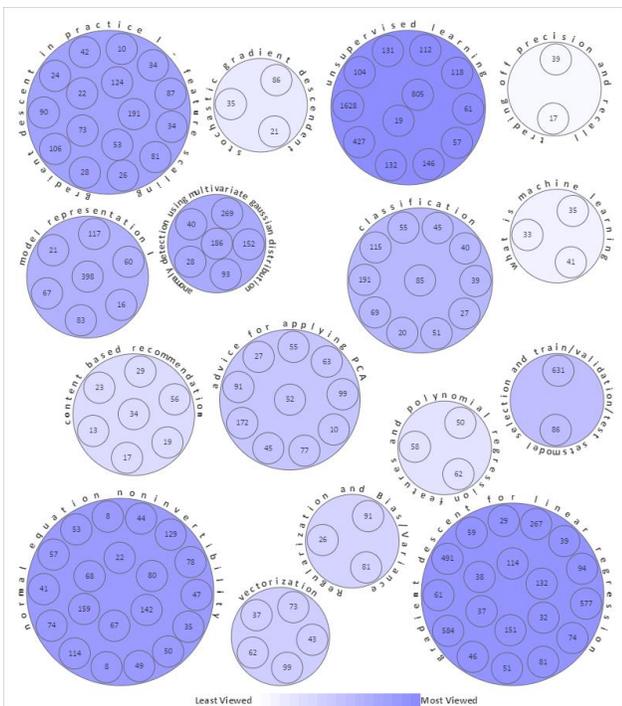


Figure 5: Relationship between topics and views in the Machine Learning course

### 3. What is the relationship between topics and instructor replies?

Instructor replies and discussion topics are moderately positively correlated in ML ( $r = 0.32$ ;  $p > 0.01$ ). However, in STAT and PSY, these two variables demonstrate statistically significant results ( $r = 0.72$ ;  $p < 0.01$  for STAT and  $r = 0.77$ ;  $p < 0.01$  for PSY). This suggests that the instructors’ intervention is more towards emergent topics which may isolate participants who have posted in other topics (i.e. declining topics). A study conducted by Dawson found that instructors primarily interact with high performing students despite isolated and low performing students being neglected irrespective of what they posts [8]. The ML course had relatively low instructor

involvement for any topics while STAT and PSY courses had a good turnaround and strong positive correlation between these two variables. The visualisation in the Figure 6 demonstrates which topics require more inputs from instructors.

This analysis supports the open question of whether the emergent topics or declining topics require more instructor intervention. However, topic-wise classification will provide benefits to the instructors in identifying and prioritise the intervention. Simultaneously, a mechanism to ‘pin’ the emergent discussions will aid to avoid repeated discussions on the same topic.

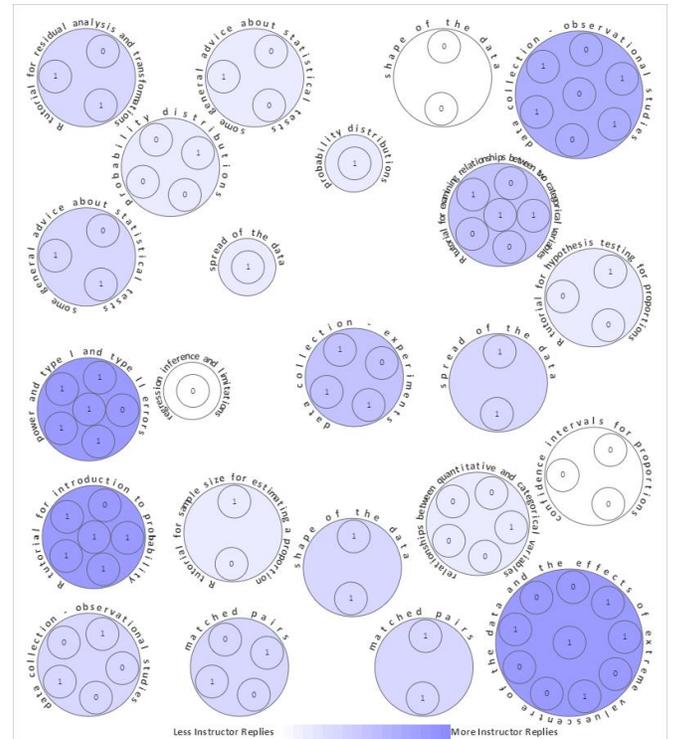


Figure 6: Relationship between topics and instructor replies in the Statistics course.

Our visualisation is currently extending to demonstrate the evolution of topics over time. The time-series analysis focuses on identifying corresponding week or set of weeks a given topic is being discussed. Some topics are discussed outside the course span (e.g. ‘diagnosis’ of Psychology course is discussed in week 9 where the course spans over 8 weeks). Timeline visualisation is helpful in identifying the topics that are being discussed either within or outside of the allocated weeks, enabling the identification of topics that are sustained throughout the course span.

This paper includes only a sample of visualisations and we have shared more visualisations based on the identified dataset here<sup>3</sup>.

In summary, topic-thread visualisation assists in understanding massive volumes of discussion data by identifying emergent discussion themes, allowing the forum facilitators to make interventions more quickly rather than by reading and responding to individual threads. Similarly, topic-wise classification is supportive of comparison across discussions in understanding unexpectedly popular topics even after their expected periods in discussion.

The work presented in this paper is intended for MOOC course staff. We believe it will reduce manual forum moderation time in answering repeated questions, allowing novel discussions to occur contributing to new knowledge construction. Despite providing valuable insights into the analysis of large scale discourse, there is still considerable room for future research. These kinds of visualisation may also provide benefit to students, depending on their experience in interpreting visual information. Therefore, we consider that a topic-wise classification of discussion posts is useful as a navigational support for students, and intend to extend this work in future to support personalised navigation and recommendation of relevant posts.

This work does not yet include an in-depth analysis of individual topics or relationship between topics. It is yet to be analysed for relationship between topics and users. Our future work will include social network analysis to identify topic-inspired interactions between learner-teacher and learner-learner (i.e. peers).

## 5. CONCLUSION

One of the primary challenges of MOOCs is to understand the massive volume of data to make inferences regarding student engagement or learning. To support this, our work analyses learner-generated discussion contents to identify emergent topics of discussions and labels them corresponding to the course lectures. This paper presents the visualisation of our topic-wise classification of discussion data, allowing the user to explore the analysis by manipulating different variables such as votes, views, instructor replies, and time-series analysis. A series of statistical analysis were performed to measure the correlation between discussion topics and other variables, and the findings were compared using the visualisation dashboard. This work provides benefit to the educational data mining and learning analytics research community through an open framework for topic analysis and visualisation of massive volume of discussion data generated regularly through MOOCs and other online learning platforms.

## 6. ACKNOWLEDGMENTS

The authors would like to acknowledge Google Inc. for supporting this project through the ‘Google MOOC Focused Research Award Scheme’.

## 7. REFERENCES

- [1] Thomas, J.J. and K.A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. 2005: IEEE Computer Society Press.
- [2] Coffrin, C., et al., *Visualizing patterns of student engagement and performance in MOOCs*, in *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*. 2014.
- [3] Keim, D., et al., *Visual Analytics: Definition, Process, and Challenges*, in *Information Visualization*. 2008, Springer Berlin Heidelberg. p. 154-175.
- [4] Arnold, K.E. and M.D. Pistilli, *Course signals at Purdue: using learning analytics to increase student success*, in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. 2012.
- [5] Mercer, N., *Sociocultural discourse analysis: analysing classroom talk as a social mode of thinking*. *Journal of Applied Linguistic*, 2004. **1**(2): p. 137-168.
- [6] Ezen-Can, A., et al., *Unsupervised modeling for understanding MOOC discussion forums: a learning analytics approach*, in *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*. 2015.
- [7] Reich, J., et al., *Computer Assisted Reading and Discovery for Student Generated Text in Massive Open Online Courses*. *Journal of Learning Analytics*, 2015. **2**(1).
- [8] Bakharia, A. and S. Dawson. *SNAPP: A Bird's-Eye View of Temporal Participant Interaction*. in *Proceeding of the 1st International Conference on Learning Analytics and Knowledge*. 2011.
- [9] Oshima, J., R. Oshima, and Y. Matsuzawa, *Knowledge building discourse explorer: a social network analysis application for knowledge building discourse*. *Educational technology research and development*, 2012. **60**(5): p. 903-921.
- [10] Chen, B., *Visualizing semantic space of online discourse: the knowledge forum case*, in *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*. 2014.
- [11] Speck, J., et al., *ForumDash: analyzing online discussion forums*, in *Proceedings of the first ACM conference on Learning @ scale conference*. 2014.
- [12] Atapattu, T. and K. Falkner. *A Framework for Topic Generation and Labeling from MOOC Discussions*. in *Third Annual ACM conference on Learning at Scale*. 2016.
- [13] Xu, Z., et al., *Visual analytics of MOOCs at maryland*, in *Proceedings of the first ACM conference on Learning @ scale conference*. 2014.
- [14] Teplovs, C., N. Fujita, and R. Vatrappu, *Generating predictive models of learner community dynamics*, in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. 2011.
- [15] Liddo, A., et al., *Discourse-centric learning analytics*, in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. 2011.
- [16] Hsiao, I. and P. Awasthi, *Topic facet modeling: semantic visual analytics for online discussion forums*, in *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*. 2015.
- [17] Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent Dirichlet Allocation*. *Journal of Machine Learning Research*, 2003. **3**: p. 993-1022.
- [18] Rossi, L.A. and O. Gnawali. *Language Independent Analysis and Classification of Discussion Threads in Coursera MOOC Forums*. in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2014)*. 2014.

# Document Segmentation for Labeling with Academic Learning Objectives

Divyanshu Bhartiya  
IBM Research  
Bangalore, India  
dibharti@in.ibm.com

Danish Contractor  
IBM Research  
New Delhi, India  
dcontrac@in.ibm.com

Sovan Biswas  
IBM Research  
Bangalore, India  
sobiswa3@in.ibm.com

Bikram Sengupta  
IBM Research  
Bangalore, India  
bsengupt@in.ibm.com

Mukesh Mohania  
IBM Research  
Melbourne, Australia  
mukeshm@au1.ibm.com

## ABSTRACT

Teaching in formal academic environments typically follows a curriculum that specifies learning objectives that need to be met at each phase of a student's academic progression. In this paper, we address the novel task of identifying document *segments* in educational material that are relevant for different learning objectives. Using a dynamic programming algorithm based on a vector space representation of sentences in a document, we automatically segment and then label document segments with learning objectives. We demonstrate the effectiveness of our approach on a real-world education data set. We further demonstrate how our system is useful for related tasks of document passage retrieval and QA using a large publicly available dataset. To the best of our knowledge we are the first to attempt the task of segmenting and labeling education materials with academic learning objectives.

## Keywords

text segmentation, document labeling, academic learning objectives, unsupervised

## 1. INTRODUCTION

The rapid growth of cost-effective smart-phones and media devices, coupled with technologies like Learning Content Management Systems, tutoring systems, digital classrooms, MOOC based eLearning systems etc. are changing the way today's students are educated. A recent survey<sup>1</sup> found that there was a 45% year-on-year uptake between 2013 and 2014 of digital content in the classroom and a nearly 82% uptake in the use of digital textbooks. Of the 400,000 K-12 students surveyed, 37% of them reported using online textbooks for their learning needs. Students and teachers frequently

<sup>1</sup>Project Tomorrow, Trends in Digital Learning 2015

search for free and open education resources available online to augment or replace existing learning material. Organizations like MERLOT<sup>2</sup> and the Open Education Consortium<sup>3</sup> offer and promote the use of free learning resources by indexing material available on the web, based only on keywords or user specified meta-data. This makes the identification of the most relevant resources difficult and time consuming. In addition, the use of manually specified meta-data can also result in poor results due to inconsistent meta-data quality, consistency and coverage. Identifying materials most suitable for a learner can be aided by tagging them with learning objectives from different curricula. However, manually labeling material with learning objectives is not scalable since learning standards can contain tens of hundreds of objectives and are prone to frequent revision. Recent work by [3] attempted to address this problem by using external resources such as Wikipedia to expand the context of learning objectives and a *tf-idf* based vector representation of documents and learning objectives. One of the limitations of the system is that it works well only when documents are relatively short in length and relate to a few learning standard objectives. The accuracy of the algorithm reduces when the documents considered are resources such as textbooks due to the dilution of the weights in the *tf-idf* based vector space model. Further, from the perspective of information access, returning a large reference book for a learning objective still burdens the user with the task of identifying the relevant portions of the book. This, therefore, does not adequately address the problem.

In this paper, we address the problem of finding document segments most relevant to learning objectives, using document segmentation [1] and segment ranking. To the best of our knowledge, we are the first to attempt the problem of segmenting and labeling education materials with academic learning objectives.

In summary, our paper makes the following contributions:

- We define the novel task of identifying and labeling document segments with academic learning objectives.

<sup>2</sup><http://www.merlot.org>

<sup>3</sup><http://www.oecconsortium.org/>

- We present the first system that identifies portions of text most relevant for a learning objective in large educational materials. We demonstrate the effectiveness of our approach on a real world education data set. We report a sentence level  $F1$  score of 0.6 and a segment level minimal match accuracy@3 of 0.9
- We demonstrate, using a large publicly available dataset, how our methods can also be used for other NLP tasks such as document passage retrieval and QA.

The rest of the paper is organized as follows: In the next section we describe related work, in section 3 we formally describe our problem statement, section 4 describes our algorithm and implementation details and section 5 presents our detailed experiments. Finally, in section 6 we conclude this paper and discuss possible directions of future work.

## 2. RELATED WORK

Broadly, our work is related to three major areas of natural language research: Text Segmentation, Query Focused Summarization and Document Passage Retrieval. We present a comparison and discussion for each of these areas below:

**Text Segmentation:** Typically, the problem of automatically chunking text into smaller meaningful units has been addressed by studying changes in vocabulary patterns [6] and building topic models[5]. In [12], the authors adapt the TextTiling algorithm from [6] to use topics instead of words. Most recently, [1] uses semantic word embeddings for the text segmentation task. While supervised approaches tend to perform better, we decided to adapt the state of the art unsupervised text segmentation method proposed in [1], due to the challenges associated with sourcing training data for supervised learning.

**Query Focused Summarization:** Focused summarization in our context [8], [10] [4] is the task of building summaries of learning materials based on learning objectives. Here, each learning objective can be treated as a *query*, and the learning materials as documents that need to be summarized. However, it is important to note that in the education domain, any such summarization needs to ensure that summarized material is presented in a way that facilitates learning. This poses additional research challenges such as automatically identifying relationships between concepts presented in the material and therefore, in this paper, we do not model our problem as a summarization task. We encourage the reader to consider it as a possible direction for future research.

**Document Passage Retrieval:** Lastly, document passage retrieval [2] is the task of fetching relevant document passages from a collection of documents based on a user query. However, such tasks typically require the passage boundaries to be well known and therefore, cannot return sub-portions that may be present within a passage or return results that span sub-parts of multiple passages.

## 3. PROBLEM STATEMENT

Typically, a learning standard consists of a hierarchical organization of learning objectives where learning objectives

are grouped by Topic, Course, Subject and Grade. For the purpose of this paper we refer to a “label” as the complete Grade (g) -> Subject (s) -> Course (c) -> Topic (t) -> Learning Objective (l) path in the learning standard.

Given a document  $\mathcal{D}$  of length  $N$  we would like to identify the most relevant segments  $\phi_{i,j}^{\{g,s,c,t,l\}}$  for a given label  $\{g, s, c, t, l\}$  where  $i, j$  denote positions in a document i.e  $i, j \in [0, N]$  and  $i < j$ . In the rest of the paper, we denote the learning objective  $\{g, s, c, t, l\}$  as  $e$  to ease notation.

Figure 1 shows chapter 2 from the the “College Physics” OpenStax textbook<sup>4</sup>. The segments (demarcated using rectangles) have been identified for two learning objectives INST1 and INST2 and occur in different portions of the book. They can even be a sub-part of an existing section in a chapter as shown for INST1.

The next section describes our algorithm for the problem of segmentation and labeling based on learning objectives.

## 4. OUR METHOD

We represent each sentence as a unit vector  $s_i$ , ( $0 \leq i \leq N - 1$ ) in a  $Dim$  dimensional space. The goal of segmentation is to find  $K$  splits in a document, denoted by  $(x_0, x_1, \dots, x_K)$ , where  $x_0 = 0$  and  $x_K = N$  and  $x_i$  denotes the line number specifying the segment boundary such that if the  $k$ th segment contains the sentence  $s_i$ , then  $x_{k-1} \leq i < x_k$ . The discovered segment  $\phi_{i,j}$  is the segment between the splits  $x_i$  and  $x_j$ . Depending on the granularity of the learning objectives and the document collection, the optimal number of splits can be set (See section 5). Let the cost function  $\psi$  for a segment  $\psi(i, j)$  measure the *internal cohesion* of the segment, ( $0 \leq i < j \leq N$ ). The segmentation score for  $K$  splits  $s = (x_0, x_1, \dots, x_K)$  can then be defined as  $\Psi$  :

$$\Psi(s) = \psi(x_0, x_1) + \psi(x_1, x_2) + \dots + \psi(x_{K-1}, x_K)$$

To find the optimal splits in the document based on the cost function  $\Psi$ , we use dynamic programming. The cost of splitting  $\Psi(N, K)$  is the cost of splitting 0 to  $N$  sentences using  $K$  splits. So,

$$\Psi(N, 1) = \psi(0, N)$$

$$\Psi(N, K) = \min_{l < N} \Psi(l, K - 1) + \psi(l, N)$$

We define the  $\psi$  function as follows:

$$\psi(i, j) = \sum_{i \leq l < j} \|s_l - \mu(i, j)\|^2$$

where  $\psi(i, j)$  is analogous to the intra-cluster distance in traditional document clustering while  $\mu(i, j)$  is a representative vector of the segment. We discuss possible forms of  $\mu$  later in this section.

**Ranking:** Each segment is represented as a normalized vector  $\mu(i, j)$  and we determine the most relevant segments to a learning objective  $e$  by ranking segments in increasing order of similarity based on cosine similarity.

$$\cos(\mu, e) = \sum_{d=1}^{Dim} \mu_d * e_d$$

<sup>4</sup><https://openstax.org/details/college-physics>

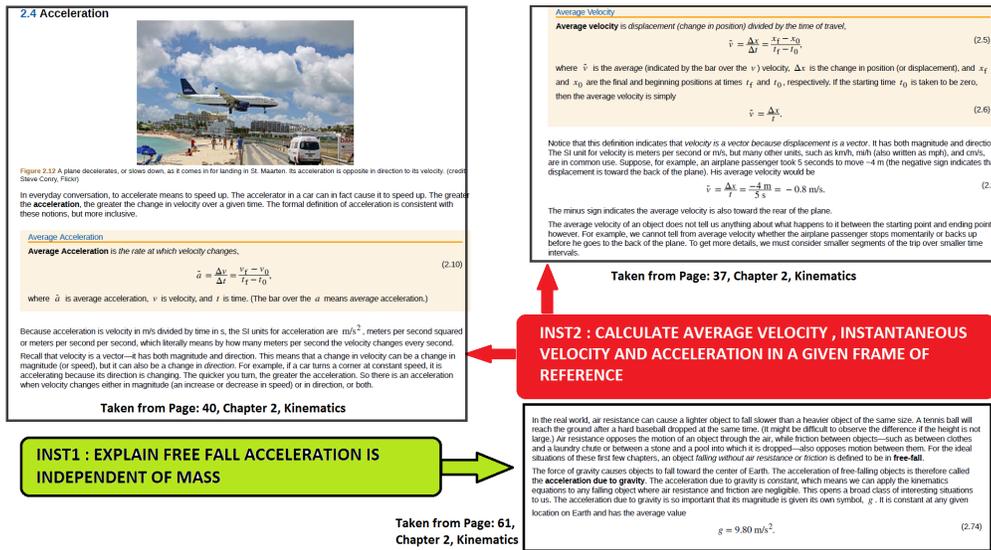


Figure 1: This image shows excerpts from chapter 2 Kinematics from the College Physics text book by OpenStax along with the segment boundaries for two learning objectives INST1 and INST2 shown in colors red and green respectively.

We then select the top  $n$  ranked segments as the segments relevant to the learning objective. In section 5.3 we describe how the number of splits  $K$  as well as the value of  $n$  can be chosen empirically given a validation data set.

We now describe different methods of constructing the document and segment vectors:

**TF-IDF:** Each sentence is represented as a bag of words, the dimensionality being the vocabulary size. Each word in a sentence  $v_i$  is weighted by its *tfidf* measure. For a word  $v_i$  in the sentence  $s_k$  of a document  $\mathcal{D}$ , the *tfidf* measure is given by :

$$tfidf(v_i)_{s_k, \mathcal{D}} = f(v_i, \mathcal{D}) \log \left( \frac{|D|}{df(v_i)} \right)$$

where  $f(v_i, d)$  is the frequency of the word  $v_i$  in the document  $d$ ,  $|D|$  being the total number of documents in our corpus and  $df(v_i)$  is the number of documents with the word  $v_i$  in it. The segment vector  $\mu(i, j)$  in this case is the mean of the sentence vectors in that segment.

**Word Vector:** We represented each sentence as a weighted combination of the word vectors in a sentence. The word-vector  $w_i$  for each word  $v_i$  is specified using Mikolov's Word2Vec[9]. <sub>$s_i$</sub>  Each sentence  $s_i$  is represented as:

$$s_i = \sum_v f(v, d) \log \left( \frac{|D|}{df(v)} \right) w_i$$

The segment vector  $\mu(i, j)$  is also the mean vector in this case.

**Fisher Vector:** Paragraph vectors[7] try to embed the sentences in a fixed dimension, but they require extensive training on the source dataset. Instead we use Fisher Vectors, which have been widely used in the vision community [11] for combining different feature vectors (word vec-

tors in our case), and were recently used for question retrieval by Zhou et.al. [15]. The word vocabulary is modeled as a Gaussian Mixture Model, since a GMM can approximate any continuous arbitrary probability density function. Let  $\lambda = \{\theta_j, \mu_j, \Sigma_j, j = 1 \dots N_G\}$  be the parameters of the GMM with  $N_G$  Gaussians. Let,  $\{w_1, w_2, \dots, w_T\}$  be the vectors for the words  $v_1, v_2, \dots, v_T$  in the sentence  $s_i$  for which we need to construct the fisher vector. We define  $\gamma_j(w_i)$  to be the probability that the word  $w_i$  is assigned the gaussian  $j$ ,

$$\gamma_j(w_i) = \frac{\theta_j \mathcal{N}(w_i | \mu_j, \Sigma_j)}{\sum_{u=1}^{N_G} \theta_u \mathcal{N}(w_i | \mu_u, \Sigma_u)}$$

We define the gradient vector as the score for a sentence,  $G_\lambda(s_i)$  [13]. To compare two sentences, Fisher Kernel is applied on these gradients,

$$\mathcal{K}(s_i, s_j) = G_\lambda(s_i) F_\lambda^{-1} G_\lambda(s_j)$$

where,  $F_\lambda$  is the Fisher Information Matrix,

$$F_\lambda = E_{x \sim p(x|\lambda)} [G_\lambda(s_i) G_\lambda(s_j)^T]$$

$F_\lambda^{-1}$  can be decomposed as  $L_\lambda^T L_\lambda$ , hence the Fisher Kernel can be decomposed to two normalized vectors,  $\Gamma_\lambda(s_i) = L_\lambda G_\lambda(s_i)$ . This  $\Gamma_\lambda(s_i)$  is the fisher vector for the sentence

$$\Gamma_{\mu_j^d}(s_i) = \frac{1}{T \sqrt{\theta_j}} \sum_{t=1}^T \gamma_j(w_t) \left( \frac{w_t^d - \mu_j^d}{\sigma_j^d} \right) \quad (1)$$

$$\Gamma_{\sigma_j^d}(s_i) = \frac{1}{T \sqrt{2\theta_j}} \sum_{t=1}^T \gamma_j(w_t) \left[ \frac{(w_t^d - \mu_j^d)^2}{(\sigma_j^d)^2} - 1 \right] \quad (2)$$

The final fisher vector is the concatenation of all  $\Gamma_{\mu_j^d}(s_i)$  and  $\Gamma_{\sigma_j^d}(s_i)$  for all  $j = 1 \dots N_G$ ,  $d = 1 \dots Dim$ , hence  $2 * N_G * Dim$  dimensional vector. We define the segment vector  $\mu(i, j)$  as the fisher vector formed by using the word vectors

in the segment, hence giving us a unified representation of the segment.

## 5. EXPERIMENTS

In this section we evaluate our method for identifying document segments suited for learning objectives.

### 5.1 Data

We made use of two data sets for our experiments:

**AKS labeled Data Set:** We use the collection of 110 Science documents used by [3] labeled with 68 learning objectives from the Academic Knowledge and Skills (AKS)<sup>5</sup>. We also used term expansions as described in [3] to increase the context of learning objectives. We further identified document segments (at the sentence level) suitable for the learning standard in each of the documents, where applicable.

To build a collection of documents covering multiple learning objectives, we simulated the creation of large academic documents such as text books, by augmenting each lecture note with 9 randomly selected lecture notes. Thus, for each of the 68 instructions that were covered in our data set, we created 5 larger documents each consisting of 10 documents from the original set, giving us a document collection of 340 large documents, with an average length of 300 sentences.

Dataset	#Docs	#Avg. Sentences	#Avg. Splits
AKS Dataset	340	300	10
WikiQA	8100	180	10

**WikiQA Dataset:** To show the general applicability of our approach on tasks such as document passage retrieval and QA, we also use the recently released WikiQA data set [14] which consists of 3047 questions sampled from Bing<sup>6</sup> query logs and associated with answers in a Wikipedia summary paragraph. As outlined in the approach above, for each of the questions, we created a larger document by including 9 other randomly selected answer passages. For each of the 2700 questions from the Train and Test collection we created 3 such documents, thus giving us 8100 documents.

### 5.2 Evaluation Metrics

We define the following metrics for our evaluation:

**MRR (Mean Reciprocal Rank) :** The MRR is defined as the reciprocal rank of the of the first correct result in a ranked list of candidate results.

**P@N (Precision@N):** Let the set of sentences in the top  $N$  segments identified be  $\Gamma^{Sys}$  and further, let the set of sentences in the gold standard be  $\Gamma^{Gold}$ . The precision@N is given by :

$$P@N = \frac{|\Gamma^{Sys} \cap \Gamma^{Gold}|}{|\Gamma^{Sys}|} \quad (3)$$

<sup>5</sup><https://publish.gwinnett.k12.ga.us/gcps/home/public/parents/content/general-info/aks>

<sup>6</sup><http://www.bing.com>

**R@N (Recall@N):** Using the same notation described above, the recall @ N is given by :

$$R@N = \frac{|\Gamma^{Sys} \cap \Gamma^{Gold}|}{|\Gamma^{Gold}|} \quad (4)$$

**F1@N (F1 Score @N):** The F1 Score@N is given by the harmonic mean of the Precision@N and Recall@N described above. **MMA@N (Minimal Match Accuracy@N)** For a collection of  $D$  labeled documents, the minimal match accuracy@N is a relaxed value of precision and is given by:

$$\frac{\sum_i^D \mathbb{1}\{|\Gamma_i^{Sys} \cap \Gamma_i^{Gold}| \geq 1\}}{D} \quad (5)$$

where  $\mathbb{1}\{\}$  is the indicator function.

### 5.3 Experimental Setup

For the AKS dataset, we calculate the *idf* using a collection of 6000 Science documents from Wikibooks<sup>7</sup> and Project Gutenberg<sup>8</sup>. For the WikiQA dataset, *idf* was calculated on the 2700 summaries in the training and test collection. Word vectors and fisher vectors were trained on the full collection of English Wikipedia articles to ensure that the Gaussian Mixture model isn't trained on a skewed dataset and can be used across universally for all kinds of english educational documents. The number of gaussians were selected based on the bayesian information criterion.<sup>9</sup>

**Choosing the number of top segments:** The number of top ranked segments  $n$  and the number of splits  $K$  both affect the accuracy of the system. For instance, if we set  $K$  to be half the total number of sentences, the resulting segments will be very small. Therefore, the value of  $n$  needs to be higher to have adequate coverage (recall). Similarly, choosing very few splits can result in large chunks, which can be problematic if the learning objectives were precise and required finer segments. Thus, the choice of  $n$  and  $K$  depends on the granularity of specification in the learning objectives as well as the nature of content in the document collection.

We use 20% of the dataset (selected at random) as the validation set for tuning the parameters  $n$  and  $k$ . By varying both  $n$  and  $K$  we can determine the value at which the system performance (measured using F1 score) is best. Figure 2 shows the variation in F1 Score for different values of  $K$  and  $n$ . For clarity of presentation, we only show this for the system using TF-IDF vectors. As can be seen, the *F1* score is best for 10 splits and choosing the 3 best segments closest to the learning objective i.e  $K = 10, n = 3$ . Figures 3 and 4 show the individual contributions to the *F1* score.

## 5.4 Results

### 5.4.1 Document Segmentation and Labeling

On performing segmentation on the AKS dataset using all three vector approaches, we observe (table 1) that the tf-idf vector representation works best. We noticed that many

<sup>7</sup><http://www.wikibooks.org>

<sup>8</sup><http://www.gutenberg.org>

<sup>9</sup>An index used for model selection  $-2L_m + m \ln n$ , where  $L_m$  is the maximized likelihood,  $m$  are the number of parameters and  $n$  is the sample size

Query Expansion		@1			@3			@5		
		P	R	F1	P	R	F1	P	R	F1
No Expansion	TFIDF	0.669	0.359	0.468	0.493	0.698	<b>0.578</b>	0.395	0.843	0.538
	WORDVEC	0.462	0.357	0.403	0.331	0.633	0.434	0.284	0.829	0.423
	FISHER	0.476	0.366	0.414	0.342	0.679	0.454	0.284	0.855	0.426
With Expansion	TFIDF	0.686	0.320	0.436	0.545	0.701	<b>0.613</b>	0.435	0.856	0.577
	WORDVEC	0.483	0.323	0.387	0.351	0.586	0.439	0.308	0.797	0.444
	FISHER	0.481	0.322	0.386	0.351	0.619	0.448	0.305	0.827	0.445

Table 1: Results on the AKS Labeled Dataset

	MRR	MMA@1	MMA@3	MMA@5
TFIDF	0.78	0.652	<b>0.905</b>	0.882
WORDVEC	0.56	0.429	0.635	0.782
FISHER	0.55	0.405	0.620	0.715

Table 2: Segment Level Results on AKS Dataset

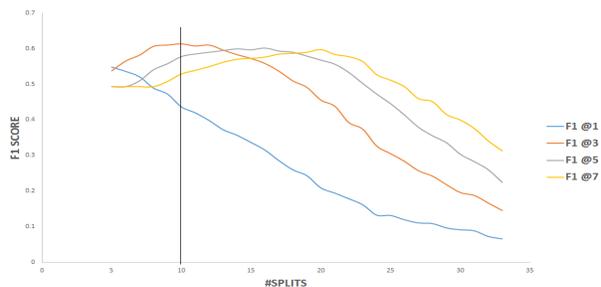


Figure 2: F1 Variation with number of segments at varying depths of retrieval. Best score at 10 segments at depth 3

of the documents in the AKS data set were very well contextualized when changing topics, thus blurring the segment boundaries. For example, in one of the documents which described “Motion in a Straight Line”, the concepts of “velocity”, “acceleration”, “position-time” graphs are intertwined and the topical drift is not easy to observe. As a result, due to the nature of documents in the collection, we hypothesize that the fisher vectors and word vectors which have been trained on large general corpora are unable to adequately distinguish some portions of the text, while the tf-idf vectors which have been tuned on the corpus better reflect the word distributions.

The precision, recall and F1 scores are calculated at the sentence level, thus making it a very strict measure. So we also report segment level accuracy, i.e. how many of the top  $n$  segments identified were relevant. A predicted segment is labeled relevant to the external query if at least 70% of the segment overlaps with the gold labeled segments. We evaluate the performance using MRR and MMA@N. Table 2 shows the segment level evaluation of our system.

#### 5.4.2 Passage Retrieval and QA

We also conducted experiments with a more discriminative dataset where the topical shift is not as hard to observe. We report (table 3) an MRR of 0.895 and P@1 of 89.4% for the passage retrieval task on each of the documents generated,

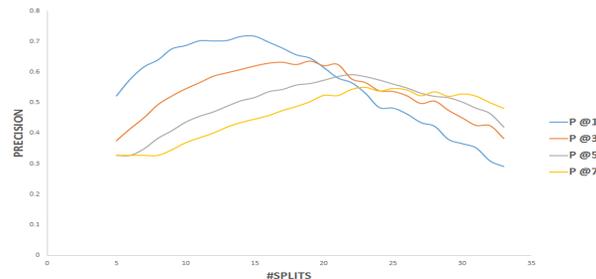


Figure 3: Precision variation with number of segments at varying depths of retrieval. Low values of  $n$  and high values of  $K$  give high precision. Increasing  $K$  while keeping  $n$  constant gives a drop in precision.

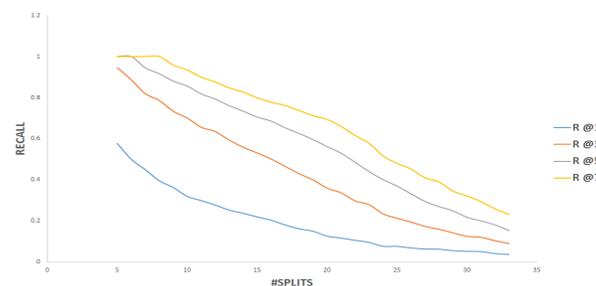


Figure 4: Recall variation with number of segments at varying depths of retrieval. Recall is higher at low values of  $K$  and high values of  $n$ , and the recall drops considerably as the number of segments  $K$  increases.

as described in section 5.1.

Further, we also describe our results on the original task, proposed with the data set, of finding the answer in a passage for a question. In our experiments we report results under two conditions: (a) First identifying the best passage and then choosing the best sentence (b) Assuming the best passage is already known and then choosing the best sentence that answers the query (original WikiQA QA task). Table 4 presents results of experiments under both these conditions. It can be seen that our system gives comparable results under both conditions. The state of the art results under condition (b) as reported in the original paper is an MRR of 0.696. Our system, though not designed for the original task, has an MRR score 10% lower than the best system reported.

	MRR	MMA@1	MMA@3	@1			@3		
				P	R	F1	P	R	F1
TFIDF	0.807	0.797	0.812	0.839	0.893	0.865	0.308	0.958	0.466
WORDVEC	<b>0.895</b>	0.877	0.913	<b>0.894</b>	0.914	0.904	0.315	0.984	0.478
FISHER	0.865	0.842	0.887	0.863	0.885	0.874	0.298	0.975	0.457

Table 3: WikiQA Passage Retrieval Results

	MRR Top Segment	MRR Gold Standard Passage
TFIDF	0.528	0.495
WORDVEC	0.548	0.586
<b>FISHER</b>	<b>0.577</b>	<b>0.597</b>

Table 4: Finding the sentence answering the question: “Top segment” uses our system to select the best passage while “Gold standard passage” uses the actual passage labeled in the data set

## 6. DISCUSSION AND CONCLUSION

In this paper we described the novel task of automatically segmenting and labeling documents with learning standard objectives. Using a state of the art dynamic programming algorithm for text segmentation, we demonstrate its use for this problem and report a sentence level  $F1$  score of 0.613 and segment level  $MMA@3$  of 0.9. We also demonstrated the effectiveness of our approach on document passage retrieval and QA tasks.

Our method is completely unsupervised and only requires a small validation set for parameter tuning. This makes our work general and easily applicable across different geographies and learning standards. Identifying document segments best suited for learning objectives is a challenging problem. For instance, portions of documents that introduce or summarize topics or build a background in an area are very hard to disambiguate for the algorithm due to the lack of observable topic shifts. Developing more sophisticated cohesion and topical diversity measures to address this problem could be an interesting direction of further research.

In future work, we would also like to explore methods that jointly segment and label documents. We also plan to use other methods of vector construction such as paragraph vectors [7] to better represent segments using a training data set as well as semi-supervised text segmentation methods.

## 7. REFERENCES

- [1] A. A. Alemi and P. Ginsparg. Text segmentation based on semantic word embeddings. *arXiv preprint arXiv:1503.05543*, 2015.
- [2] C. L. Clarke and E. L. Terra. Passage retrieval vs. document retrieval for factoid question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 427–428. ACM, 2003.
- [3] D. Contractor, K. Popat, S. Ikbali, S. Negi, B. Sengupta, and M. K. Mohania. Labeling educational content with academic learning standards. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 136–144, 2015.
- [4] H. Daumé III and D. Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics, 2006.
- [5] L. Du, J. K. Pate, and M. Johnson. Topic segmentation in an ordering-based topic model. 2015.
- [6] M. A. Hearst. Texttiling: A quantitative approach to discourse segmentation. Technical report, Citeseer, 1993.
- [7] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.
- [8] J.-P. Mei and L. Chen. Sumcr: a new subtopic-based extractive approach for text summarization. *Knowledge and information systems*, 31(3):527–545, 2012.
- [9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. word2vec, 2014.
- [10] Y. Ouyang, W. Li, S. Li, and Q. Lu. Applying regression models to query-focused multi-document summarization. *Information Processing & Management*, 47(2):227–237, 2011.
- [11] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision—ECCV 2010*, pages 143–156. Springer, 2010.
- [12] M. Riedl and C. Biemann. Text segmentation with topic models. *Journal for Language Technology and Computational Linguistics*, 27(1):47–69, 2012.
- [13] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.
- [14] Y. Yang, W.-t. Yih, and C. Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018. Citeseer, 2015.
- [15] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of ACL*, pages 250–259, 2015.

# Semi-Automatic Detection of Teacher Questions from Human-Transcripts of Audio in Live Classrooms

Nathaniel Blanchard<sup>1</sup>, Patrick J. Donnelly<sup>1</sup>, Andrew M. Olney<sup>2</sup>, Borhan Samei<sup>2</sup>, Brooke Ward<sup>3</sup>, Xiaoyi Sun<sup>3</sup>, Sean Kelly<sup>4</sup>, Martin Nystrand<sup>3</sup>, Sidney K. D'Mello<sup>1</sup>

<sup>1</sup>University of Notre Dame; <sup>2</sup>University of Memphis;

<sup>3</sup>University of Wisconsin-Madison; <sup>4</sup>University of Pittsburgh

384 Fitzpatrick Hall

Notre Dame, IN 46646, USA

[nblancha@nd.edu](mailto:nblancha@nd.edu); [sdmello@nd.edu](mailto:sdmello@nd.edu)

## ABSTRACT

We investigate automatic detection of teacher questions from automatically segmented human-transcripts of teacher audio recordings collected in live classrooms. Using a dataset of audio recordings from 11 teachers across 37 class sessions, we automatically segment teacher speech into individual teacher utterances and code each as containing a teacher question or not. We trained supervised machine learning models to detect questions using high-level natural language features extracted from human transcriptions of a random subset of 1,000 segmented utterances. The models were trained and validated independently of the teacher to ensure generalization to new teachers. We are able to detect questions with a weighted  $F_1$  score of 0.66, suggesting the feasibility of question detection on automatically segmented audio from noisy classrooms. We discuss the possibility of using automatic speech recognition to replace the human transcripts with an eye towards providing automatic feedback to teachers.

## Keywords

Automatic Speech Recognition, Natural Language Processing, Classroom Environments, Question Detection

## 1. INTRODUCTION

Teachers employ a wide array of instructional strategies in their classrooms due to individual teaching styles, requirements of the curricula, and other constraints. These strategies may include lectures, asking questions and evaluating student responses, or assigning small-group work, among many others. However, these approaches are not equally effective at promoting student achievement. Certain techniques, such as asking particular types of questions or facilitating a classroom-wide discussion, have been shown to predict student engagement and achievement growth above others [1], [2].

Research also indicates that providing teachers with feedback on their instructional practices can lead to improved student achievement [3]. But where does the feedback come from? Currently, the onus is on trained human judges who analyze teacher instruction by observing live classrooms. For example, the Nystrand and Gamoran coding scheme [4], [5] provides a general template for observers to document and analyze teacher

instructional practices. This scheme has been empirically validated in numerous studies across hundreds of middle school and high school classrooms [6]–[8]. Unfortunately, this is an expensive and labor intensive process that hinders the ability to analyze classroom instruction at scale. Instead, computational methods that can automatically analyze classroom instruction at scale are needed. We take a step in this direction by considering the possibility of detecting teacher questions in live classrooms. We focus on questions because they are a central component of dialogic instruction, often serving as a catalyst for in-depth classroom discussions and so called ‘dialogic spells’ [9].

The classroom environment provides a unique set of challenges for the automatic analysis of questions. There are also numerous constraints as discussed in detail by D’Mello et al. [10]. Briefly, the analytic approach should not be disruptive to either the teacher or the students. Secondly, it must be affordable to enable widespread adoption across classrooms. Finally, for privacy concerns, video recordings are not possible unless students can be de-identified.

We attempted to overcome these challenges by designing a system that includes a low cost, wireless headset microphone to record teachers as they move about the classroom freely. Our system accommodates various seating arrangements, classroom sizes, and room layouts, but attempts to mitigate complications due to ambient classroom noise, muffled speech, or classroom interruptions, factors that reflect the reality of real-world environments.

There is the open question as to whether the data collected in this fashion can be of sufficient quality for automatic question detection. As an initial step, we consider semi-automated question detection from human-transcripts of automatically-segmented teacher audio. If successful, the next step would be to apply our basic approach by using automatic speech recognition (ASR) in lieu of human transcriptions.

### 1.1 Related Work

Our work is related to previous attempts at automatic detection of questions from transcriptions of audio albeit outside of the noisy classroom interaction context we consider here. We limit our review to experiments that include ASR, as our ultimate goal is in full automation of question detection.

In a study attempting to detect questions in office meetings, Boakye et al. [11] trained models using the ICSI Meeting Recorder Dialog Act (MRDA) corpus, a dataset of 75 hour-long meetings recorded with headset and lapel microphones. Using an AdaBoost classifier to detect questions from human transcriptions, the authors obtained an  $F_1$  score of 67.6 by combining various NLP features.

*Space for Copyright*

Stolcke et al. [12] built a dialogic act tagger on the conversational switchboard database. A Bayesian network modeling word and trigrams discourse grammars, from human transcriptions achieved a recognition rate of 71% to detect a set of dialogic acts, such as statements, questions, apologies, or agreement (chance level 35%; human agreement 84%). The authors further attempted to distinguish questions from statements, two speech acts often confused by their model. They obtained an accuracy of 86% on a subset of their dataset containing equal proportions of questions and statements using only word features (chance accuracy 50%). This result, while promising, is based on an artificially balanced dataset of statements and questions.

Most recently, Orosanu and Jouviet [13] investigated classification of sentences labeled as either statements or questions in three French language corpora, testing on a set of 7,005 statements and 831 questions. The models accurately classified 75.5% of questions and 72.0% of statements using human transcripts. The authors compared the results of using human-annotated sentence boundaries against a semi-automatic method for boundary detection. A subset of sentences, those without prior and proceeding silences of an undefined length, were split once on the longest silence in the sentence; the remainder of the sentences were left unchanged. Semi-automatic splitting led to a 3% increase in classification errors. Although only a subset of sentences were split and there were no cases where sentences were combined, the results suggest that detecting questions from imperfect boundaries may be possible.

## 1.2 Contributions and Novelty

We describe an approach to automatically identify teacher questions from human-transcriptions of teacher audio recorded in live classrooms. We make several contributions while addressing these challenges. First, we examine a dataset of full length recordings of real world class sessions, drawn from multiple teachers and schools. Second, we only use teacher audio because it is the most scalable and practical option. Third, we automatically segment audio recordings into individual teacher utterances in a fully automated fashion and manually transcribe a subset of these utterances for use in our classification models. Fourth, we restrict our feature set to high-level natural language features that are more likely to generalize to classes on different topics rather than low-level domain-specific words. Finally, we design our models to generalize across teachers rather than optimizing to the speech patterns of individual teachers.

## 2. METHOD

### 2.1 Recording Teacher Audio

Data was collected at six rural Wisconsin middle schools during literature, language arts, and civics classes. Class sessions were taught by 11 teachers (three male; eight female) and lasted between 30 and 90 minutes. The teachers carried out their normal lesson plan, allowing the collection of a corpus of real-world samples of classrooms. Based on previous work [10], a Samson 77 Airline wireless microphone was chosen for teachers to wear while teaching. Teacher speech was captured and saved as a 16 kHz, 16-bit single channel audio file. A total of 37 class sessions were recorded on 17 separate days over a period of a year. The recordings contain a total of 32 hours and five minutes of audio.

### 2.2 Teacher Utterance Detection

Teacher speech was segmented into utterances using a voice activity detection (VAD) technique described in [14] and briefly reviewed here. Audio from the teacher's microphone was

automatically split into potential utterances, consisting of either teacher speech or other sounds (e.g., accidental microphone contact, classroom noise), based on pauses (i.e., periods of silence) between speech. The beginning of a potential utterance was automatically identified when the amplitude envelope rose above a preset threshold. The end point of the utterance was automatically identified when the amplitude envelope dropped below this threshold for at least 1000 milliseconds, a pause of one second. The threshold was set to be sufficiently low so as to capture all instances of speech, also causing a high rate of false-alarms. False alarms were eliminated by filtering all potential utterances with Bing ASR [15]. If the ASR rejected a potential utterance, then it was discarded as a non-speech segment.

We validated the effectiveness of our VAD approach in an experiment by hand coding a random subset of 1,000 potential utterances as either containing speech or not containing speech [11]. We achieved an  $F_1$  score of 0.97, which we deemed sufficiently accurate for the purposes of this study. Therefore, we applied our approach for VAD to the full dataset of 37 classroom recordings and extracted 10,080 utterances.

## 2.3 Question Coding and Transcription

We manually coded the complete set of automatically extracted utterances as containing a question or not. It should be noted that a known limitation of annotating automatically segmented speech is that each utterance may contain multiple tags (questions in this case), or conversely, a tag may be spread across over multiple utterances. This occurs because we use both a fixed amplitude envelope threshold and pause length to segment utterances, rather than creating specific thresholds for each teacher or class-session. This fully automates the VAD detection process, and allows us to test generalizability to new teachers. For this work, we allow question tags to span multiple utterances, since the entire content of question is likely to be essential to future work aimed at providing feedback to teachers.

We define a question after the question coding scheme developed by Nystrand and Gameron [4], [5], which is specific to classrooms. For example, calling on students in class (e.g., "What is the capital of Iowa [pause] Michael") is considered a question. Likewise, the teacher calling on a different student to answer the same question after evaluating the previous response (e.g., "Nope [pause] Shelby") is also considered a question. Calling a student name for other reasons, such as to discipline them, is not a question (e.g., "Steven"). Thus, question coding involves ascertaining both the context and intentionality of the utterance.

The coders were seven research assistants and researchers whose native language was English. Coders listened to the utterances in temporal order and assigned a label (question or not) to each based on the words spoken by the teacher, the teachers' tone (e.g., prosody, inflection), and the context of the previous utterance. Coders could also flag an utterance for review by a primary coder, although this occurred rarely.

As training, the coders first engaged in a task of labeling a common evaluation set of 100 utterances. These 100 utterances were selected to exemplify difficult cases. Once coding of the evaluation set was completed, the primary coder, who had considerable expertise with classroom discourse and who initially selected and coded the evaluation set, reviewed the codes. Coders were required to achieve a minimal level of agreement with the primary coder (Cohen's kappa,  $\kappa = 0.80$ ). If the agreement was lower than 0.80, then errors were discussed with the coders.

After this training task was completed, the coders coded a subset of utterances from the complete dataset. In all, 36% of the 10,080 utterances were coded as containing questions. A random subset of 117 utterances from the full dataset were selected and coded by the expert coder. Overall the coders and the primary coder obtained an agreement of  $\kappa = 0.85$  on this evaluation set.

From the full dataset of 10,080 labeled utterances, we selected a random (without replacement) subset of 1,000 utterances for manual transcription by humans. 30% of the utterances in this subset contained a question, which is slightly lower than the 36% question rate on the entire dataset.

## 2.4 Model Building

We trained and tested supervised classification models to predict if utterances contained part (or all) of a question, or did not contain a question. The model building process involved the following steps.

**Features.** Features were generated using the human transcripts for each utterance. We limited our feature set to a set of 37 generalizable NLP features to limit overfitting to teacher dialect or classroom subject/domain. These 34 features were obtained by processing each utterance with the Brill Tagger [16]. Each tagged token was examined for features (see [17] for further details) based on the semantics of various question types (e.g., causal, interpretation, disjunction) or the syntax of questions (e.g., WH-words and modal verbs). These 34 features capture key word (e.g., *why*, *how*), word categories (e.g., procedural), and parts of speech (e.g., noun, verb), and have previously been used to detect domain independent question properties associated with learning from human-transcribed questions [18]. Three additional features include proper nouns (e.g., student names), pronouns associated with teacher questions incorporating student responses (a type of question known as uptake), and pronouns not associated with uptake.

**Minority oversampling.** We supplemented *training* data with additional synthetic instances generated by the Synthetic Minority Over-sampling Technique (SMOTE) algorithm [19] in order to eliminate skew in the training set. Importantly, SMOTE was only applied to the training set and the original distributions in the testing set were not altered.

**Classification and validation.** We explored a number of classifiers: Naïve Bayes, logistic regression, random forest, J48 decision tree, J48 with Bagging, Bayesian network,  $k$ -nearest neighbor ( $k = 7, 9, \text{ and } 11$ ), and J48 decision tree, using implementations from the WEKA toolkit [20]. We also combined the classifiers with MetaCost, which penalized misclassifications of the minority class (weights of 2 and 4). All 37 features were used in the models.

We validated the classification models with leave-one-teacher-out cross-validation, in which models were built on data from 10 teachers (the training set) and validated on the held-out teacher (the testing set). The process was repeated for 11 folds so that each teacher appeared once in the testing set. This cross validation technique tests the potential of our models to generalize to new teachers in terms of variability in question asking and language.

## 3. RESULTS

The best performing model was Naïve Bayes, which achieved the overall highest  $F_1$  score (0.53) for detecting utterances containing questions (the minority class). This model achieved an overall weighted  $F_1$  score of 0.66 (see Table 1 for the confusion matrix).

Additionally, we also compared our results to a chance-model that assigned the question label at the same rate as our model, but did so randomly. We calculated the chance recall and precision for the question label as the average value per teacher over 10,000 iterations. We consider this approach to computing chance to be more informative than a naïve minority baseline model that would yield perfect recall but negligible precision. We observed an encouraging level of recall (0.61) for the question class, which reflects the model’s ability to detect questions from utterances well above both chance precision (0.32) and recall (0.42). However, we note that further refinement is needed to improve the model’s precision (0.47), which is hindered by the frequent misclassification of utterances as questions.

**Table 1. Confusion matrix of 1,000 utterance subset, showing the count and the proportion in parenthesis.**

Instances	Actual	Predicted	
		Question	Utterance
320	Question	195 (0.61)	125 (0.39)
680	Utterance	224 (0.33)	456 (0.67)

## 4. GENERAL DISCUSSION

Questions play a central role in dialogic instruction in classrooms. The importance of dialogue and discussion is widely acknowledged in research [6], [9], [20] and public policy (e.g., Common Core State Standards for Speaking and Listening). The ability to automatically detect questions for both research and teacher professional development might have important consequences in improving student engagement. Towards this goal, our current work focuses on semi-automatic prediction of individual teacher questions teacher audio recorded in live classrooms.

We demonstrated promising results with our approach, consisting of manually transcribed automatically segmented teacher speech, high-level language features, and machine learning. Our best model, validated independently of the teacher, achieved an overall  $F_1$  score of 0.66 and a  $F_1$  score for the question class of 0.53. This reflects a modest improvement in overall classification ( $F_1$  of 0.63) and a significant improvement in question detection accuracy ( $F_1$  of 0.40) over a recent state of the art model [13].

A major contribution of our work is that our models were trained and tested only on automatically, and thus imperfectly, segmented utterances. This confirms that question detection on imperfect sentence boundaries is possible, a result that furthers the work of [13], in which the authors split a subset of manually defined sentences on the longest silence in the sentence (see Section 1.1).

Despite these encouraging results, this study is not without limitations. Most importantly, we only considered manually transcribed speech in order to examine the feasibility of the automatic identification of questions derived from noisy classroom environments. To fully automate our approach we will need to incorporate ASR engines. We expect that the incorporation of noisy ASR will contribute to additional errors in classification, a possibility we are studying in ongoing work that applies automatic speech recognition (ASR) on our full dataset of 10,080 utterances.

Research [11]–[13] indicates that acoustic and contextual features may be important to capture certain difficult types of questions and we will explore the use of these features in future work. Furthermore, additional data collection which includes a second microphone that captures general classroom activity is ongoing. This second channel of audio, when combined with the recording of the teacher, will allow modelling patterns of teacher-student interactions, potentially revealing question-response patterns between teachers and students. Finally, we will extend our approach to classify the question properties defined by Nystrand and Gameron [9]. We have previously explored this task using human transcriptions of manually segmented questions [18], [21], but will extend this work using our approach that employs automatic segmentation and subsequently ASR transcriptions.

In summary, we took steps towards fully automating the detection of teacher questions from audio recordings of live classrooms. We will continue to refine and improve these models as we extend our approach to use ASR transcriptions of the utterances. The present contribution is one component of a broader effort to automate the collection and coding of classroom discourse to improve learning. The automated system is intended to generate personalized formative feedback to teachers, enabling reflection and improvement of their pedagogy, with the ultimate goal of increasing student engagement and achievement.

## 5. ACKNOWLEDGMENTS

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.

## 6. REFERENCES

- [1] S. Kelly, “Classroom discourse and the distribution of student engagement,” *Soc. Psychol. Educ.*, vol. 10, no. 3, pp. 331–352, 2007.
- [2] W. Sweigart, “Classroom talk, knowledge development, and writing,” *Res. Teach. Engl.*, vol. 25, no. 4, pp. 469–496, Dec. 1991.
- [3] M. K. Lai and S. McNaughton, “Analysis and discussion of classroom and achievement data to raise student achievement,” in *Data-based decision making in education*, Springer, 2013, pp. 23–47.
- [4] M. Nystrand, A. Gamoran, R. Kachur, and C. Prendergast, “Opening dialogue,” *Teach. Coll. Columbia Univ. N. Y. Lond.*, 1997.
- [5] A. Gamoran and S. Kelly, “Tracking, instruction, and unequal literacy in secondary school english,” *Stab. Change Am. Educ. Struct. Process Outcomes*, pp. 109–126, 2003.
- [6] A. N. Applebee, J. A. Langer, M. Nystrand, and A. Gamoran, “Discussion-Based Approaches to Developing Understanding: Classroom Instruction and Student Performance in Middle and High School English,” *Am. Educ. Res. J.*, vol. 40, no. 3, pp. 685–730, 2003.
- [7] M. Nystrand, “Research on the role of classroom discourse as it affects reading comprehension,” *Res. Teach. Engl.*, vol. 40, no. 4, pp. 392–412, May 2006.
- [8] M. Nystrand and A. Gamoran, “Instructional discourse, student engagement, and literature achievement,” *Res. Teach. Engl.*, pp. 261–290, 1991.
- [9] M. Nystrand, L. L. Wu, A. Gamoran, S. Zeiser, and D. A. Long, “Questions in time: Investigating the structure and dynamics of unfolding classroom discourse,” *Discourse Process.*, vol. 35, no. 2, pp. 135–198, 2003.
- [10] S. K. D’Mello, A. M. Olney, N. Blanchard, B. Samei, X. Sun, B. Ward, and S. Kelly, “Multimodal capture of teacher-student interactions for automated dialogic analysis in live classrooms,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, New York, NY, USA, 2015, pp. 557–566.
- [11] K. Boakye, B. Favre, and D. Hakkani-Tur, “Any questions? Automatic question detection in meetings,” in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, 2009, pp. 485–489.
- [12] A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. Van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, and M. Meteer, “Dialogue act modeling for automatic tagging and recognition of conversational speech,” *Comput. Linguist.*, vol. 26, no. 3, pp. 339–373, 2000.
- [13] L. Orosanu and D. Jouviet, “Detection of sentence modality on French automatic speech-to-text transcriptions,” in *International Conference on Natural Language and Speech Processing*, Alger, Algeria, 2015.
- [14] N. Blanchard, M. Brady, A. M. Olney, M. Glaus, X. Sun, M. Nystrand, B. Samei, S. Kelly, and S. D’Mello, “A study of automatic speech recognition in noisy classroom environments for automated dialog analysis,” in *Artificial Intelligence in Education*, 2015, pp. 23–33.
- [15] Microsoft, “The Bing Speech Recognition Control,” May 2014.
- [16] E. Brill, “A simple rule-based part of speech tagger,” in *Proceedings of the Workshop on Speech and Natural Language*, 1992, pp. 112–116.
- [17] A. Olney, M. Louwerse, E. Matthews, J. Marineau, H. Hite-Mitchell, and A. Graesser, “Utterance classification in AutoTutor,” in *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing Volume 2*, 2003, pp. 1–8.
- [18] Samei, B., Olney, A. M., Kelly, S., Nystrand, M., D’Mello, S., Blanchard, N., Sun, X., Glaus, M. & Graesser, A., “Domain independent assessment of dialogic properties of classroom discourse,” in *7th International Conference on Educational Data Mining*, 2014, pp. 233–236.
- [19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, pp. 321–357, 2011.
- [20] National Governors Association Center for Best Practices and Council of Chief State School Officers, “Common Core State Standards Speaking & Listening.” National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C., 2010.
- [21] Samei, Borhan, Olney, Andrew M., Kelly, Sean, Nystrand, Martin, D’Mello, Sidney, Blanchard, Nathaniel, and Graesser, Art, “Modeling classroom discourse: Do models of predicting dialogic instruction properties generalize across populations?,” in *Proceedings of the Eighth International Conference on Educational Data Mining*, Madrid, Spain, 2015, pp. 444–447.

# Modeling Interactions Across Skills: A Method to Construct and Compare Models Predicting the Existence of Skill Relationships

Anthony F. Botelho  
Worcester Polytechnic Institute  
100 Institute Rd.  
Worcester, MA 01609-2280  
abotelho@wpi.edu

Seth A. Adjei  
Worcester Polytechnic Institute  
100 Institute Rd.  
Worcester, MA 01609-2280  
saadjei@wpi.edu

Neil T. Heffernan  
Worcester Polytechnic Institute  
100 Institute Rd.  
Worcester, MA 01609-2280  
nth@wpi.edu

## ABSTRACT

The incorporation of prerequisite skill structures into educational systems helps to identify the order in which concepts should be presented to students to optimize student achievement. Many skills have a causal relationship in which one skill must be presented before another, indicating a strong skill relationship. Knowing this relationship can help to predict student performance and identify prerequisite arches. Skill relationships, however, are not directly measurable; instead, the relationship can be estimated by observing differences of student performance across skills. Such methods of estimation, however, seem to lack a baseline model to compare their effectiveness. If two methods of estimating the existence of a relationship yield two different values, which is the more accurate result? In this work, we propose a method of comparing models that attempt to measure the strength of skill relationships. With this method, we begin to identify those student-level covariates that provide the most accurate models predicting the existence of skill relationships. Focusing on interactions of performance across skills, we use our method to construct models to predict the existence of five strongly-related and five simulated poorly-related skill pairs. Our method is able to evaluate several models that distinguish these differences with significant accuracy gains over a null model, and provides the means to identify that interactions of student mastery provide the most significant contributions to these gains in our analysis.

## Keywords

prerequisite structures, skill relationships, feature selection, model comparison

## 1. INTRODUCTION

Many educational systems like ASSISTments and Khan Academy already implement a prerequisite structure as a suggested ordering in which skills should be presented to students. These

structures are often developed by domain experts and teachers in the field of study, and are likely to hold ground-truth. It is clear, for example, that relationships can be identified by observing skills at the problem-level; by viewing the steps required for students to complete each item, it can be known that any skills required to complete such problems can be considered prerequisites. For example, Multiplying Whole Numbers may act as a prerequisite to Greatest Common Factors, as is used in our analysis. While causality suggests a strong relationship, it is possible for two skills to relate to each other in other ways. Such relationships are less intuitive, perhaps requiring a similar thought process or sequence of steps to solve, even if the content of such tasks differ. Many causal skill arches are identifiable by domain experts by observing content, but as described, other such relationships may be missed due to their non-intuitive structures. By observing strong skill relationships identified by domain experts, we construct a method of measuring the factors that are most predictive of their existence.

We also argue that identifying strong relationships is not enough for a method of prediction to be considered adequate. Such a method should also be able to identify weak or non-existent skill relationships. It is likely that while much attention and research is placed on structuring prerequisite links, some of these are false-positives. In other words, a skill may be listed as a prerequisite, but has no true relationship to its supposed post-requisite skill. In such a case there is little or no interactions of performance. Such links must also be identified and removed or reordered in learning platforms to benefit the students.

A significant amount of research has looked at measuring the strength of skill relationships [1],[4], and even the effects such relationships have on measuring student performance [3],[10], but without understood ground truths, it is difficult to compare across these methods. Furthermore, many of these methods represent similar conceptualizations of performance inherently, or through variations of representation such as aggregation or centering. For example, “student achievement” is likely a predictor of skill relationships (achievement on a prerequisite skill will likely influence achievement on a post-requisite skill), but can be represented as the percent of problems answered correctly, mastery speed (the number of items needed to complete an assignment as is commonly used in intelligent tutoring systems), or countless other combinations of features. It will be

important to distinguish between these generalized components to avoid incorporating features that capture the same types of conceptualizations into predictive models.

This work provides a method to evaluate models that measure the strength of skill relationships, and with this model we attempt to identify which features best indicate a strong relationship between two skills. This analysis will incorporate a method of generalizing and distinguishing features that measure different aspects of learning and performance. With this methodology, we seek to answer the following two research questions:

1. What link-level features, expressed in this paper as interactions of performance between skills, are significant in predicting the existence or non-existence of skill relationships?
2. Which features are the strongest predictors of skill relationships, and does combining them make for a more accurate predictive model?

The next section of this paper will discuss some of the previous research performed on skill relationships and prerequisite structures. Then, we will discuss our theory and methodology to provide a baseline model of comparing methods of measuring skill relationships. Using this model, we then compare several commonly-used student-level features, and of the most accurate, compare several different representations of those features. Finally, we will discuss our findings and suggested future works.

## 2. PREVIOUS WORKS

The discovery and refinement of prerequisite skill structures has been an important research question in recent years. The impact of this research on educational systems cannot be overemphasized. Domain experts who design these structures need data centered methods to support the decisions they make; it is vital to have empirical data to support hypothesis regarding the order in which skills are presented as it can have a large impact on student achievement and either aid or impede the learning process. Additionally, identifying the best prerequisite skill structure will enhance student modeling; knowing a student's prior performance on prerequisite skills can help estimate that student's performance on the post-requisites. This can lead to earlier interventions for struggling students, or even help redefine mastery perhaps students who perform very well on a prerequisite requires less practice on a post-requisite, or can be given more advanced examples.

Tatsuoka, defined a data structure called the Q-Matrix, that represents the mapping of problems to skills: the rows of this matrix represent the problems, and the columns represent the skills [9]. Though the goal of the research was to diagnose the misconceptions of students, they set in motion a number of studies that have used this data structure as the first step to find prerequisite structures [2],[5],[8].

Desmarais and his colleagues developed an algorithm that finds the prerequisite relationship between questions, or items, in students' response data [6]. They compare pairs of items in a test and determine any interactions existing between

each pair. Depending on the interactions and a set of interaction-related criteria, they determine whether the two items have a prerequisite relationship between them. This approach was applied by Pavlick, et al. to analyze item-type covariances and to propose a hierarchical agglomerative clustering method to refine the tagging of items to skills [7]. Brunskel conducted a preliminary study in which they use students' noisy data to infer prerequisite structures [4]. Further research by Scheines, et al. extended a causal structure discovery algorithm in which an assumption regarding the purity of items is relaxed to reflect real data and to use that to infer prerequisite skill structure from data [8].

## 3. DATASET

The dataset<sup>1</sup> used for this study consists of real-world student data from the ASSISTments online learning platform. The raw data contains student problem logs pertaining to ten math skills from the 2014-2015 school year. These ten skills represent five skill pairs, listed in Table 1, for which domain experts identified as having a strong prerequisite relationship. While we are not limiting the usage of our proposed baseline model to just prerequisite relationships, these are the most reliable to identify due to the causal effect of content (if problems in skill B require the use of skill A to complete, a strong relationship can be identified).

**Table 1: The strong skill pairs as determined by domain experts**

Prerequisite	Post-requisite
Multiplication of Whole Numbers	Greatest Common Factor
Subtracting Integers	Order of Operations
Division of Whole Numbers	Dividing Multi-Digit Numbers
Volume of Rectangular Prisms Without Formula	Volume of Rectangular Prisms
Nets of 3D Figures	Surface Area of Rectangular Prisms

In order to identify believable ground-truth skill pairs, a survey containing 24 skill pairs for which we had sufficient student data (greater than 50 student rows) was administered to 45 teachers and domain experts who use ASSISTments. Each was asked to rate on a scale of 1 to 7, indicating the perceived qualitative strength of the relationship of each skill pair. From the survey results, five skill pairs were selected to be the strongest related links with the smallest variance in opinion scores. As we are treating these links as truth, we wanted to be highly selective of these pairs.

The resulting dataset consists of 1838 total student rows from 896 unique students. This includes two rows of data per student for each of the five skill pairs included. The first row contains information of that student's performance on the pre- and post-requisite skills, while the second row contains student performance on the prerequisite and a simulated post-requisite described further in the next section.

<sup>1</sup>The full raw and filtered datasets are available at the following link: <http://tiny.cc/veqg5x>

For each student, a feature vector was selected using common performance metrics to compare within our model. This feature vector contained eight link-level features representing the interactions between student-level prerequisite and post-requisite performance metrics. The generated link-level features observed are as described below:

#### **Percent Correct**

The mean-centered<sup>2</sup> percentage of correct responses in the prerequisite skill multiplied by the mean-centered percentage of correct responses in the post-requisite skill.

#### **First Problem Correctness (FPC)**

The binary correctness of the first response in the prerequisite skill multiplied by the binary correctness of the first response on the post-requisite skill.

#### **Mastery Speed**

The mean-centered mastery speed of the prerequisite skill, defined as the number of problems required for each student to achieve three consecutive correct responses, multiplied by the mean-centered mastery speed of the post-requisite skill. In addition to centering, these values were also winsorized to make the largest possible value 10, chosen as this is often the maximum number of daily attempts allowed within ASSISTments. All centering and winsorizing occurred before multiplying the two values.

#### **Z-Scored Percent Correct**

The z-scored<sup>3</sup> value of mean-centered percentage of correct responses in the prerequisite skill multiplied by the z-scored value of mean-centered percentage of correct responses in the post-requisite skill.

#### **Binned Mastery Speed (Bin)**

The numbered bin of mastery speed as described in [3] of the prerequisite skill multiplied by the bin of mastery speed in the second skill. Students were placed into one of five bins based on mastery speed if the assignment was completed and based on percent correct if the assignment was not completed.

#### **Z-Scored Mastery Speed**

The z-scored value of mean-centered, winsorized mastery speed in the prerequisite skill, multiplied by the z-scored value of mean-centered, winsorized mastery speed in the post-requisite skill.

#### **Bin X FPC**

The binned mastery speed value in the prerequisite skill multiplied by the binary correctness of the first response in the post-requisite skill.

#### **Percent Correct X FPC**

The mean-centered percentage of correct responses in the prerequisite skill multiplied by the binary correctness of the first response in the post-requisite skill.

<sup>2</sup>All centering of features was performed at the skill-level.

<sup>3</sup>All z-scoring was performed at the class-level.

## **4. METHODOLOGY**

The ultimate goal of this work is to provide the means of comparing models predicting the existence, or non-existence of skill relationships. Our approach to this is through the comparison and identification of features that most accurately predict these relationships. Using principal component analysis, we group similar features into more generalized conceptualizations to both compare which types of features matter when predicting relationships, but also to avoid problems of multicollinearity that may bias our estimates. Once this baseline model is established, we can construct new predictive models from the significant features and observe their accuracy in predicting the existence of skill relationships when compared to a simple null, or unconditional model.

In order to compare the usage of features against a weak or non-existent relationship, we simulated a new skill using students from the existing prerequisite skill by generating random sequences of responses. For each existing student, we randomly assign him/her a probability between 0.5 and 0.9 in order to create a random sequence of answers. For example, a student given a probability of 0.5 has a 50% chance of answering each given problem correctly. We simulate student answers until either mastery is achieved, defined as three sequentially correct responses, or the student reaches 10 problems without mastering; a value of 10 is chosen here, as many assignments in ASSISTments are given a daily limit of 10 problem attempts before asking the student to seek help or try again on another day. While we acknowledge there are many ways to accomplish this simulation step, we feel this simple method sufficiently creates a skill that has no relationship to the original prerequisite as intended. As our proposed method is intended to be used in the future to help identify undiscovered pre- or post-requisite links, we chose to use a simulated skill rather than a random existing skill to avoid the possibility of randomly selecting an undiscovered related skill. Again, we wanted to be highly selective and consider several such scenarios as we are attempting to create ground-truth values to which we can make our comparisons.

Using these two skill-pairs, one link representing a strong relationship while the other representing a non-existent relationship, we can calculate a feature vector for each student in the prerequisite skill with values from each skill-pair. We use a binary logistic regression with the existence of a relationship as the dependent variable and several link-level covariates to predict whether a skill relationship exists for each student row. The existence of a relationship can be determined then simply by majority ruling, but such calculation is not included in this work and instead observes accuracy at the student-level for a more accurate comparison.

We begin to compare commonly used student-level features in this study through two levels analysis. The first step attempts to compare groups of features, generalizing different representations of similar features into conceptual groupings. As such, we are able to view the predictive power of what we denote as initial performance, mastery, and correctness. The second experiment looks at the individual features as different representations of the overall group to compare

	Component		
	1	2	3
Percent Correct		.821	
First Problem Correctness (FPC)			.839
Mastery Speed	.969		
Z-Scored Percent Correct		.865	
Binned Mastery Speed (Bin)	.972		
Z-Scored Mastery Speed			
Bin X FPC			.873
Percent Correct X FPC		.612	

**Figure 1: The results of the PCA analysis. All features except Z-Scored Mastery Speed mapped to one of three generalized components.**

these predictors at a closer level. We can take each factor of mastery, for example, and compare their usage in several models to determine which is the most accurate predictor of the existence of skill relationships.

#### 4.1 Comparing Link-Level Features

In order to compare representations of student-level features, we must first be able to compare general conceptualizations of features to determine which provide more accurate predictions of the existence of skill relationships. We want to capture the true representations of each metric and attempt to interpret these generalizations as types of features. In order to accomplish this grouping of predictors, we use principal component analysis (PCA) to identify which student-level features correlate to and are representative of more generalized components. PCA is primarily used for dimensionality reduction as we are doing here and gives us the ability to create new variables from the component mappings. The resulting feature alignment can be seen in Figure 1. As is the case in our study, and was mentioned in the previous section, we have multiple metrics of mastery speed as well as several other features. As we can represent “mastery” in several ways, we want to know if the overall concept of mastery, as captured by the metrics used, is reliably predictive of the existence of skill relationships.

Creating a new set of predictors of these groupings, we are able to incorporate these into a binary logistic regression model to view the predictive power of each. While PCA groups similar features together based on their correlations, by viewing which features are grouped we are able to interpret and label each. From this process, we found that most of our features fell into three categories for which we have given the names “mastery,” as this consists of representations of mastery speed, “correctness,” as this consists of representations of the percentage of correct student responses, and “initial performance,” as this consists of representations of

student performance on the initial items of each skill. In addition to these three categories, we are also left with student mastery speed z-scored within student classes as a variable that did not fall under either of the three aforementioned categories; while a derivation of mastery speed, we believe that this did not correlate to the “mastery” category due to the method of standardization as it is capturing this metric in relation to students’ peers. We will readdress this case in our section of discussion.

Once these predictors are identified and created, we construct a binary logistic regression model to predict, for each student row, whether a relationship exists or not. This model will give us a significance value and coefficient for each predictor in the model, as well as an overall predictive accuracy of the model which will be used more for the next analysis.

#### 4.2 Comparing Feature Models

After being able to compare which generalized groups of features are significant predictors of the existence of skill relationships, we are able to compare the individual student-level features that fall into each category by incorporating them into separate models to observe predictive accuracy. The analysis of the first experiment is used to determine which categories are significant in predicting the existence of skill relationships. Using that information, we are able to focus on those groupings with significance to construct models that utilize factors from each grouping. The grouping of “mastery,” for example contains the factors of mastery speed and binned mastery speed, so we can construct models using each to compare differences in predictive power. To avoid problems of collinearity, no single model contains more than one factor from a single grouping. This significantly reduces the number of combinations of features to test compared to running this experiment without first grouping like features and identifying those that are significant as we did in the first experiment.

Using the significant groupings, we are able to create 17 models consisting of single, pairs, and triplets of features. A logistic regression is run on each of these models to predict the existence of a skill relationship. Of the 17 models, 10 of them produce a statistically significant prediction when compared to a null model. Ideally, our null model should produce a 50% accuracy as there is an equal number of good and bad link rows in our dataset. This is not always the case, however, as depending on the feature observed, information may be missing for a particular student; mastery speed, for example, as the number of items attempted by a student before reaching 3 consecutive correct answers, would be missing for any student that did not complete the assignment. For this reason, the predictive power of each model is described as gains in predictive accuracy, or rather, the accuracy of each model minus the accuracy of the corresponding null model.

### 5. RESULTS

The results of the first analysis are expressed in Table 2. Each of the three feature groupings of Mastery, Correctness, and Initial Performance created using PCA in addition to the Z-Scored Mastery are compared within the same model, predicting the existence of a skill relationship. As these

**Table 2: The coefficients and significance values of the generalized components analyzed. From this we can focus on models that exclude features contained in the components with no significance.**

Component	Coefficient Value (log-odds units)	Significance
Mastery	-.251	<.001***
Correctness	.015	.802
Initial Performance	.129	.037*
Z-Scored Mastery Speed	-.129	<.001***

again are link-level features describing interactions between student-level performance on prerequisite and post-requisite skills, it is difficult to draw tangible interpretations from the coefficient value, expressed in log-odds units. This coefficient, used in the logistic regression to make the predictions, describes each component’s effect on the dependent variable. For example, for each unit increase in “Mastery,” the probability that the link exists decreases. Again, as this component is an aggregation of interaction features, it is really describing an aggregation of differences of differences between student-level features making it difficult to make definitive claims regarding these values alone and were included purely to display a general trend of these components on the prediction.

From the table, we are able to determine the significance of each component on the overall prediction by viewing the corresponding p-values in the third column. Looking at these values, we can claim that the overall grouping of “Correctness” seems to have less of an impact on the predictive accuracy of the model. As this term is not significant, we can focus the remainder of our study on the remaining three components.

Table 3 illustrates the results of our second analysis comparing the models that we are able to construct with the remaining features once the “Correctness” grouping has been disregarded. This figure shows the comparative predictive accuracy of the 10 models that give statistically significant predictions as seen in Table 3. Again, these values are expressed as accuracy gains, or rather the percent accuracy increase over the null model run for each predictive model.

## 6. DISCUSSION

This work provides a baseline model of comparing student-level performance across skills to measure the strength of a skill relationship and compare the accuracy of both features and models that estimate this value. Such a model, in our experience, has not existed prior to this study. Our method attempts to identify not only the individual features that contribute to better predictions of these relationships, but also moves to generalize similar features into conceptualizations for comparison in order to minimize multicollinearity.

The principal component analysis step of our model found that all but one feature mapped to one of three components

that we have interpreted as mastery, correctness, and initial performance. It was found the z-scored mastery speed, contrary to our intuition, did not map well to the grouping of mastery. We can speculate the reason for this occurrence by altering our interpretation of the feature. Mastery speed itself is an interesting metric as it attempts to capture two dimensions of performance: a level of understanding and a rate of learning. Also, to reiterate a prior distinction, these metrics are interactions of performance across skills. By z-scoring the metric, it is capturing a contextual effect of each student in comparison with other students in the class, a distinction that appears to have a significant effect.

Observing the resulting model components from the principal component analysis in Table 2, we were able to focus our attention to those components with significant values. Correctness was the only component of that model that was found to have no statistical significance on the dependent variable. This is certainly interesting, as percent correctness and other such measures are among the most common metrics of performance. Perhaps the interaction between pre- and post-requisite percent correct is losing some predictive power from when the metric is used for other predictions of performance.

This aspect illustrates one other important finding that the distinct representations of one metric or another each contribute differently to the predictive accuracy of the models studied. Models incorporating mastery speed, for example, had no significant accuracy gains over a null model, while mastery speed binning showed considerable gains as seen in Table 3. The baseline model of comparison proposed in this study provides the means to make that distinction regarding features contained within the same generalized component grouping. As is seen in that figure, combinations of features outperform any single feature, illustrating a more robust model by capturing multiple representations of performance.

## 7. FUTURE WORK

While we have shown that our model is able to compare and identify features that contribute to higher accuracy in predicting the existence of skill relationships, we also need to stress the importance of the usage of this information. The ability to compare features is only the first step of our model’s goal. By identifying strong predictors of skill relationships that we know exist, we can apply it to other skills within ASSISTments and other systems to identify potentially new prerequisite arches, and also to better measure and predict long-term student performance, learning, and retention. Having an accurate estimate of skill relationships can help restructure prerequisite structures to provide skill sequences in an order that optimizes student learning and achievement.

The work in this paper incorporated several skills into a single dataset to make predictions. In this case, we wanted to create a method that is generalizable to some degree. While our selective skill set allows us to make some claims in terms of the accuracy these models over all skills, it may likely be the case that skill relationships are measurable in different ways for different skills. Further analysis could repeat the steps here on each one of the acquired skills in the dataset.

**Table 3: The models constructed from features in the significant generalized components. No one model contains more than a single feature from each generalized component.**

Model	Null Accuracy	Model Accuracy	Accuracy Gain	Significance
Mastery Speed (MS)	0.63	0.62	0.00	1.000
Z-Scored Mastery Speed	0.63	0.63	0.00	0.888
First Problem Correctness (FPC)	0.50	0.56	0.06	<0.001***
Binned MS	0.50	0.69	0.19	<0.001***
Bin X FPC	0.50	0.56	0.06	<0.001***
Bin, Z-Scored MS	0.50	0.71	0.21	<0.001***
MS, FPC	0.63	0.62	0.00	1.000
MS, Bin X FPC	0.63	0.62	0.00	1.000
Bin, FPC	0.50	0.69	0.19	<0.001***
Bin, Bin X FPC	0.50	0.69	0.19	<0.001***
MS, FPC, Z-Scored MS	0.63	0.63	0.00	0.754
MS, Bin X FPC, Z-Scored MS	0.63	0.63	0.00	0.979
Bin, FPC, Z-Scored MS	0.50	0.71	0.20	<0.001***
Bin, Bin X FPC, Z-Scored MS	0.50	0.71	0.21	<0.001***
MS, Z-Scored MS	0.63	0.63	0.00	0.843
FPC, Z-Scored MS	0.50	0.64	0.14	<0.001***
Bin X FPC, Z-Scored MS	0.50	0.61	0.11	<0.001***

While correctness was not significant in these results, perhaps it is significant when predicting certain types of skills. Perhaps, similar to our features, skills themselves could be generalized into conceptual types for different kinds of analysis pertaining to interactions of performance and their relationships.

The feature vectors generated for each student in our dataset captured many of the most common student-level metrics, but certainly not all of them. There are many other aspects that could be added including completion, measures of learning rate, time spent on the assignments, hint usage, and countless other variables. In addition, this study only observed interactions expressed as multiplications of these terms to describe them as link-level features. There are various other ways to represent interactions or other such transformations including differences of values, division of values, or just simply cross-feature interactions as was partially explored here by looking at Bin X FPC and Percent Correct X FPC. Such interactions model various other aspects of student performance and behavior that can be very useful in this type of relationship prediction.

The methodology presented observes models that predict the existence of skills as a binary outcome, while it can be modified to make comparisons on estimates of relationship strengths as a continuous outcome as well. The method observed model accuracy at the student level for better measurements, but it is a skill-level relationship that is being tested. One simple addition of future work could explore how to best combine the predictions at a student level to make a skill-level prediction. The methodology can then test relationships on the entire system skill structure.

## 8. ACKNOWLEDGMENTS

We acknowledge funding from the NSF (1316736, 1252297, 1109483, 1031398, 0742503, 1440753), the U.S. Dept. of Ed. GAAN (P200A120238), ONR’s “STEM Grand Challenges,” and IES (R305A120125, R305C100024).

## 9. REFERENCES

- [1] S. Adjei, D. Selent, N. Heffernan, Z. Pardos, A. Broaddus, and N. Kingston. Refining learning maps with data fitting techniques: Searching for better fitting learning maps. In *Educational Data Mining*, 2014.
- [2] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, 2005.
- [3] A. Botelho, H. Wan, and N. Heffernan. The prediction of student first response using prerequisite skills. In *Learning at Scale*, 2015.
- [4] E. Brunskill. Estimating prerequisite structure from noisy data. In *Educational Data Mining*, pages 217–222. Citeseer, 2011.
- [5] Y. Chen, P.-H. WUILLEMIN, and J.-M. Labat. Discovering prerequisite structure of skills through probabilistic association rules mining. In *The 8th International Conference on Educational Data Mining*, pages 117–124, 2015.
- [6] M. C. Desmarais, A. Maluf, and J. Liu. User-expertise modeling with empirically derived probabilistic implication networks. *User modeling and user-adapted interaction*, 5(3-4):283–315, 1995.
- [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. *Online Submission*, 2008.
- [8] R. Scheines, E. Silver, and I. Goldin. Discovering prerequisite relationships among knowledge components. In *Educational Data Mining*, 2014.
- [9] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of educational measurement*, 20(4), 1983.
- [10] H. Wan and J. B. Beck. Considering the influence of prerequisite performance on wheel spinning. In *Educational Data Mining*, 2015.

# Robust Predictive Models on MOOCs : Transferring Knowledge across Courses

## ABSTRACT

As MOOCs become a major player in modern education, questions about how to improve their effectiveness and reach are of increasing importance. If machine learning and predictive analytics techniques promise to help teachers and MOOC providers customize the learning experience for students, differences between platforms, courses and iterations pose specific challenges. In this paper, we develop a framework to define classification problems across courses, provide proof that ensembling methods allow for the development of high-performing predictive models, and show that these techniques can be used across platforms, as well as across courses. We thus build a universal framework to deploy predictive models on MOOCs and demonstrate our case on the dropout prediction problem.

## Keywords

Transfer Learning, Ensembling methods, Stacking, MOOCs, Dropout prediction

## 1. INTRODUCTION

As Massive Open Online Courses (MOOCs) continue to become a vital part of modern education, it becomes more and more necessary to increase their effectiveness and reach. Along with learning science and design, data analytics is known to be one of the fields most likely to improve this new education experience ([1]). Predictive analytics are particularly promising, allowing researchers to design real-time interventions and to adapt course content based on student behavior ([7],[9],[6],[3]).

Ideally, these predictive analytics would act in ways similar to an experienced teacher—One who is able to identify different students, and to adapt her actions accordingly. However, because the data available for training models is often significantly different than the data to which those models will be applied, it can be challenging to fully realize this promise ([8]). A predictive analytics system for MOOCs should be able to build on accumulated “past data” to make

accurate predictions about an ongoing class. Thanks to the vast offerings of MOOC databases like edX and Coursera, there is now a plethora of past data available, both across and within a given course.

But this diversity of available courses also means the goal of real-time prediction is easier set than accomplished. Courses may come from different platforms, focus on different topics, or occur at different times. They may have more or less homework, span different lengths of time, or require different levels of involvement. As platforms evolve, courses may also morph to include new information or fulfill shifting demands. Such changes typically affect the behavior of students.

This raises a number of questions and challenges for a data or learning scientist. Given data from a set of repeatedly offered MOOC courses, key questions that shape the design of relevant predictive analytics methods are as follows:

**Purpose** Can I use past courses to predict outcomes within an ongoing course?

**Data** What data should I exploit to build my predictions? Is data from a single course enough, or should I use several courses?

**Method** What method will achieve good efficacy if I use data from a single course? Several courses?

In this paper, we address the challenges inherent in building predictive models that perform well across courses. We answer the questions, mentioned above, that a MOOC analyst would ask about the prediction objectives, the data, and the methods used to build such models. We also address whether such methods are able to perform well across courses on the same platform, and on different platforms.

This paper is divided into five sections. The remainder of this first section explores the available literature regarding MOOC dropout prediction and ensembling methods in machine learning. Section 2 introduces the formal notations, assumptions, and data sets we used to prove our case. Section 3 details different methods that prove useful for building robust models that transfer well to new courses. Section 4 presents the evaluation metrics, and showcases the effectiveness our techniques on the dropout prediction problem. Section 6 evaluates the potential impact of such techniques, and summarizes the key findings and 7 conclusions.

## Literature review

Even before the recent e-learning boom, concerned researchers

have attempted to predict dropout. One major obstacle facing such attempts is the difficulty of building robust predictive algorithms. While working with early e-learning data, the authors of [7] improved the performance of their learning algorithm by merging several predictive algorithms together, namely Support Vector Machines, Neural Networks, and Boosted Trees.

Since then, almost all dropout studies have been conducted on MOOC data. Some researchers (like the authors of [9], who studied the effects of collaboration on the dropout rate of students) focus on understanding drivers of dropout among students. Others develop feature extraction processes and algorithms capable of pinpointing at-risk students before they drop out. If a MOOC is able to identify such students early enough, these researchers reason, it may be possible for educators to intervene. In [6], Halawa et. al. used basic activity features and respective performance comparison to predict dropout one week in advance. The authors of [2] included more features, as well as an integrated framework that allowed users to apply these predictive techniques to MOOC courses from various eligible platforms.

As MOOC offerings proliferate, the ability to "transfer" statistical knowledge between courses is increasingly crucial, especially if one wants to predict dropout in real time. Unfortunately, it is often difficult to take models built on past courses and apply them to new ones. In [3], Boyer and Veeramachaneni showed that models built on past courses don't always yield good predictive performance when applied to new courses.

Because there is generally only one dataset available per course, the ability to build robust models on MOOCs has naturally accompanied the rise of ensemble methods. Over the past twenty years, a flourishing predictive literature has appeared, offering various techniques for choosing and ensembling models in order to achieve high-performing predictors. A technique called "stacking" has proven particularly promising. In [5], Szeroski et. al. showed that stacking models usually perform as well as the best classifiers. They also confirmed that linear regression is well-suited to learning the metamodel, and introduced a novel approach based on tree models. The authors of [4] demonstrated the possibility of incrementally adding models to the "ensembling base" from a pool of thousands. Sakkis et. al. [10] used the stacking method to solve spam filtering problems, finding that it significantly improved performance over the benchmark.

In this paper, we explore a framework conducive to building robust predictive models applicable to MOOCs. Although we do address dropout prediction specifically, we also consider the broader possibilities for building predictive models from a set of courses.

## 2. PROBLEM SETTING AND DATA SETS

We place ourselves in the context of using past courses to build a predictive model for a unseen course. We use the term *source* courses for those courses whose data is used to build (train) the predictive models, and the term *target* course for the initially "unseen" course. We consider the general problem of predicting for each student  $i$  an outcome  $y_i^t$  at time  $t$  in the future. We suppose that we have access to information about each student's behavior through a set of features: for example, a behavioral vector  $x_i^t \in R^d$  describes the behavior of the student  $i$  at time  $t$ .

ID	Name	Platform	Students	Weeks
$C_0$	6002x13	edX	29,050	14
$C_1$	6002x12	edX	51,394	14
$C_2$	201x13	edX	12,243	9
$C_3$	3091x12	edX	24,493	12
$C_4$	3091x13	edX	12,276	14
$C_5$	aiplan_001	Coursera	9,010	5
$C_6$	aiplan_002	Coursera	6,608	5
$C_7$	aiplan_003	Coursera	5,408	5
$C_8$	animal_001	Coursera	8,577	5
$C_9$	animal_002	Coursera	5,431	5
$C_{10}$	astrotech_001	Coursera	6,251	6
$C_{11}$	codeyourself_001	Coursera	9,338	7
$C_{12}$	criticalthinking_1	Coursera	24,707	5
$C_{13}$	criticalthinking_2	Coursera	15,627	5
$C_{14}$	criticalthinking_3	Coursera	11,761	5

Figure 1: Summaries of courses used for experiments. The first set contain five courses from edX platform (Harvard-MIT), the second set contain ten courses from the EDI platform (University of Edinburgh).

We assume that the *source* courses and the *target* course share a non-empty set of behavioral features, such that we can restrict ourselves to this set when building our predictive models. As we will see below, this hypothesis is often verified in practice. In this context, our goal is to learn the statistical mapping from the behavior vector  $x^{w'}$  of a student in week  $w'$  to their particular outcome  $y^w$  in week  $w$ . To do this, we propose to learn a statistical model by leveraging data (both  $x^{w'}$  and  $y^w$ ) from previous courses.

**Data sets:** Our experiments are based on two sets of MOOC courses. The source set, on which we built and validated our methods, consists of five courses, and was provided by the edX platform. Its attributes are described in table 1. This dataset initially contained log files describing students' behavior on the platform. For each student in these courses, we extracted a set of 21 features on a weekly basis. The second, or "target," set of courses was given by the University of Edinburgh (EDI) and consists of 10 courses from Coursera, whose attributes are also described in table 1. The courses in this second set are shorter in duration (only 5 weeks), and contain less detailed features. They share only 11 features with the first set of courses.

To build a robust framework that could achieve reliable predictive models, we initially designed, trained and validated our different methods on the first set of 5 courses from edX. At the very end of this paper, we apply these models to the second set of courses. When building models on one course and applying its predictions to another, two issues must be overcome. First, the two courses might not share the same features (for example, the grade for p-sets during week 1 might be available for some courses and not for others). Second, they might not span the same number of weeks. We overcame these issues by only considering features and timespans common to all courses. Therefore, we first used 21 features and 9 weeks when we trained, tested and validated our models on the edX courses. We then restricted ourselves to an 11-feature, 5-week scheme when testing our procedure on the EDI courses.

### 3. METHODS

In this section, we describe the different approaches used to build a predictive model for dropout. We first describe common practices, and explore whether a single course can be used to build a predictive model for another course. We then explain how the aggregation of several data sources can be used to improve the predictive power of a model. Finally, we describe how a type of machine learning technique called "Ensembling methods" can be used to further boost the predictive power of models built from different courses.

#### 3.1 Naive approaches

**Simple models:** When building a supervised predictive model out of data sources, the first logical step involves training a single model on a particular dataset. Although plenty of classification algorithms exist, there is no systematic a-priori method to determine which one is best suited to a particular problem. This is the first hurdle that must be overcome when building a robust model.

The second hurdle involves choosing which prior course to train the model on. Although the first course  $s_1$  may have a distribution closer to that of our target course  $t$ ,  $s_2$  may have more samples, resulting in a better predictive model. Hence, we must choose both an algorithm and a prior course that, working together, will be most suitable for predicting outcomes in the new course.

**Merging sources:** Alternatively, one may ask, why not use all the data from all the courses? Could learning a predictive model on the concatenated data from all courses  $\{s_1, s_2, \dots, s_n\}$  result in a model that transfers better to new courses? This mitigates the problem of choosing among courses, but certainly does not solve the need to choose a modeling approach, as it raises a number of new questions. First, concatenating obscures the differences between courses, preventing a predictive model from making predictions within the environment of the original sample. Second, if the courses have different numbers of students, concatenating them can overweight the influence of the larger data sets. Though this may not be a concern in cases where all datasets are drawn from a single distribution, in our case, combining the datasets is likely to limit the overall information available.

Although those concerns could be addressed using different tricks (for example, adding a "dataset" feature to account for the particularities of models, or undersampling bigger datasets to balance their weight in the concatenated set), we instead sought a different and more systematic approach to building robust models.

#### 3.2 Ensembling methods to improve transfer of models in MOOCs

In this section, we leverage a type of machine learning technique called "Ensembling methods," often used to aggregate different predictive models. These techniques are now widely practiced after their successful deployment in the *Netflix*<sup>1</sup> challenge, in which hundreds of teams competed to build a precise recommendation system. They are used both in the industry and in public competitions, such as those held by Kaggle<sup>2</sup>, to improve the predictive power of models trained

<sup>1</sup><http://www.netflixprize.com/>

<sup>2</sup><https://www.kaggle.com/>

and tested on a single dataset<sup>3</sup>.

In this paper, we ask whether ensembling methods can in fact help in transferring models trained on one or more courses. What additional parameter tuning or methods do we have to develop to make this transference possible? Ordinarily, a data scientist uses ensembling techniques by selecting different subsets of features and training examples, learning algorithms, or parameters, and then building a set of predictors to ensemble. In the context of MOOCs, which have multiple courses, there is a natural split in the data we can exploit. We will demonstrate that in the some cases (for short term predictions), these approaches outperform the performance of the transferred predictive models built on a single course data and from a single algorithm.

We will discuss the different methods explored with respect to the three following dimensions :

- A set of pre-trained predictors  $E = \{p_1, \dots, p_n\}$
- A set of rules to combine the predictions of different algorithms. We call these rules "voting rules" and note them  $R = \{R_1, \dots, R_p\}$
- A structure  $S$ , which specifies in which order and to which predictions these rules should be applied.

**Predictors:** The first step in building a transferring method for dropout prediction is to train a set of predictive algorithms on data available from past courses. Given  $N$  source courses and  $P$  predictive models to train, this produces  $N \times P = H$  predictors  $\{p_1, \dots, p_H\}$ .

We trained four classification algorithms (RandomForest, Logistic Regression, SVM, and Nearest Neighbors) on each course. For each of these algorithms, we used 5-fold cross-validation to optimize the parameters. We note that for each of the past available courses, we left a holdout subset of 20% for a later-stage parameters optimization.

**Fusing methods:** One can combine a set of underlying predictions  $\{p_1(x), \dots, p_H(x)\}$  in infinite ways. Below, we list three common ways of ensembling that have been proven to perform well over a broad range of applications:

- Averaging ( $R_1$ ). The most common fusion method consists of averaging the predictions of different underlying predictors.

$$p_{norm}(x) = \frac{1}{H} \sum_{i=1}^H p_i(x)$$

- Normalized averaging ( $R_2$ ). When combining disparate predictive methods, some predictors might produce estimations in  $\{0.49, 0.51\}$ , while others produce estimations in  $\{0, 1\}$ . To account for the diversity of ranges from one predictor to the next, one can normalize the predictions of each predictor before averaging them.

$$p_{avg}(x) = \frac{1}{H} \sum_{i=1}^H \frac{p_i(x) - \min_{z \in t} p_i(z)}{\max_{z \in t} p_i(z) - \min_{z \in t} p_i(z)}$$

- Rank voting ( $R_3$ ) In addition to differences in the range of probabilities, may also differ in how fast they

<sup>3</sup><https://inclass.kaggle.com/c/mooc-dropout-prediction>

vary with the input. To mitigate this behavior (which might cause the overall prediction to overweight very sensitive predictors), one can rank the probabilities within the target set first, then average and normalize the resulting ranks of different .

$$p_{rank}(x) = \frac{1}{H} \sum_{i=1}^H \frac{rank(p_i(x)) - 1}{N_t}$$

where  $rank(p_i(x))$  refers to the rank of sample  $p_i(x)$  in the set  $\{p_i(z), z \in t\}$

We note that none of these techniques assume anything about the relative performance of different algorithms. We call those voting schemes “symmetric” because they treat each predictor in the same way. Our next set of methods allows us to fuse predictions by accounting for the varying performances of different predictors, and allowing us to put more weight on the “best” predictors. To identify these weights, we use the holdout subset of our source courses, and develop a method known as *stacking* as follows:

- Stacking ( $R_4$ ) We concatenate all the holdout subsets from all source courses  $X_{HO} \in R^{N_{HO} \times d}$  and apply all pretrained predictors  $\{p_1(x), \dots, p_H(x)\}$  on this dataset. The output of this procedure  $Y_{HO} \in R^{N_{HO} \times H}$  is then considered as a new training dataset. We apply a logistic regression on this output to learn the weights for each predictor.

**Structures:** The last component of an ensembling method is the structure, within which predictions are merged together. Two example structures are shown in figure 2. Structures can influence the final performance of the method. Given a set of predictors and a set of fusing methods, the

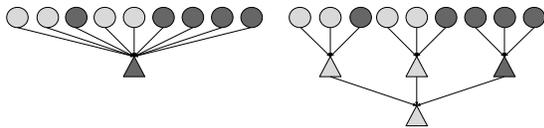


Figure 2: Illustration of two structures used to combine the same set of predictions using a simple voting rule ( $R_1$ ) (color code is 1 for blacks and 0 for whites). The two different structures result in two different predicted outcomes.

”structure” is the sequence in which said predictors are fused in order to produce the final output.

**Learning the structure:** We posit that the structure of votes could influence the performance of the overall ensembling method. Due to the potentially arbitrary number of “layers,” the number of possible structures is infinite. We restrict ourselves to structures with a high degree of symmetry. We enumerate a subset of structures in the figure 4. We then use algorithm 1 to compare the performance of the preselected structures. Our goal is to find the structure that will yield the highest performing predictor when applied to target courses.

For this comparison to be independent of the choice of target course, we consider each one of the five edX courses as the target course successively, calling them  $C_0, C_1, C_2, C_3$  and  $C_4$ . The remaining four act as source courses. We then

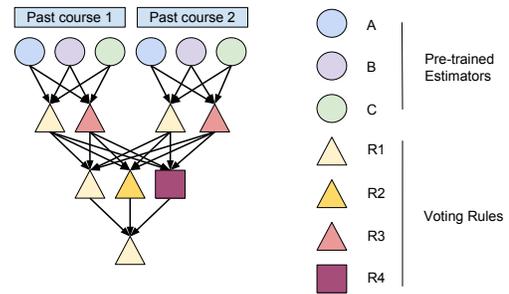


Figure 3: Example of a complex voting structure built on top of two data sources (past courses). The predictions of the different predictors are first aggregated by course then aggregated across courses.

aggregate our results by averaging the performance over the five permutations. We also remark that, in order to learn the metamodel necessary to the stacking rule, we separate all source course into a train and a validation set, as explained in algorithm 1.

**Data:** Full Data for the 5 edX courses

**Result:** Evaluate performance of structures

**for problem in  $P$  do**

**initialization :** Split each dataset into a training and a validation subset (80% - 20%);

**for  $t$  in set of courses do**

        Train each of the predictive algorithms (Random Forest, SVM, Logistic Regression, Nearest Neighbors) on each train set for the 4 remaining courses;

**for  $s$  in set of Structures do**

**if  $s$  requires training then**

                concatenate validation sets for 4 remaining courses;

                train  $s$  on this dataset;

**else**

                pass

**end**

            measure ROC AUC :  $AUC_s^{p,t}$ ;

**end**

**end**

**end**

**Algorithm 1:** Comparing performance of different ensembling structures.

## 4. RESULTS ON DROPOUT PREDICTION

MOOC platforms offer courses that span a particular length of time, typically around 12 weeks. A large cohort of students register for each of these courses, but only a fraction of this cohort usually remains at the end of the class.

We consider the common problem of predicting which students will remain in the class. Specifically, given a “current week”  $w_c$  and a “prediction week”  $w_p$  our goal is to identify which of the students present in the class at week  $w_c$  will have dropped out by week  $w_p$ . We call this particular problem  $(w_c, w_p)$ , and we remark that, given a particular course lasting  $W$  weeks, there exist exactly  $\frac{W \cdot (W-1)}{2}$  potential problems of this type.

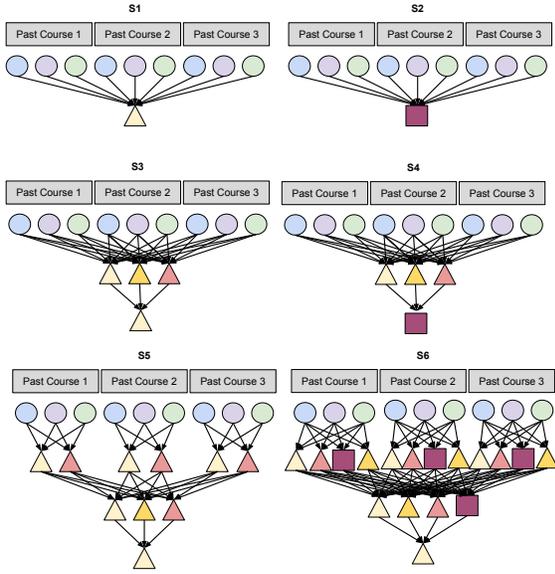


Figure 4: The six structures evaluated on the MOOCs dropout prediction problem. Only  $S2, S4$  and  $S6$  require training on the validation data because of the presence of a "stacking vote rule". (Note that for simplicity the diagrams shows only 3 courses and 3 predictive algorithms per courses whereas we used 4 and 4).

In the first to parts of this section, we use the five courses from edX (described in 1) to experiment and build our predictive models. We noted that the five courses from edX shared 21 behavioral features altogether. In the third part of this section, we show that these models indeed perform significantly better than our benchmark, even on courses from a different platform, Coursera.

#### 4.1 Performance metrics and benchmarks

**Evaluation metrics:** To measure the performance of our predictive algorithm, we rely on the AUC-ROC metric, which is commonly used in dropout prediction. Because not all courses last for the same amount of time, we restrict ourselves to problems acceptable for all courses; i.e. the set

$$P = \{(w_c, w_p) \text{ s.t. } w_c < w_p \text{ and } w_p < W_{course} \forall \text{ course}\}$$

For the five courses used in this study,  $W = 12$ , meaning we can experiment on  $|P| = 66$  different prediction problems. When comparing the performance of algorithms between problems, it becomes clear that some situations are intrinsically more difficult to predict than others. For instance, a short-term prediction problem (e.g., (6, 5)) will generally yield higher performance than a long term problem (e.g., (6, 1)). Similarly, some courses are more suited for predictions than others, due to the size of the student cohort or the volatility of students within that cohort.

To mitigate this, we *normalize* the performance, and use the following metric to measure the performance of an algorithm  $a$  on a problem  $p$  and on *target* course  $t$ :

$$DAUC_a^{p,t} = \max_{a' \in A} (AUC_{a'}^{p,t}) - AUC_a^{p,t}$$

In other words, we subtract the actual AUC of an algorithm from the best observed AUC of any other algorithm on this particular problem for this particular *target* course. In this configuration, a lower DAUC should be considered to indicate a better performance. In particular,  $DAUC_a^{p,t} = 0$  exactly means that  $a$  is the best algorithm for this particular problem and target course.

To appreciate this metric over different problems, we display both the mean and the variance of the DAUC. In order to account for the different performance on different types of problems, we introduce two sets of problems, for which we choose to average the DAUC:

#### Two subset of problems

$P$  Mean ROC AUC obtained on the 66 available problems

$P_s$  Mean ROC AUC obtained on three 'short term' prediction problems ( $\{(5, 6), (8, 9), (11, 12)\}$ )

**Benchmarks :** A simple approach to building predictive models is to train a classifier on a *source* course and use it to make predictions on the *target* course. In figure 5, we report the results obtained by training four different classification algorithms on a *source* course (for course 1 to 4) and applying it to the *target* course ( $C_0$ ). We use 5-fold cross-validation on the training set, and we tune the parameters independently for each method, each source and each prediction problems.

A more systematic approach consists of building predictive models on the concatenation of all available data. In addition to avoiding the hurdle of having to 'guess' which course should be chosen as the *source* course, this approach also allows us to leverage more (and more diverse) data to train predictive models.

As shown in figure 5, we observed that, regardless of the algorithms used, models trained on the concatenation of all available data sources always performed better than the best models trained on a single course. This is true both in terms of average DAUC over all problems in  $P$ , and in terms of variance of the DAUC across those same problems.

#### 4.2 Building robust models

##### Improvement through Merging Methods :

Concatenating the data from past courses undoubtedly improved the algorithms' predictive power, both in terms of average DAUC and variance. To further improve the average performance, and to reduce the variance of our dropout prediction system, we then leveraged the ensembling methods presented above. Instead of restricting ourselves to a choice of a single predictive algorithm, we trained four of them (SVM, Random Forest, Logistic Regression and Nearest Neighbors) and merged their predictions using a simple  $R_1$  voting rule.

Figure 5 shows the average DAUC and its variance for different algorithms, as well as their "merged" version through an  $R_1$  rule. Comparing the result obtained by the "merged" method with those of the four single algorithms, we observe that the merged method always performs comparably to the best single algorithm, beating all competitors on courses  $C_2$  and  $C_4$ , and behaving comparably on courses  $C_1$  and  $C_3$ ). This is true both in terms of average performance and in terms of variance.

Next, we apply this same  $R_1$  rule to merge the predictions built on the data concatenated from all available *source* courses. Here, the results unveil a lower DAUC average and variance for the "merged" method on the concatenated data than for any other algorithm on the same data. Moreover, this method performs better than those "merged" methods trained only on a single course, in terms of both average and variance. Through an "all-algo all-data" kind of method, we have achieved a more reliable and more accurate predictive model, on average, over all possible prediction problems. In the next section we will see that, for certain type of problems, it is possible to improve this model significantly by using a more complex type of ensembling method called stacking.

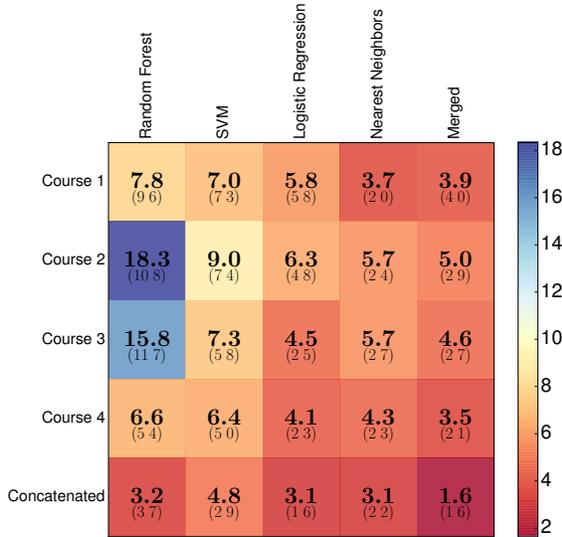


Figure 5: Average (standard deviation) of the DAUC (x100) for all prediction problems ( $P$ ) on target course  $C_0$ . The x-axis contains different predictive algorithms, the y-axis contains different data source.

### Optimizing the vote Structure:

The figures above show promising results for ensembling methods in the context of dropout predictions. This encouraged us to explore different ensembling methods to further improve the and performance and/or reliability of our dropout prediction system.

Our ensembling strategy uses all available estimators described above (those built on a single-source course as well as those built on the concatenated data). It then applies one of the manually pre-selected structures as shown in figure 4. We then use algorithm 1 to learn the structure.

Figure 6 displays the DAUC obtained by different ensembling structures according to algorithm 1. We differentiate our observations according to the subset of problems over which the average is computed ( $P, P_s$  and  $P_l$ ).

- Over all problems (average over  $P$ ), we first remark that the structure has only a very small impact on both the average performance and the variance of the predictive method. We also remark, however, that structure  $S_6$  yields slightly better results, both in terms of average DAUC and in terms of variance.

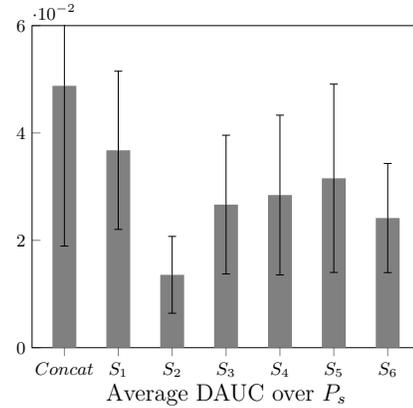
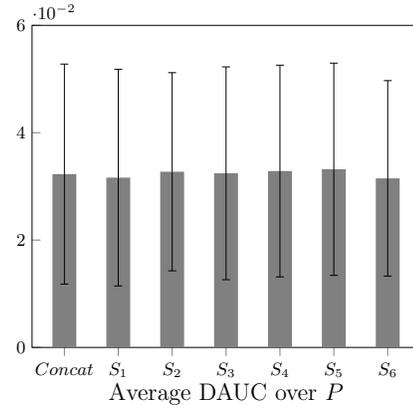


Figure 6: Average DAUC over each edX course taken as target, as computed by algorithm 1.

- Over the short-term problems, (average over  $P_s$ ), we observe a lot more difference across the different structures. By far, structure  $S_2$  is the best performer for this type of problem, with a DAUC of 0.013 on average compared to 0.049 for the merged method discussed in the previous section.
- Over long-term problems (average over  $P_l$ ) the difference between structures is significant, and thus not as big as for the short term problems. The best structure here is  $S_4$ .

### 4.3 Transferring across MOOC platforms

Having achieved robust methods for dropout prediction on different predictive problems, we now test our method on a new set of courses, composed of 10 courses from the University of Edinburgh. Rather than testing this method on the holdout course as explained in algorithm 1, this set of courses present the additional difficulty of being derived from another MOOC platform (Coursera), thus having potentially very different statistics for the features used in our models. For example, overlap between the features of our 5 first courses (from edX) and the features of those new courses is not total. Whereas our initial 5 courses shared 21 common features, they share only 12 features with this new set of courses. In figure 7 we report the DAUC obtained on average over all possible prediction problems and over all ten

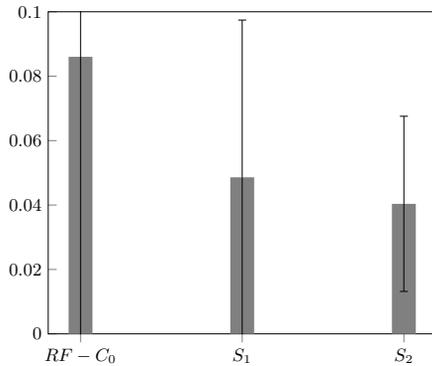


Figure 7: Average DAUC over all prediction problems on the 10 UDI courses taken as target (only edX courses are taken as source courses).

ID	Name	AUC short	AUC all
$C_5$	aiplan_001	0.82	0.75
$C_6$	aiplan_002	0.79	0.70
$C_7$	aiplan_003	0.81	0.74
$C_8$	animal_001	0.73	0.64
$C_9$	animal_002	0.75	0.67
$C_{10}$	astrotech_001	0.77	0.67
$C_{11}$	codeyourself_001	0.84	0.74
$C_{12}$	criticalthinking_1	0.71	0.63
$C_{13}$	criticalthinking_2	0.80	0.71
$C_{14}$	criticalthinking_3	0.78	0.70

Figure 8: AUC achieved by  $S_2$  ensembling method built on the five edX courses and applied on the 10 UDI courses.

courses (taken as target). We display the results for a simple Random Forest algorithm built on the first edX course, for an ensembling method based on the  $S_1$  structure, and finally for the best-performing ensembling method (from the experiment in the previous sub section) based on  $S_2$  structure. All the ensembling methods are built on top of estimators from all the five edX courses (for the four algorithms : Random Forest, SVM, Logistic Regression, Nearest Neighbor). Table 8 reports the absolute performance of the best technique ( $S_2$ ) structure in terms of average AUC across different prediction problems for each course.

We remark first that the  $S_2$  performs again significantly better than both the simple algorithm and the simple ensembling method. We also note that the absolute performance achieved by this best ensembling technique is relatively high, given the small number of features available and the different origins of the two set of courses.

## 5. KEY FINDINGS

Our key findings can be summarized in three categories, corresponding to the three sets of questions described in the introduction:

**Purpose** We showed that even though MOOC courses span different numbers of weeks and have different characteristics, one can usually find sufficient overlap between courses to perform nontrivial prediction tasks.

**Data** We showed that using more courses as training data improved the predictive power significantly. We also proved that this predictive power was sufficient to apply the model built on one particular MOOC platform to another platform.

**Method** First we showed that, both in the case of a single course model and in the case of a model built from several courses, using simple ensembling methods between algorithms significantly improved the performance. When compared to a single algorithm trained on all available courses, a simple ensemble methods improved the AUC by an average of 1.5 to 3.2 points. Secondly, we proved that in certain use cases (for instance, short term dropout prediction problems), using more complex ensembling structure can significantly boost performance. For short term prediction problems, using a  $S_2$ -like structure of ensembling resulted in no less than a 4 point AUC improvement on average.

Finally, our best method was successful when applied to a set of unseen courses. On the ten never-before-seen Coursera courses, our method obtained a 0.70 average AUC overall and a 0.78 average AUC on short term prediction problems (one week ahead). This completes our case that a high-performance predictive model can be built from a set of previous courses, and that ensembling methods appear to be a suitable framework to build such models.

## 6. DISCUSSION

When trying to estimate the actual benefit of such techniques on the real life of students and teachers on MOOC platforms, one has to make several assumptions that may only be verified after several years of implementation.

The main assumption is the possibility to reduce churn of

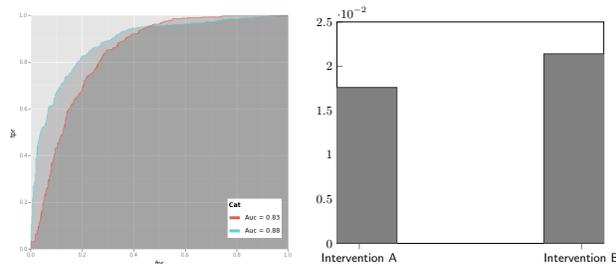


Figure 9: Estimated increase in number of students completing a typical MOOC course. Intervention A refers to an intervention based on a simple predictive model built on a course. Intervention B refers to an intervention of type  $S_2$ . See description for numerical assumptions.

students through personalized intervention. This is not obvious, as many argue that most dropout students were intrinsically not interested in the content of the class, and could therefore not be fruitfully intervened with. Most MOOC providers, however, agree that a good chunk of each cohort could be prevented from dropping out of the class thanks to some customized and well-adapted interventions. Identifying dropout students (the example describe in this paper) could enable a concrete set of interventions to be done, with extra resource help, additional videos or motivating resources particularly tailored to potential "dropout" students. For our purpose we will assume that a tailored intervention will save 1% of all potential dropout students.

Given a fixed false-positive rate, arguably necessary to design an intervention targeted for dropout students, the purpose of the predictive methods described above can be understood as the maximization of the true-positive rate: the ratio of predicted dropout students to the number of total dropout students.

Taking a weekly intervention framework, in which an intervention is conducted for potential dropout students at the end of each week, we showed in the previous section that ensembling methods (particularly the  $S_2$  structure) were able to perform around 0.05 AUC point better than other more straightforward models (particularly an "all-algo all-sources" method). In figure 9, we show the example of two ROC AUC separated by 0.05. We remark that with a constraint of 10% on the false positive rate, we obtain a difference of around 20% in the true positive rate .

Given a typical MOOC class – 10 weeks long, starting with 10.000 students, and with a typical weekly dropout rate of 20% per week we display in figure 9 the simulated data of the number of students completing the course. When an intervention based on a straightforward predictive model is simulated, it increases the number of students finishing the course by around 1.7%, whereas an  $S_2$  based predictive model would increase it by around 2.1% (an additional 50 student completions overall).

## 7. CONCLUSION

In this paper, we developed a framework to address the main challenges faced when applying predictive analytics to MOOCs: How to build models that transfer well across courses and platforms?

To do this, we used ensembling methods, as well as a broad

range of algorithms and a rigorous training procedure. We explored different variations of these techniques and reported the results obtained on a first set of five courses from the edX platform. We introduced a novel performance metric, allowing for performance comparison across prediction problems and target courses. These results show that ensembling methods improved the accuracy of prediction, both on average and in terms of variance. We also showed that "stacking" (or learning metamodels on top of a set of base predictors) can significantly boost performance in the case of short term prediction problems.

Eventually, we tested the method developed in a first part (on the first set of five courses from edX MOOC platform) on ten courses from the University of Edinburgh MOOC platform. We reported the results obtained in terms of AUC and showed that the method developed performed very well on those new courses, too.

We argue that our paper demonstrates a robust framework to develop predictive algorithms that are transferable across online courses.

## 8. REFERENCES

- [1] P. Adamopoulos. What makes a great mooc? an interdisciplinary analysis of student retention in online courses. 2013.
- [2] S. Boyer, B. U. Gelman, B. Schreck, and K. Veeramachaneni. Data science foundry for moocs. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.
- [3] S. Boyer and K. Veeramachaneni. Transfer learning for predictive models in massive open online courses. In *Artificial Intelligence in Education*, pages 54–63. Springer, 2015.
- [4] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18. ACM, 2004.
- [5] S. Džeroski and B. Ženko. Is combining classifiers with stacking better than selecting the best one? *Machine learning*, 54(3):255–273, 2004.
- [6] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and best practices in and around MOOCs*, 7, 2014.
- [7] I. Lykourantzou, I. Giannoukos, V. Nikolopoulos, G. Mparadis, and V. Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53(3):950–965, 2009.
- [8] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.
- [9] B. Poellhuber, M. Chomienne, and T. Karsenti. The effect of peer collaboration and collaborative learning on self-efficacy and persistence in a learner-paced continuous intake model. *International Journal of E-Learning & Distance Education*, 22(3):41–62, 2008.
- [10] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. *arXiv preprint cs/0106040*, 2001.

# A Comparative Analysis of Techniques for Predicting Student Performance

Hana Bydžovská  
CSU and KD Lab Faculty of Informatics  
Masaryk University, Brno  
bydzovska@fi.muni.cz

## ABSTRACT

The problem of student final grade prediction in a particular course has recently been addressed using data mining techniques. In this paper, we present two different approaches solving this task. Both approaches are validated on 138 courses which were offered to students of the Faculty of Informatics of Masaryk University between the years of 2010 and 2013. The first approach is based on classification and regression algorithms that search for patterns in study-related data and also data about students' social behavior. We prove that students' social behavior characteristics improve prediction for a quarter of courses. The second approach is based on collaborative filtering techniques. We predict the final grades based on previous achievements of similar students. The results show that both approaches reached similar average results and can be beneficially utilized for student final grade prediction. The first approach reaches significantly better results for courses with a small number of students. In contrary, the second approach achieves significantly better results for mathematical courses. We also identified groups of courses for which we are not able to predict the grades reliably. Finally, we are able to correctly identify half of all failures (that constitute less than a quarter of all grades) and predict the final grades only with the error of one degree in the grade scale.

## Keywords

Student performance prediction, student similarity, classification, regression, collaborative filtering.

## 1. INTRODUCTION

One of the key problems of educational data mining is to design student models that would predict the student performance. Once we have a reliable performance prediction, it can be used in many contexts: for identifying weak students [14], for guiding the adaptive behavior in intelligent tutoring systems [10], or for providing a feedback to students.

Our specific problem is the following: we have access to data about students, their study achievements and their behavior characteristics stored in the university information system and we want to predict students' final grades. The predictions are useful at the beginning of each semester to help students with planning their workload in the whole semester. We also beneficially use this information to design a course enrollment recommender system. The early grade prediction is more difficult since we have no a priori information about students' knowledge, skills or enthusiasm for particular courses. It has been proven [4] that the data about the activity of students during the semester improves the prediction.

The problem of the student grade prediction in a particular course has recently been addressed using data mining techniques. Researchers usually examine study-related records, e.g. the age, the gender, and the field of study [9] because of their easy

availability in university information systems. Moreover, they attempt to identify additional characteristics that can lead to better understanding of students' behavior, e.g. their habits [6] or parents' education [13]. The most typical way how to obtain such data is to conduct questionnaires. Masaryk University has more than 40,000 active students and we try to predict the grades as accurately as possible for all of them. We cannot rely on data obtained by questionnaires since they tend to have a lower response rate. Therefore, only the data originated from the Information System of Masaryk University (IS MU) are employed for our experiments.

The goal of this research is to predict students' grades with the major emphasis on the detection of students who can fail to meet the course requirements. Therefore, we are dealing with the following two main tasks:

- prediction of students' success or failure,
- prediction of the students' final grades.

In this paper, we present two different approaches moving towards our objectives. The first approach is based on the state of the art educational data mining techniques: classification and regression analysis [12]. We created an ensemble learner to utilize the strength of the both techniques. We also present a new type of data about students' social behavior originated from IS MU that can improve the predictions. The second approach is based on collaborative filtering techniques [5] applied to the educational context. We mapped the users-item-rating problem to the student-course-grade problem and predict the final grades based on previous achievements of similar students. This paper describes both approaches in detail, compares them and reports their advantages and disadvantages.

## 2. DESIGNED METHODS EVALUATION

Historical data were employed for experiments allowing us to evaluate both designed approaches. We processed data about 138 courses which were offered to the students at the Faculty of Informatics. We used only data stored in IS MU in the time of students' enrollments. We omitted freshmen students because we had no data about them in the system. The data comprised of 3,584 students. The two independent data sets were used. The training set consisted of the data collected between the years of 2010 and 2012 (37,005 instances) and was used for the identification of the most suitable methods with their settings. The test set consisted of the data from the year 2013 (11,026 instances) and was used for the validation of the methods on different data.

The following grade scale was used: 1 (excellent), 1.5 (very good), 2 (good), 2.5 (satisfactory), 3 (sufficient), 4 (failed or waived). The value 4 represents student's failure; the others represent a full completion. We evaluated approaches using the *mean absolute error* (MAE). The technique measures how close predictions are to the real outcomes. Lower values represent better

results. The measure is commonly used for grade prediction evaluation. In the educational environment, one of the most important issues is to reveal weak students. Therefore, we also computed the *sensitivity* (also called recall). Categorizing students only as successful or unsuccessful, the sensitivity measures the proportion of unsuccessful students who are correctly classified as unsuccessful. For students' success or failure prediction we also utilized *F1 score* that conveys the balance between the precision and the recall.

### 3. STUDENTS' CHARACTERISTICS

#### 3.1 Study-related Data

Classification and regression are the most often used techniques for student performance prediction [12]. Researchers usually examined study-related (SR) data. Our study-related data contained common attributes such as the gender, the year of birth, the year of admission, the number of credits gained from passed courses, or the average grades. We built a classifier for each investigated course based on the training set and evaluated the results using the 10-fold cross validation. The method that achieved best results was subsequently validated on the test set.

##### 3.1.1 Student success/failure prediction

The first task was to reveal unsuccessful students. Two prediction classes were considered: students' success (def. 1: grades 1–3) and failure (def. 2: grade 4). Widely utilized classification algorithms were employed: Support Vector Machines (SVM), Random Forests, Rule-based classifier (OneR), Trees (J48), Part, IB1, and Naive Bayes (NB). As the baseline we defined a model which always predicts failure. Table 1 confirms that SVM achieved the best performance.

**Table 1. Classification algorithms results**

Rank	Method	F1	MAE	Sensitivity
1	SVM	0.559	0.161	0.444
2	NB	0.554	0.251	0.467
3	J48	0.552	0.182	0.397
4	Random Forests	0.550	0.173	0.362
5	Part	0.543	0.202	0.417
6	IB1	0.536	0.216	0.436
7	OneR	0.508	0.183	0.321
8	Baseline	0.326	0.822	1

##### 3.1.2 Grade prediction

The regression is a commonly used technique for student grade prediction. Widely utilized regression algorithms were selected: SVM Reg., Random Forest, IBk, RepTree, Linear Regression, and Additive Regression. The baseline model predicts the average grade of the training set of a given course. The best results (see Table 2) were achieved by support vector machine (SVM Reg.).

##### 3.1.3 Conclusion

For each task, the best method was selected and an ensemble learner was built. If the classifiers (SVM or SVM Reg.) predicted the failure or the grade 4, then the ensemble learner also predicted the failure. Otherwise, it resulted in the value of the grade predicted by the SVM Reg. classifier. Finally, the overall performance of this approach could be seen in Table 3. We also

evaluated the classifiers on the test set. The results indicated that we were able to reveal almost half of the unsuccessful students even if the task was difficult due to the fact that all unsuccessful students constitute less than a quarter of all students. The prediction error was about 0.75 on average which was almost 1.5 degree in the grade scale.

**Table 2. Regression algorithms results**

Rank	Method	MAE	Sensitivity
1	SVM Reg.	0.605	0.196
2	Linear Reg.	0.615	0.152
3	Additive Reg.	0.634	0.165
4	RepTree	0.643	0.184
5	Random Forests	0.668	0.216
6	IBk	0.767	0.294
7	Baseline	0.806	0

**Table 3. Global SVM results**

Data Set	MAE	Sensitivity
Training Set	0.701	0.524
Test Set	0.744	0.414

### 3.2 Social Behavior Data

Recent researches are often based on finding additional data that can improve the prediction accuracy. Our improvements have been achieved through adding social behavior (SB) data to the original data set [1]. This specific type of data originating from IS MU described the students' behavior characteristics and their mutual cooperation. We focused on statistical data that represented an interaction among students: posts and comments in discussion forums, e-mails statistics, publication co-authoring, or files sharing. This information served as the basis for computing social ties among students and building a sociogram. From this sociogram, new features like weighted average grades of friends can be easily derived. Using Pajek [11], we also computed additional standard graph features [3] like degree (the number of the friends), weighted degree (degree weighted by the strength of ties), centrality or betweenness (the importance measure for each student in the network). Moreover, we collected data about students' disclosure from different system sections. By default, IS MU does not provide a complete list of classmates due to the students' privacy. Students have to actively disclose themselves to become visible for their classmates. We can also calculate how many times students attended courses of a certain teacher. Among others, students can also mark offered courses as favorite.

H1: Hypothesis supposes that students' social ties correlated with the students performance.

Other ensemble learners trained on data sets containing social attributes were built. The other settings were maintained. The comparison of the results can be seen in Table 4. The MAE score was slightly lower on average. However, for 32 courses in the test set, the difference in MAE was significantly better using social behavior data (min: 0.1; average: 0.178; max: 0.734). Only 5 courses achieved worse results (min. 0.1; average: 0.12; max: 0.21). For the rest courses, the difference was negligible.

**Table 4. Adding social behavior attributes to the data set**

Data Set	Attributes	MAE	Sensitivity
Training Set	SR	0.701	0.524
	SR + SB	0.629	0.528
Test Set	SR	0.744	0.414
	SR + SB	0.688	0.427

The sorted list of selected attributes was constructed. In Table 5, we present the top five social behavior attributes that significantly affected the results.

**Table 5. The most interesting social behavior attributes**

Rank	Avg. Ord.	Attribute
1	13.328	the betweenness
2	16.252	the information if the course was marked as favorite
3	18.694	the centrality
4	22.464	the weighted degree
5	29.807	the number of times when a student attended any course with the same teacher

H1 was confirmed. Data about students' behavior improved the predictions. Based on the most significant attributes, we assumed that the assistance of students' friends had increased the probability to pass the courses.

## 4. STUDENTS' GRADES

We also focused on methods utilized in recommender systems [5]. The data about user-item-rating triples were replaced by student-course-grade triples and we focused on the similarities among students' grades.

H2: Our hypothesis supposed that students' knowledge can be characterized by the grades of courses that students enrolled during their studies. Based on this information we could select students with similar interests and knowledge and subsequently predict whether a particular student has sufficient skills needed for a particular course.

### 4.1 Grade Prediction

Our preliminary work can be found in [2]. However, the approach suffered from several limitations that we overcome in this paper.

The first step was to build a similarity matrix  $G$  where rows represented students and columns represented courses. Although we predicted grades for 138 courses, the matrix  $G$  has 499 columns since we analyzed all students' grades (e.g. courses from the other faculties, courses not offered now). Grades obtained by all students from the training set formed the matrix. If a student did not attend a particular course, the corresponding cell remained empty. The aim was to complete cells defining students' grades from the investigated courses enrolled by students in 2012 (marked by symbol ?).

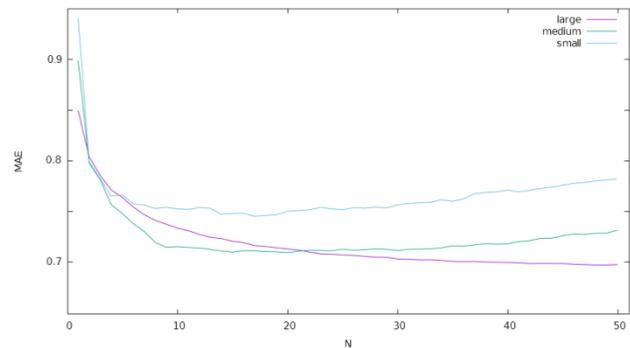
Using the vectors of grades from the matrix  $G$ , we computed the similarity between all students enrolled in a course  $c$  in 2012 and all students previously also enrolled in  $c$  in 2010 or 2011.

**Example of Matrix G**

Students / Courses	$c_1$	$c_2$	$c_3$	$c_4$
$s_1$	2	?		?
$s_2$	?	2.5	3	?
$s_3$	1		2.5	3
$s_4$		2		1.5

Widely utilized similarity metrics were used for the calculation of the students' similarity: Mean absolute difference (MAD), Root mean squared difference (RMSD), Cosine similarity (COS), and Pearson's correlation coefficient (PC). All metrics compare grades of students' shared courses. The average number of courses shared by students was 10.

Subsequently, the appropriate neighborhood of the most similar students to the examined student could be selected to influence the predicted final grade. We utilize the idea of a baseline user [7]. We selected such students to the neighborhood who were more similar to the investigated student than the investigated student was to the baseline student. We decided to calculate two types of baseline students: an average student (the average grade for each course) and a uniform student (the average grade through all courses: 2.5). The neighborhood of the top 25 students showed reasonable results. However, for smaller courses, 25 students could be all students enrolled in the course in one year. Therefore, we have decided to define three categories of courses with respect to the course occupancy: small ( $\leq 30$  students), medium (30–70 students), and large ( $\geq 70$  students). Therefore, we analyzed the suitable size of the neighborhood for courses with the different occupancy. Figure 1 shows the relationship between MAE and the cardinality of  $N$ . We selected the size of neighborhood as follows: 10 for small courses, 15 for medium courses, and 30 for large courses. In the figure, we can also see that the prediction for smaller courses was the most challenging.

**Figure 1. Relationship between MAE and the size of neighborhood with respect to the course occupancy**

The final grades were estimated from the grades of similar students belonging to the computed neighborhood. Simple methods as mean, max, median as well as advanced methods utilizing significance weighting were utilized.

Table 6 introduces the top five combinations of the similarity methods, methods for the neighborhood selection and the grade estimation functions. The method utilizing a baseline user needed a large neighborhood for each student ( $|N| = 376$  on average). In the production system, it was very important to lower the ties

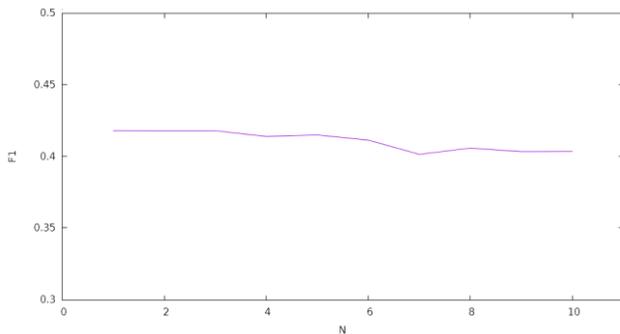
among students due to the recalculation of all similarities in the system during the course enrollment process to be up to date for students. Therefore, different neighborhood was selected even if the MAE score could be slightly higher. For efficiency reasons, we selected the third one for the implementation in the system.

**Table 6. Similarity methods comparison**

Rank	Method	N	MAE	Sensitivity
1	PC + average student + sig. weighting	376	0.648	0.248
2	PC + uniform student + sig. weighting	378	0.648	0.248
3	PC + Top  N  + sig. weighting	10/15/30	0.650	0.267
4	RMSD + Top  N  + median	10/15/30	0.651	0.211
5	PC + Top 25 + Pred	25	0.657	0.274

## 4.2 Student Success/Failure Prediction

The majority of students passed examined courses. Therefore, we searched for a smaller neighborhood in order to reveal more unsuccessful students. As you can see in Figure 2, the highest F1 was reached when we included only the most similar student. However, the method suffered by a low precision. Therefore, we predicted failure even if the method for grade prediction (3<sup>rd</sup> row Table 6) predicted grade worse than 2.4 (average grade). The precision was improved and still we found the sufficient number of unsuccessful students. The final results of methods were: MAE = 0.174, sensitivity = 0.413.



**Figure 2. Relationship between F1 and the size of the neighborhood**

## 4.3 Course similarity

Any change in the similarity matrix  $G$  could lead to the recalculation since the similarity of students was calculated from all students' grades.

H3: Our third hypothesis supposed that similar courses required similar skills of students to pass. It should decrease the computational cost and do not significantly lower the prediction accuracy when we use only grades of similar courses for predictions instead of all attended courses.

### 4.3.1 Students' grades

The collaborative filtering approach based on similarity of item to item was utilized and the *adjusted cosine similarity* was computed from the previously defined similarity matrix  $G$  for each pair of

courses. Subsequently, we utilized the average link clustering [8] to group the investigated courses based on this similarity measure. The resulted clusters defined the groups of similar courses.

Finally, when we predicted the students' grades of a certain course, we reduced the computations to the grades obtained from courses belonging to the same cluster as the investigated course. 110 of all investigated courses belonged to one of the 37 clusters. The number of courses in one cluster ranged from 2 to 15. The average number of courses in one cluster was 3. The average number of students' shared courses was also 3.

### 4.3.2 Course Characteristics

Students search for useful information about courses in the Course Catalog that help them to decide whether or not they should enroll the course. We selected different course characteristics and attempted to identify dependencies among courses. Similarity of courses  $a$  and  $b$  was defined by the weighted sum of the similarities of the selected course characteristics  $t \in T$ :

$$sim(a, b) = \sum_{t \in T} w_t \text{dist}(a_t, b_t)$$

where  $w$  defined the weight of the examined characteristic. The weights of the characteristics were set with respect to maximize the grade prediction accuracy. The similarity for each pair of courses was calculated. The selected characteristics and distance metrics  $dist$  were the following:

*Prerequisites* define a set of courses that had to be passed before students could enroll a certain course. The similarity was set to the value of 1 if the compared course belonged to the prerequisites; 0 otherwise. The weight of this characteristic was set to 1 because the prerequisites denoted a significant dependence.

*Literature* contains the recommended literature for particular courses that can be characterized by the set of assigned authors. The similarity of the set of authors  $A$  and the set of authors  $B$  is given by Jaccard's coefficient. The characteristics weight was set to the value of 0.9 due to the hypothesis that authors do not frequently publish in different fields. Therefore, the literature could constitute strong ties among courses.

The *course content* was represented by the text about the study subject and outline what students should learn in the course. We cut the STOP words from the text and utilized stemming to get the roots of the words. TF-IDF was utilized for defining the importance of each word in the texts. Subsequently, the Cosine similarity measure was used for the processing of the final vector representation of the words' importance. The characteristics weight was set to the value of 0.7.

*Teachers* of a course could be divided into two groups: lecturers and tutors. Weighted Jaccard's coefficient was used for comparing the teachers of the two courses. The weight of the lecturers was set to the value of 1 and 0.5 for seminar tutors. The weight of characteristic was set to the value of 0.6.

*Course supervisor* patronize the courses. The similarity was set to the value of 1 if the compared courses had the same supervisor; 0 otherwise. The characteristics weight was set to the value of 0.4.

When we calculated the similarity of courses by the aforementioned procedure, we could also utilize average link clustering [8]. 340 from all courses (499) belong to one of the 105 created clusters. 93 investigated courses were presented in one of the clusters. The number of courses in one cluster ranged from 2 to 22. The average number of courses in a cluster was 3. The average number of shared courses taken by students was 2.

### 4.3.3 Comparison of approaches

In comparison with the method using *all grades*, both approaches had positive effects on the number of calculations. 123 courses (from all 138) belonged to some of the created clusters and the final grades could be predicted based on the grades of only 3 other courses on average. 70 of our investigated courses belonged to different clusters using  $SC_1$  and  $SC_2$ . A slightly better MAE was obtained by the method utilizing the course characteristics for these courses (see Table 7). Therefore, when a grade is predicted, the corresponding course is searched in  $SC_2$ , then  $SC_1$ .

Table 7. Comparison of  $SC_1$  and  $SC_2$

Method	MAE	Sensitivity	Average cluster size	Shared Courses
All grades	0.687	0.402	499	10
$SC_1$	0.681	0.390	3	3
$SC_2$	0.640	0.386	3	2

## 4.4 Conclusion

H2 and H3 were confirmed. We described the novel approach for predicting the students performance (see Table 8) using only students' grades and course characteristics. It proved to be as successful as the first described approach (see Table 9). The most important contribution of this approach was that each university information system stores the data about students' grades which were needed for the prediction unlike the data about students' social behavior. We also identified course dependencies that lowered the calculation cost. Moreover, we were able to predict the final grade considering grades from only 3 other courses for the most of the investigated courses.

Table 8. Global results of the approach

Data Set	MAE	Sensitivity
Training Set	0.661	0.470
Test Set	0.685	0.418

## 5. USAGE OF THE APPROACHES

Both approaches defined in Section 3 (based on students' attributes (SBA)) and Section 4 (based on students' grades (SBG)) reached similar average results (see Table 9). However, they can differ in specific situations. Our goal was to identify course groups for which we could get trustworthy predictions and also to detect when one approach outperforms the other.

Table 9. Comparison of the both approaches

Data Set	Approach	MAE	Sensitivity
Training Set	SBA	0.629	0.528
	SBG	0.661	0.470
Test Set	SBA	0.688	0.427
	SBG	0.685	0.418

H4. Each approach is more suitable for different course groups.

We selected the following categories based on the basic course characteristics:

- difficulty – the average grade of all students' grades is 2.4. Therefore, we divided courses into two categories: easy ( $\leq 2.4$ ), and difficult ( $> 2.4$ ),

- occupancy rate – as defined in Section 4.1: small ( $\leq 30$ ), medium (30 – 70), and large ( $\geq 70$ ),
- specialization – courses divided into four groups: mathematics (M), theoretic informatics (I), applied informatics (P), and others (O).

Each investigated course belonged to one of the groups for each of the defined categories. With respect to the three aforementioned categories, we could define six (3!) tree structures which differ in the splitting order of the categories. We examined each permutation of the categories. We built full trees where courses from the training set were split subsequently by all categories. Each node stored the information about courses that belonged to it with respect to the split. Harmonic mean (HM) was calculated for each node and both approaches in order to get a suitable relationship between the sensitivity and the MAE score.

Subsequently, we examined the trees and merged branches which were not interesting in order to detect significant phenomena. Interesting branches contained at least one of the following situations:

- Difference  $> 0.1$  in HM of SBA and SBG in the node (The rule detected a significant difference in the prediction accuracy of the both approaches for the examined groups of courses.).
- Difference  $> 0.1$  in HM of the sibling nodes (The rule detected course groups that were significantly easily or with difficulties predicted than other courses from this split.).
- Difference  $> 0.1$  in HM of parent and child nodes (The rule detected the course groups that should be separated due to the significant difference in the prediction in comparison with the rest courses from the parent node.).

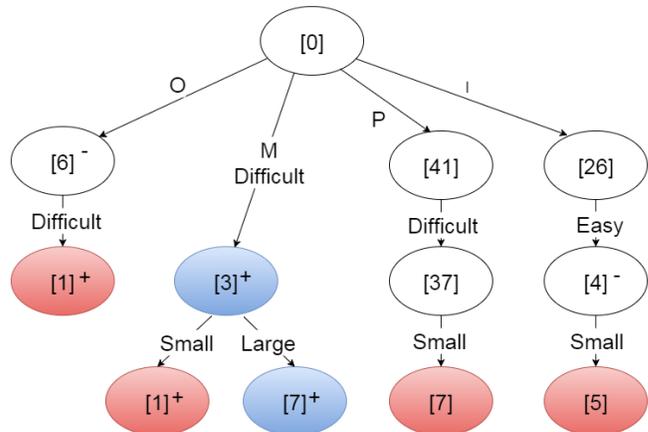


Figure 3. Resulted Tree

One of the resulted trees can be seen in Figure 3. As the figure shows, this approach had several benefits:

- Course groups that were predicted significantly better than average were identified (marked by +). It contains all mathematical courses (the main skill at the faculty of informatics can be easily predicted) and the English course.
- Course groups that were predicted significantly worse than average were identified (marked by -). It contained almost all courses belonged to the category *others* (we do not know students' general knowledge) and medium or large easy theoretic informatics courses (the grade maybe depended on the amount of the effort which could differ for each course and cannot be predicted).

- H4 was confirmed. Course groups that were predicted significantly better by the SBG approach are represented by the blue color. It covered almost all mathematics courses (except one small course). Otherwise, red nodes present better results obtained by the SGA approach. It contained the most of small courses. For the white nodes, the difference in prediction accuracy was negligible.
- Outliers were also identified. One course of the group showed different behavior than others: the course of English (path: O-difficult) was easily predictable in comparison with all courses belonged to the category *others*; one small mathematical course (M-difficult-small) differed in the approach that achieved better results in comparison with all other mathematical courses.

We applied this knowledge for prediction of the students' performance when the test set was utilized. We can easily locate any particular course in the tree and used the suitable approach that led to the better results. We also gave no predictions for courses that we were not able to predict reliably. As the results in Table 10 show, MAE was significantly improved in comparison with the state of the art method utilizing only SVM. Finally, we were able to predict the final grades with an error of one degree in the grade scale. We were also able to reveal almost a half of the unsuccessful students.

**Table 10. Final results validated on the Test set**

Approach	MAE	Sensitivity	Omitted Courses
Novel	0.609	0.436	10
SVM	0.744	0.414	0

## 6. CONCLUSION

In this paper, we focused on the problem of predicting final grades of students at the beginning of the semester with the emphasis on identifying unsuccessful students. Two different approaches were presented. Firstly, we utilized widely used classification and regression algorithms. SVM reached the best results. We also proved that data about social behavior of students improve the predictions for a quarter of courses. This approach can be beneficially utilized for the grade prediction of courses with a small number of students.

The second novel approach utilized collaborative filtering techniques and predicted grades based on the similarity of students' achievements. The advantage of this approach was that each university information system stores the data about students' grades which were needed for the prediction unlike the data about students' social behavior. We also succeeded in identifying course dependencies. Finally, we were able to predict the final grades of the investigated course by examining grades of only 3 other courses. The approach can be beneficially used for the grade prediction of mathematical courses.

We also identified groups of courses that are hardly predictable: courses with a different specialization than usual at the Faculty of Informatics, and also large informatics courses which are easy to pass. Finally, we were able to predict the final grade with the error of only one degree in the grade scale for the rest of courses. Half of students' failures were also correctly identified even if the task was difficult due to the fact that all unsuccessful grades constitute less than a quarter of all grades.

## 7. REFERENCES

- [1] Bydžovská, H. and Popelínský, L. 2014. The Influence of Social Data on Student Success Prediction. *Proceedings of the 18th International Database Engineering & Applications Symposium*, pp.374-375.
- [2] Bydžovská, H. 2015. Are Collaborative Filtering Methods Suitable for Student Performance Prediction? *Progress in Artificial Intelligence - 17th Portuguese Conference on Artificial Intelligence*, pp. 425-430.
- [3] Carrington, P., Scott, J. and Wasserman, S. 2005. Models and methods in social network analysis. Structural analysis in the social sciences. Cambridge University Press.
- [4] Koprinska, I., Stretton, J., and Yacef, K. 2015. Students at Risk: Detection and Remediation. *The 8th International Conference on Educational Data Mining (EDM 2015)*, pp. 512 – 515.
- [5] Manouselis, N. and Drachsler, H. and Vuorikari, R. and Hummel, H. and Koper, R. 2011. Recommender Systems in Technology Enhanced Learning, Recommender systems Handbook Springer Verlag 2011, pp 387-415.
- [6] Marquez-Vera, C. Romero, C. and S. Ventura. 2011. Predicting school failure using data mining. *In Proceedings of the 4th International Conference on Educational Data Mining (EDM'11)*, pp. 271-276.
- [7] Matuszyk, P., and Spiliopoulou, M. 2014. Hoeffding-CF: Neighbourhood-Based Recommendations on Reliably Similar Users. *In User Modeling, Adaptation, and Personalization*, volume 8538 of Lecture Notes in Computer Science, Springer International Publishing.
- [8] Murtagh, F. and Contreras, P. 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.
- [9] Nghe, T. N., Janecek, P. and Haddawy, P. 2007. A comparative analysis of techniques for predicting academic performance. *37th ASEE/IEEE Frontiers in Education Conference*, Milwaukee, WI 2007.
- [10] Nižnan, J., Pelánek, R., and Řihák, J. 2015. Student Models for Prior Knowledge Estimation. *The 8th International Conference on Educational Data Mining (EDM 2015)*, pp. 109-115.
- [11] Nooy, W., Mrvar, A. and Batagelj V. 2011. Exploratory Social Network Analysis with Pajek. Structural Analysis in the Social Sciences. Cambridge University Press.
- [12] Strecht, P., Cruz, L., Soares, C., Mendes-Moreira, J., and Abreu, R. 2015. A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance. *The 8th International Conference on Educational Data Mining (EDM 2015)*, pp. 392-395.
- [13] Vandamme, J.P., N. Meskens and J.F. Superby, 2007. Predicting academic performance by data mining methods. *Educ. Econ.*, 15, 405-419.
- [14] Ventura, S., Romero, C., López, M.-I., and Luna, J.-M. 2013. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education* 68, 458-472.

# Course Enrollment Recommender System

Hana Bydžovská  
CSU and KD Lab Faculty of Informatics  
Masaryk University, Brno  
bydžovska@fi.muni.cz

## ABSTRACT

One of the main problems faced by university students is to create and manage the semester course plan. In this paper, we present a course enrollment recommender system based on data mining techniques. The system mainly helps with students' enrollment decisions. More specifically, it provides recommendation of selective and optional courses with respect to students' skills, knowledge, interests and free time slots in their timetables. The system also warns students against difficult courses and reminds them mandatory study duties. We evaluate the usability of designed methods by analyzing real-world data obtained from the Information System of Masaryk University.

## Keywords

Course enrollment recommender system, student performance, prerequisites, university information system.

## 1. INTRODUCTION

Recommender systems can be used in different fields including educational environment. Such systems are mainly focused on providing high educational standard and try to enhance the process of teaching and learning [13]. They help with searching for suitable web resources [8], recommend good solutions to improve students' knowledge [4], or analyze data obtained from quizzes and provide a feedback to instructor to modify a quiz [9].

Nowadays, researchers also try to improve personalized searching for beneficial courses. The aim of several projects was to select courses in order to obtain good exam results [12] or recommend elective course modules based on previous students' enrollments using collaborative filtering techniques [6]. Other option is to utilize association rules [1] or ant colony optimization [11].

In the last few years, recommendations became more complex. Besides selecting passable courses, it is essential to recommend beneficial courses [3]. The suitability of courses was determined by the importance in all fields of the university, the ratio of connectivity among courses and by the importance in the student's field of study. Association rules were utilized for searching relationships between courses. Another approach was presented in [7]. To graduate, all defined blocks of courses must be completed by finishing a pre-defined number of courses. They utilized a flow algorithm to find the minimal set of courses that students have to pass.

In this paper, we present a pilot version of the course enrollment recommender system designed at the Faculty of Informatics Masaryk University. All methods were validated on data originated from the Information System of Masaryk University (IS MU). The data contain information on courses, templates defining the mandatory and selective courses, students, study-related attributes, and social behavior data. The designed methods predict students' final grades and recommend them interesting courses with respect to their skills, interests, and free time-slots in the timetable.

## 2. COURSE ENROLLMENT RECOMMENDER SYSTEM

### 2.1 Motivation

All students have to follow the obligations and principles stated by their university. Especially at the beginning of the study, it is hard for students to cover all the mandatory duties. At Masaryk University, all semesters are preceded by a course enrollment process. All active students have to enroll a sufficient number of courses to achieve at least the minimal pre-defined amount of credits. If they do not reach the minimum limit, they cannot proceed to the next semester. Students have to pass many courses before finishing their studies successfully. All mandatory courses must be completed. Students have to also pass several selective and optional courses. Analyzing the enrollment statistics, we found out that students prefer interesting and passable courses. Universities usually offer a large number of courses and it is difficult for students to be familiarized with all of them. They are forced to search through the entire course catalog, read many abstracts and syllabi, and compare a large amount of success rate statistics. Naturally, they often discuss courses with other students who have their own personal experiences. Obviously, the decisions they have made during the course enrollment process could significantly influence the whole study progress and the final result.

### 2.2 System Overview

The current version of the recommender system monitors the number of credits of enrolled courses to ensure successful progression to the next semester. It also reminds them to enroll all mandatory courses. Selective and optional courses are recommended according to the student's performance and interests with respect to free time slots in students' timetables. The system clarifies the decisions to students using notifications. The system also warns against enrolled courses that usually cause problems to students with similar characteristics. If the system identifies a difficult course in the student's enrollment, it informs the student about the potential issue. It allows students to focus more on this course or to revise the enrollment decision. Students can also assess each recommendation whether the recommended courses were interesting and adequately difficult. Based on these assessments, the recommendation algorithms will be modified in order to enhance the relevance of the further recommendations.

## 3. COURSE TEMPLATES

At our university, templates represent tree-like definitions of mandatory and selective courses for each field of study. The system allows checking the requirements that a student has already accomplished. The completed courses/nodes are marked with a green ring (o) and the uncompleted courses/nodes are marked with a red cross (x).

We examined 67 templates defining the study requirements for active students in the years of 2010-2013 at Faculty of Informatics. An example of a template can be seen in Figure 1.

- x **B-IN Parallel and Distributed Systems** – into all 2 (total: 52 credit(s), 10 course(s))
- x **Mandatory Courses** – into all 12 (total: 30 credit(s), 6 course(s))
  - x **SBAPR** □ Bachelor Thesis z
  - o **IA039** □ Supercomputer Architecture and Intensive Computations 1 zk (4 credit(s))
  - o **IB000** □ Induction and Recursion 1 zk (4 credit(s))
  - o **IB002** □ Design of Algorithms I zk (4 credit(s))
  - o **IB005** □ Formal Languages and Automata 1 zk (6 credit(s))
  - x **IB109** □ Design and Implementation of Parallel Systems
  - x **IV010** □ Communication and Parallelism
  - x **IV100** □ Parallel and distributed computations
  - x **IV112** □ Project on programming parallel applications
  - x **IV113** □ Introduction to Validation and Verification
  - o **MB000** □ Calculus I zk (6 credit(s))
  - o **MB001** □ Calculus II zk (6 credit(s))
- x **Selective Courses** – at least 4 from 11 (total: 22 credit(s))
  - x **IA040** □ Modal and Temporal Logics for Processes
  - x **IA058** □ Computing and Communication Networks and Their Applications
  - x **IV109** □ Modeling and Simulation
  - x **PA150** □ Advanced Operating Systems Concepts
  - x **PA151** □ Advanced Computer Networks
  - x **PA159** □ Net-Centric Computing I
  - x **PA165** □ Enterprise Applications in Java
  - o **PV017** □ Information Technology Security zk (4 credit(s))
  - x **PV065** □ UNIX – Programming and System Management I
  - o at least 1 z (credit) 2 (total: 12 credit(s), 2 course(s))
    - o **PB161** □ C++ Programming zk (6 credit(s))
    - o **PB162** □ Java zk (6 credit(s))
  - o at least 1 z (credit) 3 (total: 6 credit(s))
    - x **IV054** □ Coding, Cryptography and Cryptographic Protocols
    - x **IV111** □ Probability in Computer Science
    - o **MV011** □ Statistics I zk (6 credit(s))

Figure 1. Template of mandatory and selective courses

However, the structure of the templates is often more complicated. Each node defines how many child nodes have to be completed (all, defined by the number of credits, or defined by the number of children). The template does not enforce in which semester courses should be enrolled.

### 3.1 Which courses do students have to pass before enrolling a certain course?

Some courses have prerequisites that define what a student must meet before he or she can enroll in a certain course. At our university, prerequisites are composed of terms  $p_1 \dots p_n$  that are associated with logical operators AND(&&), OR(||). A term  $p_i$  can be a course or a compound term. Prerequisites can be transformed into the template subtree by the following rules:

- $p_i \ \&\& \ p_j \rightarrow$  new node containing  $p_i$  and  $p_j$  with the rule of fulfillment: all nodes
- $p_i \ || \ p_j \rightarrow$  new node containing  $p_i$  and  $p_j$  with the rule of fulfillment: at least one of nodes

- x **PA211** □ Advanced Topics of Cyber Security - into all 3
  - o **PV210** □ Cyber security in an organization
  - x at least 1
    - x **PA159** □ Net-Centric Computing I
    - x **PA191** □ Advanced Computer Networking
  - o **PV065** □ UNIX – Programming and System Management I

Figure 2. PA211 prerequisites: PV210 && (PA159 || PA191) && PV065

Example of such transformation can be seen in Figure 2. Each template could be extended by prerequisites courses for students to be able to count on them when creating their study plans.

### 3.2 When do students have to enroll a certain course?

Students can decide in which semester they enroll in a certain course. All graduate students that completed the template requirements were selected and the semester in which the most of them enrolled in the particular mandatory course was calculated

by Algorithm 1. Therefore, we remind courses in the proper semesters with respect to students' completed semesters.

#### Algorithm 1. Semester Selection

**Function** select\_semester(course, template):

```
sem_max = {sem ∈ semesters | ¬∃sem2: number_students (sem2,
course, template) > number_students (sem, course, template)}
if (|sem_max| == 1) then
    return sem_max[0];
else if (|sem_max| > 1) then
    return min(sem_max);
else
    return 1;
end if;
```

**Function** number\_students(semester, course, template):

return the number of students having completed the given template enrolled in the given course in the specific semester;

### 3.3 Which courses are passable for a certain student?

We focused on the problem of predicting the final grade at the beginning of the semester with the emphasis on identifying unsuccessful students. The following grade scale was used: 1 (excellent), 1.5 (very good), 2 (good), 2.5 (satisfactory), 3 (sufficient), 4 (failed or waived). The value 4 represents students' failure; the others represent a full completion.

We present two different approaches in [2]. Both approaches are validated on 138 courses which were offered to students of the Faculty of Informatics of Masaryk University between the years of 2010 and 2013. The first approach is based on classification and regression algorithms that search for patterns in study-related data and also data about students' social behavior. We prove that students' social behavior characteristics improve prediction for a quarter of courses. The second approach is based on collaborative filtering techniques. We predict the final grades based on previous achievements of similar students. We also present the novel approach how to find out which approach is better for which courses. Finally, we are able to correctly identify half of all failures (that constitute less than a quarter of all grades) and predict the final grades only with the error slightly higher than one degree in the grade scale.

Due to the prediction error, we decided to lower the granularity of predictions to the following three classes: excellent (1, 1.5), good (2, 2.5), and bad (3, 4) to prevent the recommendation of difficult courses. As it can be seen in Table 1, the mean absolute error was below 0.5 and due to the high value of sensitivity the most of unsuccessful students were revealed.

The approaches are beneficially utilized in the presented course enrollment recommender system to warn students against difficult courses and to recommend only passable optional courses. Courses with predicted grade better than bad grade are considered as passable for a student.

Table 1. Prediction Evaluation on Test set

Task	MAE	Sensitivity
Grade prediction	0.609	0.436
Excellent / good / bad prediction	0.474	0.899

## 4. SELECTIVE COURSES

Students can select different sets of selective courses from the template with respect to their skills and the course content. They have to select enough courses to fulfill the node requirements. We were interested in the student behavior, e.g. information about the most preferred courses.

### 4.1 Designed Recommendation Methods

We defined a course  $c$  for a student  $a$  as interesting by the following function:

$$f(a, c) \begin{cases} 1 & \text{if the student } a \text{ attended course } c \text{ or marked it as} \\ & \text{favorite} \\ 0 & \text{otherwise} \end{cases}$$

This characteristic defined the student's interest in the course. Therefore, each student can be characterized by a set of his or her interesting courses. We designed the following 4 algorithms to recommend courses:

**S1. The most selected courses by students with the same template.** We were interested in the student behavior, e.g. information about the most preferred courses. We computed the most frequent path of graduate students in the template. We were inspired by a simple ant colony algorithm and marked each node with the number of students that passed through. The path was computed by universal path finding Algorithm 2.

**S2. Courses enrolled by similar students.** We calculated the similarity between sets of interesting courses for each student and all graduate students that already completed the template. We utilized Jaccard's coefficient. For each student, we selected the most similar students and recommended their courses. We were searching for the proper size of the neighborhood and evaluated  $n \in [1; 25]$ . When we sorted the courses in the list by their frequency of occurrence in similar student's lists, we also explored how many of them were suitable to be recommended. We examined  $x \in [1; 10]$ .

**S3. Courses taught by favorite teacher.** Students' interesting courses were examined and favorite teachers were revealed. We considered all course lecturers and only student's tutors. The teacher's popularity was defined as the sum of all his or her courses which were considered as interesting. Considering the teacher's popularity, we recommended other teacher's courses if his or her popularity was above the threshold (2).

**S4. Courses enrolled by friends.** We examined students' social behavior characteristics and their mutual cooperation. We focused on statistical data that represented the interaction among students: explicitly expressed friendship, posts and comments in discussion forums, e-mails statistics, publication co-authoring, or files sharing. This information served as the basis for computing social ties among students by means of a sociogram [2]. From this sociogram, we were able to reveal friends ties among students. We recommended courses that friends considered as interesting and belonged to the template.

The algorithms also observed the following rules:

- Courses recommended for a particular student were limited to courses that should be enrolled in the certain semester:  

$$\text{course's semester} \leq \text{student's semester}$$
*Student's semester* was defined as the number of commenced semesters and the course's semester was defined as the

semester in which other students usually enrolled in the course calculated by Algorithm 1.

- We also did not recommend courses that belonged to the subtree of the template which students had already completed.
- Only courses that could be enrolled in the actual semester were recommended.

---

#### Algorithm 2. Finding Path in Template

---

```

Function process_node (node, template, student):
children ← children of the node;
for each child in children do
    unless (child_computed) then
        process_node(child, template, student);
    end if;
end for;
path; # calculated path
sort children in descending order by the value in the node;
for each child in children do
    path ← child;
    if (node_fulfilled(node, student)) then
        return path;
    end if;
end for;

Function node_fulfilled (node, student):
if (the given node is fulfilled by the given student) then
    return true;
else
    return false;

```

---

### 4.2 Recommendation Methods Evaluation

We can assume that students are familiar with the offer of selective courses. Therefore, offline experiments [10] can be suitable approach to evaluate previously mentioned algorithms. All students that enrolled in the semester autumn 2014 and did not complete their templates were selected: 1,444 students in total.

#### 4.2.1 Settings for the algorithm S2

Firstly, we had to evaluate suitable settings for the algorithm S2. Our task was to select suitable courses for students and subsequently detect if they enrolled in them or not. Therefore, the suitable evaluation metrics were precision and recall. To find a balance between precision and recall, the F1 score was also calculated.

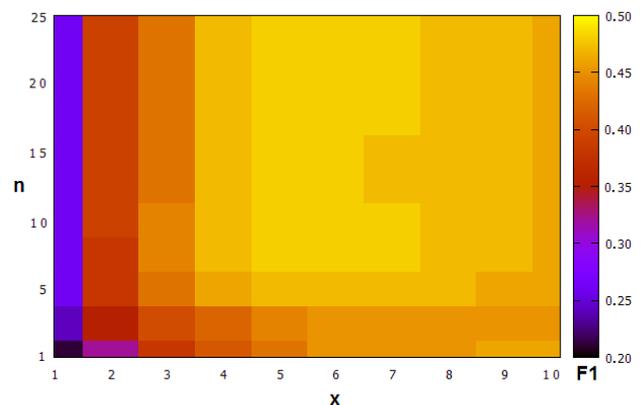


Figure 3. Relationship among the size of the neighborhood  $n$ , number of selected courses  $x$  and the value of F1 score

We selected 90% of examined students and calculated the F1 score of the recommendations. Figure 3 shows the relationship among variables  $n$ ,  $x$ , and the value of F1. Based on these findings, the following setting was selected for algorithm S2 as the most suitable:  $n = 8$ ,  $x = 5$ . This conclusion was also verified on the test set (the rest 10% of students).

#### 4.2.2 All algorithms' evaluation

We utilized all previously described algorithms to recommend courses for each student. The coverage determines the percentage of students for whom we were able to recommend at least one course.

**Table 2. Results of selective courses recommendation**

Algorithm	S1	S2	S3	S4
Coverage	0.97	0.63	0.60	0.54
Offered courses	2.97	4.81	3.85	4.43
Enrolled courses in the semester autumn 2014	1.63	2.08	1.81	1.88
Enrolled courses anytime	2.82	3.15	2.49	2.85
Precision	0.81	0.56	0.48	0.47
Recall	0.55	0.42	0.28	0.39
F1	0.66	0.48	0.35	0.43
<b>Rank</b>	<b>1</b>	<b>2</b>	<b>4</b>	<b>3</b>

The coverage of approaches differs as it can be seen in Table 2. S1 covered almost all students. In contrary, the rest of approaches recommended courses for only 60% of selected students. The average number of courses offered by each algorithm can be seen in the second row. Algorithms recommended 3-5 courses on average. The average number of courses that students really enrolled in autumn 2014 can be seen in the third row. Because the university does not define when students have to enroll courses, we extend the searching for enrollment also to the next semesters. The average number of courses that students really enrolled anytime from autumn 2014 till now can be seen in the fourth row. As it can be seen, the number of enrolled courses almost doubled in all cases. Finally, we also calculated precision and recall for all algorithms. The algorithm S1 reached the best results.

#### 4.2.3 Which courses are selected the most often?

H1: We supposed that students select easier selective courses.

For finding the easiest way to complete the template, we assessed each course using its success rate (the percentage of successful students to all students in the course). However, we had to penalize courses with a small number of students and also the courses with smart students only (with excellent average grade). Therefore, the adjusted success rate (ASR) was defined as:

$$ASR = CSR \cdot \log_4 ESAG \cdot \frac{NES}{MAX\_ENR}$$

where CSR defined the course success rate, ESAG defined the average grade of enrolled students, NES defined the number of enrolled students in a course, and MAX\_ENR was a constant for the template and defined the maximum number of students enrolled in any course from the template. We calculated the minimal adjusted success rate of courses that have to be passed in the subtree for each node of the template. Subsequently, we employed the Algorithm 2 that selected the easiest courses till the node requirements were met.

For each template  $t \in T$  we constructed the easiest path (EP) and also the most frequented path (MFP). Both paths can be represented as a set of selected courses on the path. Jaccards' coefficient (JC) was calculated to compare these sets of courses. The similarity of paths was 0.8 on average for all templates.

$$\frac{\sum_{t \in T} JC(EP, MFP)}{|T|} = 0.8$$

H1 was confirmed. Correlation of EP and MFP over all templates confirmed our hypothesis that students usually select easier selective courses.

## 5. OPTIONAL COURSES

To fulfill all study requirements, students have to obtain the pre-defined number of credits in their studies. Except credits obtained from mandatory and selective courses, they have to select optional courses. Optional courses for each student were defined as courses that do not belong to the student's template.

### 5.1 Designed Recommendation Methods

We utilized the same methodology as described in Section 4 for recommendation of selective courses. The main difference was that algorithms did not restrict courses from templates. The courses recommended by algorithms were limited to only passable courses (the predicted grade was not bad) according to the method introduced in Section 3.3.

- S1. The most selected courses by students with the same field of study.** All optional courses of all students of a certain field of study were selected. The number of students that were interested in each course was calculated and the sorted list of all courses based on the calculated value was created from the most interesting.
- S2. Courses enrolled by similar students.** We computed the student similarity with all active students and also students graduated in the last five years. The revealed courses were sorted into a list by the number of occurrences in similar students' sets of optional courses.
- S3. Courses taught by favorite teacher.** Courses were sorted into a list in decreasing order by the popularity of a teacher.
- S4. Courses enrolled by friends.** Courses were sorted into a list by the number of occurrences in friends' sets of optional courses.

### 5.2 Recommendation Methods Evaluation

As a contrary to the selective course recommendation, we supposed that students are not familiarized with all the optional courses. Therefore, the offline experiments were not sufficient evaluation technique in this case and we had to conduct a user study [10]. We contacted only selected group of students to request them to assess our recommendations.

We could approach 607 students enrolled in one of our courses in the last semester. Considering the number of students and expecting the lower response rate of students, we selected 5 top rated courses by each algorithm for each student. The coverage of approaches when the algorithm found at least one course to offer is presented in Table 3 in the first row. Only for a half of students, we revealed friends who could inspire students with interesting courses. The average number of offered courses by each algorithm can be seen in the second row. The approach which uses social ties (S4) offered only 4 courses on average.

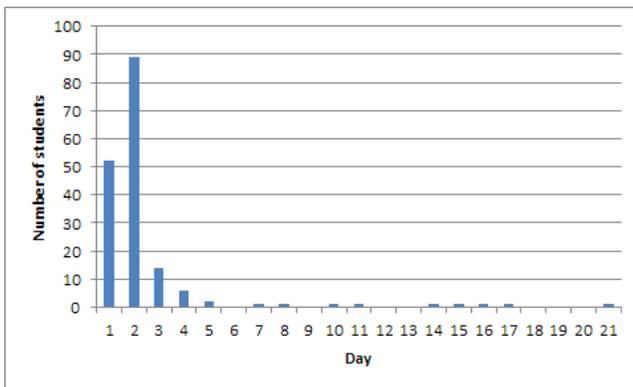
In our experiment, we offered 10 courses at maximum selected using the 2 our algorithms  $S_i$  and  $S_j$  for each student. We sorted the students in the list by their average grade in order to be

independent of students' characteristics and nearly randomly selected 2 algorithms that offered its top 5 courses each at maximum to students. We balanced the number of occurrence of each algorithm due to the low coverage of S4. We also merged the list of courses of  $S_i$  and  $S_j$  in order to not prioritize one of them in the following order:  $S_{11}, S_{j1}, S_{12}, S_{j2}, S_{13}, S_{j3}, S_{14}, S_{j4}, S_{15},$  and  $S_{j5}$ . When both algorithms selected the same course, the course appeared only once in the list. The assessment of the course was added to results for both algorithms.

**Table 3. Algorithms coverage**

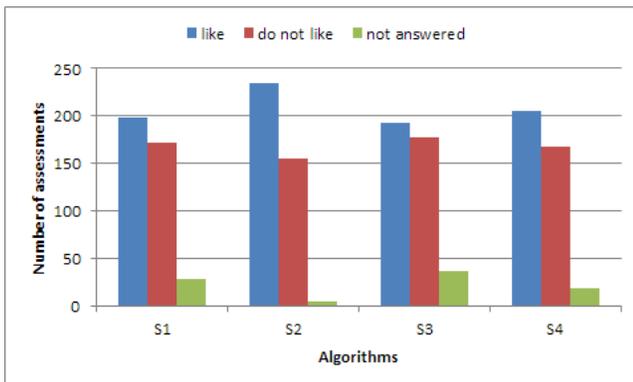
Algorithm	S1	S2	S3	S4
Coverage	1	1	0.96	0.49
Offered Courses	4.98	4.98	4.47	4.02

Subsequently, students were asked for assessing the recommendation during their course enrollment process to increase the possibility of their reaction. Students could mark courses using the following attributes: like, do not like or leave it unanswered. Overall, 172 students responded. The most of them responded in one week since the invitation (see Figure 4).



**Figure 4. Students' reaction period**

The distribution of students' reactions is shown in Figure 5. The best recommendation was offered by the algorithm S2. The algorithm is based on the similarity of students' sets of interesting courses.



**Figure 5. Assessed courses**

The number of students assessed (NSA) our algorithms was almost in balance. Each student was included twice: for each of algorithms that assessed. As it can be seen in Table 4, we obtained more assessments of courses inspired by friends' selections (S4).

It can mean that students with more social ties in the system are more active. We omitted recommendations that were not assessed. For all algorithms we obtained enough assessments to be able to properly evaluate them. We utilized the same evaluation metrics as for selective courses besides recall because we could not compute false negatives. On average for all algorithms, students liked 2-3 of 4-5 offered courses.

**Table 4. Algorithms evaluation**

Algorithm	S1	S2	S3	S4
NSA	79	79	82	99
Liked Courses	2.52	2.97	2.35	2.07
Offered Courses	5	5	4.8	3.9
Precision	0.53	0.60	0.52	0.55
<b>Rank</b>	<b>3</b>	<b>1</b>	<b>4</b>	<b>2</b>

Considering all evaluation methods, we determined the ranking of algorithms' success rate. Algorithm based on similarity of interesting courses (S2) reached the best results. However, the final solution will combine all algorithms to achieve best results.

## 6. RECOMMENDATIONS

We have designed new elements for Registration Application which might be available to all students of Masaryk University in the future. The first enhancement presents the predicted difficulty of courses to students. The predictions are computed by the method described in Section 3.3. The predicted grades correspond to the following color:

- Excellent grade – green color.
- Good grade – yellow color.
- Bad grade – red color.

All predictions are presented as the icons of corresponding color. When we have no predictions, there is no icon. We try to predict grades of courses that students enrolled or courses that we recommend to them (see Figure 6). Based on these warnings, students can concentrate on difficult courses or revise their choices depending on the planned workload in the semester.

The second improvement is the panel on the right (see Figure 6) where the recommended courses are presented. For each student we remind mandatory courses, recommend selective and optional courses selected by methods introduced in Sections 4 and 5, and also recommend their prerequisite courses. After clicking the wrench icon, the short explanation of each recommendation is displayed to increase students' trust to the system [5]. They can also assess each recommendation. Based on assessments we continuously improve our algorithms.

## 7. CONCLUSION

We presented a pilot version of course enrollment recommender system that reminds students their duties, warns them against difficult courses and recommends them potentially beneficial courses. Therefore, the system helps students with their decisions during the enrollment process at the beginning of each semester.

More specifically, we designed four algorithms suitable for the course recommendation. The first algorithm searches for the most frequently enrolled courses. The second algorithm utilizes similarities of students based on courses of their interests. The third algorithm recommends courses of students' favorite teachers. The last algorithm calculates the social ties among

students and selected courses which were interested for students' friends.

The most suitable algorithm for the selective course recommendation was the first described algorithm. Students usually selected easier courses defined in their templates. In contrary, the best results for the optional courses recommendation achieved the second algorithm utilizing students' similarities. However, we decided to employ all methods in the system due to the high students' satisfaction with recommendations. Optional courses were also recommended only if we predicted that students could pass the course and they had free time slots in the timetable for the course. We validated all designed methods on data originated from students of the Faculty of Informatics Masaryk University stored in the university information system.

We also introduced the environment that presents recommendations to students, offers them the explanations why the courses were selected, allows them to leave a feedback, warns them against difficult courses, and reminds them important events that should be accomplished, e.g. enroll in mandatory courses or enroll enough credits. The designed course enrollment recommender system will be a part of the university information system in the future.

## 8. REFERENCES

- [1] Bendakir, N. and Aimeur, E. 2006. Using Association Rules for Course Recommendation. In *Proceedings of the AAAI Workshop on Educational Data Mining*, pp. 31-40.
- [2] Bydžovská, H. 2016. A Comparative Analysis of Techniques for Predicting Student Performance. *Proceedings of the 9th International Conference on Educational Data Mining 2016* (Accepted).
- [3] Lee, J. Ch. Y. and Lee, K.-W. 2011. An intelligent course recommendation system. *Smart Computing Review*, 1(1).
- [4] Loll, F. and Pinkwart, N. 2009. Using collaborative filtering algorithms as elearning tools. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*.
- [5] O'Donovan, J. and Smyth, B. 2005. Trust in Recommender Systems. *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pp. 167-174.
- [6] O'Mahony, M. P., and Smyth, B. 2007. A recommender system for on-line course enrolment: an initial study. In *Proceedings of the ACM conference on Recommender systems (RecSys '07)*, pp. 133-136.
- [7] Parameswaran, A., Venetis, P., and Garcia-Molina, H. 2011. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Trans. Inf. Syst.*, 29(4):20:1-20:33.
- [8] Recker, M. M., Walker, A., and Wiley, D. 2004. Collaborative information filtering: A review and an educational application. *International Journal of Artificial Intelligence in Education*, Volume 14 Issue 1, pp. 3-28.
- [9] Romero, C., Zafra, A., Luna, J. M., Ventura, S. 2013. Association rule mining using genetic programming to provide feedback to instructors from multiple-choice quiz data. *Expert Systems* 30(2): 162-172.
- [10] Shani, G. & Gunawardana, A. 2011. Evaluating recommendation systems. *Recommender Systems Handbook*, pp. 257-297.
- [11] Sobacki, J. and Tomczak, J. M. 2010. Student courses recommendation using ant colony optimization. In *Proceedings of the Second international conference on Intelligent information and database systems*: pp. 124-133.
- [12] Vialardi, C., Chue, J., Peche, J. P., Alvarado, G., Vinatea, B. Estrella, J., and Ortigosa, A. 2011. A data mining approach to guide students through the enrollment process based on academic performance. *User Modeling and User-Adapted Interaction*, Volume 21, Issue 1, pp. 217-248.
- [13] Vuorikari, R., Hummel, H., Manouselis, N., Drachler, H., and Koper, R. 2011. Recommender Systems in technology enhanced learning. In *Recommender systems Handbook*, pp. 387-415. Spriger Verlag.

The screenshot displays a user interface for course management. On the left, a table titled "Courses currently registered for or enrolled in:" lists various courses with their details and enrollment status. On the right, a sidebar titled "Recommended courses" is divided into "Selective courses" and "Optional courses", each listing suggested courses with their respective details and icons.

Course	Further information	Enrolled
<a href="#">FI:IA006</a> Selected topics on automata theory Thu 16:00-17:50 <a href="#">D1</a> Group: <a href="#">IA006/06</a> each odd Wednesday 12:00-13:50 <a href="#">B410</a>		yes zk 5 credit(s)
<a href="#">FI:IA067</a> Informatics Colloquium Tue 14:00-15:50 <a href="#">D2</a>		yes z 1 credit(s)
<a href="#">FI:MA007</a> Mathematical Logic Tue 16:00-17:50 <a href="#">D1</a> Group: <a href="#">MA007/04</a> each odd Wednesday 14:00-15:50 <a href="#">C525</a>		yes zk 5 credit(s)
<a href="#">FI:MA010</a> Graph Theory Thu 12:00-13:50 <a href="#">D1</a> Group: <a href="#">MA010/01</a> each odd Monday 12:00-13:50 <a href="#">B410</a>		yes zk 5 credit(s)
<a href="#">FI:PA017</a> Software Engineering II Thu 14:00-15:50 <a href="#">D3</a>		yes zk 4 credit(s)
<a href="#">FI:PA159</a> Net-Centric Computing I Tue 10:00-11:50 <a href="#">D3</a>		yes zk 4 credit(s)
<a href="#">FI:PV028</a> Applied Information Systems Fri 8:00-9:50 <a href="#">D2</a>		yes k 3 credit(s)
Total		27 credit(s) [k: 1; z: 1; zk: 5]

**Recommended courses**

**Selective courses**

- [FI:IV113](#) Validation and Verification  
Wed 16:00-17:50 [A218](#)

**Optional courses**

- [FI:PV254](#) Recommender Systems  
Wed 14:00-15:50 [C416](#)
- [FI:IV107](#) Bioinformatics I  
Wed 8:00-9:50 [C525](#)
- [FI:PB172](#) Systems Biology Seminar  
Fri 10:00-11:50 [A418](#)
- [FI:IA080](#) Knowledge Discovery  
Wed 8:00-9:50 [C513](#)
- [FI:PV211](#) Information Retrieval  
Wed 8:00-9:50 [D3](#)
- [FI:MV011](#) Statistics I  
Wed 10:00-11:50 [D1](#)

Figure 6. Demonstration of Interface

# Data-driven Automated Induction of Prerequisite Structure Graphs

Devendra Singh Chaplot  
School of Computer Science  
Carnegie Mellon University  
dchaplot@cs.cmu.edu

Yiming Yang  
School of Computer Science  
Carnegie Mellon University  
yiming@cs.cmu.edu

Jaime Carbonell  
School of Computer Science  
Carnegie Mellon University  
jgc@cs.cmu.edu

Kenneth R. Koedinger  
School of Computer Science  
Carnegie Mellon University  
koedinger@cmu.edu

## ABSTRACT

With the growing popularity of MOOCs and sharp trend of digitalizing education, there is a huge amount of free digital educational material on the web along with the activity logs of large number of participating students. However, this data is largely unstructured and there is hardly any information about the relationship between material from different sources. We propose a generic algorithm to use educational material and student activity data from heterogeneous sources to create a Prerequisite Structure Graph (PSG). A PSG is a directed acyclic graph, where the nodes are educational units and the edges specify the pairwise ordering of the units in effective teaching by instructors or for effective learning by students. We propose an unsupervised approach utilizing both text content and student data, which outperforms to supervised methods (utilizing only text content) on the task of estimating a PSG.

## 1. INTRODUCTION

Students need prior knowledge for thorough understanding of educational content. This need imparts an implicit order in learning educational concepts. Determining this order requires significant human time and effort. Furthermore, relying on expert knowledge to determine this order is subject to inconsistencies due to ‘expert blind spot’ [8]. We aim to leverage free educational material on the web, and huge amount of student activity logs associated with them, to create a universal Prerequisite Structure Graph (PSG). We define PSG as a directed acyclic graph, where the nodes are the universal concepts in an educational domain and the edges specify the pairwise ordering of concepts in effective teaching by instructors or for effective learning by students. The proposed unsupervised methods utilize both textual content and student performance data to perform better than supervised methods utilizing textual content. They can be

generalized to find the learning order between any pair of educational elements from heterogeneous resources, at any level of granularity (courses, units, modules, skills, etc.).

The rest of the paper is divided as follows. The related work pertaining to the proposed methods is discussed in Section 2. Section 3 describes the dataset used for experiments. Performance-based and text-based unsupervised induction of a PSG are described in Sections 4 and 5, respectively. We describe the method of combining text-based and performance-based approaches in Section 6. Experiments and results are presented in Section 7. In Section 8, we analyze whether the concepts extracted by proposed methods are meaningful. Conclusions and future directions are covered in Section 9.

## 2. RELATED WORK

Currently, the construction of Concept Graphs majorly depends on manual work of domain experts. Recent work by [10] on Concept-Graph-Learning (CGL), focuses on determining the relationship between different University courses and MOOCs by inferring concepts from course descriptions. The proposed methods are completely unsupervised as compared to supervised CGL which requires partial instructor-specified links. One other recent work includes extracting a concept-hierarchy from textbooks [9], where the focus is only on extracting the hierarchies between concepts and the learning is only done at the concept level. We differentiate ourselves from this work with the fact that we learn the prerequisite relationships between educational concepts rather than hierarchies, and our method is generalizable to any granularity of educational elements.

Another indicator of prerequisite links between educational elements is student performance. An early approach to inferring prerequisite graphs from student performance data is knowledge spaces [2], which uses associations between student success on different classes of tasks to infer prerequisite relationships. The essential idea is that if students are highly likely to get tasks of type A correct (e.g., finding least common multiples) conditioned on getting tasks of type B correct (e.g., adding fractions with unlike denominators) but not the other way around (i.e., many students that can find common multiples fail at adding fractions), then A is a pre-

requisite of B. Subsequently, algorithms for inferring cognitive models of student learning from data have been developed and it is possible to infer prerequisite relationships from the results of these models [1]. The methods we propose are different as we utilize not only the student performance data, but also student activity data along with large amounts of text in course material. Also, previous approaches assume that there is no learning between attempts at different problems, which is suitable for standardized testing scenario but not true for student performance logs of complete courses.

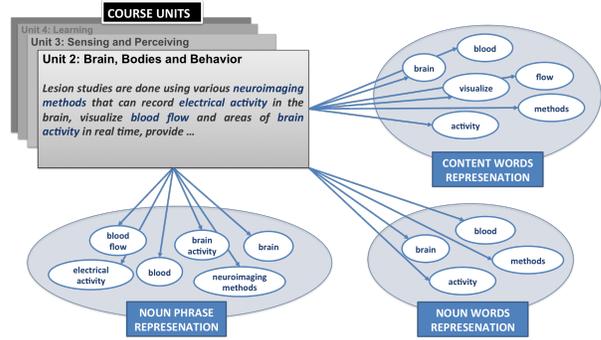
### 3. DATASET

We use the text content and student activity and quiz performance data from Georgia Tech’s “Introduction to Psychology” MOOC which uses content from the Open Learning Initiative of Carnegie Mellon University [6]. The course spans over 12 weeks and a major topic of Psychology (like intelligence, personality, psychological disorders, etc.) was covered in each week of class. For each week, the text content from the corresponding unit(s) was extracted. The unit(s) covered in each week are shown in Table 1. On an average, each unit contained 12545 word instances with a standard deviation of 3730. For simplicity, we will use Unit  $i$  to denote the content covered in Week  $i$ , although the content covered in week  $i$  might include multiple units in the course. Besides the text inside course units, we also used text in the weekly quizzes separately to evaluate our text-based methods.

The course also contained ungraded practice activities within each unit. At the end of each week (from week 1 to week 11), students were assessed by a high stakes quiz containing questions from content covered in the corresponding week. The dataset includes the number of interactive activities and quiz scores of 1154 students for each week.

This dataset is ideal for our analysis since it has both the textual data of course material and the student activity and quiz performance data. We aim to predict prerequisite links between weeks using this data, which will imply prerequisite links between corresponding units. For example, a prerequisite link from Unit 9 to Unit 11 implies prerequisite link from Personality to Disorders, or in other words, a student who has learned Personality will be better able to learn Disorders.

For evaluation, the dataset was first annotated by three non-experts who determined whether a prerequisite link exists between content covered in any two units. If a prerequisite link exists from Unit  $i$  to Unit  $j$ , we call it a positive link, and conversely, if there is no link, it is called a negative link. The average percentage agreement for positive links between each pair of annotator was 29.6% while the percentage agreement for positive links among all the annotators was 18.7%. Since the inter-annotator agreement was very low, we got the dataset annotated by a domain expert. All the links marked positive by all three non-expert annotators were also marked positive by the domain expert, except one link. Finally, we took 15 links marked positive and domain expert, and 1 more link marked positive by all non-expert annotators as the set of positive links. Therefore, among 110 possible links, 16 links were labeled as positive and rest negative. Note that 55 out of 110 possible links are backward (i.e. from Unit  $i$  to Unit  $j$  such that  $i > j$ ), which



**Figure 1:** Example of three types of concept space representation schemes: Content Words, Noun Words and Noun Phrases.

should be implicitly negative, but we will not use the information about the ordering of units in any of the proposed methods so that our methods are generalizable to any pair of educational elements: modules, chapters or whole courses.

Week	Unit(s) Covered
1	Introduction and Methods
2	Brains, Bodies, and Behavior
3	Sensing & Perceiving
4	Learning
5	Memory
6	Language and Intelligence
7	Lifespan development
8	Emotion and Motivation
9	Personality
10	Psychology in Our Social Lives
11	Disorders

**Table 1:** Unit(s) covered in each week of “Introduction to Psychology” course.

### 4. TEXT-BASED METHODS

Each educational unit consists of a set of canonical educational concepts. The text content in each educational unit can be used to find the concepts involved in it. The set of concepts in all units is defined as the universal concept space [10]. We define three concept space representation schemes as follows:

- **Content Words Representation (Word):** The set of content words (Nouns, Verbs, Adjectives and Adverbs) occurring in the course content is used as the concept space. The words are lemmatized using MIT Java Wordnet Interface (JWI) [3].
- **Noun Words Representation (Noun):** In this representation scheme, we only use set of nouns occurring in the course content as the concept space rather than all content words. These are again lemmatized using MIT JWI.
- **Noun-Phrase Representation (NP):** In this representation scheme, the set of noun phrases (of depth less than 5) occurring in the course content is used as the concept space.

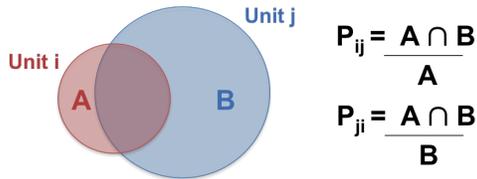


Figure 2: Overlap Method

An example of these three types of representation schemes is shown in Figure 1. The Concept space can be represented using other schemes such as Sparse Coding of Words and Distributed Word Embedding, but these produce latent concepts, which are not human understandable. Furthermore, previous results indicate word-based Representation scheme is more effective than latent concept based representation schemes [10].

Let the total number of the concepts in the concept space be  $p$ . Then the educational content in each unit can be represented by a  $p$ -dimensional vector, where each element is the frequency of corresponding concept (word, noun or noun-phrase) in the text content of the unit. The concept frequency can be normalized using the following quantities:

- **Collection Frequency (CF):** Total number of occurrences of the word in the collection or in our case, course. This normalizes concept frequencies such that all concepts are given equal weightage.
- **Document Frequency (DF):** Number of documents or in our case, units, that contain the concept. This gives less weightage to words occurring in most units such as module, learning objective, psychology, etc.
- **Wordnet Frequency (WF):** The frequency of word given in WordNet which represents the frequency of word in naturally occurring domain-independent text. This re-scales the frequencies such that domain-specific psychology terms have more weightage than generic terms.

We first describe an unsupervised method which determines prerequisite links based on only the text overlap between educational units. The key idea is that course unit  $u_i$  is a prerequisite of  $u_j$  to the extent that  $u_i$  is a probabilistic subset of  $u_j$  (i.e., most concepts involved in  $u_i$  are mostly involved in  $u_j$ ) and  $u_j$  is not a probabilistic subset of  $u_i$  (i.e., most concepts involved in  $u_j$  are not involved in  $u_i$ ). This idea of using asymmetry in computing the probabilistic subset is motivated by the theory of knowledge spaces [2], but we use text information rather than performance data.

Let  $x_i$  be a vector denoting the concept space representation of unit  $u_i$ . The length of this vector is the total number of the concepts. Each element of this vector is the frequency of the concept in the unit or one of the normalized versions of concept frequency (CF, DF or WF). The intuitive gloss on how we compute the probability that  $x_i$  is a probabilistic subset of  $x_j$  is by dividing the size of the intersection of  $x_i$  and  $x_j$  by the size of  $x_i$  ( $A$  is a subset of  $B$  if  $A \cap B = A$  and less so to the extent that  $A \cap B < A$ , see Figure 2).

Mathematically, we define  $P_{ij}$  as the ratio of sum of elements of pairwise minimum of  $x_i$  and  $x_j$  to the sum of elements of  $x_i$ :

$$P_{ij} = \frac{\text{sum}(\min(x_i, x_j))}{\text{sum}(x_i)} \quad (1)$$

Then  $P_{ij}$  is the weight of the prerequisite link from unit  $i$  to unit  $j$ , which ranges from 0 to 1.

## 5. PERFORMANCE-BASED METHODS

Our particular approach for unsupervised induction of PSG based on student performance data grows out of recent analysis of student performance [6] which concludes that interactive activities are more indicative of learning gains than video watching or online text reading. In subsequent analysis, it was found that student learning within a course unit is more highly predicted by their activity within that unit than within other units [7]. However, there is an additional learning outcome boost associated with greater activities before a target unit, but not with greater activities after that unit. This result is consistent with there being prerequisite relationships between prior and later units and was the inspiration for new algorithm development on performance-based PSG inference.

The key idea behind the proposed performance-based methods is that more the activity in unit  $i$  predicts success in unit  $j$ , the more likely is unit  $i$  a prerequisite of unit  $j$ . This means that if students who do more activities in week  $i$  perform better in the week  $j$  quiz, as compared to students who do fewer activities in week  $i$ , then there is an evidence for a prerequisite link from content in week  $i$  to week  $j$ . Let

$y_j$  be Quiz Scores in week  $j$ ,

$x_i$  be the number of interactive activities done in week  $i$ , and

$w_{ij}$  be the parameters denoting the effect of activities in week  $i$  on quiz in week  $j$ , which we want to estimate.

The value of the parameter is the strength of corresponding prerequisite relationship.

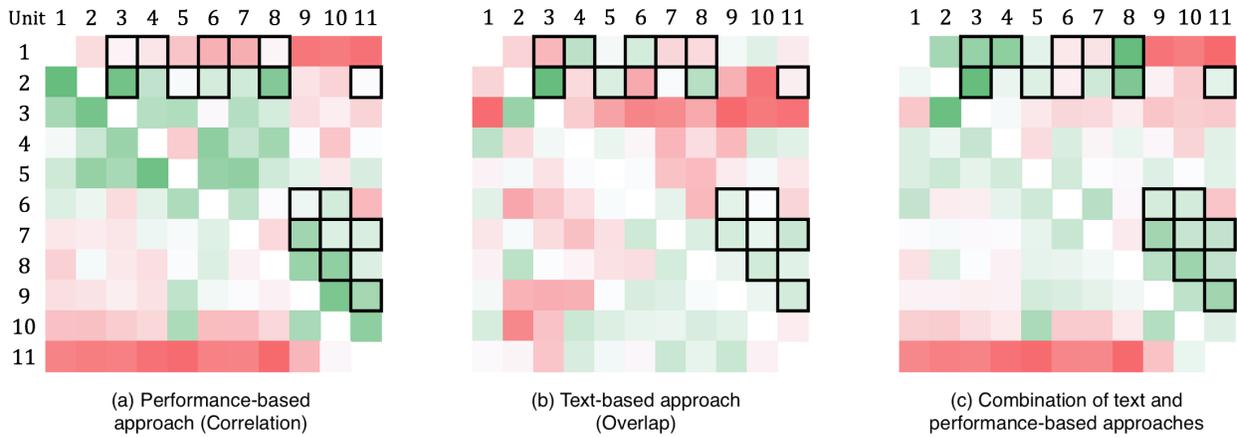
We define two methods for predicting prerequisite links using student performance data:

- **Correlation:** The effect of activities in week  $i$  on the performance in week  $j$  is estimated by the correlation between the number of activities by students in week  $i$  and the quiz scores of students in week  $j$  quiz. Let  $\rho(X, Y)$  be the Pearson correlation coefficient between  $X$  and  $Y$ . Then,

$$w_{ij} = \rho(x_i, y_j) = \frac{\text{cov}(x_i, y_j)}{\sigma_{x_i} \sigma_{y_j}}$$

- **Multiple Linear Regression:** We compute a linear regression for student quiz scores across the 11 units of the course where the dependent variable is student quiz score for the target unit and the independent variables are number of activities students do within each unit. Let  $\mathbf{w}_j = [w_{1j}, w_{2j}, \dots, w_{11j}]$ , be a vector denoting the effects of activities in all weeks on quiz score of week  $j$ . We define multiple linear regression using lasso regularization as follows:

$$\mathbf{w}_j^* = \underset{\mathbf{w}_j}{\text{argmin}} \sum_n (y_j - \mathbf{x}^T \mathbf{w}_j)^2 + \lambda \|\mathbf{w}_j\|$$



**Figure 3:** The heat map of strength of links from Unit  $i$  to Unit  $j$  for (a) Performance-based (Correlation) approach, (b) Text-based (Overlap) approach and (c) Combination of both. The black boxes represent the prerequisite links labeled by domain experts. Note that there is no link from Unit 7 to Unit 10, even though it appears to be surrounded by a black box.

Method Name	Method Type	Data Utilized	MAP	AUC
Regression	Unsupervised	Performance	0.562	0.571
Correlation	Unsupervised	Performance	0.604	0.720
Overlap	Unsupervised	Quiz Text	0.693	0.700
Overlap	Unsupervised	Unit Text	0.743	0.710
Overlap + Corr	Unsupervised	Performance & Quiz Text	0.798	0.820
Overlap + Corr	Unsupervised	Performance & Unit Text	<b>0.837</b>	<b>0.840</b>
CGL[10]	Supervised	Unit Text & Labeled links	0.747	0.820

**Table 2:** Comparison of all methods

## 6. COMBINING TEXT-BASED AND PERFORMANCE-BASED METHODS

We observed that most of the prediction errors in unsupervised text-based and performance-based methods were due to false-positives. This is because the dataset is imbalanced towards negative class with 85.45% negative labels. Unsupervised systems lacking this information predict positive and negative instance without any prior bias. In order to reduce the errors due to false positives, we propose to predict a positive link only when both methods indicate a positive link.

We get two square matrices of dimension equal to the number of units in the course, one each from text-based and performance-based methods. The  $(i, j)^{th}$  element of these matrices represents the weight of the prerequisite link from unit  $i$  to unit  $j$  obtained from the corresponding method. We combine the two methods by first forcing diagonal entries (self-links) to be 0, then standardizing both the matrices such that both have zero mean and equal variance and then just applying a pairwise minimum over these standardized matrices. This approach predicts a link between any ordered pair of units only if both methods suggest that there should

be a link between them. The combination of both methods using a pairwise minimum operation performed better than combination using pairwise summation, pairwise maximum and pairwise product. We also explored more complex models for combination, but found no evidence to justify model complexity.

## 7. EXPERIMENTS & RESULTS

We gathered and annotated the dataset for experiments as described in Section 3. For evaluation, we used macro-averaged Mean Average Precision (MAP) [5] and Area under ROC Curve (AUC) [4], which are popular metrics in ranked list retrieval and link detection evaluations [10].

The first two rows in Table 2 show the performance of two proposed performance-based methods: Multiple Linear Regression and Correlation. As the Correlation method performed better than Regression method (MAP 0.604 vs 0.562 and AUC 0.720 vs 0.571), we will use Correlation method for combining with text-based methods. The third and fourth column in Table 3 show the performance of text-based Overlap method over different concept space representation types and normalization types. The last two columns of this table show the performance of Overlap method combined with Correlation method. We compare this combined method to supervised Concept Graph Learning algorithm (CGL) [10]. The best results of all methods are summarized in Table 2, which shows that the unsupervised method which combines text-based and performance-based approaches outperforms supervised concept graph learning algorithm by a considerable margin (MAP 0.837 vs 0.747 and AUC 0.840 vs 0.820). As seen Table 3, the combined method performs better than CGL for most concept space representation and normalization types. Note that as compared to supervised CGL method, the proposed method (‘Overlap+Corr’) utilizes performance data in addition to the text content in educational material but doesn’t require labeled links from experts. The results in Table 3 also suggest that on an average, Noun Phrase concept space representation works best for all text-based methods, although there is no clear winner among Normalization types.

Method Name		Overlap		CGL		Overlap+Corr	
Method Type		Text Unsupervised		Text Supervised		Perf+Text Unsupervised	
Rep Type	Norm Type	MAP	AUC	MAP	AUC	MAP	AUC
Word	None	0.656	0.640	0.685	0.789	0.686	0.750
	CF	0.667	0.680	0.742	0.805	0.717	0.800
	DF	0.638	0.660	0.638	0.766	0.836	0.830
	WF	0.693	0.660	0.676	0.781	0.730	0.800
NP	None	0.661	0.680	0.722	0.789	0.745	0.810
	CF	0.703	0.710	<b>0.747</b>	<b>0.820</b>	0.792	0.820
	DF	<b>0.743</b>	<b>0.710</b>	0.572	0.773	<b>0.837</b>	<b>0.840</b>
	WF	0.717	0.710	0.743	0.805	0.748	0.820
Nouns	None	0.734	0.670	0.751	0.805	0.746	0.820
	CF	0.681	0.680	0.687	0.797	0.821	0.810
	DF	0.721	0.680	0.535	0.766	0.755	0.830
	WF	0.738	0.680	0.696	0.797	0.748	0.820

**Table 3:** Comparison of different concept space representation schemes (Rep Type) and different Normalization schemes (Norm Type) over different text-based methods. CF, DF and WF refer to Collection Frequency, Document Frequency and WordNet Frequency, respectively, as described in Section 4. The best AUC and MAP scores for each method are marked in **bold**.

We analyzed the weights of links predicted by different methods to understand how the combination of text and performance based methods affects our prediction. Figure 3 shows a heat map of strength of links between all pairs of Units. Each  $(i, j)^{th}$  element in the matrix represents the strength of link from Unit  $i$  to Unit  $j$ , where green is denoting higher strength and red is denoting lower. Note that the heat of the colors is determined by relative value of the weights in one matrix and not absolute values across matrices. This is because AUC and MAP metrics evaluate relative value of predicted weights rather than absolute values. The black boxes represent the prerequisite links labeled by experts. The figure indicates that the estimates of performance and text-based approaches compliment each other to give better estimates when combined.

## 8. DISCUSSIONS

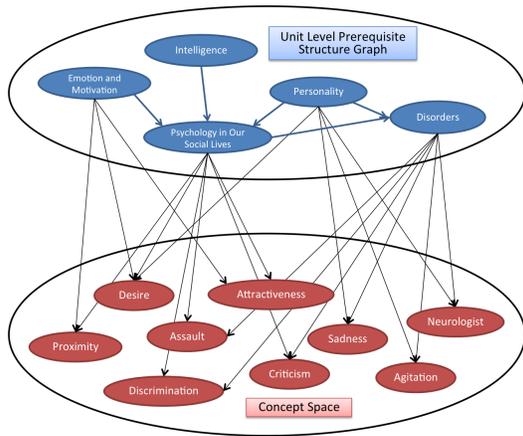
Figure 4 demonstrates a subset of prerequisite links identified by the proposed method and a subset of overlapping concepts occurring in them in the concept space. We would like to analyze whether the concepts identified by the proposed method are meaningful. Consider the relationship between Unit 11, ‘Emotion and Motivation’ and Unit 13, ‘Psychology in Our Social Lives’. All the proposed methods estimate significant weights for link from Unit 11 to Unit 13. Figure 6 shows a part of the concept space representation using Content Words Representation scheme for these units. Overlap method indicates a strong prerequisite link from Unit 11 to Unit 13 due to significant overlap between the concepts in these units. Looking into the contents of these units, the Unit 11, ‘Emotion and Motivation’ consists of ‘Human Motivation’ module which involves understanding the motivation behind sexual behavior. It introduces concepts of ‘attractiveness’, ‘proximity’ and ‘similarity’ as motivating factors behind sexual interest. Unit 13, ‘Psychology in Our Social Lives’ requires the understanding of these concepts in order to understand ‘Interpersonal Attraction’ in ‘Close Relationships’ module. Since there are more

concepts in Unit 13 like ‘personality’, ‘aggression’, ‘stimulus’, ‘judgment’, etc. which are not present in Unit 11,  $P_{11,13}$  is greater than  $P_{13,11}$ . Thus, the concepts extracted by the proposed Concept Representation schemes appear to be interpretable and meaningful.

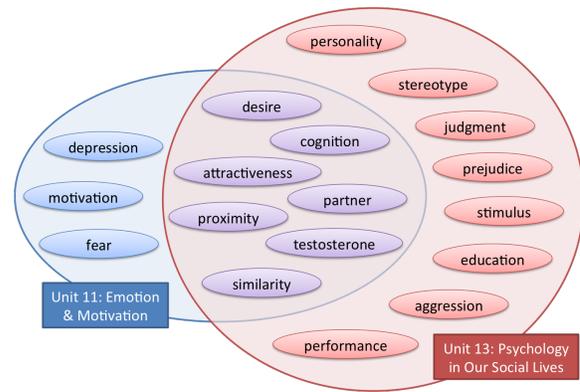
Similarly, we also try to interpret the performance-based results by inspecting the text of the interactive activities within the course. For example, the interactive activities in ‘Human Motivation’ module correspond to understanding concepts of ‘attractiveness’, ‘proximity’ and ‘similarity’. The quiz at the end of unit on ‘Psychology in Our Social Lives’ also contains a question about role of proximity and similarity in interpersonal attraction. Therefore, the students who do more activities in week 8 (involving Unit 11 content) perform better on the week 10 (involving Unit 13) quiz (as compared to students who do fewer week 8 activities) and thus, performance-based approaches identify this relationship. Figure 5 shows the average number of activities of students in prior units as a function of their quiz scores in later unit for set of positive and negative links. The average number of activities in prerequisite units is greater than non-prerequisite units for all quiz scores which is a possible explanation of the effectiveness of performance-based methods. Also, the correlation between number of activities and quiz scores suggests that interactive activities are indicative of learning gains.

## 9. CONCLUSIONS & FUTURE WORK

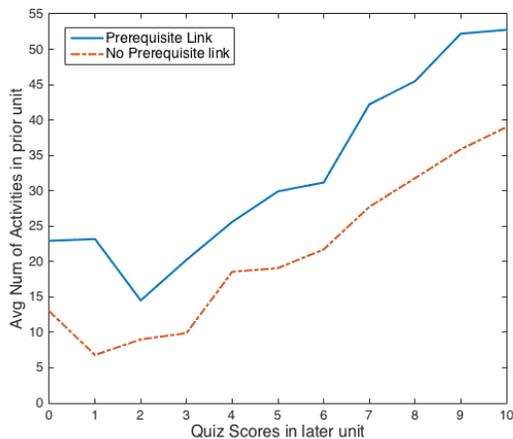
We proposed completely unsupervised methods to leverage freely available textual content in educational resources and student performance & activity data for predicting prerequisite structure graph between arbitrary educational resources. Three different concept space representation schemes have been used for text-based methods with a variety of normalization methods for concept frequencies. We also show that when unsupervised text-based and performance-based methods are combined, they supplement each other to outper-



**Figure 4:** Demonstration of prerequisite links between different units in ‘Introduction to Psychology’ Course and a subset of overlapping concepts.



**Figure 6:** Demonstration of overlap of concepts between units on ‘Emotion and Motivation’ and ‘Psychology in Our Social Lives’ and prediction of prerequisite link using Overlap method.



**Figure 5:** The average number of activities of students in prerequisite units as a function of their quiz scores in post-requisite unit.

form sophisticated supervised methods. Concepts extracted using the proposed representation schemes seem to be interpretable and meaningful from educational perspective.

While the results are encouraging, a limitation of the current work is the size of the dataset. Although the text content in the course and student activity and performance data is rich, the number of positive prerequisite relations in the dataset is low. Validation of proposed methods on diverse educational data from different courses is required to test their generalizability and scalability. Furthermore, conducting a long-term user-study involving students to verify if the predicted prerequisites help them improve their performance over a course, would be useful.

## 10. REFERENCES

[1] M. C. Desmarais, A. Maluf, and J. Liu. User-expertise modeling with empirically derived probabilistic implication networks. *User modeling and user-adapted*

*interaction*, 5(3-4):283–315, 1995.

[2] J.-P. Doignon and J.-C. Falmagne. Spaces for the assessment of knowledge. *International journal of man-machine studies*, 23(2):175–196, 1985.

[3] M. A. Finlayson. Java libraries for accessing the princeton wordnet: Comparison and evaluation. In *Proceedings of the 7th Global Wordnet Conference, Tartu, Estonia*, 2014.

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

[5] K. Kishida. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan, 2005.

[6] K. R. Koedinger, J. Kim, J. Z. Jia, E. A. McLaughlin, and N. L. Bier. Learning is not a spectator sport: Doing is better than watching for learning from a mooc. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 111–120, 2015.

[7] K. R. Koedinger, E. A. McLaughlin, J. Z. Jia, and N. L. Bier. Is the doer effect a causal relationship? how can we tell and why it’s important. In *Proceedings of the Sixth International Learning Analytics & Knowledge Conference*, 2016.

[8] M. J. Nathan, K. R. Koedinger, and M. W. Alibali. Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the Third International Conference on Cognitive Science*, pages 644–648. Citeseer, 2001.

[9] S. Wang, C. Liang, Z. Wu, K. Williams, B. Pursel, B. Brautigam, S. Saul, H. Williams, K. Bowen, and C. L. Giles. Concept hierarchy extraction from textbooks. In *Proceedings of the 2015 ACM Symposium on Document Engineering, DocEng ’15*, pages 147–156, New York, NY, USA, 2015. ACM.

[10] Y. Yang, H. Liu, J. Carbonell, and W. Ma. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 159–168. ACM, 2015.

# Exploring Learning Management System Interaction Data: Combining Data-driven and Theory-driven Approaches

Hongkyu Choi, Ji Eun Lee, Won-joon Hong, Kyumin Lee, Mimi Recker, Andy Walker  
Utah State University  
Logan, UT, USA

{hongkyu.choi, jieun.lee, wonjoon.hong}@aggiemail.usu.edu  
{kyumin.lee, mimi.recker, andy.walker}@usu.edu

## ABSTRACT

This research connects several data-driven educational data mining approaches to a framework for interaction developed in educational research. In particular, 10 million usage data points collected by a Learning Management System used by students and teachers in 450 online undergraduate courses were analyzed with this framework. A range of educational data mining techniques were employed, including K-means clustering, multiple regression, and classification, to both explore and predict student final grades and course completion rates. Findings show that support for the overall model varied with the way data were mapped to the framework (e.g., static vs. temporal features) and the analysis technique used (with clustering and classification providing more useful insights).

## Keywords

Learning Management System, Interactions in Online Learning, Clustering, Prediction

## 1. INTRODUCTION

Educational data mining (EDM) studies have typically relied upon data-driven techniques in order to extract useful patterns and information from large-scale educational datasets [11]. While these data-driven approaches have provided important contributions, some have argued that their inherent a-theoretic nature may fall short in terms of providing insight into the development of educational theory and practice [6]. As such, more studies are needed that better connect EDM findings to educational theory, research, and practice.

To address this need, this paper integrates a theory-driven approach with a data-driven approach to explore student learning outcomes, activities, and patterns as they interact with course content using a popular Learning Management System (LMS), called Canvas. Specifically, for the theory-driven approach, we apply an interaction framework [2] to explore how patterns in the LMS data are related to student

final grades and course completion rates at a course level – a macro-perspective. Here, we use K-means clustering and multiple regression analysis. For the data-driven approach, we build classifiers based on machine learning algorithms to predict a student's final grade and whether a student will complete a course or not, providing a micro-perspective.

In particular, we conducted three tasks by addressing following research questions: 1) How many clusters of courses are found based on users' interaction patterns? Are there relationships between individual interaction clusters and course features (size, content, level)? 2) Do the interaction patterns significantly predict student final grades and course completion rates? 3) Can we build effective classifiers to predict an individual student's final grade and whether each student will complete a course? Are the pre-built classifiers still robust and effective for the next semester's data? How many weeks in a semester are needed to discover low performing students or non-course completers (i.e., who may drop out a course)?

## 2. BACKGROUND

### 2.1 Interaction in Online Learning

Interaction has long been a significant research topic in the field of educational technology. Nonetheless, it remains a hard concept to define, as it is multifaceted and complex [1, 7]. Some researchers have taken a more restrictive view by excluding non-human factors, and focusing only on human interactions [5]. However, others argued that both human and non-human interactions are integral aspects of the educational experience [1, 2, 4]. Further, supporting various combinations of interaction among teacher, student and the content can help foster a community of inquiry in online learning [4].

In particular, Moore [7] categorized interaction into three types: (i) learner-content interaction, (ii) learner-instructor interaction and (iii) learner-learner interaction. Anderson and Garrison [2] expanded Moore's categorization by differentiating between teacher-content and student-content interaction. In their final model, teacher-content (TC) interaction refers to teachers creating content and learning activities. Student-content (SC) interaction refers to students' interactions with various forms of educational content including reading texts, completing assignments, and working on projects. Student-teacher (ST) interaction includes both asynchronous and synchronous communication between students and teachers. Finally, student-student (SS) interaction

**Table 1: Characteristics of 450 courses.**

Course characteristics		Courses	Percent
STEM Non-STEM	STEM	116	25.8%
	Non-STEM	334	74.2%
Course size	Small (<21)	107	23.8%
	Med (<51)	210	46.7%
	Large (51+)	133	29.5%
Course level	1000 level	156	34.7%
	2000 level	79	17.5%
	3000 level	157	34.9%
	4000 level	58	12.9%

refers to interaction between individual students.

There have been several empirical studies investigating the relationships between different types of interaction and student learning. For example, Bernard et al. [3] conducted a meta-analysis on the effects of the three types of interactions (i.e., SC, ST and SS) on student performance in online learning. They found that the effects of SS interaction and SC interaction were significantly larger than the effect of ST interaction in terms of student performance.

In this paper, we use this interaction framework to explore how interaction is related to student performance and course completion rates in online courses by analyzing and exploring LMS interaction data.

## 2.2 Educational Data Mining in Learning Management Systems

A LMS provides a wide range of features to support interactions between students, teachers, and content [9]. Moreover, the LMS typically captures interactions with these features in various formats and at diverse granularity levels. The most widely used methods in EDM studies using LMS data are prediction, clustering, and distillation for human judgment (visualization) [10]. Prior studies have found that usage variables related to SS interaction (i.e., the number of discussion messages posted) and SC interaction (i.e., the number of completed assignments) were significant predictors of student performance [6, 12].

However, prior studies using LMS data analyzed student-level data, rather than looking at the various levels and kinds of interactions between teachers, students, and contents. In this paper, we used course level data as well as individual student level data to provide both macro- and micro-perspectives on interactions between students, teacher, and contents in online learning. In this way, our research complements the existing research base.

## 3. DATASET AND METHODS

### 3.1 Dataset

For the present study, data were extracted from the Canvas LMS deployed at a mid-sized public university located in the western U.S. The LMS automatically captures all teacher and student online interactions. Note that an academic support unit at the university extracted and anonymized these data, and Institutional Review Board (IRB) approved using the data for research purposes.

We conducted data preprocessing by transforming raw data into an appropriate shape for analysis. First, we performed

data cleaning in the following three steps: 1) selected courses offered between Fall 2014 and Spring 2015; 2) selected only online undergraduate courses; and 3) excluded low enrollment courses (i.e., the number of enrolled students is less than 5). After conducting the data cleaning process, our dataset consisted of 450 courses including 10,576,718 interactions, and anonymized 21,171 student profiles (8,844 distinct student profiles) and 450 teacher profiles (228 distinct teacher profiles).

Table 1 shows the number of courses in our dataset, categorized by STEM vs. non-STEM, size, and course level. 25.8% courses are Science, Technology, Engineering, and Mathematic (STEM) courses. A full range of course sizes is represented and is centered around medium-sized enrollments (i.e., 21-50 students). The largest number of courses is 1000 level (34.7%) and 3000 level (34.9%) courses.

### 3.2 Data Mining Methods and Features

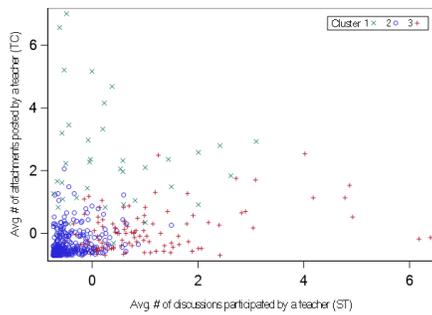
In this study, we used three data mining methods for three tasks – one method for each task: (i) K-means clustering to find groups of courses each of which has similar interaction patterns at a course level; (ii) multiple regression to measure the relationship between each interaction feature/variable and average student final grade and course completion rates at a course level; and (iii) classification algorithms to predict each student’s final grade and whether the student will complete a course or not. The first two methods provided a macro perspective focusing on courses, while the last method provided a micro perspective focusing on individual students.

**Task 1.** We used K-means clustering to identify how online courses were clustered based on interaction patterns. We used the PROC FASTCLUS method in SAS, as missing values were replaced with an adjusted distance using the non-missing values [8]. We used Euclidean distance to measure distance between each node (i.e., a course) and a centroid. To find the optimal  $K$ , we examined the agglomeration schedule to determine the optimal number of clusters.

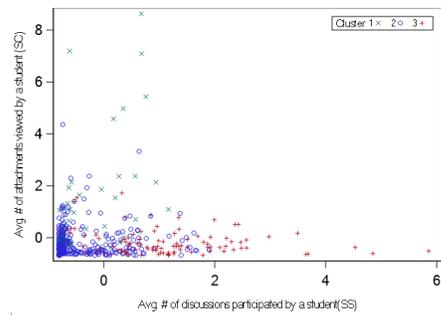
**Task 2.** We conducted multiple regressions using SAS to test whether each interaction type significantly predicted outcome variables – average final grades and course completion rates.

For Tasks 1 and 2, we grouped Canvas features (variables) into four categories (TC, SC, SS, ST) based on Anderson and Garrison’s interaction framework [2]. Table 2 presents four categories associated with the Canvas features, and each feature’s mean, standard deviation (SD) and minimum and maximum values obtained from the 450 courses.

**Task 3.** We applied classification algorithms (i.e., SVM, Random Forest, J48 and AdaBoost) to predict each student’s final grade and whether the student will complete a course or not. Effectiveness of classifiers depends on quality of features. For this task, we used 129 features consisting of 52 static features and 77 temporal features as shown in Table 3. These features consisted of not only the main interaction features that we used in the first and second tasks (while they were average values in the first and second tasks, individual student feature values were used in the third task),



(a) ST interaction vs. TC interaction (z-transformed data).



(b) SS interaction vs. SC interaction (z-transformed data).

Figure 1: Scatter plots showing how courses in clusters are distributed differently.

Table 2: Descriptive statistics of 450 courses analyzed by 12 interaction features associated with four categories.

Category	Features	Mean	SD	Min-Max
Teacher-Content	Avg. # of attachments posted by a teacher (tc_atta)	15.97	22.86	0-176
	Avg. # of discussion topics posted by a teacher (tc_disc)	18.55	15.54	0-107
	Avg. # of wiki topics posted by a teacher (tc_wiki)	13.58	13.96	0-74
	Avg. # of quizzes posted by a teacher (tc_quiz)	9.72	9.48	0-56
	Avg. # of assignments posted by a teacher (tc_assi)	15.30	12.97	0-75
Student-Content	Avg. # of attachments viewed by a student (sc_atta)	118.19	174.57	0-1,625
	Avg. # of discussions viewed by a student (sc_disc)	48.05	44.88	0-296
	Avg. # of wiki viewed by a student (sc_wiki)	54.42	51.92	0-387
	Avg. ratio of completed quiz by a student (sc_quiz)	0.88	0.12	0.10-1
Student-Student	Avg. ratio of completed assignments by a student (sc_assi)	0.78	0.16	0.10-1
Student-Student	Avg. # of discussions participated by a student (ss_disc)	12.21	15.13	0-101
Student-Teacher	Avg. # of discussions participated by a teacher (st_disc)	50.15	68.63	0-489

but also additional features (e.g., the number of views of the grade and announcement pages, course information and temporal features). In particular, temporal features were extracted from a series of daily snapshots of each student’s interaction record. Given a course and interaction information of a student who took the course, we represented the student by using the 129 features.

## 4. EXPERIMENTAL RESULTS

In the previous section, we described our dataset and three data mining methods for conducting three tasks. In this section, we present results of these experiments using each of the methods for each task.

Table 3: 129 Features extracted from each student and each corresponding course.

Static Features	
Features	Features
Course level and Department offering the course	2
Total # of views and total # of participation by a student	2
# of views and participation in each of the 24 items by a student	48
Temporal Features	
Features	Features
Total # of participated weeks (i.e., we add +1 if a student did participation at least once in a week)	1
Mean and standard deviation of weekly view count and weekly participation count	4
Each week’s view count and participation count	36
Accumulated weekly view count and accumulated weekly participation count	36

### 4.1 Task 1: Clustering Courses and Analyzing Characteristics of Clusters

In Task 1, our research goal was to cluster courses based on interaction patterns and analyze characteristics of the clusters. First, we standardized the interaction features/variables (raw scores) by following the recommendation in the literature [8]. The raw scores were z-transformed to a mean of 0 and standard deviation of 1 for either the course or semester level data.

K-means clustering requires an input  $K$ . To make sure we chose an optimal  $K$ , we examined the agglomeration schedule. The demarcation point indicated that  $K = 3$  would produce the optimal result. Clusters 1, 2 and 3 contained 41, 300 and 109 courses, respectively. The root mean squared standard deviations (RMSSTD) for each cluster were 1.32, 0.71, 0.98 respectively, indicating that the courses in cluster 1 are more widely dispersed than the others.

We further drew two scatter plots to help understand characteristics of the three clusters as shown in Figure 1. Figure 1(a) represents a scatter plot of ST interaction ( $st\_disc$ ) vs. TC ( $tc\_atta$ ) interaction. Courses in cluster 1 had higher TC interaction than those in the other clusters, whereas courses in cluster 3 had higher ST interaction than the other two clusters. Figure 1(b) shows a scatter plot of SS interaction ( $ss\_disc$ ) vs. SC interaction ( $sc\_atta$ ). Courses in cluster 1 showed higher student-content interaction than the other two clusters. On the contrary, courses in cluster 3 showed higher SS interaction than the other two clusters.

**Table 4: Means and standard deviations of clusters. \* indicates the highest value among the three clusters.**

Feature	Cluster 1 (n=41) Content- interaction		Cluster 2 (n=300) Low- interaction		Cluster 3 (n=109) Inter-person interaction	
	M	SD	M	SD	M	SD
tc_atta	2.12	1.78	-0.32	0.44	0.09	0.67
tc_disc	0.26	0.96	-0.44	0.59	1.1	1.04
tc_wiki	1.53	1.31	-0.37	0.64	0.43	0.98
tc_quiz	0.68	1.32	-0.05	0.99	-0.12	0.76
tc_assi	0.38	1.23	-0.28	0.77	0.62	1.14
(T-C) mean	0.99*	0.66	-0.29	0.43	0.42	0.55
sc_atta	1.47	2.27	-0.14	0.62	-0.18	0.47
sc_disc	-0.04	0.52	-0.46	0.55	1.22	1.02
sc_wiki	1.8	1.62	-0.23	0.68	-0.07	0.7
sc_quiz	-0.18	1.04	0.02	0.92	0.02	1.19
sc_assi	-0.18	1.15	0.03	1.04	-0.01	0.84
(S-C) mean	0.57*	0.85	-0.16	0.46	0.2	0.42
(S-S)	-0.2	0.59	-0.38	0.58	1.05*	1.22
(S-T)	0.29	1.02	-0.43	0.33	1.07*	1.33
<b>final grades</b>	<b>2.77</b>	<b>0.59</b>	<b>3.01</b>	<b>0.57</b>	<b>3.05*</b>	<b>0.38</b>
<b>complet. rates</b>	<b>84.04</b>	<b>12.95</b>	<b>86.84</b>	<b>12.75</b>	<b>88.09*</b>	<b>9.18</b>

Next, we examined descriptive statistics for the predictors and outcome variables (final grades and completion rates) for each cluster as shown in Table 4<sup>1</sup>. The results showed that cluster 1, dubbed “*Content-Interaction courses*”, had the highest means for both TC interaction ( $M = 0.99$ ,  $SD = 0.66$ ) and SC interaction ( $M = 0.57$ ,  $SD = 0.85$ ). Cluster 2, dubbed “*Low-Interaction courses*”, had the lowest means for all interaction variables. Lastly, cluster 3, dubbed “*Inter-person Interaction*”, had higher means for SS interaction ( $M = 1.05$ ,  $SD = 1.22$ ) and ST interaction ( $M = 1.07$ ,  $SD = 1.33$ ). The analysis revealed that courses in each cluster had different course emphases: content interaction in cluster 1, non-interaction in cluster 2, and person interaction in cluster 3.

Then, we compared the three clusters in terms of average student final grades and course completion rates. As shown in Table 4, the cluster 3 had the highest mean in student final grades ( $M = 3.05$ ,  $SD = 0.38$ ) and course completion rates ( $M = 88.09$ ,  $SD = 9.18$ ) among the three clusters. The cluster 1 had the lowest mean in student final grades ( $M = 2.77$ ,  $SD = 0.59$ ) and course completion rates ( $M = 84.04$ ,  $SD = 12.95$ ). This finding reveals that the positive impact of courses focusing on interactions between participants.

Next, we conducted chi-squared tests to compare STEM and Non-STEM courses in the three clusters. As shown in Table 5, the distribution of the STEM and Non-STEM courses was significantly different across the three clusters,  $\chi^2(6, N = 450) = 7.80$ ,  $p < .05$ . STEM courses were infrequent overall, but even more scarce in the cluster 3.

Then, we analyzed how many courses in the three clusters

<sup>1</sup>The meaning of each feature’s acronym is described in Table 2.

**Table 5: The number of STEM and Non-STEM courses in three clusters.**

Cluster	Non-STEM	STEM	Total
C1	29 (70.7%)	12 (29.3%)	41
C2	21 (71.0%)	87 (29.0%)	300
C3	92 (84.4%)	17 (15.6%)	109
Total	334	116	450

**Table 6: The number of small, medium, large courses in three clusters.**

Cluster	Small	Medium	Large	Total
C1	13(31.7%)	13(31.7%)	15(36.6%)	41
C2	78(26.0%)	130(43.3%)	92(30.7%)	300
C3	16(14.6%)	67(61.4%)	26(24.0%)	109
Total	107	210	133	450

had small, medium and large enrollments. Table 6 shows the analytical results. The result of a chi-squared test showed significant differences among the three clusters,  $\chi^2(4, N = 450) = 15.31$ ,  $p < .05$ . The cluster 1 had the largest proportion of large courses, whereas the cluster 3 had the smallest proportion of large courses. The findings suggest that promoting interaction among participants is rarer in large courses.

Lastly, we examined how many courses in the three clusters were at the 1000, 2000, 3000 and 4000 levels. A chi-squared test found no significant differences in the distribution of the course levels among the clusters,  $\chi^2(6, N = 450) = 8.79$ ,  $p > .05$ .

## 4.2 Task 2: Prediction Using Multiple Regression Analysis

In task 2, first we conducted a multiple regression analysis to examine the influence of interaction features or feature category listed in Table 2 in predicting average student final grades in each course. Table 7 shows regression results of significant variables. The results indicated that the explanatory variables accounted for a modest 15.8% of the variance ( $R^2 = 0.16$ ,  $F(12, 411) = 6.41$ ,  $p < .05$ ). Several significant and negative predictors were found in teacher-content interaction. In particular, as *tc\_disc*, *tc\_wiki*, and *tc\_assi* increased, final grades tended to decrease. Findings in the student-content interaction category were the opposite. Final grades tended to increase when *sc\_quiz* and *sc\_assi* increased and the same is true in the student-teacher interaction category.

A second multiple regression analysis was conducted to test the influence of each interaction feature or each feature category on course completion rates. The explained variance was a modest at 15.7% ( $R^2 = 0.16$ ,  $F(12, 411) = 6.64$ ). Only a single teacher-content variable *tc\_wiki* was negatively significant. Student-content interaction features *sc\_quiz* and *sc\_assi* were significant and positive again in relation to course completion rates. Taken together, these findings suggest that certain teacher activities related to content were less productive, whereas student activities related to content were more positively productive in both final grades and course completion rates.

Table 7: Multiple regression results (\* indicates the feature is significant at the 0.05 level, and the table includes only significant features).

Category	Feature	final grades					completion rates				
		<i>B</i>	<i>SE(B)</i>	$\beta$	t	p	<i>B</i>	<i>SE(B)</i>	$\beta$	t	p
Intercept		0.000	0.089	0.000	29.600	0.001	0.000	0.089	0.000	29.600	<.0001
Teacher-Content Interaction	tc_disc	-0.006	0.003	-0.177*	-2.240	0.026	-0.078	0.060	-0.059	-0.990	0.324
	tc_wiki	-0.011	0.002	-0.295*	-4.540	0.001	-0.241	0.054	-0.202*	-3.710	0.000
	tc_assi	0.004	0.002	0.106*	1.970	0.050	0.037	0.048	0.033	0.690	0.490
Student-Content Interaction	sc_wiki	0.001	0.001	0.141*	2.140	0.033	0.125	0.015	0.029	1.900	0.058
	sc_quiz	0.003	0.001	0.164*	3.250	0.001	0.284	0.019	0.107*	5.650	<.0001
	sc_assi	0.003	0.001	0.177*	3.530	0.001	0.115	0.019	0.044*	2.290	0.023
Student-Teacher Interaction	st_disc	0.001	0.001	0.160*	2.340	0.020	0.130	0.011	0.022	1.910	0.057

Table 8: Feature Sets

Feature Set	Features (# of features)
A	Course level and department offering the course, total # of views and total # of participation (4)
B	feature set A + # of views and participation in each of the 24 items by a student (52)
C	feature set B + total # of participated weeks (53)
D	feature set C + mean and standard deviation of weekly view count and weekly participation count (57)
E	feature set D + each week’s view count and participation count, and accumulated weekly view count and participation count (129)

### 4.3 Task 3: Predicting Individual Student’s Final Grade and Course Completion

So far, experiments in Tasks 1 and 2 were conducted at the course levels, providing a macro perspective. Now we turn to building classifiers to predict individual student’s final grade and course completion (i.e., whether the student will complete the course or not) by using a data-driven approach, providing a micro perspective, and then evaluating effectiveness of the classifiers. In task 3, predicting a student’s final grade means predicting whether the student will belong to a high performance group (i.e., obtaining one of A, A-, B+, B and B-) or a low performance group (i.e., obtaining one of C+, C, C-, D+, D, F and W).

#### 4.3.1 Prediction in 2014 Fall Semester Dataset

In this experiment, we used the 2014 Fall semester dataset consisting of 229 courses with 4,314,425 interactions and anonymized 10,003 student profiles. To build highly accurate classifiers, proposing and using features which have significant distinguishing power is important. To test this, the 129 features listed in Table 3 were sampled to make five feature sets entitled feature sets A, B, C, D and E as shown in Table 8. As we chose from feature set A to E, the number of features increased by including the previous features but also additional features. Feature sets A and B consisted of only static features, while feature sets C, D and E consisted of static features and temporal features.

Since we didn’t know apriori which classification algorithm would perform the best, we chose 4 popular classification algorithms – SVM, Random Forest, J48 and AdaBoost. Given the 2014 Fall semester dataset, we did 10-fold cross-validation by dividing the dataset to 10 sub-samples. Each sub-sample

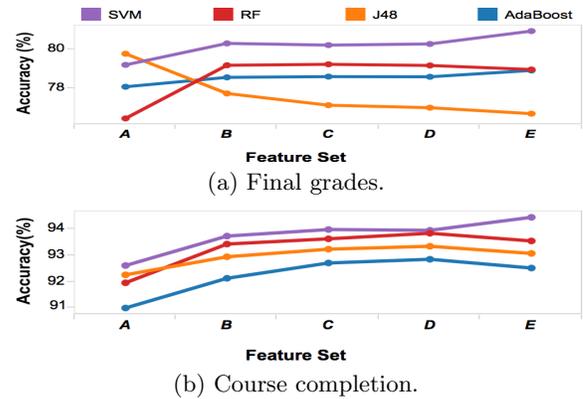


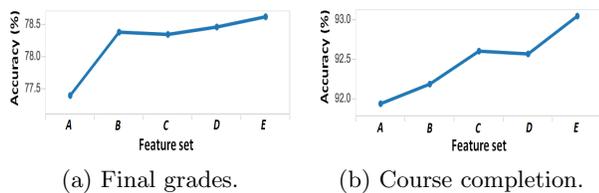
Figure 2: Prediction results of SVM, Random Forest, J48 and AdaBoost based classifiers with five feature sets.

became a test set, the other 9 sub-samples became a training set. We conducted a classification experiment for each of the 10 pairs of training and test sets. Then, we averaged the 10 classification results. We repeated this process for each classification algorithm.

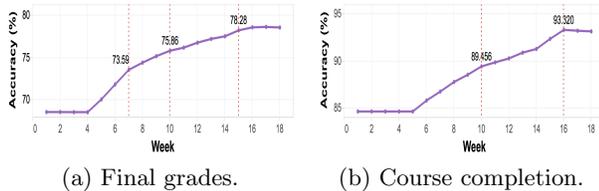
Figure 2 shows prediction results for final grades/performance groups and course completions. SVM based classifier outperformed Random Forest, J48 and AdaBoost based classifiers, achieving 80.95% accuracy, 0.79 F-measure and 0.72 AUC in final grade prediction and 94.41% accuracy, 0.94 F-measure and 0.85 AUC in course completion prediction. As we added more features (changing from feature set A to E), SVM classifier’s accuracy has increased in both predictions. Compared with the *baseline*, which was measured by a percent of the majority class instances and achieved 68% accuracy in final grade prediction and 84% in course completion prediction, our SVM based classifier improved 19% ( $= \frac{80.95}{68} - 1$ ) accuracy in final grades prediction, and 12.4% ( $= \frac{94.41}{84} - 1$ ) accuracy in course completion prediction.

#### 4.3.2 Robustness of Our Prediction Model

In Section 4.3.1, we evaluated effectiveness of our classification approach for both final grades prediction and course completion prediction. Now we are interested in how much the pre-built model is robust when we apply it to data generated in the future (i.e., future semesters). To simulate this scenario, we used the 2014 Fall semester dataset as a



**Figure 3: Prediction results obtained by applying SVM-based classifiers trained by 2014 Fall dataset to 2015 Spring dataset.**



**Figure 4: Prediction results over time.**

training set and the 2015 Spring semester dataset as a test set (consisting of 221 courses with 6,262,293 interactions and anonymized 11,168 student profiles). We built a SVM-based classifier and predicted each student's final grade and course completion in the test set.

Figure 3 shows prediction results as we used feature set *A* to *E*. Again, using all the features (feature set *E*) produced the best results, achieving 78.64% accuracy and 0.682 AUC in final grades prediction and 93.06% accuracy and 0.817 AUC in course completion prediction. Compared with the previous experimental results in Section 4.3.1, there were only small reductions – 2.31% (final grades) and 1.35% (course completion). The experimental results confirmed that our proposed approach is robust and can be applied to future semesters.

### 4.3.3 Early Prediction

The previous experimental results showed that our approach was effective in predicting final grades and course completion. In practice, it is better to produce prediction earlier so that a tool/system can automatically identify and alert which students are at risk of receiving a low grade or dropping out of a course thereby requiring intervention by a teacher. To address this need, we used daily snapshot of data including student profiles, course information and interaction logs, and then simulated the scenario by building a SVM-based classifier in each week. In other words, we built a classifier and evaluated its performance in each week. By doing this, we examined how the classifier's performance changed over time, and when we could achieve a reasonable accuracy.

Figure 4 shows prediction results in the 2014 Fall dataset. In final grades prediction, when we built classifiers in the 7th week, 10th week and 15th week, we achieved 73.59%, 75.86% and 78.28% accuracy, respectively. Similarly, in course completion prediction, we achieved 89.4% and 93.3% accuracy in 10th week and 16th week, respectively. Overall, adding more data improved performance of our classifiers. This study reveals that it is possible to detect students early who have a higher chance of receiving low grades or dropping out

a course.

## 5. CONCLUSIONS

The purpose of this study was to explore relationships between theoretically defined constructs extracted from a Learning Management System and student learning outcomes. Three different tasks employing three different methods were used to explore these relationships. The first two tasks were conducted at the macro-level and thus aligned with a theory-driven approach, whereas the last task at the micro level aligned with a data-driven approach.

Results from the cluster analysis revealed that courses with high inter-person (SS, ST) interaction had higher final grades and completion rates than courses in the other clusters (low-interaction and content-interaction), aligning with results from previous studies [6, 12]. Results also suggested that STEM and large courses tended to exhibit fewer of these productive interactions. The micro-level, data-driven machine learning analysis using prediction with SVM enabled the discovery of at-risk students with high accuracy. It achieved the best performance when all temporal features (complete feature set) were taken into consideration and was robust when predicting future data.

In sum, for this dataset comprised of LMS interactions drawn from online undergraduate courses, the interaction framework was useful for interpreting at both macro and micro levels.

## 6. REFERENCES

- [1] T. Anderson. Modes of interaction in distance education: Recent developments and research questions. *Handbook of distance education*, pages 129–144, 2003.
- [2] T. D. Anderson and D. R. Garrison. Learning in a networked world: New roles and responsibilities. 1998.
- [3] R. M. Bernard, P. C. Abrami, E. Borokhovski, C. A. Wade, R. M. Tamim, M. A. Surkes, and E. C. Bethel. A meta-analysis of three types of interaction treatments in distance education. *Review of Educational Research*, 79(3):1243–1289, 2009.
- [4] D. R. Garrison and M. Cleveland-Innes. Facilitating cognitive presence in online learning: Interaction is not enough. *The American Journal of Distance Education*, 19(3), 2005.
- [5] D. Laurillard. 8 new technologies, students and the curriculum. *Higher education re-formed*, 2000.
- [6] L. P. Macfadyen and S. Dawson. Numbers are not enough. why e-learning analytics failed to inform an institutional strategic plan. *Educational Technology & Society*, 15(3), 2012.
- [7] M. G. Moore. Editorial: Three types of interaction. *American Journal of Distance Education*, 3(2):1–7, 1989.
- [8] E. Reiss, S. Archer, R. Armacost, Y. Sun, and Y. Fu. Using sas® proc cluster to determine university benchmarking peers. *SESUG, Savannah GA, September*, 2010.
- [9] C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura. Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1), 2013.
- [10] C. Romero, S. Ventura, and E. García. Data mining in course management systems: Moodle case study and tutorial. *Comput. Educ.*, 51(1):368–384, Aug. 2008.
- [11] G. Siemens and R. S. d Baker. Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, 2012.
- [12] T. Yu and I.-H. Jo. Educational technology approach toward learning analytics: Relationship between student online behavior and learning performance in higher education. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*, 2014.

# A Comparison of Automatic Teaching Strategies for Heterogeneous Student Populations

Benjamin Clement  
Inria, France  
benjamin.clement@inria.fr

Pierre-Yves Oudeyer  
Inria, France  
pierre-yves.oudeyer@inria.fr

Manuel Lopes  
Inria, France  
INESC-ID, Instituto Superior  
Técnico, Portugal  
manuel.lopes@inria.fr

## ABSTRACT

Online planning of good teaching sequences has the potential to provide a truly personalized teaching experience with a huge impact on the motivation and learning of students. In this work we compare two main approaches to achieve such a goal, POMDPs that can find an optimal long-term path, and Multi-armed bandits that optimize policies locally and greedily but that are computationally more efficient while requiring a simpler learner model. Even with the availability of data from several tutoring systems, it is never possible to have a highly accurate student model or one that is tuned for each particular student. We study what is the impact of the quality of the student model on the final results obtained with the two algorithms. Our hypothesis is that the higher flexibility of multi-armed bandits in terms of the complexity and precision of the student model will compensate for the lack of longer term planning featured in POMDPs. We present several simulated results showing the limits and robustness of each approach and a comparison of heterogeneous populations of students.

## 1. INTRODUCTION

The current advances and ubiquity of learning and teaching technologies have the potential to improve education accessibility and personalization. Intelligent Tutoring Systems (ITS) have been proposed to make education more accessible, more effective, and as a way to provide useful objective metrics on learning [1].

A major aspect of personalized education is to be able to identify the current level of students and how to address particular difficulties in the student learning process. The goal is to be able to choose online the activity that better addresses the challenges being encountered by each particular student. Even two students with the same knowledge will require different activities to progress further due to their previous experience, cognitive skills or preferences. This is a difficult challenge because as ITS are encountering the students for the first time, it is difficult to know what is

the impact of each activity on their progress. A commonly used method is to exploit a population-wide model on how students learn and assume that they are all similar. The personalization in such an approach is limited to adapting to student's knowledge levels but assumes that the impact of each exercise is the same for all students with the same knowledge levels.

Different methods have been proposed to handle this problem. One popular and well-known method is the Partially Observable Markov Decision Process (POMDP) framework which has been proposed in different ways to select the optimal activities to propose to a learner [13]. This framework can find the optimal teaching trajectories for a given teaching scenario model if an accurate student model is provided which is not always possible in practice. The main drawback is the high computational complexity and as a consequence, only the simplest cases can be solved exactly. Another method explored recently to select optimized activities is to use the Multi-Arm Bandit (MAB) framework to personalize sequences of pedagogical activities [6]. These methods optimize learning in the short term (rather than in the long-term) and rely on much simpler student models while being computationally very efficient.

In this paper, we compare the POMDP framework and the MAB framework (specifically the algorithm ZPDES already evaluated in real classrooms [6]). We first introduce a student model used to compare the different algorithms. We then propose ways to model the heterogeneity in classrooms by considering that different students will have not only different learning parameters but also that they might have different dependencies between the different knowledge components (KCs). Our experiments will evaluate how well a MAB can approach the optimal solution of a POMDP, and how the different algorithms behave when encountering a heterogeneous group of student.

## 2. RELATED WORK

In this work we are interested in the impact of the quality of the student models on the quality of the sequences of activities chosen by online algorithms.

There are several approaches to automatically choose exercises based on the current knowledge level of students. We are here particularly interested in optimization methods that rely on minimal prior assumptions about the students or the knowledge domain.

One option already explored is the use of a partial-observable Markov decision process (POMDP) [13], [14]. POMDPs offer an appealing theoretical framework that guarantees an optimal long-term solution for a planning problem. However, in general, as the computational complexity is high, it is practically impossible to find an exact solution to the problem. Some approximate solutions in the domain of ITS have considered the use of aggregations of states instead of tracking the full knowledge components. Another drawback is that POMDPs require a precise student model for which the policy is optimized. If the real student encountered deviates from this model, then the optimality properties are lost.

A more recent approach is to use the Multi-Arm Bandit (MAB) framework to manage pedagogical activities [6]. MABs have the advantage of being extremely computationally efficient and rely on very weak student models. The main drawback is that there is no long-term planning of the best sequence of activities relying on an exploration-exploitation tradeoff to find the best path. Aware of such problem, authors of one such algorithm considered that standard MAB needs to be complemented with a weakly specified knowledge graph to provide a long-term view on the optimization [6].

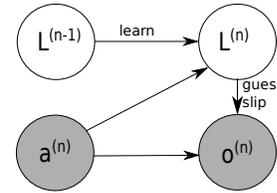
As noted, before optimizing the sequence of exercises it is important to have some knowledge about the impact of a given exercise in the learning of the KCs, and also to be able to track what each student already masters. A large part of ITS research has been on the modeling aspects of the cognitive and student models. A seminal work on this topic was the *Knowledge Tracing* framework [7] which builds a detailed cognitive model of the student, of its learning processes by considering a set of independent KCs, the probability of learning them and the probability of correct or wrong answer in exercises that relies on those KCs. More recent methods extend this framework to a bayesian probabilistic approach [12, 15] improving the performance and understanding of those methods. Recent methods have started to consider how to learn such models, and variants of it, allowing to simultaneously discover the relation between activities and KC, e.g. [8, 2, 5, 9].

As discussed these methods require an accurate knowledge of how students learn and require to track their mastery of each KC. For this, it is necessary to learn the constraints between different KC, exercises and KC. Given students' particularities, it is impossible for a teacher to understand all the difficulties and strengths of individual students and provide an accurate student model manually. Even with the recent advances on model learning, there are several challenges in identifying parameters that best describe each individual student. These models have many parameters, and identifying all such parameters for a single student is a very hard problem due to the lack of data, often making the problem intractable. In most cases it is even impossible to identify some of the parameters [3, 4]. In the general case, it results in inaccurate models that cannot be exploited for individualized learning. Another problem is that these planning methods are for a population of students and not for a particular student and this has already been proven to be suboptimal [11].

### 3. STUDENT MODELS

#### 3.1 Student model

In this section, we will present the student model we will use, also called learner model in literature. We want a generative model that can simultaneously be used to predict students behaviour, model their knowledge acquisition and track their mastery level. For this, we built a student model, shown in Fig.1 similar to the Knowledge Tracing framework [10] and its variants. Similarly to [9], we include extra features in our model. We are particular interested in more realistic cases where each KC might depend on other KCs. In most cases it is assumed that each exercise just depends on one KC and that they are independent, this is not realistic most of the time, and such dependencies have a strong impact on the learning sequences generated by the different algorithms.



**Figure 1: Graphical model of the Student model, with  $L^{(n)}$  the hidden state of the student at step  $n$ ,  $a^{(n)}$  activity proposed, and  $o^{(n)}$  the result obtained by the student.**

We consider a situation where a student has a set of  $m$  KCs  $K_i$  to learn. A student's state at step  $n$  is represented by the state of each KC,  $L^{(n)} = K_1^{(n)}, \dots, K_m^{(n)}$ , the global model is described on figure Fig.1. Each KC is defined by his state, mastered ( $K_i = 1$ ) or not mastered ( $K_i = 0$ ). For each KC, there is an initial probability of mastering it  $p(K_i^{(0)} = 1)$  which is always null in our experiments to make students learn all the KCs through activities. The emission probabilities are defined by the guess probability, i.e performing correctly without mastering the skill, and the slip probability, i.e performing incorrectly despite mastering the knowledge. These probabilities are constant. Finally  $p(K_i^{(n)} = 1 | L^{(n-1)}, a^{(n)})$  defines the probability of transition from not mastered to mastered  $K_i$  while doing activity  $a$  at step  $n$  and depending of the constraints between KCs and their states. An activity can be represented as a vector  $a = \alpha_1, \dots, \alpha_m$  where  $\alpha_i = 1$  if the activity allows to acquire  $K_i$ ,  $\alpha_i = 0$  else. The transition probability to learn a given KC  $K_i$  at step  $n$  is given by the following formula:

$$p(K_i^{(n)} = 1 | L^{(n-1)}, a^{(n)}) = \alpha_i(\beta_{i,i} + \sum_{j \neq i}^m \beta_{i,j} K_j^{(n-1)}) \quad (1)$$

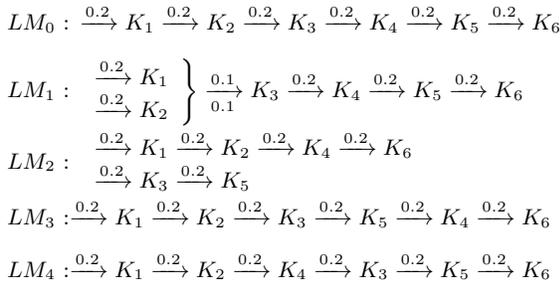
Where  $\beta_{i,i}$  represent the probability to learn  $K_i$  without considering other KCs and  $\beta_{i,j}$  represent the impact of the KC  $K_j$  on the probability to learn  $K_i$ . If a given KC does not need other KCs to be learned, the term  $\sum_{j \neq i}^m \beta_{i,j} K_j$  is null.

For more simplicity, in our experiments, an activity  $a$  can provide an opportunity to acquire only one KC which induces an isomorphism between the knowledge space and the activity space.

### 3.2 Models of populations

The previous model can be used to describe a single student or an average model of a population. Our goal is to understand the impact that the diversity of students has when the given sequence is optimized considering the same parameters for all students. We will achieve such goal by considering a canonical model and then make two types of disruptions: i) change the probabilities between the variables; ii) change the knowledge graph.

The first way is to disrupt the parameters in the model, i.e. the probability of transition, guess, and slip. To do that, we consider that each parameter is sampled from a gaussian distribution. We can change the variance to increase the heterogeneity of the population. With a variance null, all the population has the same parameters. The second way is to change the knowledge graph that changes the dependencies between the different knowledge. This type of disruption can be small like adding or removing a dependency, or it can be as critical as rearranging completely the organization of the knowledge dependencies. These two types of disruption are combined in our experiments.



**Figure 2: Knowledge graphs used in the simulations.**  $LM_0$  is the nominal knowledge graph, with  $LM_1$  and  $LM_2$  introducing small disruptions in the pre-requirements between KCs.  $LM_3$  and  $LM_4$  represent more critical disruptions that change the overall order of KCs.

We used multiple knowledge graphs, shown in Fig.2. The arrows represent the dependencies between KCs. For example,  $LM_0$  represents a graph where the constraints between the different KC are ordered in a linear way. Here,  $\beta_{1,1} = \beta_{2,1} = \beta_{3,2} = \beta_{4,3} = \beta_{5,4} = \beta_{6,5} = 0.2$  and all the others values of  $\beta_{i,j}$  are null. We then created several different transformations and variants to model different needs of the students in terms of the order of the different KC.

$LM_1$  and  $LM_2$  follow approximately the same overall sequence of KC, but considering two initial branches for the different KC.  $LM_1$  considers that  $KC_1$  and  $KC_2$  are independent and any of them allows to learn  $KC_3$ . In these knowledge graphs, we can expect that optimizing for one will also work for the other as the overall sequence of KC is respected, even if the strategy is no longer optimal. We also created more critical disruptions in the knowledge graph.  $LM_3$  and  $LM_4$  present an inversion between two KCs. For  $LM_3$ ,  $KC_4$  and  $KC_5$  are inverted, what radically change the overall sequence of KCs. For  $LM_4$ , it is  $K_3$  and  $K_4$  that are inverted.

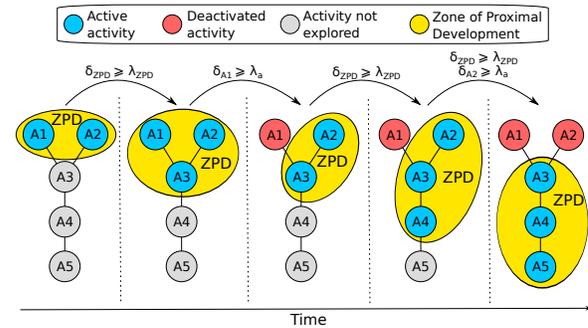
## 4. OPTIMIZING LEARNING POLICIES

### 4.1 Partially Observed Markov Decision Process (POMDP)

POMDP is a markovian decision process where the state is hidden and can only be inferred indirectly from the observations. A POMDP consists of a tuple  $\langle S, A, Z, T, R, O, \gamma \rangle$  with  $S$  the state space,  $A$  the action space and  $Z$  the observation space.  $T$  is the transition model, it gives the probabilities  $p(s'|s, a)$  of transitioning from state  $s$  to state  $s'$  with the action  $a$ .  $O$  is the observation model, it gives the probabilities  $p(z|s, a)$  of having the observation  $z$  when action  $a$  is made in state  $s$ .  $R$  the cost model, it specifies the cost  $r(s, a)$  of choosing action  $a$  in state  $s$ , and the discount factor  $\gamma$  gives the relation between immediate costs and delayed costs. With all these components, the solution of a POMDP is a policy that optimizes total discounted future reward.

This framework has been already used in the context of ITS [13]. The learner's mastery is the hidden state  $s$ , learning is the transition between states, the probabilities that the learner gives a good answer are given by the observation model of the observation {correct, incorrect}. We use Perseus [14] as solver to find the optimal policy for our POMDP problem.

### 4.2 Zone of Proximal Development and Empirical Success (ZPDES)



**Figure 3: ZPDES exploration of an activity graph, with  $\delta_{ZPD}$  the success rate over all active activities,  $\lambda_{ZPD}$  the threshold to expand the ZPD,  $\delta_{Ax}$  the success rate for the activity  $Ax$ , and  $\lambda_a$  the threshold to reach to deactivate an activity.**

Here we present the recently introduced algorithm Zone of Proximal Development and Empirical Success (ZPDES) that is based on multi-armed bandits [6]. The idea of the algorithm is presented in Fig.3 and summarized in Alg.1. The algorithm follows an activity graph but goes through it in a stochastic way. ZPDES is initialized with a certain number of activities defined as starting activities. At each point in time, ZPDES has a set of activities, called the zone of proximal development, that can be proposed to the student which is adapted depending on student result. In the experiments presented here, we make small changes in the activation/deactivation mechanism of the original algorithm. When the recent student success rate over all active activi-

ties  $\delta_{ZPD}$  reaches a value  $\lambda_{ZPD}$ , the graph is expanded to explore another activity and when the recent success rate for a particular activity  $\delta_{a_i}$  is higher than a threshold  $\lambda_a$ , this activity can be removed from the active list. This two threshold allow to partially configure the exploration behaviour of the algorithm. Inside the set of active activities, ZPDES proposes exercises proportionally to the recent learning progress obtained by that activity. The activity graph following the same structure than the knowledge graph, we can directly configure ZPDES with the same knowledge graph used to configure POMDP.

---

**Algorithm 1** ZPDES algorithm

---

**Require:** Set of  $n_a$  activities  $A$   
**Require:**  $\zeta$  rate of exploration  
**Require:** distribution for parameter exploration  $\xi_u$

- 1: Initialize of quality  $w_a$  uniformly
- 2: **while** learning **do**
- 3:   Initialize ZPD
- 4:   {Generate exercise:}
- 5:   **for**  $a \in ZPD$  **do**
- 6:      $\tilde{w}_a = \frac{w_a}{\sum_j w_j}$
- 7:      $p_a = \tilde{w}_a(1 - \zeta) + \zeta\xi_u$
- 8:     Sample  $a$  proportional to  $p_a$
- 9:   **end for**
- 10:   Propose activity  $a$
- 11:   Get student answer  $C_t$  and compute reward:
- 12:    $r = \sum_{k=t-d/2}^t \frac{C_k}{d/2} - \sum_{k=t-d}^{t-d/2} \frac{C_k}{d-d/2}$
- 13:    $w_a \leftarrow \beta w_a + \eta r$  {Update quality of activity}
- 14:   Update ZPD based on activity graph and success rates
- 15: **end while**

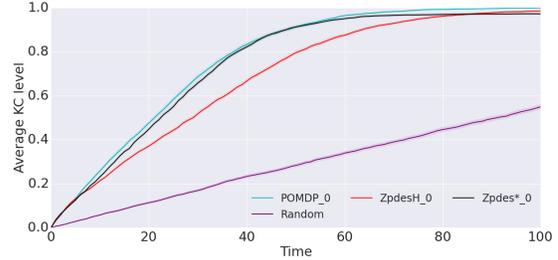
---

## 5. EXPERIMENTS

The goal of our experiments is to compare the impact of the knowledge about the students on the online algorithms for choosing exercises, namely POMDP and ZPDES. We will proceed to change the heterogeneity of the student populations and see how much disruption each algorithm is able to adapt. Our comparative measure of performance is the average skill level overall knowledge and over time, for all the students in the population.

We will compare the results obtained with two algorithms: POMDP and ZPDES. Each algorithm will have different variants based on the knowledge included on each of them. POMDP relies on a knowledge graph and the parameters of such graph. Each variant of  $POMDP_x$  is characterized by a specific student model used to find the optimal policy. ZPDES has as information the knowledge graph, and some parameters describing how to traverse this graph, no particular assumption is made about the probabilities of knowledge acquisition.  $ZPDES_x^H$  is a variant of ZPDES with the corresponding graph  $x$  and using the parameters that were used in an other experiment in a real world situation [6] mostly hand-tuned with the help of a pedagogical expert.  $ZPDES_x^*$  will also use the graph  $x$  but the parameters to traverse the graph are optimized for that particular graph using a greed search. During the optimization, we saw that the majority of parameters present average results and only extreme parameters gave critical results.

**Single model results.** The first experiment will do a sanity check to evaluate each algorithm in conditions where each student is the same in the population and each algorithm is configured for this model of student. We expect POMDP to have the best results and we want to see how far ZPDES will be from the optimal solution. A Random strategy which selects one activity randomly among all possible is also presented in this first experiment to see the gain of the algorithms.



**Figure 4:** Evolution of the average skill level for 600 students modeled with  $LM 0$  which activity are managed by POMDP, ZPDES\*, ZPDES<sup>H</sup> configured for  $LM 0$ . Shaded area represents the standard error of the mean.

Fig.4 shows the comparison of POMDP, ZPDES\*, ZPDES<sup>H</sup> and Random with a population of 600 students modelled with the knowledge graphs  $LM 0$ . We can see POMDP is the best for all the models, closely followed by ZPDES\*. ZPDES<sup>H</sup> give a slower learning than the two others. Unsurprisingly, for one particular model, POMDP has the best performance. The optimized ZPDES is very close in performance to POMDP. The results are similar for models 1, 2, 3 and 4, the curves are not presented here for space reason. We can thus verify that the combination of knowledge graphs and the activity exploration rules provides a space of policies that is close to the optimal POMDP one. ZPDES<sup>H</sup> present the slowest population learning among the algorithms but as its configuration was not optimized for any particular model we can expect such result.

These results show that the algorithms behave as expected and that ZPDES has the potential to be close to the optimal POMDP solution.

**Multi model results.** We will now present the main results of this work with the comparison between POMDP, ZPDES\* and ZPDES<sup>H</sup> when confronted with heterogeneous populations of students. The protocol of the experiments is as follows. First we provide each algorithm with the information about a specific population of students and then we test the capability of the algorithms to address a different and diverse population of students. As described earlier, each algorithm is given information about a particular student model  $x$ , POMDP <sub>$x$</sub>  receives the graph and the student model parameters, ZPDES\* <sub>$x$</sub>  receives the graph and exploration parameters optimized for that same graph, ZPDES<sup>H</sup> <sub>$x$</sub>  receives the graph and standard parameters for the graph exploration. We test different versions of each algorithm with a population composed of students following 3 differ-

**Table 1: Performance position of each algorithm configuration for each setup. The rank of each algorithm configuration, and the average rank of each algorithm is presented for steps 50 and 200.**

Students 0,1,2 / Alg config 0,1,2				
Algorithm	Rank t 50		Rank t 200	
	Per conf	Average	Per conf	Average
POMDP <sub>0</sub>	1		1	
POMDP <sub>1</sub>	3	1	2	2
POMDP <sub>2</sub>	4		3	
ZPDES <sub>0</sub> <sup>H</sup>	3		1	
ZPDES <sub>1</sub> <sup>H</sup>	3	3	1	1
ZPDES <sub>2</sub> <sup>H</sup>	6		3	
ZPDES <sub>0</sub> <sup>*</sup>	2		1	
ZPDES <sub>1</sub> <sup>*</sup>	3	2	2	2
ZPDES <sub>2</sub> <sup>*</sup>	5		3	

Students 0,3,4 / Alg config 0,3,4				
Algorithm	Rank t 50		Rank t 200	
	Per conf	Average	Per conf	Average
POMDP <sub>0</sub>	1		2	
POMDP <sub>3</sub>	2	1	3	2
POMDP <sub>4</sub>	4		5	
ZPDES <sub>0</sub> <sup>H</sup>	2		1	
ZPDES <sub>3</sub> <sup>H</sup>	3	2	2	1
ZPDES <sub>4</sub> <sup>H</sup>	4		3	
ZPDES <sub>0</sub> <sup>*</sup>	2		2	
ZPDES <sub>3</sub> <sup>*</sup>	3	2	4	2
ZPDES <sub>4</sub> <sup>*</sup>	4		4	

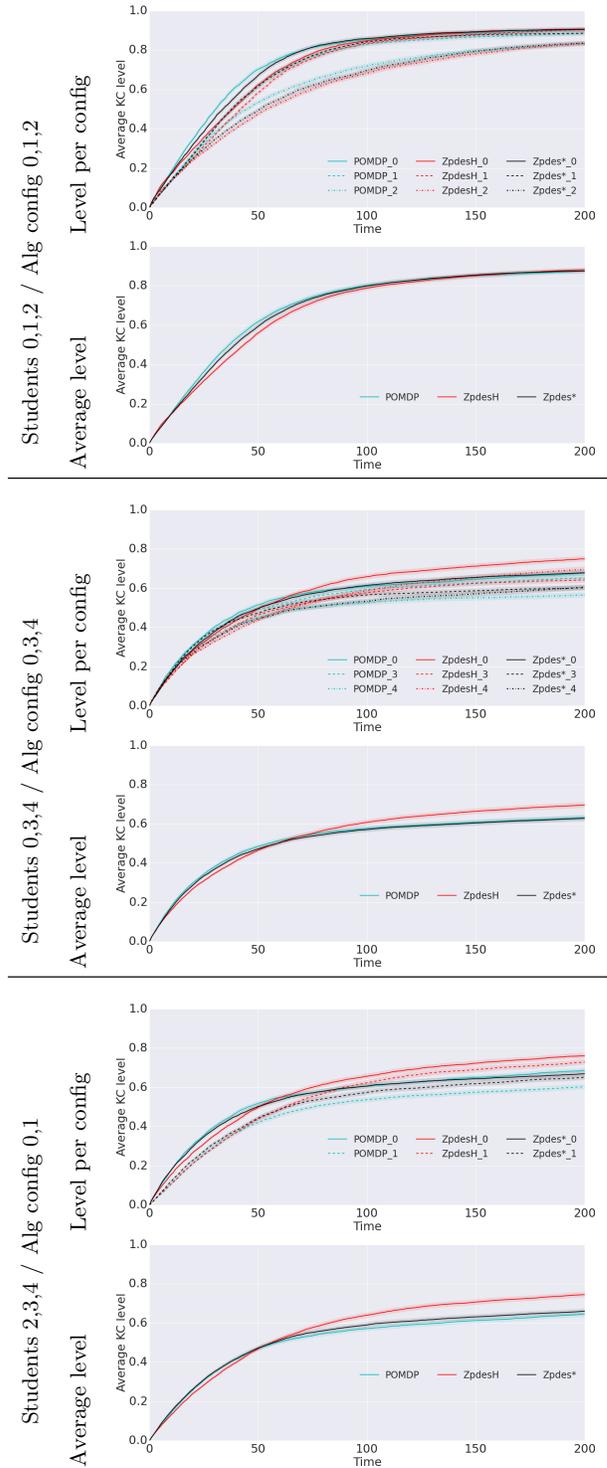
  

Students 2,3,4 / Alg config 0,1				
Algorithm	Rank t 50		Rank t 200	
	Per conf	Average	Per conf	Average
POMDP <sub>0</sub>	1	1	3	2
POMDP <sub>1</sub>	4		6	
ZPDES <sub>0</sub> <sup>H</sup>	2	1	1	1
ZPDES <sub>1</sub> <sup>H</sup>	3		2	
ZPDES <sub>0</sub> <sup>*</sup>	2	1	4	2
ZPDES <sub>1</sub> <sup>*</sup>	3		5	

ent knowledge graphs. The probabilistic parameters of the student models in the population follow a gaussian distribution. There is 200 students per graphs for a total of 600 students.

On figure 5 we can see the evolution of the average mastery level for all KCs. The table 1 presents the ranking of each version of the algorithms and the average ranking of each algorithm at step 50 and 200 according to the curves comparison for each setup  $LM_{0,1,2}$ ,  $LM_{0,3,4}$ , and  $LM_{2,3,4}$ . The table 2 presents the statistical significance tests at step 50 and 200 for each setup and what is the best methods if the results are statistically significant.

By comparing the different p-values, we can see that the differences between POMDP and ZPDES\* are never significant, but it's not the case for ZPDES<sup>H</sup>. For the models  $LM_{0,1,2}$ , at step 50, ZPDES<sup>H</sup> drops behind the two others, but it catches up rapidly with the two others and present the same results at step 200. So for models which are close to each other, the 3 algorithms present almost the same result.



**Figure 5: Evolution of the average skill level for 600 students with POMDP, ZPDES\*, ZPDES<sup>H</sup>. For each curve, the number attached to the algorithm's name indicate what knowledge graph has been used to configure the algorithm. Each curve shows the average KC level of the student population over time for each algorithm configuration. In general ZPDES have better results than POMDP. Shaded area represents the standard error of the mean.**

**Table 2: ANOVA p-values for each setup to verify if the differences in the KC level distribution according to each algorithm are statistically significant with the best algorithms in parenthesis when it is significant. We note P for POMDP and Z for ZPDES**

LM	P/Z*		P/Z <sup>H</sup>		Z*/Z <sup>H</sup>	
	t 50	t 200	t 50	t 200	t 50	t 200
0,1,2	.075	.95	<b>10<sup>-6</sup></b> (P)	.82	<b>.003</b> (Z*)	.87
0,3,4	.24	.90	.17	<b>10<sup>-5</sup></b> (Z <sup>H</sup> )	.89	<b>10<sup>-4</sup></b> (Z <sup>H</sup> )
2,3,4	.31	.30	.18	<b>10<sup>-5</sup></b> (Z <sup>H</sup> )	.77	<b>10<sup>-7</sup></b> (Z <sup>H</sup> )

For the models  $LM_{0,3,4}$ , observations are different. At step 50, all the algorithms seem to have approximately the same performance, even if ZPDES<sup>H</sup> seems a bit behind but it's not significant (p-values at 0.17 and 0.89). But with time, it takes the lead and achieves the best performance at 200 steps. So when there are two models critically different from another, ZPDES<sup>H</sup> presents the best results. For the last case, the population is constituted of students following  $LM_{2,3,4}$  models, and the algorithms are configured for models  $LM_{0,1}$ . As for the previous case there is no differences at step 50 but ZPDES<sup>H</sup> presents the best results at step 200.

ZPDES<sup>H</sup> provides the best result because its exploration parameters were not optimized for any particular knowledge graph, giving it higher adaptability and less constrains in the exploration. For a particular type of student model it will present worse performance than POMDP or ZPDES\*, but for a heterogeneous population, ZPDES<sup>H</sup>, being more adaptable, has the best performance.

## 6. CONCLUSION

In this work we considered student models where the knowledge components can have constraints among each other, allowing to model some kind of pre-requisites. Under different student models we can find an optimal teaching sequence using POMDP. Another alternative is the use of the recently proposed method ZPDES that is computationally more efficient but without optimality guarantees. Our goal was to test how robust each of these methods is in relation with ill-estimated parameters of the models, or even wrongly estimated relations between KCs. This corresponds to the more realistic case of heterogeneous classes of students.

We showed that for the trivial situation where the students are perfectly modeled with the student model, ZPDES can achieve the same performance as the POMDP. For heterogeneous populations again ZPDES can achieve solutions similar to POMDP. The best algorithm was using ZPDES that uses parameters that are not optimized for no population in particular. By having more flexibility in the exploration it becomes more robust to changes in the population.

We conclude that multi-armed bandits, when combined with an activity graph, are a best choice in comparison with POMDPs due to its computational efficiency and reliance on simpler student models.

The code to generate the graphics and the results is available at: [github.com/flowersteam/kidlearn/tree/edm2016](https://github.com/flowersteam/kidlearn/tree/edm2016), follow the README.

## 7. REFERENCES

- [1] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.
- [2] Ryan SJ Baker, Albert T Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415, 2008.
- [3] Joseph E Beck and Kai-min Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*. 2007.
- [4] Joseph E Beck and Xiaolu Xiong. Limits to accuracy: How well can we do at student modeling? In *Educational Data Mining*, 2013.
- [5] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis—a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, 2006.
- [6] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Multi-Armed Bandits for Intelligent Tutoring Systems. *Journal of Educational Data Mining (JEDM)*, 7(2):20–48, June 2015.
- [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] José P González-Brenes and Jack Mostow. Dynamic cognitive tracing: Towards unified discovery of student and cognitive models. In *EDM*, pages 49–56, 2012.
- [9] JP González-Brenes, Yun Huang, and Peter Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Inter. Conf. on Educational Data Mining*, 2014.
- [10] K.R. Koedinger, J.R. Anderson, W.H. Hadley, M.A. Mark, et al. Intelligent tutoring goes to school in the big city. *Inter. Journal of Artificial Intelligence in Education (IJAIED)*, 8:30–43, 1997.
- [11] J.I. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Inter. Conf. on Educational Data Mining (EDM)*, 2012.
- [12] Kai min Chang, Joseph Beck, Jack Mostow, and Albert Corbett. A bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent Tutoring Systems*, 2006.
- [13] A. Rafferty, E. Brunskill, T. Griffiths, and P. Shafto. Faster teaching by pomdp planning. In *Artificial Intelligence in Education*, pages 280–287, 2011.
- [14] Matthijs T. J. Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.
- [15] Michael Villano. Probabilistic student models: Bayesian belief networks and knowledge space theory. In *Intelligent Tutoring Systems (ITS'92)*, 1992.

# Automatic Assessment of Constructed Response Data in a Chemistry Tutor

Scott Crossley  
Kristopher Kyle  
Georgia State University  
Atlanta, GA 30303  
scrossley@gsu.edu  
kkyle@student.gsu.edu

Jodi Davenport  
WestEd  
San Francisco, CA 94107  
jdavenport@wested.org

Danielle S. McNamara  
Arizona State Univ.  
Tempe, AZ, 85287  
dsmcnama@asu.edu

## ABSTRACT

This study introduces the Constructed Response Analysis Tool (CRAT), a freely available tool to automatically assess student responses in online tutoring systems. The study tests CRAT on a dataset of chemistry responses collected in the ChemVLab+. The findings indicate that CRAT can differentiate and classify student responses based on semantic overlap with student input and indices related to word frequency, text content, and lexical sophistication. Overall, the findings suggest that more accurate student responses show greater overlap with the content learned, include more academic function words, contain greater content that is descriptive, and includes more specific and familiar words.

## Keywords

Natural language processing, on-line tutors, constructed response scoring

## 1. INTRODUCTION

For science education to be more effective, students should move beyond memorizing facts and procedures and toward gaining deeper conceptual understanding that allows them to both apply scientific knowledge to explain new phenomena and to design investigations. The Next Generation Science Standards [1], offer a new vision of science instruction that integrates science practices, disciplinary core ideas, and cross cutting concepts, such as scale, energy, and patterns that unify different fields. However, assessing learning of these interconnected strands is challenging using traditional, multiple-choice items. Constructed responses, as well as more novel types of assessments provide students with important opportunities to demonstrate reasoning, explanation, and inquiry skills and are thus an important educational tool [2].

One problem with constructed responses are associated scoring costs [3]. A possible solution to these costs can be found in automated scoring tools that can reduce the need for human scoring and potentially increase scoring consistency [4]. In this study, we introduce a freely available natural language processing (NLP) tool called the Constructed Response Analysis Tool (CRAT) that can automatically score constructed responses in domain specific learning environments. We conduct a pilot study that tests the efficacy of CRAT to score student responses to a

domain specific question in an on-line chemistry tutoring system by comparing scoring models developed by CRAT to human ratings of constructed responses.

### 1.1 Assessing student understanding

Simulations and games provide rich environments for students to learn science and demonstrate their understanding of scientific principles [5]. Such games and simulations can be included in online systems that allow for just-in-time feedback. The dynamic feedback found in online systems affords students the opportunity to confront misconceptions and provides information about areas of struggle or mastery that teachers can use as formative assessments that influence instructional decision making. However, the utility of feedback depends on the ability of an online system to provide an accurate diagnosis of student understanding. Though multiple choice and student behaviors in simulation environments may be readily scored using constraint-based model tutors [6], interpreting and accurately scoring constructed responses in science education has proven much more challenging [2]. These challenges have led researchers to develop content-based automated scoring systems that demonstrate medium to high agreement with human scores. These systems show promise for a number of domains (e.g., math, reading, psychology, biology) and a number of student levels (i.e., middle school, high school, college) [7, 8, 9].

### 1.2 Current Study

The goal of this study is to introduce CRAT and examine its potential to automatically assign accuracy scores to student constructed responses from an on-line tutor. Constructed responses were collected in the ChemVLab+ tutoring system (chemvlab.org) and scored by expert raters. We used the Constructed Response Analysis Tool (CRAT) to calculate linguistic features related to text content, text summarization, and lexical sophistication and used these linguistic features to predict the human scores.

## 2. METHOD

### 2.1 ChemVLab+

The ChemVLab+ is an on-line tutoring system that provides students with opportunities to apply chemistry knowledge to meaningful contexts and to receive immediate, individualized tutoring. Of interest in the current study are the four stoichiometry activities contained within ChemVLab+. The activities engage students in a variety of problem-solving tasks using interactive simulations including a virtual chemistry lab. At the end of each activity, students respond to one to three open-ended questions (i.e., constructed responses) designed to evaluate their ability to synthesize the information they had learned. The four stoichiometry activities included a total of 10 questions.

## 2.2 Participants

A total of 1392 high school chemistry students from the classes of thirteen teachers in the California bay area used the Stoichiometry module. Students used the online activities as part of their normal coursework.

## 2.3 Human Scores of Constructed Responses

All constructed responses were coded by two independent raters familiar with the chemistry content. Coders used an annotated rubric that described criteria for each score and provided examples of responses receiving those scores. Reliability of scoring varied across the questions, and interrater reliability ranged from Cohen's  $\kappa = 0.55$  to .92. Each question had three possible scores, except for the two lowest reliability questions, (items 1 and 2.1), which had four possible scores. When the highest two scores in these questions were collapsed, interrater reliability increased from 0.56 to 0.68 for item 1 and from 0.59 to 0.69 for item 2.2.

## 2.4 Selection of Constructed Responses

We selected student constructed responses from question 1 in the stoichiometry lab to test CRAT. The question had the greatest number of student answers ( $n = 1374$ ). The question asked students to explain the relationship between the amount of sugar, the volume of the drink, and concentration of the sports drink.

## 2.5 CRAT

CRAT is an easy to use constructed response analysis engine that calculates indices related to a) the linguistic and semantic similarities between a source text and a constructed response, b) the linguistic sophistication of a constructed response, and c) text properties (e.g., length and syntactic categories). It is freely available, cross-platform, and is accessed via a graphic user interface (GUI). The similarity indices include lexical similarity calculated using key word overlap, synonym overlap, and latent semantic analysis (LSA) similarity [10] and phrasal similarity calculated using key bigram and trigram overlap and key part of speech sensitive slot-grams (e.g., a trigram with an open slot such as *into the \_\_\_\_*). The constructed response sophistication indices include psycholinguistic word information indices (e.g., concreteness and familiarity [11, 12]), lexical frequency and range (words that occur in a wider range of texts) indices based on the British National corpus (BNC [13]) and the Corpus of Contemporary American English (COCA [14]), and syntactic categories (e.g., number of adjectives and nouns). For COCA, CRAT reports on frequency and range indices for a number of different genres including academic, newspaper, and fiction genres. Selected index features are outlined below. See <http://www.soletlab.com> to download the tool and to access the complete list of indices.

### 2.5.1 Function and content word only indices

CRAT indices generally consider all words in a text. CRAT also includes index variants that include only the content words (e.g., nouns, verbs, adjectives, adverbs) and only the function words (e.g., determiners, prepositions, etc.). Content word indices and function word indices are designed to provide more fine-grained analyses, and have been shown to be more predictive, in some cases, than when all words are considered in an index [15].

### 2.5.2 Text and sentence minimum indices

CRAT indices generally comprise the average score for all instances of a feature across an entire text. Additionally, CRAT calculates index variants that comprise average minimum scores

for each sentence in a text in order to assess smaller texts that may be a single sentence in length.

### 2.5.3 Key word exclusion indices

In addition to the index variants outlined above, constructed response sophistication indices include variants that exclude words that occur more frequently in the source text than would be expected (i.e., words that are "key"). The key word exclusion index variants were included to minimize interference from sophisticated language in the source text on the constructed response produced.

### 2.5.4 Latent Semantic Analysis Weighting

One variable that can affect LSA similarity scores is the weighting scheme employed. CRAT includes LSA variants calculated from the TASA corpus using normalized weighting, rare words dominated weighting, and frequent words dominated weighting. Normalized weighting considers all words in a reference corpus equally. Rare words dominated weighting assign higher scores to words that occur infrequently in the reference corpus. Frequent words dominated weighting assigns higher scores to words that frequently occur in the reference corpus [16].

## 2.6 Summary Input

CRAT is a domain specific tool and uses system input (i.e., source texts) to develop knowledge spaces for the domain of interest. The source texts used to develop knowledge spaces can be textbooks, lecture notes, presentations, or any type of text that generalizes expected knowledge on the part of the student. For this analysis, we used the hints provided to the students during specific activities within the ChemVLab+ system. These hints provide an overview of the input the student received and are designed to provide informational hints to students if they are unable to generate the information individually. The hints available to students in question 1 of the stoichiometry lab comprised over 5,000 words and focused specifically on the relationship between sugar, volume, and concentration in a sports drink.

## 2.7 Statistical Analysis

The indices reported by CRAT that yielded non-normal distributions were removed. A multivariate analysis of variance (MANOVA) was conducted to examine which indices reported differences between the three levels of scores for each student response (incomplete or incorrect, partially correct, and correct responses). The MANOVA was followed by stepwise discriminant function analysis (DFA) using the selected normally distributed indices from CRAT that demonstrated significant differences between responses that were incorrect or incomplete, partially correct, and correct and did not exhibit multicollinearity ( $r > .90$ ) with other CRAT indices. In the case of multicollinearity between indices, the index demonstrating the largest effect size was retained in the analysis. The DFA was used to develop an algorithm to predict group membership through a discriminant function co-efficient. A DFA model was first developed for the entire corpus of constructed responses. This model was then used to predict group membership of the constructed responses using leave-one-out-cross-validation (LOOCV) in order to ensure that the model was stable across the dataset.

## 3. RESULTS

### 3.1 MANOVA

A MANOVA was conducted using the NLP indices calculated by CRAT as the dependent variables and the human scores of the student responses as the independent variables. Of the 759 indices

**Table 1: Descriptive statistics and MANOVA results for CRAT variables**

Index	Incomplete/incorrect Mean (SD)	Partially correct Mean (SD)	Correct Mean (SD)	<i>F</i>	$\eta^2$
Semantic similarity (LSA) response and input (rare word dominated)	0.362 (0.159)	0.458 (0.111)	0.499 (0.079)	102.799**	0.131
Semantic similarity (LSA) response and input (frequent word dominated)	0.403 (0.155)	0.5 (0.113)	0.531 (0.096)	95.432**	0.122
Academic frequency COCA function words	24524.248 (16585.406)	36788.308 (13168.904)	34324.442 (11401.743)	76.716**	0.101
Written frequency (BNC) function words	1.000 (0.441)	1.227 (0.291)	1.25 (0.256)	53.237**	0.072
Percentage of adjectives	0.086 (0.082)	0.112 (0.069)	0.135 (0.074)	38.42**	0.053
Academic range (COCA) all words	-0.494 (0.254)	-0.401 (0.114)	-0.411 (0.096)	24.093**	0.034
Number of words	24.417 (29.134)	33.476 (53.923)	38.618 (39.975)	16.736**	0.024
Range (SUBTLEXus) content words (no key words)	3737.317 (1693.106)	3227.84 (1437.09)	3213.191 (1105.223)	15.819**	0.023
Academic frequency (COCA) content words sentence minimum	0.743 (0.705)	0.941 (0.532)	0.922 (0.487)	12.386**	0.018
Word familiarity (MRC) sentence minimum	497.207 (206.379)	560.031 (126.208)	529.915 (165.372)	10.534**	0.015
Percent content words	0.635 (0.147)	0.597 (0.085)	0.606 (0.091)	9.621**	0.014
Word familiarity (MRC) content words (no key words)	465.777 (132.451)	483.335 (87.545)	495.526 (77.668)	6.393*	0.009
Range (COCA all words sentence minimum)	-1.937 (0.143)	-1.96 (0.083)	-1.956 (0.08)	4.063*	0.006
Academic range (COCA; no key words)	0.712 (0.081)	0.693 (0.076)	0.689 (0.137)	3.865*	0.006

\*  $p < .05$ , \*\*  $p < .001$

**Table 2. Confusion matrix for DFA results for classifying scored responses**

		Incomplete/incorrect	Partially correct	Correct	$F_1$ score
Whole set	Incomplete/incorrect	<b>605</b>	202	138	0.755
	Partially correct	31	<b>119</b>	60	0.400
	Correct	21	67	<b>129</b>	0.474
		Incomplete/incorrect	Partially correct	Correct	$F_1$ score
LOOCV	Incomplete/incorrect	<b>603</b>	203	139	0.752
	Partially correct	33	<b>113</b>	64	0.379
	Correct	22	70	<b>125</b>	0.459

reported by CRAT, 96 of these indices were normally distributed and not multi-collinear with one another. Of these 96 indices, 85 of the indices reported significant differences in the MANOVA analysis. These indices were related to overlap between the constructed response and the input received in the tutor, lexical sophistication, response length, response descriptiveness, and percentage of content words in the response. These indices were used in the subsequent DFA.

### 3.2 Discriminant Function Analysis

A stepwise DFA using the 85 indices selected through the MANOVA retained 14 variables related to semantic overlap between response and input, text descriptiveness, lexical sophistication, response length, and the use of content words. The indices retained in the DFA along with their means, standard deviations, *F* scores, *p* values, and effect sizes are reported in Table 1.

The results demonstrate that the DFA using these 14 indices correctly allocated 853 of the 1372 student responses in the total set,  $\chi^2$  (df=4) = 393.169  $p < .001$ , for an accuracy of 62.2%. For the leave-one-out cross-validation (LOOCV), the discriminant analysis allocated 841 of the 1372 texts for an accuracy of

61.3% (see the confusion matrix reported in Table 2 for results and  $F_1$  scores). The Cohen's Kappa measure of agreement between the predicted and actual class label was 0.404, demonstrating moderate agreement.

## 4. DISCUSSION

This analysis provides an initial assessment of the extent to which the linguistic indices reported by the Constructed Response Analysis Tool (CRAT) are predictive of constructed responses. We examined student constructed responses to a single question in the ChemVLab+ system related to stoichiometry. We found that 86 CRAT indices demonstrated differences between the three levels of human ratings (incomplete/incorrect, partially correct, and correct) and 14 of these variables were significant predictors of human scores in a DFA with a reported accuracy of 62%. The results suggest that the CRAT tool can be used to automatically classify student constructed responses based on human ratings of response accuracy. While preliminary, the results support the use of NLP tools in constructed response scoring and point toward specific linguistic features that can be used to predict human ratings of accuracy for student constructed responses.

The discriminant function analysis indicated that the strongest predictors of human accuracy scores were related to semantic similarity between the constructed response and the knowledge space provided (i.e., the available student hints in the ChemVLab+). The results indicated that student responses that had a higher semantic overlap with the hints were more likely to be correct or partially correct. These results held for rare word and frequent word LSA overlap. This suggests that students whose responses better represent the semantic space of the domain are more likely to produce correct responses.

Beyond semantic overlap with the hints, the next strongest predictors of human scores of student responses were related to the frequency of function words. These indices indicated that students who used more frequent function words were rated as having higher response scores (for both academic and written frequency). This likely indicates that students who used function words that occur more frequently in written contexts (i.e., academic writing and writing in general) construct more accurate responses. Thus, more successful students were those who were more likely to use writing styles frequent in academic English.

More successful answers also differed in the properties of the words they contained. More accurate answers were more descriptive in that they contained a greater number of adjectives. Though longer, successful answers contained fewer content words (i.e., they contained more function words). Successful answers contained more specific words (i.e., words that demonstrated a lower range score) and also contained more familiar and frequent words.

The model developed in this pilot study reports a level of accuracy that is appropriate to provide automated feedback to users in a tutoring system such as ChemVLab+. This feedback could include a summative score to provide users with an overall assessment of the quality of the constructed response. In addition, the model could be used to provide formative feedback to users in terms of language use (i.e., the use of academic language) and appropriate content (i.e., is writer covering the content of the question appropriately). Such feedback could be used by students to revise their responses and engage more deeply with the system. However, we would caution against using the reported model in high stakes assessments where accuracy is at a premium, although this advice should be empirically tested on a number of high stakes test corpora.

CRAT differs from many other scoring systems in that it is domain specific. Domain specificity has advantages as many of the key word and semantic indices can be trained on targeted content that increases construct validity and ensures that topic adherence on the part of the student remains an important component of constructed response scoring. Training the system, however, requires source texts that provide background about the topic. In some cases, these texts may be difficult to transfer to text files (in the case of lectures) or they may not exist within a system, limiting the generalizability of CRAT across a number of system.

Lastly, it remains an open question if a model trained on one area of chemistry will transfer to another area of chemistry or to domains outside of chemistry. For instance, the model developed here needs to be tested on similar but not overlapping chemistry topics and questions to test the model's generalizability within a macro-domain (e.g., with chemistry questions that address molecular equilibrium and acid bases). In

addition, the model should be tested on domains outside of chemistry to assess whether constructed responses in various domains can be accurately scored based on a combination of semantic and keyword overlap between the response and the source and the use of academic language by system users.

## 5. CONCLUSION

This study introduces a freely available tool for constructed response scoring and tests the tool on a dataset of chemistry responses collected in the ChemVLab+. The findings indicate that the Constructed Response Analysis Tool (CRAT) can differentiate and classify student responses based on semantic overlap with text input, syntactic categories, text length, and lexical sophistication indices. Overall, the findings suggest that successful student responses contain greater overlap with the content learned and use more academic function words, more words in general, more descriptive words, and more familiar and frequent words that are also more specific.

Additional studies will be conducted to refine and continue to develop CRAT. For example, a future direction includes assessing the value of including indices of semantic overlap that use Latent Dirichlet allocation (LDA) spaces, allowing for topic modeling along with semantic graph analyses. CRAT also needs to be tested on additional constructed responses, including responses from a variety of domains. Lastly, the models developed using the CRAT tool should be assessed for application in providing feedback to users in instructional systems. Such follow up studies will provide additional information about the reliability of CRAT and the linguistic features within CRAT that are predictive of human ratings of constructed responses within different domains and on-line learning environments.

## 6. ACKNOWLEDGMENTS

This research was supported in part by the Institute for Education Sciences and National Science Foundation (IES R305A080589, IES R305A100069, IES R305G20018-02, DRL-1418072, and DRL-1418378). Ideas expressed in this material are those of the authors and do not necessarily reflect the views of the IES or the NSF.

## 7. REFERENCES

- [1] NGSS Lead States. 2013. *Next Generation Science Standards: For States, By States*. Washington, DC: The National Academies Press.
- [2] Liu, O. L., Brew, C., Blackmore, J., Gerard, L., Madhok, J., & Linn, M. C. (2014). Automated Scoring of Constructed-Response Science Items: Prospects and Obstacles. *Educational Measurement: Issues and Practice*, 33(2), 19-28.
- [3] Wainer, H., & Thissen, D. (1993). Combining multiple-choice and constructed response test scores: Toward a Marxist theory of test construction. *Applied Measurement in Education*, 6, 103-118.
- [4] Williamson, D., Xi, X., & Breyer, J. (2012). A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, 31(1), 2-13.
- [5] Honey, M. A., & Hilton, M. (Eds.). (2010). *Learning science through computer games and simulations*. Washington, DC: National Academies Press.
- [6] Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are

- going. *User Modeling and User-Adapted Interaction*, 22(1-2), 39-72.
- [7] Attali, Y., & Powers, D. (2008). Effect of immediate feedback and revision on psychometric properties of open-ended GRE subject test items. GRE Board Research Rep. No. 04-05; ETS RR-08-21. Princeton, NJ: Educational Testing Service.
- [8] Bennett, R. E., & Sebrechts, M. M. (1996). The accuracy of expert-system diagnoses of mathematical problem solutions. *Applied Measurement in Education*, 9, 133–150.
- [9] Wang, H.-C., Chang, C.-Y., & Li, T.-Y. (2005). Automated scoring for creative problem-solving ability with ideation-explanation modeling. In *Proceedings of the Thirteenth International Conference on Computers in Education* (pp. 522–529). Singapore: IOS Press.
- [10] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3), 259-284.
- [11] Brysbaert, M., Warriner, A.B., & Kuperman, V. (2013). Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*. doi:10.3758/s13428-013-0403-5
- [12] Coltheart, M. (1981). The MRC psycholinguistic database. *Quarterly Journal of Experimental Psychology*, 33A, 497-505. doi:10.1080/14640748108400805
- [13] British National Corpus, version 3 (BNC XML ed.). (2007). Retrieved from <http://www.natcorp.ox.ac.uk>
- [14] Davies, M. (2010). The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and linguistic computing*, 25(4), 447-464.
- [15] Kyle, K. & Crossley, S. A. (2015). Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly* 49(4), pp. 757-786. doi: 10.1002/tesq.194
- [16] McNamara, D. S., Cai, Z., & Louwerson, M. M. (2007). Optimizing LSA measures of cohesion. *Handbook of latent semantic analysis*, 379-400.

# Choosing versus Receiving Feedback: The Impact of Feedback Valence on Learning in an Assessment Game

Maria Cutumisu  
University of Alberta

6-102 Education North, Edmonton, AB T6G 2G5  
(780) 492 5211; cutumisu@ualberta.ca

Daniel L. Schwartz  
Stanford University

485 Lasuen Mall, Stanford, CA 94305  
(650) 725 5480; danls@stanford.edu

## ABSTRACT

Studies examining feedback in educational settings have largely focused on feedback that is received, rather than chosen, by students. This study investigates whether adult participants learn more from choosing rather than receiving feedback from virtual characters in a digital poster design task. We employed a yoked study design and two versions of an online game-based assessment, Posterlet, to compare the learning outcomes of N=264 Mechanical Turk adults in two conditions: when they *chose* the feedback valence versus when they *received* the same feedback valence and order. In Posterlet, players design posters and learn graphic design principles from feedback. We found that the more the participants chose critical feedback, the more time they spent designing posters, but there were no differences in learning, revision, and time spent designing posters between conditions. In each condition, critical feedback correlated with performance and revision, suggesting that feedback valence is important for performance, regardless of being a choice.

## Keywords

feedback valence, choice, assessment, game, learning

## 1. INTRODUCTION

A central goal of education is to prepare independent learners [16]. Previously, we operationalized this goal by a) identifying promising behaviors for autonomous learning that would reveal how students learned and b) creating novel choice-based digital assessment games that measured these behaviors. For instance, we measured students' choices to seek critical feedback and to revise, and we found that students who were more willing to seek critical feedback also learned more [4]. We examine learning choices (e.g., seeking social feedback), because such learning strategies can support ongoing learning, adapting to new challenges, and, ultimately, learning *how* to learn. These types of design thinking competencies, together with collaboration, persistence, and creativity, are crucial for 21<sup>st</sup>-century challenges, yet they are not formally assessed in schools [1, 21]. There are two main reasons why we need to measure learning behaviors. First, learning behaviors or attitudes enable learners to solve problems even when they do not have the domain knowledge skills to do so (e.g., collaborate with a partner from a different discipline). Second, current self-assessment techniques are not gender neutral: even though women and men scored similarly on a science exam (they had similar skills), women underestimated while men overestimated their performance (their attitudes did not match their skills; [7]). Such self-regulated learning behaviors [10] are worth investigating because revised self-assessment interventions may increase female representation in science, technology, engineering, and mathematics and could help create gender-inclusive 21<sup>st</sup>-century learning and assessment environments.

We previously examined the feedback valence (i.e., critical versus confirmatory) and its impact on performance and learning. In this study we examine for the first time the effect of feedback agency (i.e., choosing versus receiving). Our objective is to investigate the effect of choosing versus receiving feedback on learning, by comparing learning outcomes between participants who choose feedback and those who receive the same amount, valence, and order of feedback. We outline related work and theoretical perspectives that guide our research. Then, we describe our assessment environment, Posterlet, an online game designed to collect and assess participants' feedback and revision choices. We also created and presented a modified version of this game to accommodate the situation in which feedback is assigned to the learner in a principled way that mirrors the feedback chosen in the original Posterlet version. We then present evidence of the impact of choosing *versus* receiving feedback on learning outcomes, as well as theoretical and practical implications of this research.

We examine the impact of feedback choice and valence on learning by posing the following research questions:

- 1) Does critical feedback correlate with learning outcomes?
- 2) Are there learning outcome differences between choosing and receiving feedback?
- 3) Are there design duration differences between choosing and receiving feedback?
- 4) Are there gender differences on the measures by condition?

## 2. RELATED WORK

We distinguish several themes in the literature related to the theoretical perspectives that guide this research.

**Choice-based Assessments.** Traditional assessments measure learners' knowledge at the end of instruction, focusing on knowledge accuracy but providing little information about learners' readiness to learn new things. Vygotsky highlighted the importance of measuring learning processes [23], rather than only learning outcomes, to achieve deeper insights into students' potential to learn on their own. Schwartz and Bransford advocated *preparation for future learning* (PFL) assessments [19], which create learning opportunities during the assessment. Our research draws from work on *constructivist assessments* [20] and *choice-based assessments* [18]. Both these assessments build upon PFL assessments and measure not only learners' knowledge outcomes but also their learning processes (e.g., choices about what, when, and how to learn). For example, Posterlet [4], an online game that collects players' choices to seek critical feedback and to revise while they design posters, constitutes an instance of a choice-based assessment. The design of Posterlet is guided by the three core principles of choice-based assessments: *typical performance* (assessments need to capture every-day learning behaviors, not

test performance), *PFL* (assessments need to offer learning opportunities with measurable outcomes; [2]), and *choice* (assessments need to collect free learning choices that do not hinder the learners' ability to complete the assessments). Specifically, Posterlet provides players with a 10-15 minute fun game experience, with a chance to learn graphic design principles and to safely explore choices to seek critical feedback and revise, before applying them in more high-stakes situations. Concomitantly, Posterlet provides researchers with a way to track players' behaviors and learning outcomes to infer how prepared players are to learn on their own in new learning situations.

**Confirmatory versus Critical Feedback.** In educational contexts, feedback is defined as information related to a person's performance or understanding [11] and it is predominantly assigned by a teacher or a computer rather than chosen by the learner. There are some exceptions, but they pertain to help seeking [17] rather than specifically to feedback seeking. Here, we are mainly interested to investigate whether being given a choice about how to learn (i.e., choosing versus receiving feedback) has any impact on learning outcomes and other learning behaviors. In addition to feedback choice, the feedback literature provides some indication of the importance of feedback valence. For instance, critical feedback yields mixed results for performance [13], but studies of organizations show that most new ideas need critical constructive feedback to become successful [15]. A first challenge is that feedback is often absent from ideation environments. A second challenge is that critical feedback is even more elusive in such environments and it runs the risk of ego threat that causes people to reject instead of heed the feedback [11]. This suggests that attitudes towards seeking critical feedback are worth exploring. However, there is no evidence that the choice of critical feedback is as important as simply assigning critical feedback to the learner. Thus, we designed a variation of Posterlet and we employed a reduced-length game version for comparison to address this issue.

**Choosing versus Receiving Feedback.** Traditionally, most studies focused on supervised feedback, where the teacher assigned feedback to the student. However, in many situations, people need to actively seek feedback. Little is known about the implications of students' feedback choices on their learning or about variables that influence students' feedback choices, but researchers acknowledge the importance of the mechanisms underlying feedback for learning. For instance, Zimmerman [24] included "responsiveness to self-oriented feedback" among three critical features of students' self-regulated learning strategies. The effect of actively choosing rather than passively receiving critical feedback for learning raises interesting psychological questions. For example, patients who had control over their level of pain medication chose lower doses than those prescribed by medical staff [12]. Similarly, having a choice over critical feedback may act as a buffer against ego threat. Further, if learners are assigned critical feedback, would that lead to less learning than if they chose it? Consumer research provides corroborating evidence directly relevant to our prior research regarding the choice between confirmatory and critical feedback. Researchers found that novices sought confirmatory feedback more often, whereas experts sought critical feedback more often [9]. However, in contrast to our research, they did not measure learning outcomes.

### 3. POSTERLET

We employed two versions of the Posterlet game [4] to carry out our experiment. Participants playing the games assumed the identity of a school committee member in charge with designing a

poster for each of the two booths advertising events for the school's Fun Fair. The effectiveness of each designed poster (i.e., the number of visitors attracted by the booth) is quantified by the number of tickets sold, which is displayed when the poster is submitted. Posterlet also measures the number of times critical feedback is chosen or received, depending on condition, and the player's choices to revise posters across the game. After designing each poster, the player chooses three virtual characters out of a focus group to find out what they think about the poster. In the Choose condition, the player clicks on one box ("I like" or "I don't like") above each character. For example, in Figure 1, a participant in the Choose condition has first selected critical feedback from the lion and then confirmatory feedback from the elephant, but no feedback from the panda yet.



**Figure 1. In the Choose condition, the player has first chosen critical feedback from the lion, confirmatory feedback from the elephant, and no feedback from the panda yet.**

In the Receive condition, the player clicks on the "Click for feedback" box to reveal a feedback valence assigned by the game. For example, in Figure 2, a Receive condition participant has first clicked on the elephant's "Click for feedback" box (revealing critical feedback), then on the ostrich's "Click for feedback" box (revealing confirmatory feedback). The amount of critical feedback chosen or assigned (depending on the condition) is Posterlet's first key measure. After reading the feedback, the player has a choice to revise or submit the poster. The number of revised posters is Posterlet's second key measure. The game's feedback system generates feedback by analyzing each poster against 21 graphic design principles provided by a graphic artist and organized into three broad categories: information (e.g., the poster should include the date of the event), readability (e.g., the color contrast between the text and the background should be high), and space use (e.g., the space used by images needs to be within 30% and 70% of the poster's surface).



**Figure 2. In the Receive condition, the player has first clicked on the elephant and received critical feedback, then on the ostrich and received confirmatory feedback.**

It computes each poster’s quality (i.e., the number of tickets sold) and it includes a priority scheme to ensure a balanced representation of these categories in the feedback. The critical and confirmatory feedback phrases are equivalent in length and informational content. For example, if a player omits the day of the fair, the critical feedback is: “You need to tell them what day the fair is.” Otherwise, the confirmatory feedback is: “It’s good you told them what day the fair is.”, as shown in Figure 2.

## 4. METHOD

### 4.1 Participants, Procedures, Data Sources, and Experimental Overview

Participants (see Table 1) are N=264 Mechanical Turk adults randomly assigned to either the Choose or the Receive condition. Choose condition participants played a version of Posterlet that collected their feedback choices, while Receive condition participants played a modified Posterlet version that did not offer a feedback choice. In a one-to-one yoked experimental design, each participant in the Receive condition was assigned the feedback valence, number, and order of the feedback chosen by a matched Choose condition participant. Participants played a two-poster version of the Posterlet game individually, corresponding to their assigned condition, with a five-minute time limit on each poster or revision. Then, they completed an individual online posttest. The participants in the Choose condition were presented with a choice regarding the valence of their feedback. For instance, Figure 1 illustrates the feedback choices of a participant in the Choose condition: the participant chose a critical feedback from the lion and then a confirmatory feedback from the elephant. The Receive Condition participants were assigned the feedback valence of paired Choose condition participants, in the same order in which feedback was chosen by those paired participants. The game also collected participants’ revision choices and computed the participants’ poster performance (i.e., the quality of all their posters). Posterlet tracked the amount of critical feedback out of a maximum of 6 (3 feedback opportunities x 2 posters), as well as the amount of revisions out of a maximum of 2 (1 revision opportunity x 2 posters). A separate posttest measured the graphic design principles learned by participants in both conditions.

Table 1. Number of participants in each condition by gender

Cond.	Gender		Age Range	M <sub>age</sub> (SD <sub>age</sub> )
	F	M		
Choose	54	78	19-69	32.26 (9.53)
Receive	61	71	19-63	33.30 (10.40)
<b>Total</b>	<b>115</b>	<b>149</b>	<b>19-69</b>	<b>32.78 (9.96)</b>

For instance, Figure 2 illustrates the feedback selection of a participant in the Receive condition: the participant was first assigned critical feedback and then confirmatory feedback, just like the participant in the Choose condition illustrated in Figure 1.

In the Choose condition, participants played Posterlet for an average of M=7 minutes (SD=3.11) and then completed the posttest for an average of M=6 minutes (SD=2.24). In the Receive condition, participants played Posterlet for an average of M=7 minutes (SD=2.91) and then completed the posttest for an average of M=7 minutes (SD=2.54). This study is correlational and experimental, aiming to determine whether having a choice about one’s feedback valence aids in learning or in choosing to revise one’s work. It compares adults who exercised a choice regarding

the valence of their feedback (Choice condition) to adults who were assigned their feedback valence (Receive condition).

## 4.2 Dependent Measures

### 4.2.1 Feedback Valence and Revision Choices

**Critical Feedback** measures the number of “I don’t like” boxes chosen or received by the player across the game (0-6). **Confirmatory Feedback** measures the number of “I like” boxes chosen or received, equivalent to 6 minus *Critical Feedback* (0-6), since there are six total feedback choices across the game. **Revision** measures the number of posters a player revised (0-2).

### 4.2.2 Design Duration

We measured the time a participant spent designing each poster, from the moment a booth theme was clicked to the moment the “Test” button was pressed.

### 4.2.3 Learning Outcomes

**Poster Quality** measures the poster performance, summing the poster quality across posters. The quality of each poster is the sum of the scores for each of the 21 features: 1 if a feature is always used correctly, 0 if a feature is not on the poster, and -1 if a feature is used incorrectly. Thus, the score of any individual poster ranges from -21 to 21, while *Poster Quality* from -42 to 42.

A posttest assessed learning of the graphic principles. The overall *Posttest* score represents the sum of the normalized scores of the *Recognition* and *Principle Selection* measures.

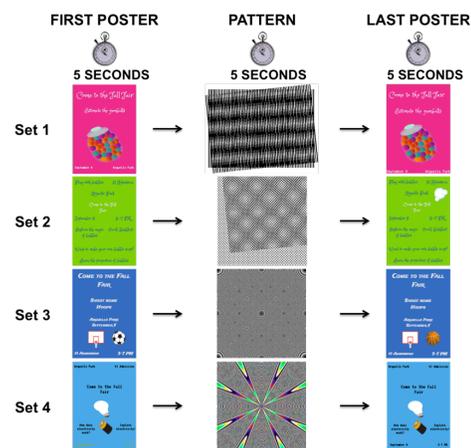


Figure 3. The *Recognition* posttest questions.

**Recognition** comprised four sets of posters (Figure 3). For each set, participants’ task was to judge whether the quality of the second poster was the same/better/worse compared to the quality of the first poster and to provide a brief written explanation for their decision. A distractor image was inserted between the two posters to ensure that memory was not playing a role [22]. Participants were guided through a mini-tutorial and a trial poster comparison, in which pictures succeeded automatically on a five-second timer. Each correct answer is scored with one point, while each incorrect answer is scored with zero points. This measure sums up only the correct answers, thus ranging from zero to four. **Principle Selection** comprised two 10-item design principle checklist questions (Figure 4). A point was awarded/subtracted for each correct/incorrect answer and scores were summed up.

Posttest: Principle Selection Questions

**Good Features Question**

What are some **good** things you notice about this poster?

- Includes all the important information
- All graphics convey information about the booth
- All the words stand out from the background color
- The font sizes are not too big or small
- The font styles are all easy to read
- No text is hidden behind other graphics/text
- All text is an appropriate distance from the edge of the poster
- All graphics are an appropriate distance from the edge of the poster
- Poster makes good use of the space
- There is good spacing between text and graphics

**Bad Features Question**

What are some **bad** things you notice about this poster?

- Does not include all the important information
- Graphics do not convey information about the booth
- Some of the words do not stand out from the background color
- There is a problem with font size being too big or small
- At least one of the font styles is not easy to read
- Some text is hidden behind graphics or text
- Some text is not an appropriate distance from the edge of the poster
- Some graphics are not an appropriate distance from the edge of the poster
- Poster does not make good use of the space
- There is bad spacing between text and graphics

Figure 4. The Principle Selection posttest questions.

## 5. RESULTS

### 5.1 Does critical feedback correlate with learning outcomes?

We examined poster performance and design principle learning. Table 2 and Table 3 show the zero-order Pearson correlations by condition. Critical Feedback and Revision correlated with Poster Quality and strongly with each other. We consider Poster Quality a learning measure, due to participants' improvement across the game [*Choose*: round<sub>1</sub>=10.64 (SD=5.0), round<sub>2</sub>=11.76 (SD=4.5), Wilks' Lambda=.92, partial eta squared=.08, F(1,131)=11.67,  $p < .01$ ; *Receive*: round<sub>1</sub>=10.68 (SD=6.0), round<sub>2</sub>=11.67 (SD=5.4), Wilks' Lambda=.96, partial eta squared=.04, F(1,131)=5.89,  $p < .05$ ]. Revision correlated with Posttest and Design Duration. Poster Quality correlated with Posttest, supporting the learning measures' internal validity. In the Choose condition, Critical Feedback correlated with Design Duration.

Table 2: Correlations between critical feedback, revision, and learning outcomes for the *Choose* condition

Measures (N=132)	Revision	Poster Quality	Posttest	Design Duration
Critical Fb.	.62**	.25**	.08	.32**
Revision	--	.23**	.21*	.39**
PosterQuality		--	.27**	.39**

\*\*  $p < .01$ , \*  $p < .05$

Table 3: Correlations between critical feedback, revision, and learning outcomes for the *Receive* condition

Measures (N=132)	Revision	Poster Quality	Posttest	Design Duration
Critical Fb.	.58**	.18*	.13	.16
Revision	--	.24**	.21*	.36**
PosterQuality		--	.21*	.38**

\*\*  $p < .01$ , \*  $p < .05$

We entered Critical Feedback and Revision in regressions to determine if they were independent predictors of the learning

outcomes. In the Choose condition, for Poster Quality, the model was significant [F(2,129)=5.10,  $p < .01$ ,  $R^2 = .07$ , Adjusted  $R^2 = .06$ ], but Critical Feedback [ $t(129)=1.6$ ,  $p = .11$ ] and Revision [ $t(129)=1.6$ ,  $p = .25$ ] were not predictors. For Posttest, the model was significant [F(2,129)=3.33,  $p = .04$ ,  $R^2 = .05$ , Adjusted  $R^2 = .03$ ], Revision was a predictor:  $t(129)=2.38$ ,  $p = .02$ , but Critical Feedback:  $t(129)=-.71$ ,  $p = .48$  was not. In the Receive condition, for Poster Quality, the model was significant [F(2,129)=4.23,  $p = .02$ ,  $R^2 = .06$ , Adjusted  $R^2 = .05$ ], Revision was a marginally significant predictor:  $t(129)=1.99$ ,  $p < .05$ , but Critical Feedback:  $t(129)=-.58$ ,  $p = .56$  was not. The Posttest model was not significant.

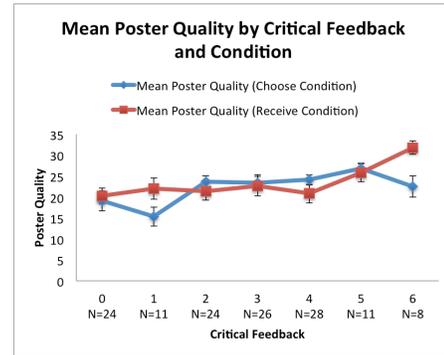


Figure 5. Poster Quality by Critical Feedback and condition.

### 5.2 Are there learning outcome differences between choosing and receiving feedback?

T-test analyses revealed no differences in Poster Quality [ $M_{Choose} = 22.39$  (SD=8.71),  $M_{Receive} = 22.36$  (SD=10.4),  $t(262) = .03$ ,  $p = .97$ ], Posttest [ $M_{Choose} = .10$  (SD=1.53),  $M_{Receive} = .04$  (SD=1.45),  $t(262) = .32$ ,  $p = .75$ ], and Revision [ $M_{Choose} = .80$  (SD=.87),  $M_{Receive} = .93$  (SD=.82),  $t(262) = -1.24$ ,  $p = .22$ ] between conditions. Figure 5, Figure 6, and Figure 7 plot our measures across the game as a function of critical feedback (from 0 to 6) by condition. Error bars represent one standard error. The x-axis shows the range of critical feedback and the number of participants for each amount of critical feedback (e.g., N=26 participants chose/received 3 pieces of critical feedback across all posters). Regressions of *critical feedback*, *condition*, and *critical feedback by condition* on learning and revision revealed no interactions of critical feedback and condition with our measures.

### 5.3 Are there design duration differences between choosing and receiving feedback?

A t-test analysis revealed no differences in Design Duration (time in seconds spent designing posters) between conditions [ $M_{Choose} = 401.30$  (SD=186.39) and  $M_{Receive} = 394.44$  (SD=174.94),  $t(262) = .31$ ,  $p = .76$ ]. Figure 8 plots participants' poster design time across the game as a function of critical feedback (from 0 to 6) by condition.

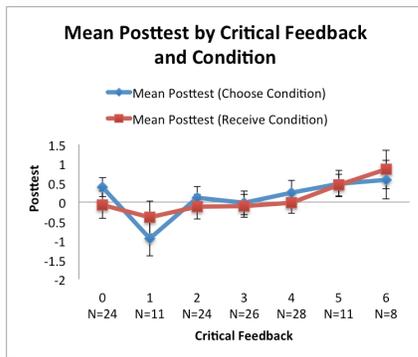


Figure 6. Posttest by Critical Feedback and condition.

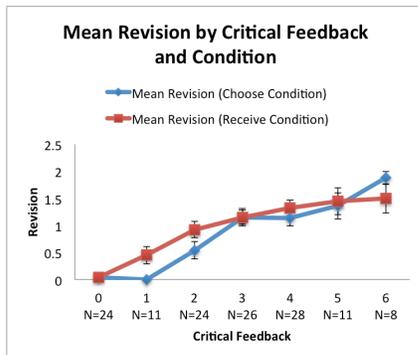


Figure 7. Revision by Critical Feedback and condition.

### 5.4 Are there any gender differences?

In the Receive condition, we found that females [ $M=433.28$  ( $SD=176.84$ ),  $t(130)=2.41$ ,  $p=.02$ ] spent more time designing posters than males [ $M=361.07$  ( $SD=167.40$ )]. There were no gender differences by condition on any of the rest of the measures (Revision, Poster Quality, and Posttest).

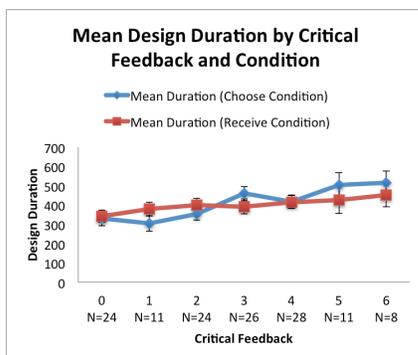


Figure 8. Design Duration by Critical Feedback and condition.

## 6. DISCUSSION

This is a first-of-kind examination of both the agency (choosing versus receiving) and the valence (critical versus confirmatory) of feedback and their impact on performance and learning. We found that, in each condition, the amount of critical feedback (either chosen or received) correlated with participants' performance on the poster design task. Consistent with our previous findings [3, 4], critical, rather than confirmatory, feedback seems beneficial for learning. Also, the choice to revise was beneficial for

performance and learning outcomes and it strongly correlated with critical feedback (chosen or received). We found no differences between conditions in any of the measures outlined in this paper. These results held when we compared the measures by gender in each condition, although in the Receive condition, females spent more time designing posters than males. This indicates that these types of behavioral assessments of learning have the potential to be gender neutral. The next step would be to design more such dynamic assessments to evaluate other behaviors, such as self-assessment. Designing gender-neutral assessments that embed both skills and learning behaviors would bring us closer to determining the knowledge, skills, and delivery methods required to foster independent learners in the 21<sup>st</sup> century, as well as ways to ensure gender equality, especially when only 14.1% of North American computer science bachelor's degree graduates are female [25]. Our study points to critical, rather than confirmatory, feedback being beneficial for learning, regardless of being chosen or assigned. It also points to ways of designing assessments that measure learning behaviors equally regardless of gender. Finally, in the Choose condition, the more the participants chose critical feedback, the more time they spent designing posters. The relation between critical feedback and revision, as well as between critical feedback and poster quality, was stronger and more stable in the Choose condition, pointing to motivational factors of choosing versus receiving critical feedback for performance. More research is needed to elucidate this motivational aspect.

People's choices of critical feedback can be influenced by a wide range of factors. For instance, the perception of a trait as fixed may lead to avoidance of negative feedback [5]. Additionally, compared to a growth mindset (an incremental theory of intelligence - the belief that intelligence can be developed over time), a fixed mindset (an entity theory of intelligence - the belief that intelligence is fixed) was found to be associated with decreased attention to corrective feedback or errors [14]. However, the results of this study suggest that there is no underlying variable (e.g., desire to learn, self-confidence, growth mindset [6, 8], etc.) that drives the effect of critical feedback. People who choose critical feedback more often may exhibit one or more of these variables, yet, despite that, assigning the same amount of feedback leads to the same results as other factors that may cause them to choose critical feedback. Consequently, it seems that such factors (e.g., deep beliefs or personal attributions, such as "I am a learner") do not need to be changed to help people reap the benefits of constructive criticism. Learner beliefs do not mediate the benefits of receiving constructive criticism. One potential implication is the possibility to change people's beliefs about seeking critical feedback without having to change their broad beliefs about themselves as learners, which we also demonstrated in a separate study [3]: fairly straightforward instruction to seek social feedback (i.e., opinions of others) transferred to Posterlet and, consequently, students learned more.

Our study's limitations are associated with conducting Mechanical Turk experiments with a large population: (1) a maximum of five minutes allotted per poster, which may have hindered the discovery of some of the game's features (e.g., that the poster background color can be changed) and (2) a maximum of two game levels, which offered participants at most six pieces of feedback from which to learn graphic design principles, which may not have overlapped with the four principles included on the posttest (feedback content varied, depending on each participant's poster, but the posttest questions were the same for all participants). The latter is one possible explanation for the lack of correlation between critical feedback and posttest. Alternatively,

participants examined each poster for only five seconds and, if they missed one of the two posters in a set, they could not have accurately answered any of the questions about that set. Thus, we plan to compare this study's Choose condition data with data from the first two levels of previous three-level Posterlet game studies. That way, we may predict participant behaviors on the third game level, to potentially detect differences between conditions in our measures that are not apparent currently.

## 7. CONCLUSIONS

We modified a choice-based assessment game to measure learning when participants are offered a choice about the valence of their feedback and when they are assigned their feedback valence. The data enabled a novel examination of choosing versus receiving confirmatory *versus* critical feedback with regards to learning outcomes. We found that the more the participants chose critical feedback in the Choose condition, the more time they spent designing posters. There were no differences in learning outcomes (performance on the poster design task and learning of the graphic design principles), choice to revise, or time spent designing posters between participants who chose feedback and those who received the same amount, valence, and order of feedback. We plan a similar study with middle-school and college students to explore instruction and assessment implications. These studies could inform teachers to create environments in which students feel encouraged to engage more with critical feedback (proactively or reactively), even in open-ended tasks as digital poster design. The flexibility of such short assessments focused on specific choices (e.g., feedback seeking) enables the development and evaluation of a variety of instruction models. Concomitantly, researchers can design pedagogical interventions and learning environments that embed such assessments to empower all learners, regardless of gender, to be innovative, confident, and prepared for the challenges of the 21<sup>st</sup> century.

## 8. ACKNOWLEDGMENTS

We thank the Gordon and Betty Moore foundation and the NSF (Grant # 1228831), Jacob Haigh for assistance with the online setup, as well as the Mechanical Turk participants.

## 9. REFERENCES

- [1] Black, P., & Wiliam, D. (2009). Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability*, 21(1), 5–31.
- [2] Bransford, J. D., & Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education*, 24, 61–100.
- [3] Conlin, L., Chin, D. B., Blair, K. P., Cutumisu, M., & Schwartz, D. L. (2015). Guardian Angels of Our Better Nature: Finding Evidence of the Benefits of Design Thinking. In Proc. of *ASEE*, June 14-17, Seattle, WA, USA.
- [4] Cutumisu, M., Blair, K. P., Chin, D. B., & Schwartz, D. L. (2015). Posterlet: A Game-Based Assessment of Children's Choices to Seek Feedback and to Revise. *Journal of Learning Analytics*, 2(1), 49-71.
- [5] Dunning, D. (1995). Trait importance and modifiability as factors influencing self-assessment and self-enhancement motives. *Personality and Social Psychology Bulletin*, 21.
- [6] Dweck, C. S., & Leggett, E. L. (1988). A social-cognitive approach to motivation and personality. *Psychological Review*, 95(2), 256-273.
- [7] Ehrlinger, J., & Dunning, D. (2003). How chronic self-views influence (and potentially mislead) estimates of performance. *J of Personality & Social Psychology*, 84(1), 5.
- [8] Ehrlinger, J., Mitchum, A. L., & Dweck, C. S. (2016). Understanding overconfidence: Theories of intelligence, preferential attention, and distorted self-assessment. *Journal of Experimental Social Psychology*, 63, 94-100.
- [9] Finkelstein, S. R., & Fishbach, A. (2012). Tell me what I did wrong: experts seek and respond to negative feedback. *Journal of Consumer Research*, 39(1), 22–38.
- [10] Garrison, D. R. (1997). Self-directed learning: Toward a comprehensive model. *Adult Ed. Quarterly*, 48(1), 18–33.
- [11] Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81–112.
- [12] Haydon, M. L., Larson, D., Reed, E., Shrivastava, V. K., Preslicka, C. W., & Nageotte, M. P. (2011). Obstetric outcomes and maternal satisfaction in nulliparous women using patient-controlled epidural analgesia. *American Journal of Obstetrics and Gynecology*, 205(3), 271-e1.
- [13] Kluger, A. N., & DeNisi, A. (1998). Feedback interventions: toward the understanding of a double-edged sword. *Current Directions in Psychological Science*, 7(3), 67–72.
- [14] Mangels, J., Butterfield, B., Lamb, J., Good, C., & Dweck, C. (2006). Why do beliefs about intelligence influence learning success? A social cognitive neuroscience model. *Social Cognitive and Affective neuroscience*, 1(2).
- [15] March, J. (2008). *Explorations in organizations*. Stanford University Press.
- [16] Piaget, J. (1964). Quoted by Eleanor Duckworth in "Piaget Rediscovered: A Report of the Conference on Cognitive Studies and Curriculum Development", *Cognitive Studies and Curriculum Development*, New York.
- [17] Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, 21(2), 267-280.
- [18] Schwartz, D. L., & Arena, D. (2009). Choice-based assessments for the digital age. *MacArthur 21st Century Learning and Assessment Project*.
- [19] Schwartz, D. L., & Bransford, J. D. (1998). A time for telling. *Cognition and Instruction*, 16, 475–522.
- [20] Schwartz, D. L., Lindgren, R., & Lewis, S. (2009). Constructivism in an age of non-constructivist assessments. *Constructivist Instruction*, 34-61.
- [21] Shute, V. J., & Ke, F. (2012). Games, learning, and assessment. In *Assessment in Game-Based Learning*, 43-58.
- [22] Standing, L. (1973). Learning 10,000 pictures. *Quarterly Journal of Experimental Psychology*, 25, 207-222.
- [23] Vygotsky, L. S. (1934). *The collected works of LS Vygotsky: Problems of the theory and history of psychology*.
- [24] Zimmerman, B. J. (1990). Self-regulated learning and academic achievement: An overview. *Educational Psychologist*, 25(1), 3–17.
- [25] Zweben, S., & Bizot, B. (2015). 2014 Taulbee Survey. *Computing Research News*, May 2015, 27(5), p. 20.

# Course Content Analysis: An Initiative Step toward Learning Object Recommendation Systems for MOOC Learners

Yiling Dai  
Graduate School of  
Informatics  
Kyoto University  
Yoshida-Honmachi, Sakyo-ku  
Kyoto, Japan  
daiyiling@db.soc.i.kyoto-  
u.ac.jp

Yasuhito Asano  
Graduate School of  
Informatics  
Kyoto University  
Yoshida-Honmachi, Sakyo-ku  
Kyoto, Japan  
asano@i.kyoto-u.ac.jp

Masatoshi Yoshikawa  
Graduate School of  
Informatics  
Kyoto University  
Yoshida-Honmachi, Sakyo-ku  
Kyoto, Japan  
yoshikawa@i.kyoto-  
u.ac.jp

## ABSTRACT

With the accelerating development of open education, low-cost online learning resources, such as Massive Open Online Courses (MOOCs), are reaching a wide audience around the world. However, when faced with these appealing but overwhelming learning resources, learners are prone making rash learning decisions, which may be either excessive or insufficient to their learning capacities. To avoid the mismatch between learners and learning objects, we propose a supporting system that recommends a personalized path of learning objects for a given learner. In realizing this system, a domain knowledge structure is necessary to connect learners' information and learning objects. As an initiative step, we employ the Labeled Latent Dirichlet Allocation method to predict how the content of a course is distributed over different categories in the domain. We conduct experiments by utilizing course syllabi as course content, and curriculum guidelines as domain knowledge. The predicting performance is improved when involving external texts related to the concerned domain knowledge unit.

## 1. INTRODUCTION

Nowadays, pedagogically condensed free online resources are playing an increasingly more important role of facilitating self-learning. Among those resources, Massive Open Online Courses (MOOCs) are engendering a revolutionary change in higher education by distributing digital versions of university courses to everyone at a relatively low cost. Courses about Computer Science on Edx (one of the largest MOOC platforms), reached over 600,000 listeners during the period from 2012 autumn to 2014 summer [6], which hardly ever occurs on real campuses. However, compared with their popularity among audience, the low completion rate of courses

(e.g, 7% of the MOOCs on Edx mentioned above) begs the question—how many learners have truly benefited from receiving MOOCs? It appears that MOOCs have a way to go to achieve its original goal of making education accessible to everyone.

Rather than not being able to receive traditional education, many users utilize MOOCs out of pure curiosity toward subjects, or to complement their academic lives or career development [2]. In addition, the occupations of MOOC users are diverse, from students, writers, and engineers to housewives [2]. This type of utilization of MOOCs sets a higher requirement in terms of learner's self-motivation and self-regulation. Consequently, many users have reflected that they did not have sufficient spare time to catch on to the process of MOOCs, or simply became stuck on the overwhelming learning contents [2].

An intuitive question concerning that how we can help to maintain this precious enthusiasm of refreshing one's knowledge, motives this paper. We hold the view that finding the "just right" learning objects for respective individuals paves the way toward a successful learning experience. This belief is also in agreement with the opinion of [4], which underlines the importance of personalization, especially in the context of online learning. Specifically, "just right" means that the learning objects fit both the learning objective and learning ability of a given learner. In the context of self-learning, where more flexibility is given to a learner for him to decide what to learn, the adaptation to learning objectives deserve greater investigation than before. Concerning the method used to accomplish personalization in learning, previous studies have shown a trend of utilizing expert manpower or learner performance data to extract internal relationships among knowledge itself and external relationships between knowledge and learner mastery, which may not work when promoting personalized learning on a massive scale.

In this paper, we propose the idea of a novel supporting system that automatically recommends an appropriate set of learning objects with cues of learning priority to a given learner. This system is expected to outperform existing

adaptive learning systems on addressing heterogeneous course materials automatically and on adapting learning objects to learners before they start to learn. As an initiative task, a course content analysis is conducted to crystallize the realization of the supporting system. We employ the Labeled Latent Dirichlet Allocation method to predict how the content of a course is distributed over different domain knowledge categories. Course syllabus texts are utilized as course content, and the knowledge listed in curriculum guidelines are utilized as domain knowledge. To improve the accuracy of predictions, we extend the content of the curriculum guideline by integrating external texts retrieved from search engines.

The remainder of this paper is structured as follows: Section 2 summarizes related work with regard to personalized learning and knowledge representation. In section 3, an illustration and the framework of the supporting system are sketched. Then, we present the results and observations of a course content analysis. Finally, we discuss on future work.

## 2. RELATED WORK

### 2.1 Personalized learning

What we call personalized learning is named differently in previous studies, e.g., adaptive learning/education, individualized learning/education, and intelligent tutoring systems; however, they all share the main concern of adapting learning materials to individual learners. In this paper, we adopt the phrase “personalized learning” to capture all these related studies and use “personalize”, “individualize”, “adapt” interchangeably.

Personalized learning is described as “learning tailored to the specific requirements and preferences of the individual” in [11]. Although not forming a fixed definition of personalized learning, many studies attempt to adapt learning to specific learners. [4] demonstrated a hypermedia textbook that can provide direct guidance and adaptive navigation support to learners. Similarly, [15] developed a topic-based adaptive learning system that directs the learner to the appropriate learning object by providing navigational cues. Moreover, [16] broadened the adaptation from a single source of personalization information to learning achievements and learning styles at one time. [8] presented an e-learning system that recommends learning items by detecting frequent learning sequences and similar learners. [9] proposed another approach of generating adaptive course content using concept filters.

A shared architecture of a personalized learning system that can be observed consists of three parts: Domain model, Learner model and Adaptation model. The domain model constructs all the knowledge units of learning materials in a common space, and its complexity varies based on the application contexts. The learner model is a projection of a learner’s learning state (i.e., mastery level of knowledge, learning objective, and learning style) onto the structure of knowledge that is defined in the domain model. The adaptation model functions as a recommend of the next learning target basing on the updated learner state. This adaptation in learning environments occurs at different levels. [11] categorized this adaptation as follows: Adaptive Interaction, which occurs during the interactions between learners and

the system; Adaptive Course Delivery, which intends to tailor learning materials to a given learner; Content Discovery and Assembly, which involve the collecting of learning materials from potential sources or repositories; Adaptive Collaboration Support, which supports communication in the learning process.

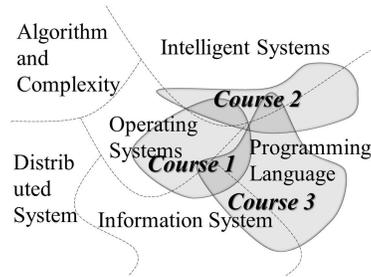
In the context of self-learning, “why I want to learn”, “what I want to learn”, “what outcomes I am expecting”, things usually being told to the learner by the curriculum, must be determined by the learner himself. As a result, we consider that the information-seeking phase before starting to learn becomes a key to a successful learning experience. We provide a learning object recommendation system that the learners can resort to when they are faced with overwhelming learning resources. Compared with a branch of studies [10, 1, 19] that implement the adaptation by redirecting the learner to an optimal learning path using tracked learner performance, our approach focuses on a more macro level of adaptation, which occurs beforehand and addresses the learning object with a larger granularity (i.e., a lecture). According to [11]’s categorization of adaptation, our system stands in an overlapping area of Adaptive Course Delivery and Content Discovery and Assembly, thereby distinguishing itself from other adaptive learning/tutoring systems.

### 2.2 Automatic domain representation

The construction of domain knowledge is a key step in accommodating a personalized learning system. However, previous studies [4, 15, 16, 8, 9, 10, 1, 19] show a substantial reliance on expert efforts, whose systems require the instructors to define strictly structured course materials for the concerned system. This is so time-consuming and platform dependent that it is unsuitable when addressing a large amount of distributed learning materials. An automatic and interoperable knowledge representation and assemble are thus desired.

In the context of learning, knowledge representation refers to the process of editing knowledge in a more visually sound and retrievable manner based on its hierarchical or dependent relationships. Previous studies relating to this concept can be divided into two types according to their approaches, and we name them prior approaches and post approaches. A prior approach means extracting the relationships between knowledge units based on the structure defined by the instructor. For example, [3] utilized the content and structure of a textbook to extract the relationships between concepts based on their co-occurrence conditions. [5] exploited the extraction of prerequisite relationships of learning objects by conducting semantic analysis on Wikipedia articles. Regarding the post approach, in which the structure of knowledge is modified by the learner reactions on these learning objects, [17] and [18] attempted to detect prerequisite relationships between knowledge units by utilizing a considerable amount of learner achievement data. Their studies are based on the rationale that knowledge units that are statistically “always” mistaken by the learners should be learned before the ones that are not so.

In this paper, we emphasize the preprocess of learning (i.e., seeking information and making a learning plan), which occurs before a substantial amount of learner performance data



**Figure 1: Illustration of our supporting system—the course map**

are available. Thus, our research falls into the category of prior approaches. Previous studies [3, 5] have employed various Natural Language Processing techniques to extract relationships between knowledge units. However, the results remain modest in addressing heterogeneous learning materials at scale; a proliferation of this stream of research is needed.

### 3. OUR SUPPORTING SYSTEM

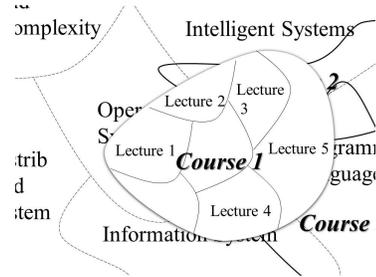
As discussed in the previous section, in the context of self-learning, support for a learner determining what to learn and how to learn is sensible. Except for a learner’s learning ability, which has received a fair discussion in previous research, we consider the estimation of the learner’s learning objective. Regarding the level of personalization in this learning environment, we highlight the phase of assembling learning materials from distributed learning resources. As a consequence, we suppose that learners will benefit from our system before they enter the real learning process when offered a tailored path of learning objects that fits their learning needs and ability.

#### 3.1 An illustration of the system

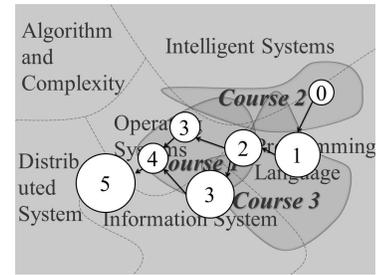
To explain our supporting system more vividly, we present an illustration of a final usage of the system. The target user of our system will not be constrained to a specific group of learners; however, the learners who will benefit the most from our system are those who are planning to challenge some unfamiliar subject. Then, we can imagine a virtual learner, a college student majoring in social science, who is wondering how data mining techniques will assist in analyzing his collected data.

First, he may simply input a keyword “data mining”. Instead of returning a ranked list of relevant courses, which is normal in existing MOOC search engines, our system will answer the query dynamically by starting with a map of relevant courses to that query. As shown in Figure 1, the shapes circled using a dotted line with titles (e.g., “Intelligent Systems”) on them refer to the predefined structure of the domain knowledge. In addition, the shape circled using a solid line represent a course that contains the knowledge in that place.

Then, the learner responds to the first reply differently. He may want to obtain details of some highly similar courses or seek a more holistic view of this domain to determine what these courses mean to his learning task. If the learner



**Figure 2: Illustration of our supporting system—the detailed course information**



**Figure 3: Illustration of our supporting system—a learning path**

chooses to zoom in to course 1, then he will obtain a detailed view of the content of course 1. As shown in Figure 2, the topics covered in course 1 will be shown in the unit of a lecture.

We suppose that the learner will not be satisfied until he can make a confident decision on what and how to learn. Therefore, he will continue interacting with our system, during which time his learning characteristics will be recorded. Finally, the recorded learner information will be used to recommend a tailored learning path for the learner (see Figure 3). The path consists of a set of learning objects that are chained according to the dependent relationships between the knowledge they cover. For well-prepared learners, the path will exclude materials he already knows and will cover a narrowed down knowledge set in the depth. For novice learners, in this case, the path will cover a wider range of knowledge and will start from the very simple knowledge units.

#### 3.2 The architecture of the system

To realize the system illustrated above, the architecture is threefold—domain model, learner model, and personalization model. The domain model conducts the task of locating the learning objects of courses in the knowledge structure of the domain. The learner model tracks learner information about his learning objective, background knowledge, and learning preferences according to the knowledge structure. The personalization model specifies the appropriate learning objects based on predefined criteria. Among them, the construction of domain knowledge and the mapping of course content determine how to estimate learner information and what learning objects to recommend. Thus, it is reasonable

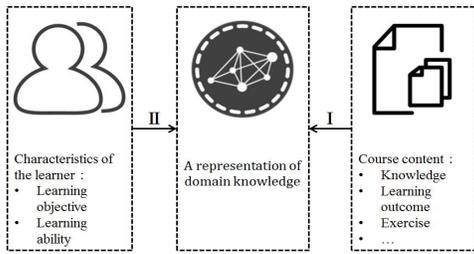


Figure 4: The architecture of proposed system

to exploit the domain model as a primary task. The following part of this paper describes a course content analysis and discusses its potential for equipping the domain model.

## 4. COURSE CONTENT ANALYSIS

### 4.1 Overview

As a primary task for matching course contents to a domain knowledge base, we extracted knowledge coverage of a given course by projecting its syllabus text onto a curricular guideline in the domain. A syllabus functions as a summary of the course content, which makes it suitable for our method. In addition, a curricular guideline generally contains important topics in the domain, which can be utilized as a reference of the domain knowledge. Specifically, we utilized the curriculum guideline *Computer Science Curricula 2013* (CS2013) [14] published by IEEE-CS and ACM, which attempts to provide instructional cues of knowledge that should be included in an undergraduate program. In CS2013, both classic and frontier topics in this domain are described in *Body of Knowledge* (BoK). BoK is compiled in a hierarchical structure wherein the smallest granularity of knowledge is a *topic*, and each *topic* belongs to a *Knowledge Unit* (KU), and each KU further belongs to a *Knowledge Area* (KA). In total, 18 KAs and 163 KUs are formed to categorize knowledge in the domain of Computer Science. A simplified example of KA-KU-Topic knowledge structure in CS2013 is shown in Table 1.

This semi-structured BoK has been used to analyze the curricula of different educational institutions [7, 13]. In an attempt to obtain an overall picture of Informatics programs in Japan, [7] conducted a judgement of knowledge coverage on syllabi by referring to curriculum guidelines. [13] employed a supervised Latent Dirichlet Allocation (LDA) method to extract KA coverages of a course using the text of its syllabus. From the above studies, it is reasonable to use curriculum guidelines as a knowledge base to form predictions of course knowledge coverage in an automated manner. However, it is not sufficient to recommend learning objects when solely using the knowledge coverage of a course at the level of KA. Therefore, we attempt to extract knowledge coverage of a course at a further fragmented level—KU in this case.

We adopt the topic model, Labeled Latent Dirichlet Allocation (Labeled LDA) to extract the knowledge coverage. Labeled LDA is designed to specify multiple dimensions of a given text that correspond to manually labeled tags [12]. In CS2013, exemplar courses with knowledge distribution information show that a course generally contains knowledge

Table 1: KA-KU-Topic knowledge structure in CS2013 [14]

KA	KU	Topics
Algorithms and Complexity (AL)	Basic Analysis	• Big O notation • ...
	Algorithm Strategies	• Greedy algorithms • ...
	...	• ...

Table 2: An example of syllabus information in CS2013 [14]

What is covered in the course?
• The modeling process
• Two system dynamic tool tutorials
• Computational error
• ...

from more than one KA or KU. Therefore, this method is suited when addressing a syllabus text that is labeled with multiple predefined tags—KA/KU in this case.

Considering that topics listed in BoK are highly compact representations of knowledge, we resort to external texts to complement the content of BoK. Specifically, we integrated snippet information retrieved from queries of a KU to improve the accuracy of predictions.

### 4.2 Dataset

81 exemplar courses, whose course information and knowledge distributions are assigned by the course instructor, are included. As shown in Table 2, the answer to the question “What is covered in the course?” is viewed as the syllabus information of a course. In addition, the information offered by the instructor on how the lecture hours of a course are allocated to each KA and KU is referred to as the ground truth of our method (e.g., 35.5 hours in CN, 3 hours in IS,...). After excluding malformed course information, 73 exemplar courses were used in the course content analysis.

Regarding the external texts, we threw 3 types of queries to retrieve snippet texts of websites from Google Custom Search API. The queries are formed by using: (1) KU title alone, (2) KA and KU title, (3) KU title and its top 3 representative terms (chosen by their tf-idf values, which represent an effective as an indicator of the importance of a term over a set of documents). 10 snippet texts were complemented to the content of each KU.

### 4.3 Procedures

#### 4.3.1 Training set

As a trial analysis, we exploit the predictability of curriculum guidelines by conducting experiments with different training sets. Among all the experiments, 30 exemplar course syllabi were chosen randomly as the testing set. Concerning the training set, we set 2 variables, forming 8 patterns, to improve the accuracy of predictions. The first variable denotes whether manually labeled syllabus texts are used in the training set or BoK texts alone are used. The

**Table 3: Experiment id**

	BoK	BoK_Snippet1	BoK_Snippet2	BoK_Snippet3
BoK	KA-1-0	KA-1-1	KA-1-2	KA-1-3
BoK+Course Syllabus	KA-2-0	KA-2-1	KA-2-2	KA-2-3

second denotes what type of snippet texts are used, with “0” denoting using BoK texts alone.

Table 3 presents the naming of the experiments according to their content of the training set. The names of experiments for the prediction of KU knowledge coverage follow the same naming scheme. We conduct all 8 experiments on predicting knowledge coverage at the level of both KA and KU, and we add “KA” or “KU” to the experiment id to indicate the different targets.

### 4.3.2 Evaluation

To evaluate the predicted probabilities over KAs/KUs of a syllabus, we apply the Normalized Discounted Cumulative Gain (nDCG), which is used to evaluate the relevance of a document rank to a given query in classic Information Retrieval (IR). We choose the nDCG because it addresses relevance as a non-binary value, which is better suited to our case where the relevance of a document corresponds to lecture hours. For each course, we compare the ranked list of KAs/KUs that is predicted by our method, with the ranked list of KAs/KUs that is allocated by the course instructor. The computation is conducted using the following equations :

$$\begin{cases} G_c[i] = rel_c[i] \\ DCG_c[k] = \sum_{i=1}^k \frac{G_c[i]}{\log_2 i+1} \\ nDCG_c[k] = \frac{DCG_c[k]}{IDCG_c[k]} \end{cases} \quad (1)$$

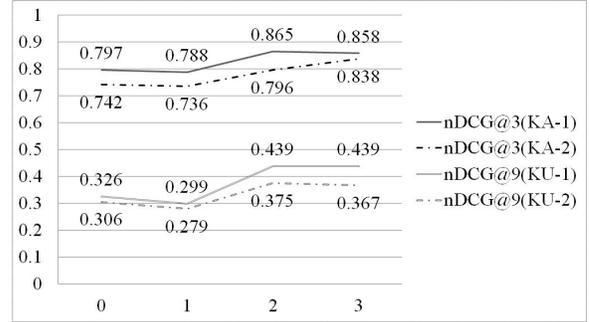
Here,  $rel_c[i]$  denotes the lecture hours allocated to the  $i^{\text{th}}$  KA/KU for a given course; DCG denotes the discounted cumulative gain of the ranked KA/KU list that is predicted by our method, and IDCG denotes the one of the ranked KA/KU list assigned by the course instructor.

## 4.4 Results

We utilized the *Stanford Topic Modeling Toolbox* to compute the KA/KU distributions of a syllabus and the Python library *Scikit Learn* to compute the tf-idf value of each term appearing in a BoK. Other data processes, such as the computation of the nDCG, are implemented in Python. Concerning the most representative terms for each KU, we chose the top three terms from a vocabulary of 2486 non-stopword terms. Because the average number of KAs that a course covers assigned by the instructor is 2.67, being 9.04 for KU, we focus on the nDCG value of  $k = 3$  for KA, of  $k = 9$  for KU. The results for each experiment are shown in Figure 5.

## 4.5 Discussion

As observed in Figure 5, all the nDCG values of the experiments with a training set containing BoK texts alone are higher than those with a training set consisting of both BoK texts and exemplar course syllabus texts. In our data set, all the BoK texts are annotated with one label, whereas exemplar course syllabus texts are annotated with multiple



**Figure 5: The nDCG values of each experiment. The vertical axis denotes the value of nDCG, which varies from 0 to 1. The horizontal axis denotes the second variable with regard to the naming of the experiments—the type of snippet texts used in training set.**

labels. This unbalanced number of labels in the training set may reduce the precision of prediction obtained using Labeled LDA. However, from a positive perspective, this result indicates the potential of only using pre-collected documents of domain knowledge instead of collecting annotated course syllabi when predicting the knowledge coverage of a given course.

Two types of snippet texts exhibit a positive effect on predicting KA/KU knowledge coverage. They are snippet texts queried from KU titles with their corresponding KA title and snippet texts queried from KU titles with their top 3 representative terms. For example, nDCG@3 of KA-1-2 and KA-1-3 are notably higher than those of KA-1-0. A similar trend can also be observed in the case of predicting KUs. In contrast, nDCG@3 of KA-1-1 are lower than those of KA-1-0, which indicates that the external texts obtained from the KU title query drag down the performance of our model. One possible reason that can be inferred is that a sole KU title can produce substantial noise when it is used without context. For example, “processing” has a much broader meaning than that in the context of “Computational Science”. Other ambiguous KU titles, such as “Basic Logic” and “Data, Information, and Knowledge”, are prone to increasing the prevalence of this type of mistake. Overall, queries consisting of KA titles and KU titles or KU titles and their keywords provide effective and relevant texts when predicting knowledge coverage.

To seek deeper factors that may contribute to the correctness of a prediction, we examined an exemplar course syllabus and compared it with BoK and external texts. We found:

- Some synonymous or semantically similar phrases (e.g.,

“strategies for choosing...” and “apply...”) may not be detected by our method.

- There exist internal relationships between KUs (e.g., KU “Processing” under KA “Computational Science” overlaps with KU “Algorithms and Design” under KA “Software Development Fundamentals”), which may mislead the prediction of KUs.
- An increase in performance in predicting KAs may not guarantee an improvement in predicting KUs. Because in some cases, the improvement in predicting KAs is achieved by assigning a probability to an incorrect KU under the KA.

## 5. CONCLUSION AND FUTURE WORK

Summarizing, we proposed a supporting system that recommends an effective and efficient path of learning objects for a given individual. To realize this system, a threefold architecture is needed—Domain model, Learner model and Adaptation Model. As an initiative step, we conducted a course content analysis, in which Labeled LDA was utilized to predict the knowledge coverage of a course. The result provided the positive indication that involving external explanatory texts on domain knowledge facilitates the prediction of the knowledge coverage of unknown course syllabi. However, the precision of the the current experiment needs further improvement in addressing texts semantically. Specifically, a bigram or trigram method is expected to perform better than the unigram method. In addition, separate nouns and noun-phrases may increase the precision. From a holistic perspective, we also need to consider the estimation of learner characteristics when constructing domain knowledge bases. For example, a framework of knowledge that connects knowledge itself with its learning outcomes may be instrumental in mapping learning objects to learners.

## 6. ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 15K00423 and the Kayamori Foundation of Informational Science Advancement.

## 7. REFERENCES

- [1] T. Barnes. Q-matrix Method: Mining Student Response Data for Knowledge. Technical report, 2005.
- [2] Y. Belanger and J. Thornton. Bioelectricity: A Quantitative Approach Duke University’s First MOOC. Report, 2013.
- [3] R. J. C. Bose, O. Deshmukh, and B. Ravindra. Discovering Concept Maps from Textual Sources. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.
- [4] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: a tool for development adaptive courseware. *Computer Networks and ISDN Systems*, 30(1-7):291–300, 1998.
- [5] F. Gaspiretti, C. Limongelli, and F. Sciarone. Exploiting wikipedia for discovering prerequisite relationships among learning objects. In *Proceedings of International Conference on Information Technology Based Higher Education and Training*, pages 1–6, 2015.
- [6] A. D. Ho, I. Chuang, J. Reich, C. A. Coleman, J. Whitehill, C. G. Northcutt, J. J. Williams, J. D. Hansen, G. Lopez, and R. Petersen. HarvardX and MITx: Two Years of Open Online Courses Fall 2012-Summer 2014. SSRN Scholarly Paper ID 2586847, Social Science Research Network, 2015.
- [7] I. Kiyoshi et al. Investigation on the Educational Contents among Informational Science and Engineering Departments by Using Syllabus (Intermediate Report). Technical Report 6, Information Processing Society of Japan, 2010.
- [8] A. Klačnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3):885–899, 2011.
- [9] F. W. B. Li, R. W. H. Lau, and P. Dharmendran. An Adaptive Course Generation Framework. *Int. J. Distance Educ. Technol.*, 8(3):47–64, July 2010.
- [10] N. Matsuda, T. Furukawa, N. Bier, and C. Faloutsos. Machine beats experts: Automatic discovery of skill models for data-driven online course refinement. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.
- [11] A. Paramythis and S. Loidl-Reisinger. Adaptive Learning Environments and eLearning Standards. *ELECTRONIC JOURNAL OF ELEARNING, EJEL: VOL 2. ISSUE, 2*:181–194, 2004.
- [12] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256. Association for Computational Linguistics, 2009.
- [13] T. Sekiya, Y. Matsuda, and K. Yamaguchi. Curriculum analysis of CS departments based on CS2013 by simplified, supervised LDA. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 330–339, 2015.
- [14] I. C. Society. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Technical report, ACM, 2013. 999133.
- [15] S. Sosnovsky and P. Brusilovsky. Evaluation of topic-based adaptation and student modeling in QuizGuide. *User Modeling and User-Adapted Interaction*, 25(4):371–424, 2015.
- [16] J. C. R. Tseng, H.-C. Chu, G.-J. Hwang, and C.-C. Tsai. Development of an adaptive learning system with two sources of personalization information. *Computers & Education*, 51(2):776–786, 2008.
- [17] S.-S. Tseng, P.-C. Sue, J.-M. Su, J.-F. Weng, and W.-N. Tsai. A new approach for constructing the concept map. *Computers & Education*, 49(3):691–707, 2007.
- [18] A. Vuong, T. Nixon, and B. Towle. A method for finding prerequisites within a curriculum. In *Proceedings of the 4th International Conference on Educational Data Mining*, 2011.
- [19] J. Řihák, R. Pelánek, and J. Nižnan. Student Models for Prior Knowledge Estimation. In *Proceedings of the 8th International Conference on Educational Data Mining*, pages 109–116, 2015.

# Student Emotion, Co-occurrence, and Dropout in a MOOC Context

John Dillon  
Univ. of Notre Dame  
jdillon5@nd.edu

Nigel Bosch  
Univ. of Notre Dame  
pbosch1@nd.edu

Malolan Chetlur  
IBM Research, India  
mchetlur@in.ibm.com

Nirandika Wanigasekara  
IBM Research, India  
nwaniga4@in.ibm.com

G. Alex Ambrose  
Univ. of Notre Dame  
gambrose@nd.edu

Bikram Sengupta  
IBM Research, India  
bsengupt@in.ibm.com

Sidney K. D'Mello  
Univ. of Notre Dame  
sdmello@nd.edu

## ABSTRACT

This paper discusses self-reported emotions experienced by students in a Massive Open Online Course (MOOC) learning context. Emotions have been previously shown to be related to learning in classrooms and laboratory studies and have even been leveraged to improve learning. In this study, frequently occurring discrete emotions as well as frequently, co-occurring pairs of emotions were analyzed during learning with a MOOC. Both discrete and co-occurring emotions were related to students dropping out of the course, illustrating the importance of student emotion in a MOOC context.

## Keywords

MOOC; affective computing; course completion.

## 1. INTRODUCTION

Emotion is one of the key aspects of the learning process [9,22]. It influences learning in a variety of ways [12], both positively (e.g., when a student feels engaged [19]) and negatively (e.g., during boredom [6,19]). These connections between emotion and cognition can be leveraged to improve learning [10]. For example, a dialog-based, intelligent tutor that adjusts its dialog to address negative emotions can improve learning for low-knowledge students [11]. Indeed, the relationship between emotion and learning has been researched in a variety of digital learning contexts in both laboratory studies and classroom studies [1,5,9]. There are, however, additional learning contexts in which the relationship between emotion and learning is less clear. In this study we focus on the role of emotion as it relates to student dropout in the context of a Massive Open Online Course (MOOC).

MOOCs are an online learning context that has recently become popular worldwide [18]. MOOCs provide education access to large groups of people, many of whom are often non-traditional students. Little is known about the relationship between emotions and learning in a MOOC context. Some initial work toward examining emotion in MOOCs indicated that some emotions were related to dropout [13]. However, these results were derived from retrospective reports of emotion after a course rather than reports in the moment, i.e., *during* the course. Similarly, studies have used MOOC discussion forums and clickstream data to infer student emotions such as *Confusion* and *Frustration* based on researchers' judgments of how these emotions are manifested [16,27], but there was no measurement of the emotions from the students themselves.

The current paper expands on this limited research, addressing key open questions about student emotions gathered from self-reports at different points in a MOOC. We explore a range of emotions, including *Anger*, *Boredom*, *Confusion*, *Contentment*, *Disappointment*, *Enjoyment*, *Frustration*, *Hope*, *Hopelessness*, *Isolation*, *Pride*, *Relief*, *Sadness*, and *Shame*, while also focusing on the relationship between *Anxiety* and learning statistics (the focus of the MOOC in this study) [8,17].

We also consider the possibility of co-occurring emotions. Decades ago, Izard et al. [14] considered the possibility that certain emotions may be experienced in concert with other emotions, rather than individually. Experimental research has shown this to be the case in some situations, for example with induced emotions and even with emotions experienced during everyday life [3,21]. In the context of learning, Bosch and D'Mello [4] studied novice programmers' emotions and found *Confusion* co-occurred with *Frustration*, while *Curiosity* co-occurred with *Engagement*. The degree of co-occurrence of *Curiosity* and *Engagement* was positively correlated with learning ( $r = .226$ ) after accounting for individual occurrences, thereby highlighting the importance of examining co-occurring emotions.

In addition to tracking the incidence of emotions and co-occurrence pairs, we also consider how emotions are related to key educational outcomes. Early studies of MOOC data and student behavior [7,26], have often focused on "dropout" as both a problem and a key outcome. Recently, some have questioned the validity of dropout as a metric of outcome assessment [13]. However, Yang et al. [26] have noted, for instance, that the very low completion rates of MOOCs should signal some concern. Researchers have used log data to predict student dropout [15,23] as part of a larger effort aimed at better understanding student dropout from MOOCs and, in turn, improving the MOOC learning experience to reduce dropout. Here, we consider the relationship between students' self-reported emotions and course dropout.

To our knowledge, this is the first study to measure a range of self-reported student emotions in a MOOC context. We believe that the opportunity to study student emotion with large courses in the wild offers a valuable addition to previous work that has focused more on laboratory settings or traditional classroom environments. We address three related questions in this research:

- Q1. What emotions do students experience in a MOOC?
- Q2. Which emotion pairs co-occur more than chance?
- Q3. How do individual and co-occurring emotions relate to dropout?

## 2. METHOD AND COURSE SETUP

“I Heart Stats” was an introductory Statistics MOOC offered by a university in the Midwestern United States. One goal of the course was to alleviate student anxiety towards statistics. In this regard it was a prime opportunity to analyze student affect in a MOOC setting, while also providing an opportunity to study student affect at scale in the wild.

This MOOC contained eight modules covering topics ranging from levels of measurement to ANOVA. Modules were designed to be completed in sequential order. Nevertheless, all modules were released to students at the same time, so students were free to complete the modules at their own pace and in whatever order they desired.

We used a “Pick-Two” list of 15 discrete emotions (Figure 1) to measure student affect. In addition to the typical set of learning-centered affective states like *Confusion* and *Boredom* [9], the list included several additional emotions, such as *Enjoyment*, *Pride*, *Isolation*, *Hope*, and *Shame*. These emotions were, in part, selected from Pekrun’s description of academic emotions [20]. One limitation of this emotion list was that *Neutral* was not included. Students were prompted to report emotions at the start of even-numbered modules (0, 2, 4, 6) as well as at the end of module 8 (last module). We only collected affect reports on every other module to minimize intrusion.

Of the 24,279 students from 183 different countries enrolled in the course, 3,591 students reported exactly two emotions on at least one module. These 3,591 students constituted the sample in this study. Students were able to report greater or fewer than two emotions, but because we were interested in co-occurrence, we excluded responses that did not consist of exactly two emotions.

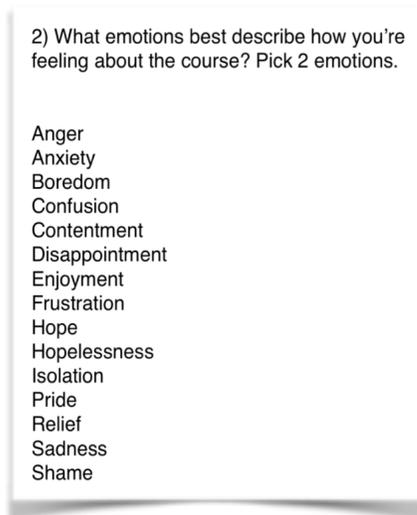


Figure 1. Emotion reporting list

In addition to five “course-level” affect surveys, in which students reported their emotions in relation to the course as a whole, we also included seven “content-level” surveys. These content-level surveys were spread throughout the course and prompted students to report their emotions in response to different video lectures and problem sets. These are two common content-delivery methods for MOOCs, thereby providing a preliminary understanding of student affect when completing these two activities.

## 3. RESULTS

We used both the course-level and content-level students self-reported emotions to answer our research questions (see Introduction).

### Q1. What emotions do students experience in a MOOC?

Figure 2 presents the aggregated proportions of each reported emotion across all five course-level surveys. We note that *Hope* and *Enjoyment* were the most frequently reported emotions. Other frequently reported emotions were *Contentment*, *Anxiety*, and *Pride*, while *Shame*, *Disappointment*, *Isolation*, *Anger*, and *Sadness* were rarely reported.

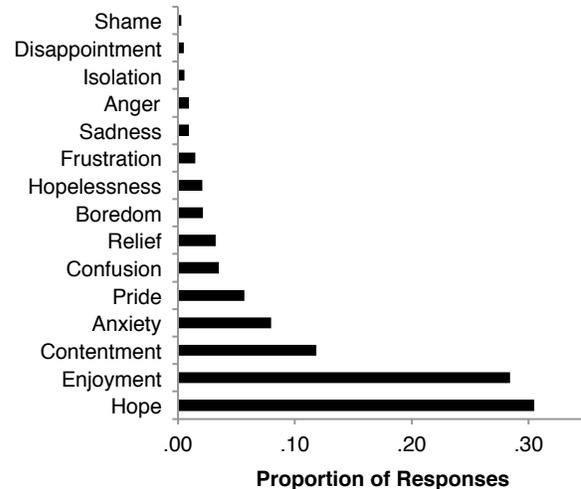


Figure 2. Proportions of self-reported emotions

These results differ from the recent D’Mello meta-analysis [9], where the studies rarely included emotions such as *Hope*, *Enjoyment*, and *Contentment*. However, the focus there was on short one-on-one interactions during learning with technology. A different set of emotions appear to be playing a critical role in the MOOC context, so context clearly matters. It is, however, difficult to separate context differences from measurement differences in the present study.

In addition to the course-level emotion surveys, we also included content-level affect surveys to assess self-reported emotion in relation to specific segments of content that may elicit different emotional responses. We selected 4 content-level affect surveys to highlight different affective states across video and problem set sections of content. Two of the activities were instruction videos and the other two were homework and practice problem sets. We excluded emotions that occurred in less than 1% of the responses for each specific activity. In addition, since all of the content for this course was released at the same time, we use log timestamps to ensure that: 1) Students engaged with the activity, 2) Students answered the activity-specific affect question *after* their engagement with the activity, and 3) Students did not take more than 1 hour following the last activity log to complete the emotion survey.

Figure 3 presents the emotion proportion distributions for four learning activities. The results indicated that unlike the course-level emotion reports, *Enjoyment* was more frequent than *Hope*. Further, while *Anxiety* was the fourth most commonly reported

emotion at a course-level, it was far less prominent at the content level.

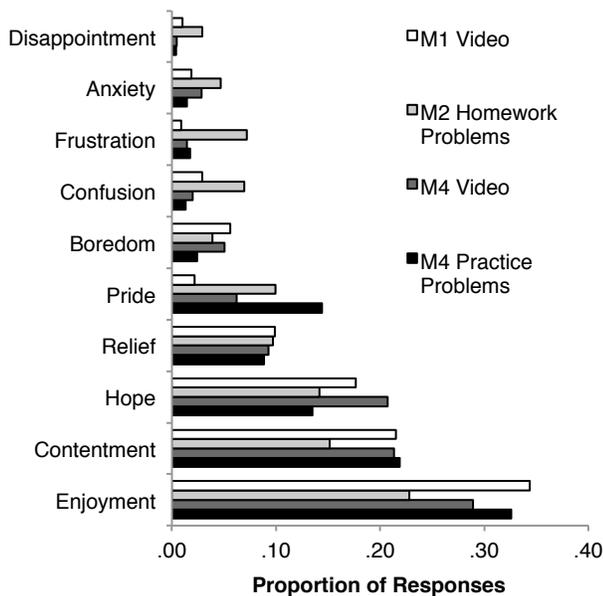


Figure 3. Proportion of emotion self-reports by activity type

We also note that the content-level emotions varied with regard to certain activities. For instance, *Pride* was reported nearly 10 times more frequently in response to Module 4 Practice Problems than in Module 1 Video. *Frustration*, *Confusion*, and *Anxiety* were quite prominent during Module 2 Homework Problem compared with Module 4 Video. *Relief*, on the other hand, did not fluctuate substantially among these four content-level reports. *Hope* was more frequently reported in both of the video activities, while *Pride* was more frequently reported in the problem sets. Further research is needed to determine if indeed students expressed *Pride* more frequently in contexts of achievement such as completing a problem set. We would also need to consider a larger set of activities to establish if certain emotions occur more frequently and significantly among certain genres of content.

These course-level affect surveys highlight that students experience different emotions during different types of content in a MOOC. If MOOCs are able identify the prominent emotions associated with various types of content such as videos and problem sets, then instructors and course designers can provide appropriate support to learners when needed.

## Q2. Which emotion pairs co-occur more than chance?

Bosch and D’Mello [4] investigated co-occurrence of emotions in a computerized learning environment. In their study, they employed a retrospective judgment protocol without any interruptions during the learning session. They determined which co-occurring emotions occurred more than chance by computing Lift scores [24] for each emotion pair. Lift is a technique from association rule learning that can be used to compare the observed co-occurrence of emotions to the level expected by chance. Lift of a pair of emotions (X, Y) is defined as ratio of  $\Pr(X \text{ and } Y)$  to  $\Pr(X) \cdot \Pr(Y)$ .

We identified co-occurring course-level emotions as follows. First, we only considered responses with exactly two emotion

reports. Second, we only considered affective states that occurred at least 1% of the time. This restricted our analysis to *Anxiety*, *Boredom*, *Confusion*, *Contentment*, *Enjoyment*, *Frustration*, *Hope*, and *Pride*. Lift scores were calculated for all pairwise combinations of the above emotions. We used random sampling without replacement (1,000 iterations) and a sample size of 3,000 to compute 95% bootstrapped confidence intervals for the Lift scores. Lift scores above 1.0 with confidence intervals that do not overlap with 1.0 are considered to occur more frequently than chance.

We computed Lift scores for all 5 course-level affect reports. There were 92 distinct co-occurring emotions and a total of 5,189 emotion pairs as reported by 3,591 learners. The results are shown in Table 1. We note that only 5 out of the possible 92 emotion combinations co-occurred at levels above chance and these mainly involved the learning-centered affective states of *Confusion*, *Frustration*, *Boredom*, and *Anxiety*. The *Confusion + Frustration* pair had the highest Lift score, which is consistent with [4] despite considerable differences in the temporal resolution of the analyses. Somewhat surprising is the fact that *Boredom* co-occurred with both *Confusion* and *Frustration*, but this might be attributed to the coarse-grained nature of the emotion self-reports (e.g., *Boredom* could occur for some activities and *Confusion* for others within the same session).

Table 1. Lift of frequently co-occurring emotion combinations

Emotion Pair	Mean (SD)	Confidence Interval
Anxiety + Frustration	1.22 (0.17)	(1.21, 1.22)
Boredom + Confusion	1.06 (0.23)	(1.05, 1.06)
Boredom + Frustration	1.39 (0.43)	(1.39, 1.4)
Confusion + Frustration	3.22 (0.41)	(3.21, 3.23)

## Q3. How do individual and co-occurring emotions relate to dropout?

We coded a student as having “dropped out” if he or she had no interaction events in the last module (Module 8). Table 2 presents partial Spearman’s *rho* between dropout and course-level discrete emotions that comprised at least 1% of the data and corresponding exceeding chance. We partialled out the number of emotion reports per student in order to control for the steep rate of attrition and subsequent dropout bias in our data.

The results indicated that *Anxiety*, *Confusion*, and *Frustration* were significantly positively correlated with dropout, which is what we would expect. It was surprising, however, that *Hope* was also positively correlated with dropout, suggesting that these hopeful students might have become disillusioned by the MOOC. *Relief* was weakly negatively related to dropout, albeit non-significantly.

Table 2. Partial correlations between affect reports and dropout

Emotion/ Combination	$\rho$	$p$
<b>Anxiety</b>	<b>.155</b>	<b>.000</b>
Boredom	.004	.954
<b>Confusion</b>	<b>.122</b>	<b>.019</b>
Contentment	-.035	.243
Enjoyment	-.028	.184
<b>Frustration</b>	<b>.251</b>	<b>.003</b>
<b>Hope</b>	<b>.046</b>	<b>.018</b>
Pride	.034	.476
Relief	-.081	.145
Anxiety + Frustration	.107	.458
Boredom + Confusion	-.088	.684
Boredom + Frustration	-.018	.956
Confusion + Frustration	.177	.263

The most valuable payoffs of this study for learning scientists and MOOC designers are the positive, though weak, correlations between *Frustration*, *Anxiety*, *Confusion* and dropout. The next step is to identify the causes or partial causes of those negative emotions. For example, students reported three times more *Frustration* in Module 2 Homework Problems than in other selected activities, suggesting that the homework problems in this module might need deeper consideration.

#### 4. DISCUSSION

We recorded student affect in a MOOC setting and analyzed them with respect to both individual emotions and co-occurring pairs. This study marks the first large-scale analysis of self-reported emotion in a MOOC context. We found that students experience a rather diverse set of emotions while completing a MOOC in comparison with previous work that has focused on lab- or in-class learning. Particularly interesting was the finding that *Hope*, *Enjoyment*, and *Contentment* were the most frequently reported emotions in the MOOC context, given that they are rare in shorter learning sessions studied in previous work [9].

We also found that some emotions fluctuate depending on MOOC content. This is an especially valuable finding for both instructional designers and researchers. From a learning design perspective, if we know how students are affectively reacting to different types of content, we can adjust the course materials accordingly.

Our findings also contribute to the dropout problem in MOOCs. Despite researchers capacity to predict dropout [25,26], we still lack a robust understanding of student dropout. We identified specific emotions and emotion combinations that correlate with student dropout, yielding an affective perspective to the dropout problem.

#### 5. LIMITATIONS AND FUTURE WORK

There are several limitations with this exploratory study. First, the content was released to students all at once, so they could complete the course in any order they desired. This limits the feasibility of temporal analysis of the data. Second, since this study was based on a live course, we could not ask students to self-report their affective states as frequently as in a lab setting.

This limits use of the data for more fine-grained sequential analyses.

Our analyses also point to several opportunities for future work. One promising avenue is sensor-free affect detection for MOOCs [2]. It would be valuable to model student emotion based entirely on clickstream data provided by edX and other online learning platforms. This would allow for far more frequent affect measurement and more timely affect intervention. If, for instance, we know, based on log data, that a student is *Frustrated*, and we know that *Frustration* correlated with dropout, we can launch pedagogical scaffolds to help the student manage his or her *Frustration*.

A second opportunity for future work is to analyze changes in emotions across the time. There are many questions that can be asked along this front. How do emotions change over the duration of an activity, a session, or the entire course? What is the affective trajectory of a successful MOOC student? Further research is needed to map emotion trajectories over the duration of the course so that we better understand the relationships between emotions, their temporal dynamics, and educational outcomes.

#### 6. ACKNOWLEDGMENTS

We would like to thank Crystal DeJaegher and Xiaojing Duan in the Office of Digital Learning at the University of Notre Dame for their assistance in the design of this MOOC and collection of these data. D’Mello was supported by the National Science Foundation (NSF) (DRL 1235958 and IIS 1523091). 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.

#### 7. REFERENCES

1. Ivon Arroyo, David G. Cooper, Winslow Burleson, Beverly Park Wolf, Kasia Muldner, and Robert Christopherson. 2009. Emotion sensors go to school. *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, IOS Press, 17–24.
2. Ryan Baker, Sujith M. Gowda, Michael Wixon, et al. 2012. Towards sensor-free affect detection in cognitive tutor algebra. *Proceedings of the 5th International Conference on Educational Data Mining*, 126–133.
3. Lisa Feldman Barrett. 1998. Discrete emotions or dimensions? The role of valence focus and arousal focus. *Cognition & Emotion* 12, 4: 579–599.
4. Nigel Bosch and Sidney D’Mello. 2014. Co-occurring affective states in automated computer programming education. *Proceedings of the Workshop on AI-supported Education for Computer Science (AIEDCS) at the 12th International Conference on Intelligent Tutoring Systems*, 21–30.
5. Nigel Bosch, Sidney D’Mello, Ryan Baker, et al. 2015. Automatic detection of learning-centered affective states in the wild. *Proceedings of the 2015 International Conference on Intelligent User Interfaces (IUI 2015)*, New York, NY: ACM, 379–388.
6. Nigel Bosch, Sidney D’Mello, and Caitlin Mills. 2013. What emotions do novices experience during their first computer programming learning session? *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED 2013)*, Berlin Heidelberg: Springer-Verlag, 11–20.
7. Lori Breslow, David E. Pritchard, Jennifer DeBoer, Glenda S. Stump, Andrew D. Ho, and Daniel T. Seaton. 2013.

- Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment* 8: 13–25.
8. Peter K. H. Chew and Denise B. Dillon. 2014. Statistics anxiety update: Refining the construct and recommendations for a new research agenda. *Perspectives on Psychological Science* 9, 2: 196–208.
  9. Sidney D'Mello. 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.
  10. Sidney D'Mello, Nathan Blanchard, Ryan Baker, Jaclyn Ocumpaugh, and Keith Brawner. 2014. I feel your pain: A selective review of affect-sensitive instructional strategies. In *Design Recommendations for Intelligent Tutoring Systems - Volume 2: Instructional Management*, Robert Sottolare, Art Graesser, Xiangen Hu and Benjamin Goldberg (eds.). 35–48.
  11. Sidney D'Mello, Blair Lehman, and Art Graesser. 2011. A motivationally supportive affect-sensitive AutoTutor. In *New Perspectives on Affect and Learning Technologies*, Rafael A. Calvo and Sidney K. D'Mello (eds.). Springer New York, 113–126.
  12. K. Fiedler and S. Beier. 2014. Affect and cognitive processes in educational contexts. *International handbook of emotions in education*: 36–56.
  13. Christian Gütl, Rocael Hernández Rizzardini, Vanessa Chang, and Miguel Morales. 2014. Attrition in MOOC: Lessons learned from drop-out students. In *Learning Technology for Education in Cloud. MOOC and Big Data*, Lorna Uden, Jane Sinclair, Yu-Hui Tao and Dario Liberona (eds.). Springer International Publishing, 37–48.
  14. Carroll E. Izard and Edmund S. Bartlett. 1972. *Patterns of emotions: A new analysis of anxiety and depression*. Academic Press, Oxford, England.
  15. Suhang Jiang, Mark Warschauer, Adrienne E. Williams, Diane O'Dowd, and Katerina Schenke. 2014. Predicting MOOC performance with week 1 behavior. *Proceedings of the 7th International Conference on Educational Data Mining*, 273–275.
  16. Derick Leony, Pedro J. Muñoz-Merino, José A. Ruipérez-Valiente, Abelardo Pardo, and Carlos Delgado Kloos. 2015. Detection and evaluation of emotions in massive open online courses. *Journal of Universal Computer Science* 21, 5: 638–655.
  17. Anthony J. Onwuegbuzie, Denise Da Ros, and Joseph M. Ryan. 1997. The components of statistics anxiety: A phenomenological study. *Focus on Learning Problems in Mathematics* 19, 4: 11–35.
  18. Laura Pappano. 2012. The year of the MOOC. *The New York Times*.
  19. Zachary A. Pardos, Ryan S. J. D. Baker, Maria O. C. Z. San Pedro, Sujith M. Gowda, and Supreeth M. Gowda. 2013. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, ACM, 117–124.
  20. Reinhard Pekrun, Thomas Goetz, Wolfram Titz, and Raymond P. Perry. 2002. Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational Psychologist* 37, 2: 91–105.
  21. Janet Polivy. 1981. On the induction of emotion in the laboratory: Discrete moods or multiple affect states? *Journal of Personality and Social Psychology* 41, 4: 803–817.
  22. Paul Schutz and Reinhard Pekrun (eds.). 2007. *Emotion in Education*. Academic Press, San Diego, CA.
  23. Tanmay Sinha, Patrick Jermann, Nan Li, and Pierre Dillenbourg. 2014. Your click decides your fate: Inferring information processing and attrition behavior from MOOC video clickstream interactions. *arXiv:1407.7131 [cs]*.
  24. Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. 2002. Selecting the Right Interestingness Measure for Association Patterns. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 32–41.
  25. Colin Taylor, Kalyan Veeramachaneni, and Una-May O'Reilly. 2014. Likely to stop? Predicting stopout in massive open online courses. *arXiv:1408.3382 [cs]*.
  26. Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rose. 2013. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. *Proceedings of the 2013 NIPS Data-driven education workshop*, 1–8.
  27. Diyi Yang, Miaomiao Wen, Iris Howley, Robert Kraut, and Carolyn Rose. 2015. Exploring the effect of confusion in discussion forums of massive open online courses. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, ACM, 121–130.

# Semi-Markov model for simulating MOOC students

Louis Faucon, Łukasz Kidziński, Pierre Dillenbourg  
Computer Human Interaction in Learning and Instruction  
École Polytechnique Fédérale de Lausanne  
{louis.faucon,lukasz.kidzninski,pierre.dillenbourg}@epfl.ch

## ABSTRACT

Large-scale experiments are often expensive and time consuming. Although Massive Online Open Courses (MOOCs) provide a solid and consistent framework for learning analytics, MOOC practitioners are still reluctant to risk resources in experiments. In this study, we suggest a methodology for simulating MOOC students, which allow estimation of distributions, before implementing a large-scale experiment.

To this end, we employ generative models to draw independent samples of artificial students in Monte Carlo simulations. We use Semi-Markov Chains for modeling student's activities and Expectation-Maximization algorithm for fitting the model. From the fitted model, we generate simulated students whose processes of weekly activities are similar to these of the real students.

## Keywords

MOOCs; simulation of students; generative models; Expectation-Maximization; Semi-Markov chains; Bayesian statistics

## 1. INTRODUCTION

Vast amounts of data which we gather and analyse in modern learning environments allow us to build models of unprecedented scale and accuracy. This phenomenon, in parallel with developments in computer science, gave rise to new possibilities of inference from educational environments. In particular, the growing field of Simulated Learners [8, 11, 14] provides us with tools for inference from educational simulations.

Inference from any simulations is bounded by the predefined level of abstraction of the analysis. In the context of Massive Online Open Courses (MOOCs), on one hand as an educational institution we have access to only a handful of MOOCs, on another hand, we have data as granular as student's clickstream in a video player. We are therefore obliged to model granularity robustly, depending on the availability of the data. We argue that understanding the properties of the statistical methodology at hand is crucial for successful inference.

We propose a probabilistic model, based on extended version of Markov Chains, called semi-Markov Chains. In the model, we can balance the complexity of the structure and the number of parameters to estimate by cross-validating its parameters. We present an algorithm for fitting the model as well as illustrative examples of the fit on a set of MOOCs.

The contributions of this paper are threefold. First, **we investigate to what extent Semi-Markov chains can be used to describe behavioural patterns of students (RQ1)**. Second, since our model implicitly divides users into clusters, **we analyse if these clusters are interpretable (RQ2)**. Third, **we analyse how these models can be used to infer distributions of events (RQ3)**.

## 2. RELATED WORK

Modeling students is a key concept in learning analytics and educational research in general. Researchers build models predicting motivation and cognition, based on student's goals [19] or they predict goals by motivational traits [7]. Large datasets allow researchers to find predictive power of seemingly slightly related signals like the length of pauses in a video [12] or potentially noisy signals like head movement in the classroom [16].

### 2.1 Generative models in MOOCs

All the aforementioned models are focused on prediction and belong to the class of so-called discriminative models. In this study, we suggest a generative model, which allow us not only to predict, but also to generate observations from the estimated distribution. These models capture the probability structure of input variables and the flow of the processes. Several generative models in MOOCs have been applied, e.g. to forums [3].

Among many generative models that can be encountered in educational research, Markov models were employed for visualization [5], for modeling engagement [17] and for modeling students retention [1].

### 2.2 Simulated learner

The area of simulating students' behaviour lays on the intersection of cognitive science and artificial intelligence. Examples of applications of simulation of students can be found even outside computer science, where the teacher simulates student's response in order to self-improve instructional skills [18]. An acknowledged example of the usage of simulating humans [9] for education deals with simulations of patients behaviour for training medicine students.

Emergence of Internet and new data storage techniques allow re-

searchers to collect and analyse massive amounts of information about the users. Researchers employ simulations for clustering students [13]. For a review of earlier techniques we refer to [2]. We motivate our methodology by the advancements of user modeling in web context [4], as we find this environment conceptually close to the environment of a MOOC.

### 3. GENERAL FRAMEWORK

#### 3.1 Dataset

From our internal MOOC database, aggregating data from Coursera and edX, we extracted events for 61 EPFL courses. The raw data contained approximately 23 million events for 500,000 students, arranged in tuples:  $\langle \text{StudentID}, \text{CourseID}, \text{EventType}, \text{Timestamp} \rangle$ . The *EventType* describes the type of an activity and takes one of four possible values presented in Table 1. We choose these events as the most discriminative actions from the key areas: learning, validation and community engagement. Note that our modelling technique can be easily extended to cover other types of events.

Abbreviation	Description	Proportion
VideoPlay	watching a video	51%
Submission	submitting an assignment	33%
ForumView	visiting the forum	15%
ForumPost	posting on the forum	1%

**Table 1: Distribution of events in the dataset.**

For the analysis we developed our own Python implementation of the algorithm fitting the model<sup>1</sup>. In Section 5 we explain the algorithm in detail. Since 23 million events can still fit in memory of a single computer, we did not require a specific computing architecture to perform the analysis. However, given the considerable size of the dataset, the algorithm takes several minutes to run.

#### 3.2 Definitions

We start with a general framework, in which student’s activity in any MOOC can be very precisely described. Next, we elevate abstraction of the model by adding assumptions simplifying the analysis. Our goal is to introduce a model whose complexity can be adapted to the structure of a course and the amount of available data.

We consider a model in which students behaviour is described in a sequential manner by the type of activity they perform and the time they wait between two sessions. Furthermore, as most of the students perform at most 1 MOOC session per day, we choose a daily granularity of actions.

A sequence of student’s daily activity is described as a list of ‘active events’ (VideoPlay, Submission, ForumView and ForumPost) followed by a ‘end of the day event’ (EndOfDay) or only a EndOfDay in the case the student did not perform any activity the given day. The formal definition of the model is following:

**The set of all students  $\mathcal{S}$ :** We use the symbol  $s \in \mathcal{S}$  to designate an individual student.

**The set  $\mathbf{A}$  of all types of activities:** For this study we chose a set of four types of events: { VideoPlay, Submission, ForumView,

<sup>1</sup>Our implementation is available under <https://github.com/lfaucou/edm2016-mooc-simulator>

ForumPost }. We add to this set one special type of event, EndOfDay. This event corresponds to the end of interactions with MOOCs on a given day. We use the symbol  $a \in \mathbf{A}$  to designate any type of activity. One can extend the set of activities to other events if needed for certain application.

Note that we do not specify the regular ‘end of a course’ event, since we only model the behaviour within the limited time-frame of a course and we treat the last day of the course as the last day of the process. Therefore, each student who went through the whole course without dropping out has just a EndOfDay event on the last day of the course. Number of EndOfDay events is therefore equal to the number of days of the course.

**The random sequential variable  $\mathbf{X}_1^{(s)}, \mathbf{X}_2^{(s)}, \dots, \mathbf{X}_n^{(s)}$**  represents the sequence of activities of one student  $s$ . Each  $\mathbf{X}_i^{(s)} \in \mathbf{A}$  and the sequence stops after an EndOfDay when the student reaches the end of the course. We denote the length of the sequence for a student  $s$  as  $n^{(s)}$ . The observation of one student activity along one MOOC is thus a **realization** of the random sequence  $\mathbf{X}$ .

**The probability distribution  $\mathbf{P}$ :** In general, for each student  $s \in \mathcal{S}$  we can model the  $i$ -th event  $\mathbf{X}_i^{(s)}$  with a probability distribution

$$\mathbf{P}^{(s)}(\mathbf{X}_i^{(s)} = a \mid \mathbf{X}_{i-1}^{(s)}, \mathbf{X}_{i-2}^{(s)}, \dots, \mathbf{X}_1^{(s)}, \mathbf{C}_s),$$

where  $a \in \mathbf{A}$ ,  $\mathbf{X}_1^{(s)}, \dots, \mathbf{X}_{i-1}^{(s)}$  are the previous events of that student and  $\mathbf{C}_s$  are personal characteristics of the student.

This distribution represents the student’s behaviour profile and allows to generate typical sequences of activities. Our main objective is to model this distribution as accurately as possible, given the limited information. The accurate distribution would allow us to draw samples of students.

#### 3.3 Assumptions

As discussed in the previous section, assessing  $\mathbf{P}^{(s)}$  is unfeasible due to dependence on too many events in the past and due to the lack of information on personal student features. In order to fit a probabilistic model we need to relax these dependencies. We introduce following assumptions:

- A1** Students’ behaviours fit into a small number of natural categories of behaviour.
- A2** The type of activity depends only on his previous activity and not on old past activities.

Assumption **A1** maps the space of all possible students’ characteristics into a limited number of categories, which are much easier to attribute. Many studies on MOOCs explicitly classify students into a small number of categories [10], students are divided between ‘Viewers’ who only watch videos, ‘Forum Actives’ who share with their peers in the MOOC discussion forum and ‘Completers’ who succeed in the assignments. As we present in the next section, our method is based on unsupervised clustering, where groups emerge in the way optimal in terms of maximum likelihood of the model.

Assumption **A2** we impose that only the last activity has an impact on the current activity. This assumption is more constraining, but since the complexity of history grows exponentially with the number of steps and, in order to be able to estimate parameters, we have to

reduce the search space. This simplification is usually called the ‘Markov assumption’.

Apart from technical assumptions required for Markov Models, we impose other assumptions for convenience. First, we do not consider length of events, so the VideoPlay event is only the moment when a student starts watching a video. Second, if the series of events happens during midnight, still an event EndOfDay is added to the sequence.

## 4. PROBABILISTIC MODELING

### 4.1 Soft clustering

In Section 3 we proposed a simplified framework, in which we assume that there are only a few different possible classes of students (A1). We enumerate clusters  $1, 2, \dots, K$ . For each student  $s \in \mathcal{S}$  we introduce a probability distribution  $\mu_k^{(s)}$  which describes probability that the student belongs to the behaviour classes  $k$ , for  $k \in \{1, 2, \dots, K\}$ .

This technique is often referred to as *soft clustering*, *weighted clustering* or *fuzzy clustering* [15]. Instead of discrete cluster assignment, as for example in  $K$ -means, we obtain for each student a probability distribution among the clusters. These probabilities can be intuitively seen as our certainty that the student belongs to a given cluster.

### 4.2 Semi-Markov Chain

Assumption (A2), i.e. dependence only on the last state, allows us to model the process Markov Chains. Formally, in the definition of distribution of the next event we can drop dependence of the events which occurred before the current one, i.e. we identify

$$\mathbf{P}^{(s)}(\mathbf{X}_i^{(s)} | \mathbf{X}_1^{(s)}, \dots, \mathbf{X}_{i-1}^{(s)}) = P^{(s)}(\mathbf{X}_i^{(s)} | \mathbf{X}_{i-1}^{(s)})$$

A preliminary analysis revealed an important weakness of using classic Markov Models in our context. A traditional Markov model considers that a student is equally likely to stop watching videos when they have watched one, as when they have already watched ten videos. In practice, students watch videos sequentially and Markov Model does not capture appropriately the number of events in the sequence.

To remedy this issue we employed Semi-Markov Models (also called Markov Renewal Processes). The key feature of this model is that it allows to replace the self-loops (transitions from one event type to itself) in the Markov Chain, by a probability distribution of the number of repetition of a given state.

In Semi-Markov Models, we still need to choose a parametric distribution, but we have more freedom than in traditional Markov Chain. Markov Chain implicitly assumes that probability of staying in the same state is the largest for 1 step and decreases with number of steps. However, we would expect that 1 is not the most probable number of repetition at least for a particular group of students. This phenomenon can be captured by, for example, Poisson distribution, which proved to be more accurate in our preliminary analysis. Thus, for an event  $a \in A$  and a class  $k$  we model the number of repeated events  $R_a^k$  by

$$\mathcal{P}(R_a^k = r) = \frac{e^{-\lambda_a^k} (\lambda_a^k)^r}{r!}$$

where  $r$  is the number of repetitions and  $\lambda_a^k$  is the average number of repetition and needs to be estimated from the data for each  $k$  and  $a$ .

To illustrate that the Poisson distribution improves the model, let us consider an example. Suppose we expect that some group of students connects to a MOOC twice a week, with approximately three days interval between connections. In that case, the average number of repetitions of the EndOfDay event is 3. Simple Markov Model, accurately models the average to be 3 but implicitly assumes that the majority of students gets only 1 repetition. Semi-Markov model with Poisson distribution also gives the average equal to 3 and the distribution is concentrated around 3.

## 5. FITTING THE MODEL

### 5.1 Algorithm

The Expectation-Maximisation (EM) algorithm has been introduced in 1977 in [6]. The goal of this iterative technique is to compute the parameters that maximize the likelihood of a given probabilistic model. The EM algorithm has been proven to converge at least to a local minimum. This minimum depends on the initialization point, thus multiple runs with different random initialisations are often used in practice in order to increase the chances of finding the global minimum.

In this study we use the EM algorithm for unsupervised learning. Neither the parameters of the latent classes nor the repartition of the students are known at the beginning and the algorithm has to estimate both quantities at once. In our settings, we define for each  $k \in \{1, 2, \dots, K\}$  and states  $a$  and  $b$ :

- $p_{b \rightarrow a}^{(k)}$ , the probability that a student with the behaviour profile  $k$  performs the activity  $a$  after the activity  $b$ :

$$p_{b \rightarrow a}^{(k)} = \mathbf{P}(\mathbf{X}_i = a | \mathbf{X}_{i-1} = b)$$

- $\lambda_a^{(k)}$ , the average number of repetitions of an event  $a$  from a student of profile  $k$ .

- $\mu_k^{(s)}$ , the probability that a student  $s$  belongs to the profile  $k$ .

We can thus compute the likelihood of the observed sequence, as a function of cluster repartition and parameters of Markov Chains by

$$likelihood = \prod_{s \in \mathcal{S}} \left[ \sum_{k=1}^K \mu_k^{(s)} \prod_{(a,b,r) \in \mathbf{T}_s} p_{b \rightarrow a}^{(k)} \mathcal{P}_{\lambda_a^{(k)}}(r) \right], \quad (1)$$

where  $\mathbf{T}_s$  is the set of tuples  $(a, b, r) \in \mathbf{A} \times \mathbf{A} \times \mathbf{N}$  corresponding to transitions from activity  $b$  to activity  $a$  with  $r$  repetitions of activity  $a$ . The goal of the algorithm is to find the parameters that maximize the likelihood.

In the first stage, the algorithm initialize randomly  $K$  profiles. Next, it iteratively improves the *likelihood*, by alternating two steps as described below. In each step it modifies the repartition or the Markov chain parameters.

**Initialization:** The initialization consists in choosing randomly either the  $p_{b \rightarrow a}^{(k)}$  and  $\lambda_a^{(k)}$  or the  $\mu_k^{(s)}$ . In our algorithm, we start

with the  $\mu_k^{(s)}$ . This can be done by generating a random number  $k^*$  from 1 to  $K$  for each student  $s$  and by setting

$$\mu_k^{(s)} = \begin{cases} 1 & \text{if } k = k^* \\ 0 & \text{otherwise.} \end{cases}$$

**Iterations:** The iteration phase has two steps. First, we compute the optimal values for  $p_{b \rightarrow a}^{(k)}$  and  $\lambda_a^{(k)}$  given that  $\mu_k^{(s)}$  are fixed (equations (2) and (3)).

$$p_{b \rightarrow a}^{(k)} = \frac{\sum_{s \in \mathcal{S}} \sum_{(a,b,-) \in \mathbf{T}_s} \mu_k^{(s)}}{\sum_{s \in \mathcal{S}} \sum_{(-,b,-) \in \mathbf{T}_s} \mu_k^{(s)}} \quad (2)$$

$$\lambda_a^{(k)} = \frac{\sum_{s \in \mathcal{S}} \sum_{(a,-,r) \in \mathbf{T}_s} r \mu_k^{(s)}}{\sum_{s \in \mathcal{S}} \sum_{(a,-,-) \in \mathbf{T}_s} \mu_k^{(s)}} \quad (3)$$

Next, we compute the new values of  $\mu_k^{(s)}$  according to the new  $p_{b \rightarrow a}^{(k)}$  and  $\lambda_a^{(k)}$  (equations (4)).

$$\mu_k^{(s)} = \frac{\prod_{(a,b,r) \in \mathbf{T}_s} p_{b \rightarrow a}^{(k)} \mathcal{P}_{\lambda_a^{(k)}}(r)}{\sum_{c=1}^K \prod_{(a,b,r) \in \mathbf{T}_s} p_{b \rightarrow a}^{(c)} \mathcal{P}_{\lambda_a^{(c)}}(r)} \quad (4)$$

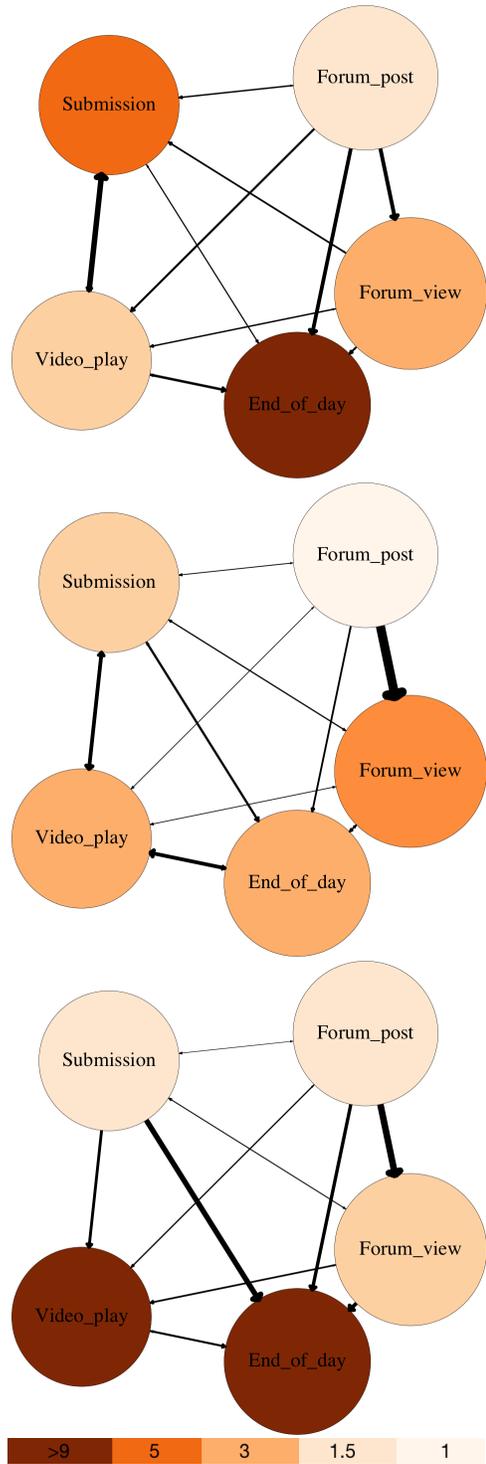
Intuitively, in the first step we compute the parameters of the latent classes given the repartition of the students and in the second step we recompute the repartition from the new classes parameters.

## 5.2 Example: Interpretation clusters (K=3)

Before we present the results for the choice of the number of clusters, in this section, we illustrate the behaviour of the algorithm and the model when the number of clusters is small ( $K = 3$ ). Although in this case we may lose important variability among groups of students, small number of clusters allows us to visualise the Semi-Markov models and interpret each of the clusters.

The visualizations of the Semi-Markov models on Figure 1 can reveal general characteristics of students' behaviours. For example, Profiles 1 and 3 are in general less active as they have more EndOfDay events. On the contrary, Profile 3 has a very high average number of repetition on VideoPlay and considerable probability to go back to EndOfDay events. This means that students of this cluster are not fully engaged in all MOOC activities.

A more insightful way to analyse and interpret the differences is to generate sequences of events and compare the outcomes. We can compute the expected number of videos watched or the expected number of post on the forum directly from simulated sequences. Table 2 shows the average number of several types of events for 100 simulated students (average from 10000 simulations) over four weeks generated with the three Markov models from Figure 1. For



**Figure 1: Three graphical representations of behaviour profiles extracted by the EM algorithm. From top to bottom: profiles 1, 2 and 3 (thickness: transition probability; color: average number of repetitions)**

example, we can see that students of Profile 1 participate in the collaborative activities of the MOOC more rarely, but engage in the assignments more than in watching the videos. This might indicate

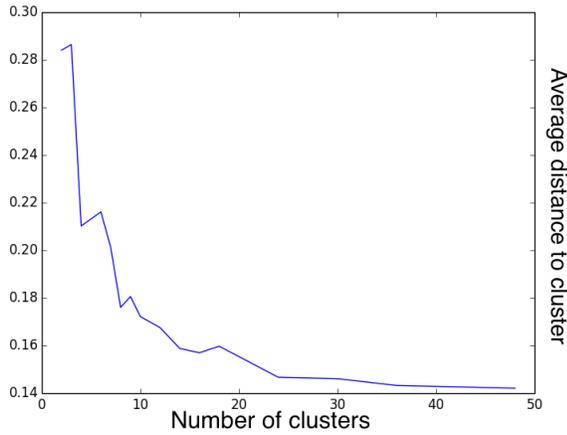
that they already have a good understanding of the content of the course and do not need to spend more time on studying. To fully investigate this hypothesis, further analysis should be conducted.

Profiles	1	2	3
Watched Videos	1060	3133	2363
Submissions	1535	2423	442
Forum Visits	68	1711	255
Forum posts	3	96	15

**Table 2: Average number of events for 100 students over the first four weeks of the MOOC**

### 5.3 Choice of the parameter K

A common challenge of unsupervised learning and fitting a probabilistic model is finding the correct number of classes. In our case, the similarity of the algorithm with other clustering techniques such as the K-means leads to the "elbow heuristic", often used in practice. The idea is to choose the number of clusters large enough to explain a large part of the variability, but such that a greater number of clusters would not explain substantially more.



**Figure 2: Average distance of students from their model for different number of classes**

In order to confirm the result of this first measure of quality, we designed another measure described in the equation (5). The goal is to quantify how the students sequences diverge from their attributed cluster. In the equation,  $|A|$  is the cardinality of the set of possible activities,  $p_s(a)$  is the probability of finding the activity  $a$  if we take uniformly at random an activity of student  $s$  and  $p_k(a)$  is the probability of finding the activity  $a$  if we take uniformly at random an activity from a sequence generated by the class  $k$ .

$$d^2(s, k) = \frac{1}{|A|} \sum_{a \in A} (p_s(a) - p_k(a))^2 \quad (5)$$

This distance measure shows an elbow shape for the same values of  $K$  between 10 and 15 as it can be seen on Figure 2. We conclude that MOOC students from our dataset can be meaningfully clustered into 10 – 15 different classes.

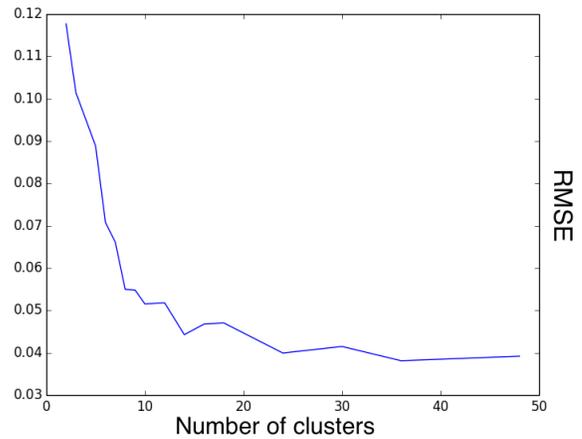
## 6. SIMULATIONS

With a model fitted with the EM algorithm at hand, the algorithm repartitioned students and chose parameters of a Semi-Markov Chain for each of the clusters. Since both the repartition and the Semi-Markov Chains are generative, we can draw samples from the fitted distribution, i.e. we can simulate the students. We run the simulations and show a possible way to measure the validity of the results.

To validate potential value of simulations, we first propose a simple accuracy measure. In equation (6),  $P_{real}(|a| > n)$  represents the probability that a student performs more than  $n$  events of type  $a$  during the time of the MOOC.  $|a|$  is the count of events of type  $a$ .  $P_{sim}(|a| > n)$  represents the same probability but for a simulated student. In the measure we chose the value  $N = 50$  because it covers most of the variability in the students activity sequences and is not too large as still 19% of the students have an activity with more than 50 repetitions.

$$MSE = \frac{1}{(|A| - 1) * N} \sum_{a \in A} \sum_{n < N} (P_{real}(|a| > n) - P_{sim}(|a| > n))^2 \quad (6)$$

In order to prove the correctness of the modeling method, we divided our dataset into a training set and a test set for validating the results. The first step is to run the algorithm on the training set with several parameter  $K$  and then, use the computed parameters to simulate a new population of students and finally compare this population with the students from the testing set. In Figure 3 we can see that the fit does not improve much after  $K = 15$ , because too high number of clusters makes the algorithm learn mostly the noise from the random actions of the students instead of their real intrinsic behavioural patterns.



**Figure 3: Measure of accuracy of a simulation for different number of classes**

The small error proves that the distribution obtained from simulations is close to the original distribution. This implies that the model properly trained on small sample of students or on just few first events, can be extrapolated by simulation to further events or larger samples.

In an experimental setup, simulations with varying initial conditions of the model (e.x. probabilities of transitions) can give us distributions of events at the later state. Knowing probability distributions of the results of two conditions allows to estimate sample sizes needed for finding statistical evidence of the investigated effect.

## 7. DISCUSSION

In Section 5 we showed that Semi-Markov chains can be successfully applied to describe behavioural patterns of students (RQ1). In Section 5.2, a simple study with reduced number of clusters prove their potential interpretability (RQ2). In Section 6, we discuss how these models can be used to infer distributions of events (RQ3).

Our method has two main limitations. They can be further relaxed with additional data or with incorporation of domain knowledge.

**The Homogeneity of the Markov process:** The Markov assumption was introduced for reducing the number of parameters of our model. It is a strong simplification, which entails some drawbacks. This assumption implicitly requires that student behave with exactly the same transition matrix during the whole course. The motivation to keep learning should increase when getting closer to the end of the course and thus the dropout rate decreases, which cannot be capture by our method. A good way to overcome this weakness is to use inhomogeneous Markov models with transitions probabilities that are functions of time.

**Differences between courses:** The quality of the videos, the level of difficulty of the assignments or the discussion topics in the forums are all factors that can greatly influence the behaviour of a student. None of these were included in our model. We hypothesize that adding external annotations that would impact the transition probabilities of our Markov models could help solve this problem. As for now, our model can be used to compare courses. For example, if we run the algorithm on two MOOCs and realise that the Video Watchers of one course have a lower engagement, that shows a lower quality of video content while differences for the Forum Follower may reveal differences on the quality of the Forum discussions.

## 8. REFERENCES

- [1] Girish Balakrishnan and Derrick Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 2013.
- [2] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.
- [3] Christopher G Brinton, Mung Chiang, Sonal Jain, HK Lam, Zhenming Liu, and Felix Ming Fai Wong. Learning about social learning in moocs: From statistical analysis to generative model. *Learning Technologies, IEEE Transactions on*, 7(4):346–359, 2014.
- [4] Ed H Chi, Peter Pirolli, Kim Chen, and James Pitkow. Using information scent to model user information needs and actions and the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 490–497. ACM, 2001.
- [5] Carleton Coffrin, Linda Corrin, Paula de Barba, and Gregor Kennedy. Visualizing patterns of student engagement and performance in moocs. In *Proceedings of the fourth international conference on learning analytics and knowledge*, pages 83–92. ACM, 2014.
- [6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.
- [7] Andrew J Elliot and Todd M Thrash. Approach-avoidance motivation in personality: approach and avoidance temperaments and goals. *Journal of personality and social psychology*, 82(5):804, 2002.
- [8] José P González-Brenes and Yun 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.
- [9] James A Gordon, William M Wilkerson, David Williamson Shaffer, and Elizabeth G Armstrong. "practicing" medicine without risk: students' and educators' responses to high-fidelity patient simulation. *Academic Medicine*, 76(5):469–472, 2001.
- [10] René F Kizilcec, Chris Piech, and Emily 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.
- [11] Kenneth R Koedinger, Noboru Matsuda, Christopher J MacLellan, and Elizabeth A McLaughlin. Methods for evaluating simulated learners: Examples from simstudent. *17th International Conference on Artificial Intelligence in Education AIED*, 5:45–54, 2015.
- [12] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. How do in-video interactions reflect perceived video difficulty? In *Proceedings of the European MOOCs Stakeholder Summit 2015*, number EPFL-CONF-207968, pages 112–121. PAU Education, 2015.
- [13] Ran Liu and Kenneth R Koedinger. Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities. In *Educational Data Mining 2015*. EDM, 2015.
- [14] Gord McCalla and John Champaign. Aied 2013 simulated learners workshop. In *Artificial Intelligence in Education*, pages 954–955. Springer, 2013.
- [15] Richard Nock and Frank Nielsen. On weighting clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8):1223–1235, 2006.
- [16] Mirko Raca, Łukasz Kidziński, and Pierre Dillenbourg. Translating head motion into attention-towards processing of student's body-language. In *Proceedings of the 8th International Conference on Educational Data Mining*, number EPFL-CONF-207803, 2015.
- [17] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. Modeling learner engagement in moocs using probabilistic soft logic. In *NIPS Workshop on Data Driven Education*, 2013.
- [18] Ipke Wachsmuth and Jens-Holger Lorenz. Sharpening one's diagnostic skill by simulating students' error behaviors. *Focus on learning problems in mathematics*, 9(2), 1987.
- [19] Christopher A Wolters. Advancing achievement goal theory: Using goal structures and goal orientations to predict students' motivation, cognition, and achievement. *Journal of educational psychology*, 96(2):236, 2004.

# Investigating Gender Differences on Homework in Middle School Mathematics

Mingyu Feng  
SRI International  
333 Ravenswood Ave  
Menlo Park, CA 94025  
1-650-859-2756  
mingyu.feng@sri.com

Jeremy Roschelle  
SRI International  
333 Ravenswood Ave  
Menlo Park, CA 94025  
1-650-859-3049  
jeremy.roschelle@sri.com

Craig Mason  
University of Maine  
5766 Shibles Hall  
Orono, ME 04469  
1-207-581-9059  
craig.mason@maine.edu

Ruchi Bhanot  
SRI International  
333 Ravenswood Ave  
Menlo Park, CA 94025  
1-650-859-5381  
ruchi.bhanot@sri.com

## ABSTRACT

Recent studies [10, 23] using US nationwide databases showed high school boys spent significantly less time doing homework than girls, based on their responses to questionnaires and surveys. To investigate gender differences in homework in middle school, in this paper, we analyzed computer log data and standardized test scores of more than 1,000 7<sup>th</sup> grade students who participated in a large-scale randomized controlled online homework efficacy study. Students used the ASSISTments platform to do their homework for a school year. Our results suggested no significant difference between the time the two genders spent on homework overall. There was a marginally significant difference on homework time between genders in the high performing group only. When examining the system-student interaction data, we found significant difference between boys and girls in their help-seeking behaviors. In addition, we found out that boys have benefited from the online homework intervention more than girls.

## Keywords

Gender gap, homework, online homework intervention

## 1. INTRODUCTION

Studies have investigated gender differences in homework completion rates, learning habits, and technology use outside of school. The investigation into gender differences found that girls spend more time on homework [36], including math [28]. Further, research has also shown that girls are more likely to spend time regulating study habits (e.g., time management, engaging in emotion self-regulation while doing homework) [13, 16, 37, 38, 39]. This was especially true with girls receiving family help while doing homework [35, 36]. With regards to gender differences in technology out of school, research clearly indicates boys have an advantage over girls with using technology for more varied reasons (e.g., programming, gaming, or internet surfing) than girls (e.g., drawing) [33] and more frequently as well [12, 17, 22, 24, 27, 34]. This gender-based advantage extends to girls' attitudes towards computer usage. Girls tend to exhibit lower self-efficacy beliefs about their use of computers [21, 33]. At the same time, studies also document parent support as a critical mitigating factor that can increase girls' use of and experience with

computers [21, 33].

More recently, two studies, [10] and [23] suggested that boys spend less time on homework than girls. Based on the PISA 2012 Database, [23] shows that around the globe, 15-year-old boys are overwhelmingly less likely than girls to spend time doing homework, which may in part explain why they are more likely to struggle academically. The study has been widely cited in recent press coverage (e.g. [26]). In the U.S., boys on average spend 1.8 hours less time per week doing homework than girls. When considering boys and girls who spend the same amount of time doing homework, the gender gap in mathematics achievement is wider. [10] examined data from American Time Use Survey (AUS) responses. They showed that high school girls spent statistically significantly more time (17 minutes per day) on homework than male high schoolers, even after controlling for SES indicators, daily activities and other factors. Furthermore, the gap for time spent on homework is largest among high-achieving students.

These studies illustrate that achievement gaps between genders' use of homework does exist. However, we noticed almost all studies on gender differences in homework use self-reported measures. PISA 2012 asked students to report how much time per week they spend doing homework by teachers. [10] used students' non-school study time using time diary data from 2003-2013 waves of the AUS and transcript data from the Educational Longitudinal Study of 2002 (ELS). Our literature search shows that there is a serious need for rigorous homework research on homework in K-12 settings. The existing studies are mostly correlational survey studies with thousands of students that relate homework time, academic-, and non-academic outcomes.

Our online homework study, which is a rigorously designed, randomized controlled experiment, gives us a unique opportunity to study the gender gap using more objective data sources of homework—computer logs from an online platform that support *middle school* students doing math homework. In this paper, examine the difference between genders in middle school mathematics on

- homework time
- the amount of problems completed by each gender
- homework performance
- how each gender interacted with the system
- whether there was any difference in the outcome measure between the two genders
- which gender benefited more from the technology-based intervention

## 2. BACKGROUND

### 2.1 Online homework study

Research has been conducted to study the role and practices of homework and its relationship with student learning, particularly for mathematics (e.g. [2, 3, 5, 8, 9, 19, 20, 28, 29]). The link between homework assignments and student achievement is far from clear across the board, as noted by Cooper and others [30]. Although some studies show that students—and especially struggling students—could benefit from middle school mathematics homework, they may not benefit under typical conditions. Technology-based learning environment, such as ASSISTments, provides ways to make homework more adaptive and productive for the students who could benefit most. These environments can also do some of the bookkeeping and help teachers to keep track the progress of their students. They enable teachers to assign customized homework to their students. For example, while doing homework in ASSISTments, students receive support including immediate feedback on the accuracy of their answers, as well as extensive tutoring. With these supports in place, students may complete more homework and learn more while doing homework. Teachers may be freed from the tedium of grading homework and be able to instead focus their energies adjusting and differentiating instruction.

SRI International, in conjunction with the University of Maine and Worcester Polytechnic Institute (the developer of the ASSISTments platform) conducted a multiyear randomized controlled efficacy trial at the school level. The study was conducted in 44 schools in the state of Maine, where one-to-one computing has been well-established for over 10 years. This experiment tested the hypothesis that the ASSISTments homework support improves student mathematics outcomes and will also examine impacts for struggling students and other important demographic groups. Schools in the study were randomly assigned to treatment or control (i.e. “business as usual”) conditions. The intervention was implemented in 7<sup>th</sup> grade classrooms in treatment schools over 2 consecutive years. In the control condition, teachers and students continue with their existing homework practices. In the treatment condition, teachers received professional development and used ASSISTments in the first year to become proficient with the system and then teachers used ASSISTments with a new cohort of students in the second year which is considered the “experiment year”. At the end of the experiment year, students were administered the TerraNova Common Core math test to provide data on student achievement in mathematics. TerraNova is a norm-referenced achievement test that is nationally normed. It generates scaled scores (ranging from 400 to 900 points) and achievement-level information that include five levels of performance proficiency (1: Starting-out; 2: Progressing; 3: Near Proficient; 4: Proficient; 5: Advanced).

### 2.2 Key features of ASSISTments platform

ASSISTments ([www.assistments.org](http://www.assistments.org)) [6, 14] is an online tutoring system that provides “formative assessments that assist.” Teachers choose (or manually add) homework items in ASSISTments and students can complete their homework online. As students do homework in ASSISTments, they receive feedback on the accuracy of their answers. Some problem types provide hints to help students improve their answers, or help decompose multistep problems into parts (so-called “scaffolding questions”)

Marty surveyed 24 students and asked them to name their favorite fruit. The circle graph below shows the results of his survey.

Which fruit was the favorite of exactly 6 of the students?

Students' Favorite Fruits

Select one:

bananas

grapes

oranges

apples

✖ Sorry, try again: "bananas" is not correct

Submit Answer

Break this problem into steps

First let's make a ratio in the form of a fraction. Comment on this problem

Which of the following is the correct ratio for the six students who like a particular fruit to all the students surveyed? (students / total students)

Our ratio will be: small group of students / all students in survey

Comment on this hint

Select one:

6/24

24/6

18/24

24/18

Submit Answer

Show hint 2 of 3

Figure 1. Screen shots of an 7<sup>th</sup> grade item in ASSISTments that provides correctness feedback and breaks the problem into steps.

(see Figure 1). Teachers may choose to assign problem sets called “skill builders” that address individual math concepts and skills at grade level and are organized to promote mastery learning. Every night, ASSISTments servers generate customized, cognitive diagnostic reports. The reports show teachers homework completion rates, performance data for each student on every problem and each math skill covered in the assignment, which questions and/or skills were particularly challenging for, and what the common wrong answers were. The report is emailed to teachers early in the morning for their review. This data allows teachers to make real-time, informed decisions about what and how they teach, and it is ideally used to guide homework review practices in class.

The usage model of the online homework study specifies that teachers who used ASSISTments in the study were expected to assign approximately 20 minutes of homework in ASSISTments for a minimum of three nights per week (making adjustments as needed to accommodate district and school homework policy).

## 3. EXPLORING GENDER DIFFERENCES IN HOMEWORK TIME, BEHAVIORS, AND PERFORMANCE

The data used in this section includes ASSISTments system logs of 1033 7<sup>th</sup> grade students, including 514 boys and 519 girls, who participated the second year of the homework study in the treatment condition. Also included in the data are their TerraNova test scores including both scaled scores and their performance levels. These students used ASSISTments for homework for the whole school year.

### 3.1 Features

We started the data analysis by constructing features that represent student’s intensity of use, performance, and behaviors while working in ASSISTments. Below, we list all the features.

- mins\_s: Total number of minutes students spent on homework in the year
- probs\_c: Total number of problems completed
- perc: Average percent correct over all assignments
- hint\_c: Average number of hint requests per problem

- *attempt\_c*: Average number of attempts<sup>1</sup> per problem
- *bottom\_hint\_c*: Average number of bottom-out hint<sup>2</sup> requests per problem
- *comp\_perc*: % of homework assignments completed on time
- *late\_perc*: % of assignments completed but late

Two features, *mins\_s* and *probs\_c* are measures of intensity of use of ASSISTments. *perc* is a measure of student’s performance on homework problems. Some other system features (*hint\_c*, *attempt\_c*, and *bottom\_hint\_c*) capture students’ interaction with the system while doing homework, including their help-seeking behaviors (*hint\_c* and *bottom\_hint\_c*) and the number of attempts they made before getting a correct answer (*attempt\_c*). The last two features show whether they complete their homework on time or late (*comp\_perc*, *late\_perc*) as opposed to not completing an assignment at all.

### 3.2 Visual exploration of homework time

Research has shown that spending more time doing homework is better for academic achievement [3, 28, 30, 32]. Additional research has also shown that homework time is associated with many factors that may have a positive effect on academic success such as motivation or academic interest [4, 15] and parent involvement [1, 25]. Therefore, we started with an exploratory analysis focusing on the time students have spent on doing homework in ASSISTments. We observed relatively weak positive relationships<sup>3</sup> ( $.2 < r < .4$ ) between students’ TerraNova scaled scores and system use and performance indicators (*mins\_s*, *probs\_c*, and *perc*), suggesting students who spent more time on homework and completed problems scored higher on the TerraNova test. When we examined the usage data closely, we found that students spent a wide range of time on homework in ASSISTments in the school year (ranging from 2 to 4,238 minutes, mean = 640, standard deviation = 784), and amount of use varies a lot by schools (65% of the variance in *mins\_s* is accounted by schools). Although homework practice is expected to differ across teachers and schools, the large variance is to some extent surprising, as the research team has specified a desired use model and has expressed the expectations clearly to all teachers in the treatment schools. On the other hand, this result confirms our previous findings on implementation fidelity from the previous 2013-14 school year where adherence, exposure, and uptake of users varied by teachers [7].

Next, we further explored the relationship between homework time and students’ achievement outcomes. We found that higher-performing students tend to spend more time on homework. Girls seem to spend relatively more time on homework than boys do, except in the middle level of achievement. The difference is most notable in the “5: Advanced” level.

Then we compared the TerraNova performances of boys and girls who spend similar amounts of time on homework. We found that there are more girls than boys who spent a significant amount of time on homework (defined as over 3,200 minutes in the school

year). Unlike [23], however, we didn’t see big gender gaps in mathematics achievement (Figure 3).

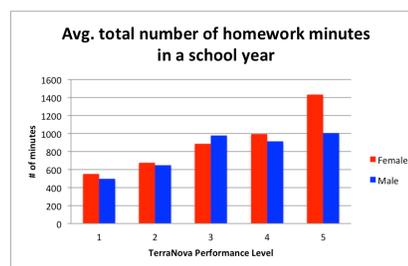


Figure 2. Bar graph comparing the average homework time by students in each TerraNova performance level, split by gender

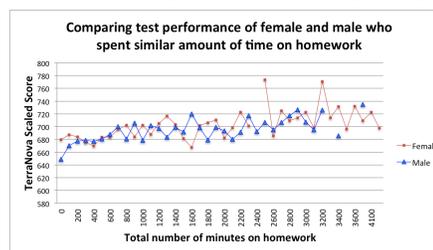


Figure 3. Plot comparing TerraNova performance of boys and girls who spend similar amount of time on homework

### 3.3 Modeling gender difference on usage and homework performance

Table 1 shows the descriptive statistics of all the features by gender. We noticed that the mean difference between the two genders were high on the two features, *mins\_s* and *probs\_c*, yet standard deviations on those measures were also quite high. We understood the extent to which schools create variation in homework behaviors: differences in the amount of homework assigned between teachers and schools, possible variations in homework review processes, and differences in teachers’ completion policies. Since these factors could affect students’ performance and/or behavior when doing homework, we trained a series of 3-Level Hierarchical Linear Regression models (HLM) (students nested in classes and classes in schools) to account for the difference in schools’ and teachers’ homework assignment practices. We used each feature as a dependent variable and use *gender* of students as the predictor (male = 0, female = 1).

Table 1. Descriptive statistics of features by gender

Features	Male		Female	
	Mean	Stdev	Mean	Stdev
<i>mins_s</i>	820.337	759.742	874.755	807.623
<i>probs_c</i>	703.214	592.099	770.734	621.226
<i>perc</i>	0.740	0.115	0.744	0.117
<i>hint_c</i>	0.115	0.143	0.094	0.103
<i>attempt_c</i>	1.403	0.281	1.375	0.248
<i>bottom_hint_c</i>	0.072	0.074	0.061	0.068
<i>comp_perc</i>	0.614	0.284	0.646	0.259
<i>late_perc</i>	0.129	0.12	0.14	0.127

As shown in Table 2, the results suggest that overall there is no significant difference between girls and boys in terms of the amount of time they spent on homework or the number of problems they completed. Furthermore, there is no difference between the two genders in their rates of correctly answered

<sup>1</sup> The system doesn’t limit the number of answers a student could attempt on a problem.

<sup>2</sup> When using ASSISTments in the practice and learning modes (as opposed to testing mode), the system requires that every problem has to be answered correctly in order for students to move to the next one. Bottom-out hints in ASSISTments reveal the correct answer to students so that they won’t get stuck.

<sup>3</sup> No other correlations were noticed

problems in ASSISTments. Girls tend to complete more assignments on time than boys, but the difference is only marginally significant ( $p = .086$ ). However, interestingly, girls and boys interacted with the system differently; girls made fewer hint requests and fewer attempts on problems, and they also requested fewer bottom-out hints as compared to boys in the same classes.

**Table 2. HLM Results Overall – Predictor: Female**

Dependent Variable	Difference	<i>p</i>
mins_s	22.482	0.350
probs_c	27.219	0.177
perc	0.006	0.351
hint_c	-0.018	0.005**
attempt_c	-0.039	0.005**
bottom_hint_c	-0.011	0.002**
comp_perc	0.015	0.086
late_perc	-0.000	0.907

Inspired by Gershenson & Holt (2015) and Figure 3 shown above, we were interested to see whether there was any interaction effect between gender and students' performance levels. Thus, we split the students into 3 groups based on their performance level on the TerraNova test. We then trained similar HLM models within each group of students, and the results are shown in Table 3.

- **Progressing or Below:** performance levels = 1 or 2; N = 328 (male: 145, female: 183)
- **Near Proficient:** performance levels = 3; N = 368 (male: 165, female: 203)
- **Proficient or Above:** performance levels = 4 or 5; N = 337 (male: 166, female: 171)

We observed trends with regard to how students interact with the system in both the *Near Proficient* and *Proficient or Above* groups. The trends are consistent with the overall trend: girls requested significantly fewer regular hints or bottom-out hints, and made fewer attempts on problems. Results regarding assignment completion status are mixed. Low-performing girls completed fewer assignments after they were due than low-performing boys did; yet in the *Near Proficient* group, girls completed more assignments late than boys did. In the *Proficient or Above* group, girls were more likely to complete assignments on time. Interestingly, we noticed a marginally significant difference in *mins\_s* in the *Proficient or Above* group, suggesting high-performing girls spent more time on homework than high performing boys. This result is in consistent with [10], but the latitude of difference is not as big.

#### 4. WHICH GENDER BENEFITED MORE FROM TECHNOLOGY-BASED HOMEWORK INTERVENTION?

One of the research questions of the online homework study is to investigate whether the impact of the ASSISTments on learning

outcomes differ by student demographic characteristics. Here we present the analysis that was conducted to examine which gender benefited more from online homework intervention. A different dataset was used for this analysis. Students from the control condition were included in this dataset in order to detect the interaction between intervention and gender, which increased the total number of students to 2,756 from 44 schools. Only students' assigned condition, gender, their 6<sup>th</sup> grade state test scores, and their TerraNova scaled scores were included in this dataset. TerraNova scores were used as dependent variable. 3-level HLM models were employed in all the analysis.

We first ran a basic model that includes prior achievement (6<sup>th</sup> grade math state test scores) and gender (male=1, female=0) as predictors of TerraNova scaled scores to examine effects of gender. The HLM model for the analysis of effect of gender is illustrated below.

Level-1 model:

$$TScore = \beta_{0j} + \beta_{1j}*(PriorMath) + \beta_{2j}*(Male) + r$$

Level-2 model:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + \gamma_{01}*(Trx) + u_0 \\ \beta_{1j} &= \gamma_{10} \\ \beta_{2j} &= \gamma_{20} \end{aligned}$$

In this model, TScore is the student's scaled score from the TerraNova test. *Trx* is a school-level indicator of the school being in the treatment condition (0=Control, 1=Treatment). Student-level variables. *PriorMath* is a student-level variable, representing the student's 6th grade math state test score. *Male* is a student-level variable, indicating the student's gender (0=Female, 1=Male). The model showed that students in the treatment condition scored 10.26 points higher than control students and males scored 5.21 points lower than females. Both effects are statistically significant ( $p < .001$ ). To help understand the difference, we referred to TerraNova technical norms published by CTB. The norms showed that the average yearly growth from 7<sup>th</sup> to 8<sup>th</sup> grade is about 10 points in scaled score.

**Table 4. HLM Results on Intervention and Gender Effect**

Gender	Control	Treatment	Difference
Females	683.21	693.46	10.26
Males	677.99	688.25	10.26
Difference	-5.21	-5.21	

Then we augmented the basic model by adding an interaction term between treatment and gender. The augmented model is illustrated below.

Level-1 model:

$$TScore = \beta_{0j} + \beta_{1j}*(PriorMath) + \beta_{2j}*(Male) + r$$

Level-2 model:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + \gamma_{01}*(Trx) + u_0 \\ \beta_{1j} &= \gamma_{10} \\ \beta_{2j} &= \gamma_{20} + \gamma_{21}*(Trx) + u_1 \end{aligned}$$

**Table 3. HLM Results By Groups – Predictor: Female**

Dependent Variable	Progressing or Below		Near Proficient		Proficient or Above	
	Difference	<i>p</i>	Difference	<i>p</i>	Difference	<i>p</i>
mins_s	8.565	0.859	-18.195	0.684	58.401	0.073
probs_c	34.421	0.378	-17.773	0.638	34.694	0.178
perc	-0.008	0.569	0.005	0.632	0.013	0.058
hint_c	-0.015	0.239	-0.024	0.061	-0.012	0.064
attempt_c	-0.017	0.56	-0.066	0.005**	-0.033	0.075
bottom_hint_c	-0.003	0.685	-0.016	0.007**	-0.013	0.002**
comp_perc	0.021	0.185	-0.001	0.929	0.027	0.055
late_perc	-0.021	0.032*	0.018	0.034*	-0.005	0.521

In this model, the effect of ASSISTments intervention was found to vary by gender ( $\gamma_{21}=7.476$ ,  $t(42)=2.232$ ,  $p = 0.031$ ). As shown in Table 5, boys in the control group scored 9.61 points lower than girls in the control group, but boys in the treatment condition scored only 2.13 points lower than girls in the same group. Girls in the treatment group scored 6.73 points higher than those in the control group (which was not significant after adding in the interaction), while boys in the treatment group scored 14.21 points higher than those in the control group. In essence, the intervention helped close the gender gap between girls and boys for standardized test achievement and boys have benefited more from the intervention than girls.

**Table 5. HLM Results on Intervention and Gender Interaction Effect**

Gender	Control	Treatment	Difference
Females	685.20	691.93	6.73
Males	675.59	689.79	14.21
Difference	-9.61	-2.13	

## 5. CONCLUSIONS AND FUTURE STUDIES

In this paper, we examined the difference between genders in middle school mathematics on homework time, the amount of problems completed by each gender, homework performance, and how each gender interacted with the system, using computer system log data from an online homework intervention. We also answered two research questions regarding which gender benefited more from a technology-based intervention supporting homework. Our results suggested no significant difference between the time the two genders spent homework overall. Among students who performed proficiently or above on the end-of-year standardized test, girls have spent more time on homework than boys, and the difference was marginally significant. We also found out that when using ASSISTments for homework, girls and boys differed in their help-seeking and problem-attempting behaviors. Girls requested less hints, made less number of attempts on problems, and they also requested less amount of bottom-out hints that would reveal the correct answers to problems. Our findings suggested that the intervention closed gender gaps in mathematics achievement in 7<sup>th</sup> grade and boys benefited from the online homework intervention more than girls.

We speculated on the reasons why boys have benefited more from the technology-based intervention. One reason could be boys in the study were more comfortable with using technologies, similar to what has been reported in earlier research. We also checked to see if there was any difference between the two genders in prior achievement. Using a simple *t*-test, there was no gender difference in 6th grade state math test scores (Female average score =645, Male average score=644,  $p=0.252$ ).

Researchers have been able to identify factors that impact this relationship between time spent doing homework and academic achievement. It was found that the quality of time spent on a task, i.e., homework, is a more critical predictor of student learning than the total number of minutes spent on the task. For instance, time on task or perseverance manifested with low distraction rates is positively correlated with achievement [30]. Other factors, especially the effort students put into homework and how frequently they do homework are far more reliable and positive predictors of student achievement [28, 30, 32]. As a follow-up study, we plan to look at student behaviors when working in in the system more closely, taking sequence and time into account. We plan to study help-seeking and problem-attempting behaviors at action level and to see whether there are any the sequential pattern

of actions taken, and whether there is between girls and boys. For instance, did boys ask for hints/bottom-out hints right away, while girls took time to persevere through challenging homework problems before requesting for assistance? We also plan to build a dataset including students' frequency of logging in each day and each week and the duration of the working sessions by gender, and see how such features predict student learning. Such studies will help the field better understand gender differences in STEM learning, esp. in out-of-classroom settings. The findings can be informative for the development of behavior detectors in online learning systems like ASSISTments so that the systems can provide interventions to improve learning outcomes and close gender gaps.

We recognize the limitations in our study. We have no access to information, such as parent involvement, their extra-curricular activities, etc. that may affect student's homework completion rates, their behaviors when doing homework, or their performance. Nor do we have access to student's attitudes towards mathematics, technology or homework. All of these limit our ability to explain the differences we've discovered. The results presented in this paper were based on data from 7<sup>th</sup> grade students who are younger than the high school students who have been the focus of [23] and [10]. It would be a reasonable next step to extend such kind of study to elementary students and see if there might exist a trajectory in the gender differences in homework.

## 6. ACKNOWLEDGMENTS

This material is based upon work supported by the Institute of Educational Sciences (IES) of U.S. Department of Education under Grant Number R305A120125. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the IES.

## 7. REFERENCES

- [1] Bhanot, R., Jovanovic, J. (2005). Do parents' academic gender stereotypes influence whether they intrude on their children's homework? *Sex Roles*, 52(3)(9/10), 597-607.
- [2] Cooper, H. (2007). *The battle over homework* (3rd ed.). Thousand Oaks, CA: Corwin Press.
- [3] Cooper, H., Robinson, J. C., & Patall, E. A. (2006). Does homework improve academic achievement? A synthesis of research, 1987–2003. *Review of Educational Research*, 76(1), 1–62.
- [4] Eccles, J. & Wigfield, A. (2002). Motivational beliefs, values, and goals. *Annual Review of Psychology*, 53(1), 109-132.
- [5] Eren, O., & Henderson, D. (2011). Are we Wasting Our Children's Time by Giving Them More Homework? *Economics of Education Review*, 30(5), 950-961.
- [6] Feng, M., Heffernan, N., and Koedinger, K. (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 journal)*. 19(3), 243-266, August, 2009.
- [7] Feng, M., Roschelle, R., Murphy, R. & Heffernan, N. (2014). Using Analytics for Improving Implementation Fidelity in a Large Scale Efficacy Trial. In *Proc. ICLS 2014*. International Society of the Learning Sciences. pp. 527-534.
- [8] Fernández-Alonso, R., Suárez-Álvarez, J., & Muñiz, J. (2015, March 16). Adolescents' Homework Performance in Mathematics and Science: Personal Factors and Teaching Practices. *Journal of Educational Psychology*.

- [9] Galloway, M. K., & Pope, D. (2007). Hazardous homework? The relationship between homework, goal orientation, and well-being in adolescence. *Encounter*, 20, 25–31.
- [10] Gershenson, S. & Holt, S. (2015). Gender gaps in high school students homework time. *Education Researcher*, Voc. 44, No. 8, Pp432-441.
- [11] Gill, B. & Schlossman S. (2003). A Nation At Rest: The American Way of Homework. *Educational Evaluation and Policy Analysis*, 25(3).
- [12] Hakkarainen, K., Ilo`maki, L., Lipponen, L., Muukkonen, H., Rahikainen, M., Tuominen, T., et al. (2000). Students' skills and practices of using ICT: Results of a national assessment in Finland. *Computers and Education*, 34(2), 103–117.
- [13] Harris, S., Nixon, J., & Rudduck, J. (1993). School work, homework and gender. *Gender and Education*, 5(1), 3-14.
- [14] Heffernan, N. & Heffernan, C. (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), 470-497.
- [15] Hidi, S., & Renning, K.A. (2006). The four-phase model of interest development. *Educational Psychologist*, 41(2), 111-127.
- [16] Honigsfeld, A., & Dunn, R. (2003). High school male and female learning-style similarities and differences in diverse nations. *Journal of Educational Research*, 96(4), 195-206.
- [17] Janssen Reinen, I. J., & Plomp, T. (1997). Information technology and gender equality: A contradiction in terminis? *Computers and Education*, 28(2), 65–78.
- [18] Juster, T. F., Ono, H., & Stafford, F. P. (2004). *Changing Times Of American Youth: 1981-2003*. Institute for Social Research University of Michigan.
- [19] Maltese, A.V., Robert, H.T., and Fan, X. (2012). When Is Homework Worth the Time? Evaluating the Association Between Homework and Achievement in High School Science and Math. *The High School Journal*, October/November 2012: 52-72.
- [20] Marzano, R. J., & Pickering, D. J. (2007). The case for and against homework. *Educational Leadership*, 64, 74–79.
- [21] Meelissen, M. R. M., & Drent, M. (2007). Gender differences in computer attitudes: Does the school matter? *Computers in Human Behavior*. doi:10.1016/j.chb.2007.03.001.
- [22] Nelson, L. J., & Cooper, J. (1997). Gender differences in children's reactions to success and failure with computers. *Computers in Human Behavior*, 13(2), 247–267.
- [23] OECD (2015). *The ABC of Gender Equality in Education: Aptitude, Behavior, Confidence*. PISA, OECD Publishing. <http://dx.doi.org/10.1787/9789264229945-en>
- [24] Papastergiou, M., & Solomonidou, C. (2005). Gender issues in internet access and favourite internet activities among Greek high school pupils inside and outside school. *Computers and Education*, 44(4), 377–393.
- [25] Ramey, G & Ramey, V. (2010). The rug rat race. *Brookings Papers on Economic Activity*, 41(1), 129-199.
- [26] Rushoway, K. (March, 2015) Retrieved from [http://www.ourwindsor.ca/news-story/54609\\_64-boys-do-less-homework-than-girls-global-study-finds/](http://www.ourwindsor.ca/news-story/54609_64-boys-do-less-homework-than-girls-global-study-finds/)
- [27] Selwyn, N. (1998). The effect of using a home computer on students' educational use of IT. *Computers and Education*, 31(2), 211–277.
- [28] Trautwein, U. (2007). The homework-achievement relation reconsidered: Differentiating homework time, homework frequency, and homework effort. *Learning and Instruction*, 17, 372–388. doi: 10.1016/j.learninstruc.2007.02.009.
- [29] Trautwein, U., Koller, O., Schmitz, B., & Baumert, J. (2002). Do homework assignments enhance achievement? A multilevel analysis in 7th-grade mathematics. *Contemporary Educational Psychology*, 27.1: 26-50.
- [30] Trautwein, U., & Koller, O. (2003a). The relationship between homework and achievement: still much of a mystery. *Educational Psychology Review*, 15, 115e145.
- [31] Trautwein, U., & Koller, O. (2003b). Time investment does not always pay off: the role of self-regulatory strategies in homework execution. *Psychologie*, 17, 199e209.
- [32] Trautwein, U., Ludtke, O., Schnyder, I., & Niggli, A. (2006). Predicting homework effort: support for a domain-specific, multilevel homework model. *Journal of Educational Psychology*, 98, 438e456.
- [33] Vekiri, I., & Chronaki, A. (2008). Gender issues in technology use: Perceived social support, computer self-efficacy and value beliefs, and computer use beyond school. *Computers & education*, 51(3), 1392-1404.
- [34] Volman, M., van Eck, E., Heemskerk, I., & Kuiper, E. (2005). New technologies, new differences. Gender and ethnic differences in pupils' use of ICT in primary and secondary education. *Computers and Education*, 24(1), 35–55.
- [35] Xu, J. (2006). Gender and Homework Management Reported by High School Students. *Educational Psychology*, 26(1), 73-91. doi: 10.1080/01443410500341023
- [36] Xu, J. (2007). Middle-School Homework Management: More than just gender and family involvement. *Educational Psychology*, 27(2), 173-189.
- [37] Xu, J., & Corno, L. (2006, March 10). Gender, family help, and homework management reported by rural middle school students. *Journal of Research in Rural Education*, 21(2). Retrieved [date] from <http://jrre.psu.edu/articles/21-2.pdf>.
- [38] Younger, M., & Warrington, M. (1996). Differential achievement of girls and boys at GCSE: Some observations from the perspective of one school. *British Journal of Sociology of Education*, 17(3), 299-313.
- [39] Zimmerman, B. J., & Martinez-Pons, M. (1990). Student differences in self-regulated learning: Relating grade, sex, and giftedness to self-efficacy and strategy use. *Journal of Educational Psychology*, 82(1), 51-59.

# Investigating Difficult Topics in a Data Structures Course Using Item Response Theory and Logged Data Analysis\*

Eric Fouh  
Department of Computer  
Science & Engineering  
Lehigh University  
Bethlehem, PA 18015  
efouh@cse.lehigh.edu

Mohammed F. Farghally  
Department of Computer  
Science  
Virginia Tech  
Blacksburg, VA 24061  
mfseddik@vt.edu

Sally Hamouda  
Department of Computer  
Science  
Virginia Tech  
Blacksburg, VA 24061  
sallyh84@vt.edu

Kyu Han Koh  
Department of Computer  
Science  
Virginia Tech  
Blacksburg, VA 24061  
kyuhan@vt.edu

Clifford A. Shaffer  
Department of Computer  
Science  
Virginia Tech  
Blacksburg, VA 24061  
shaffer@vt.edu

## ABSTRACT

We present an analysis of log data from a semester's use of the OpenDSA eTextbook system with the goal of determining the most difficult course topics in a data structures course. While experienced instructors can identify which topics students most struggle with, this often comes only after much time and effort, and does not provide real-time analysis that might benefit an intelligent tutoring system. Our factors included the fraction of wrong answers given by student, results from Item Response Theory, and the rate of model answer and hint use by students. We grouped exercises by topic covered to yield a list of topics associated with the harder exercises. We found that a majority of these exercises were related to algorithm analysis topics. We compared our results to responses given by a sample of experienced instructors, and found that the automated results match the expert opinions reasonably well. We investigated reasons that might explain the over-representation of algorithm analysis among the difficult topics, and hypothesize that visualizations might help to better present this material.

## Keywords

Item Response Theory, learning analytics, eTextbooks, algorithm analysis, data structures and algorithms

---

\*(Does NOT produce the permission block, copyright information nor page numbering). For use with ACM\_PROC\_ARTICLE-SP.CLS. Supported by ACM.

## 1. INTRODUCTION

Knowing what topics are challenging to students helps educators better allocate course resources. We present techniques to automatically determine topics that are most challenging based on student interactions within the OpenDSA eTextbook system [9, 10]. While experienced instructors can identify which topics students most struggle with, automated measures can be useful for a variety of reasons. 1) Identifying key topics takes a lot of time and effort on the part of instructors; 2) They can help instructors teaching new material or with a new approach; 3) They can be used by an intelligent tutoring system (ITS) to automatically direct more instruction to a topic; and 4) They can help find, confirm, and quantify relationships and provide new insights that might be missed even by experienced instructors.

Our study focuses on a post-CS2 data structures and algorithms course (henceforth referred to as "CS3"). We used two approaches to identify difficult course topics. The first is Item Response Theory (IRT), a latent trait models (LTM) technique to analyze student responses to problems. LTM assumes that test performance can be predicted by specific traits or characteristics [13]. IRT provides a model-based association between item responses and the characteristic assessed by a test [7]. The second approach consisted of an analysis of student interactions with exercises to identify harder exercises. We investigated the incidence of guessing, the use of hints, and the level of interactions with embedded model answers by students when solving exercises.

We found that the most difficult topics in the CS3 course are related to algorithm analysis. While this is not surprising to us, we also investigated possible reasons that might explain the topics' difficulty. Based on our study, we present some suggestions on how to make such topics more accessible to students.

## 2. RELATED WORK

IRT [19] examines test behavior at the item level, and provides feedback on the relative difficulty of the various ques-

tions. Many IRT models have been developed with the assumption of 0 or 1 assigned to each response. We adopted the one parameter (1PL) or Rasch model [16] to characterize items and examinees. In 1PL, the probability of a positive response from a student is a function of item difficulty and is modeled as  $P_i(\theta) = \frac{\exp(\theta - b_i)}{1 + \exp(\theta - b_i)}$ .  $P_i$  is the probability of a correct response to item  $i$ .  $\theta$  refers to the latent trait (this is often called *ability*) assessed by the items being analyzed.  $b_i$  represents the difficulty of item  $i$ .

IRT has been used to evaluate students' coding abilities in an introductory programming course [3]. The authors used students' code scores to build a 1PL Rasch model. They found that students with previous knowledge had a statistically significant higher performance than students with no previous knowledge [3]. IRT was also used to analyze midterm exam questions for an introductory CS course [18]. The goal was to improve the assessment for future semesters by studying questions' item characteristic curves. IRT has been used for problem selection and recommendation in ITS [14]. The authors built a model based on a combination of IRT and collaborative filtering to automatically select problems.

We know of few efforts to identify difficult topics in CS3 courses, as most such work typically has focused on introductory courses [5, 6, 11]. Brusilovsky et al [4] sent a questionnaire to CS educators asking them to report topics that they consider critical to learn, as well as topics that are hard to learn (for students) and hard to teach (for instructors). Instructors' ( $n = 61$ ) five most difficult-to-learn topics included pointers, recursion, polymorphism, memory allocation, and parameter passing. The five most difficult to teach topics included recursion, pointers, error handling, algorithms, and polymorphism. Many of these topics are covered in CS3, but it is typically not the first time that students will have seen them.

### 3. EXERCISE ANALYSIS

OpenDSA provides a collection of online, open-source tutorials that combine textbook-quality text with algorithm visualizations, randomly generated instances of interactive examples, and exercises to provide students with unlimited practice. Content within OpenDSA is organized into modules, each focusing on a specific topic such as Quicksort or Closed Hashing. The modules contain a wide variety of exercises. Some require that the student manipulate a data structure to show the changes that an algorithm would make on it. We refer to these as "proficiency exercises" (PE exercises). This type of exercise was pioneered in the TRAKLA2 system [15]. OpenDSA uses the Khan Academy (KA) exercise framework<sup>1</sup> to provide multiple choice, T/F, and short answer exercises. We also use the KA framework to implement simpler proficiency exercises.

We studied 143 student participants enrolled in a CS3 course at Virginia Tech during Fall 2014. OpenDSA was used as the main textbook, and students had until the end of the semester to complete the OpenDSA exercises. OpenDSA exercises accounted for 20% of the course final grade.

<sup>1</sup><http://github.com/Khan/khan-exercises>

### 3.1 Analysis of correct answer ratios

Our goal is to assign a value to each OpenDSA exercise in terms of "relative difficulty". We seek to find which exercises are relatively difficult for average ability students. From this, we hope to deduce which topics are most difficult for students. This in turn might lead us to refocus our instructional efforts, or come up with new interventions and presentation approaches. Unfortunately, it is not a simple matter to tell whether a question is difficult. OpenDSA works on a mastery-based system, meaning that students can repeat a question until they get it correct. As a result, most students earn full credit on almost all exercises. To confuse the situation further, as is typical with online courseware, some exercises can be "gamed" [1]. In our case, this happens when students repeatedly reload the current page until they get an easier problem instance to solve (though the system is implemented in ways to discourage other forms of guessing on any given question [9]). For these reasons, we cannot simply count how many students got an exercise correct. Instead, we developed alternative definitions for difficulty.

We analyzed OpenDSA exercises with respect to the ratio of correct to incorrect answers as a measure of exercise difficulty, that is, harder exercises should show a lower correct attempt ratio. To assess student performance, we use the fraction  $r = \frac{\text{\#of correct attempts}}{\text{\#of total attempts}}$ . For each exercise, we compute the difficulty level ( $dl$ ) as  $dl = 1 - \frac{\sum_{i=1}^n r_i}{n}$  where  $n$  is the number of students and  $r$  is the ratio of correct attempts. Similar metrics have been used previously to assess exercise difficulty. In [2], the authors used "how many attempts it takes for a student to determine the correct answer once they have made their initial mistake" as a measure of exercise difficulty for logic exercises. History of attempts coupled with IRT was also used in [17] to estimate exercise difficulty for an ITS.

We ranked the exercises by their  $dl$  and grouped them into quartiles.  $dl$  scores ranged from 0 to 0.72. Exercises in the 4<sup>th</sup> quartile ( $dl > 0.25$ ) consist mainly of exercises covering concepts related to algorithm analysis (22 out of 26 in that quartile), and one was a code writing question. Exercises in the 3<sup>th</sup> quartile ( $0.13 \leq dl \leq 0.25$ ) covered mainly (14 out of 25) the mechanics of algorithms or data structures. Ten of these exercises covered course concepts. Exercises in the 2<sup>nd</sup> quartile ( $0.05 \leq dl < 0.13$ ) covered mainly (23 out of 25) the mechanics of algorithms or data structures. The other two were summary exercises covering lists and the introduction chapter. All exercises in the 1<sup>st</sup> quartile ( $dl < 0.05$ ) covered algorithms or data structures mechanics. These results indicate that students did not seem to have difficulty completing tasks related to the behavior and the mechanics of algorithms and data structures. They seem to have the hardest time mastering algorithms analysis concepts.

#### 3.1.1 IRT analysis

To perform IRT analysis we must dichotomize the answers. We awarded 1 point for  $r \geq 0.75$  and 0 point for  $r < 0.75$ . We analyzed each chapter independently, considering all exercises in a chapter as part of an assignment. We used R statistical software (ltm package) and built a 1PL model for our investigation. For each OpenDSA chapter, we computed the item characteristic curves (ICC), item information curves

(IIC), and test information curves (TIF). For each curve, the  $x$ -axis represents the students' ability from  $-4$  to  $4$ , where  $x = 0$  means average ability. ICC shows the probability of a score of 1, given a student's ability. IIC shows how much information each exercise can tell us about a student's ability. TIF shows how reliable the overall test (or a collection of exercises) is at distinguishing students with different ability. Harder tests would better distinguish between students with above-average ability, while easier tests would better distinguish between students with below-average ability.

An ICC graph lets us see the probability of getting a score of 1 for students with average ability. Harder exercises will have  $P_i(0) < 0.5$ . In Figure 1, we see that for three of the most difficult exercises, the probability that a student with average ability will get a score of 1 is less than 0.5, indicating that those exercises distinguish students with average ability from those with above average ability, but do little to distinguish weaker from average students. On the other hand, an easy question on the binary search algorithm has a graph  $P_i(\theta) = 1$ . Thus it does not give us any information about students' ability. The curves for the easier exercises shown in Figure 2 show differences between students with below average ability in contrast with average and above average ability ( $\theta \geq 0$ ). Another possible interpretation of this result is that these exercises are relatively good at differentiating students who studied from those who did not. The TIF graph is a combination of all IIC curves, and indicates the overall performance of the test.

**Algorithm analysis chapter exercises:** Most students did not fare well on exercises in the introductory chapter on algorithm analysis, as shown in Figures 1 and 2. Thus these exercises gave us information about which students have above-average ability.

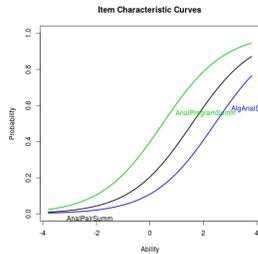


Figure 1: Algorithm analysis ICC

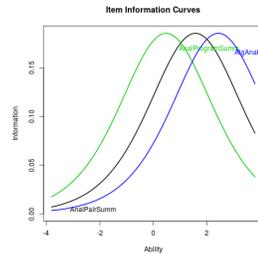


Figure 2: Algorithm analysis IIC

**Linear Structures exercises:** These students were already familiar with linear structures, since these are taught in prerequisite courses. Students could easily get a score of 1 by our difficulty measure for most problems in this chapter, and so help to identify students with below average ability ( $x < 0$ ). However, three exercises appeared to be not so easy for students. They covered list overhead concepts (a new topic for them), array list concepts, and a small programming exercise. Students who did poorly (bottom quarter) on these exercises scored an average 65 on Midterm 1 compared to 76 for the rest of the class (a significant difference at  $\alpha = 0.05$ ). They received an average score of 73 on Midterm 2 compared to 79 for the rest of the students (a significant difference at  $\alpha = 0.05$ ). They scored an average

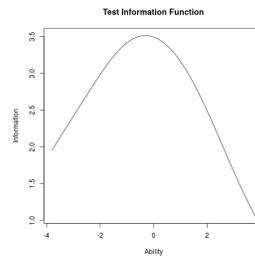


Figure 3: Sorting TIF

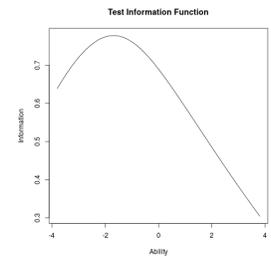


Figure 4: Binary trees TIF

of 106 on the final, compared to 112 for the rest of the class, not a statistically significant difference.

**Sorting exercises:** The sorting chapter has the most exercises, with varying difficulty levels. Summary exercises covering more advanced sorting algorithms (quicksort, radix sort, mergesort, and heapsort) seemed to provide more information about students with above average ability ( $x > 0$ ). Overall, the sorting chapter exercises seemed to provide a good range of easy to difficult exercises, and provided good information to distinguish between students with different ability levels (TIF curve maximum at ability = 0).

**Binary tree exercises:** Binary trees are typically first introduced in CS2 courses. Only three exercises appeared to be difficult for students. These involved writing a recursive function to traverse a tree, questions on heaps, and computing tree space overhead. Exercises in this chapter provided us with information about students with below average ability (TIF curve maximum at ability  $< 0$ ).

**Hashing and graph exercises:** As with other topics, proficiency exercises were relatively easy for the students, while questions on the concepts and analysis were more difficult. The graph chapter only had algorithm proficiency exercises and so were not challenging to students. Therefore, the exercises gave us information to distinguish students with low ability (TIF curve maximum at ability  $> 0$ ).

We identified 21 (out of 100) exercises with IIC maximum at ability  $\geq 0$ . 19 of those exercises cover the algorithm analysis portions of the different topics. The IRT analysis for all OpenDSA exercises given to students enrolled in the course revealed the following. Across chapters, exercises related to algorithm analysis had IIC curve maximums at ability  $< 0$ . Exercises that required students to solve small programming problems also scored as “difficult” by our metric because they tended to require multiple submissions to complete.

### 3.2 Using Hints and Guessing

Our analysis metric for “incorrect attempts” does not differentiate between using a hint or submitting an incorrect answer. So we looked in more detail at the types of incorrect submission for each exercise. We analyzed OpenDSA exercises with respect to the number of hints used, and the appearance of a trial-and-error strategy to “guess” the answers. Harder exercises are expected to display a higher rate of hints use and/or trial-and-error.

Exercises using the KA framework (multiple choice, T/F, fill-in-the-blank, and one-step proficiency exercises) generate a series of question instances on the topic. The student must get a certain number correct (typically five) to complete the exercise. One point is deducted from the student's credit toward this requirement when they submit an incorrect answer, to discourage guessing. Students can also take one or more hints that explain the answer to the question. In this case, the attempt is not graded (no point is awarded or deducted toward the threshold).

To analyze exercises based on students' hint use, we computed the hint ratio  $hr = \frac{\# \text{ of hints used}}{\# \text{ of total attempts}}$  for each KA exercise. Four exercises are potential outliers as measured by  $hr$ , related to quicksort, hashing, calculating overhead for trees, and calculating overhead for lists. To analyze exercises based on the rate of trial-and-error, we calculated the incorrect ratio  $ir = \frac{\# \text{ of incorrect answers}}{\# \text{ of total attempts}}$  for each KA exercise. Inspecting exercises in the fourth quartile (exercises in the highest 25% incorrect ratio), we found that they are related to the topics algorithm analysis, heaps, quicksort, radixsort, shellsort, and heapsort.

The seven exercises shown in Table 1 had high hint or high incorrect answer ratios. They relate to topics covering mathematical background and runtime analysis of quicksort, hashing, and shellsort. 45% of students heavily (third quartile and up for all exercises) used hints, and provided many incorrect answers when solving these seven exercises. We found that most exercises with low incorrect answer and hint ratios are for stacks, arrays, and lists. These are topics that most students know from previous courses. When using high rate of hint use as a measure of exercise difficulty, we found that exercises related to algorithm analysis and mathematics topics appeared to be more "difficult". Algorithm analysis was also identified as difficult by IRT analysis.

**Table 1: IR and HR for difficult exercises**

Exercise	$hr$	$ir$	Topic
ListOverhead	0.93	0.6	List Overhead Analysis
TreeOverheadSumm	0.78	0.73	Tree Overhead Analysis
QuicksortSumm	0.32	0.67	Quicksort Analysis
AlgAnalSumm	0.24	0.77	Algorithm Analysis
MthBgSumm	0.25	0.63	Mathematical background
ShellsortSumm	0.16	0.61	Shellsort
QuicksortPartitionPRO	0.27	0.58	Quicksort's partition

### 3.3 Model Answer Use and Exercise Reset

Algorithm proficiency exercises require students to reproduce the major steps of an algorithm. Proficiency exercises come with a "model" answer that can be viewed at any time (though doing so voids that problem instance for credit, and so the student must do another problem instance). The student can click a "reset" button to get a new problem instance. We analyzed OpenDSA exercises with respect to model an-

swer use and "reset" as a measure of (exercise) difficulty. Students are expected to reset or view model answers more for harder exercises. For each proficiency exercise, we analyzed the number of student attempts and the frequency of student access to the model answer dialog. Our analysis showed that heap and quicksort exercises have a model answer view rate approaching or exceeding 50%, which is greater than the mean ( $\mu = 25.5$ ) plus one standard deviation ( $\sigma = 16$ ) of the rates distribution. This finding indicates that these exercises are relatively more difficult compared to other proficiency exercises.

We also investigated student activity log data to learn when students access the model answer box by computing: (i)% of students who tried the exercise, then opened the model answer dialog before they received enough points to get credit for the exercise; % of students who opened the model answer dialog before attempting the exercise; and % of students who opened the model answer dialog after they received proficiency credit for the exercise.

A model answer shows how to solve a problem with less detail, while slideshows and visualizations (available to the students before attempting the exercise) carefully explain the concepts. We tried to determine if students use model answers as a substitute for viewing slideshows and visualizations. For the heap exercises, we found that about 35% of the students attempted an exercise before going through any slideshow included in the section. This result indicates that students might be using model answers (on certain topics) because they overlook and/or rush through visualizations when studying. We found that a majority of students (62% on average) opened the model answer before attempting the heap exercises. For the quicksort exercise, we found that most students (67%) opened the model answer dialog after an incorrect attempt. 24% of students opened the model answer dialog before attempting the exercise.

For each proficiency exercise, we looked at the percentage of students who returned back to solve the exercise after receiving proficiency credit. We found that exercises with a high model answer view rate have a lower level of post-proficiency attempts. 27% of students solved them post-proficiency, compared to almost 50% for other exercises. This is somewhat surprising, as students presumably use an exercise post-proficiency in order to study the material for exams. We might have expected the most difficult exercises to be targets for additional study.

We computed the ratio of correct attempts over number of reset button clicks, and the ratio of all attempts over number of reset button clicks. The correlation between the two ratios was  $r^2 = 0.99$ . Exercises with lowest ratios (bottom 25%) were related to quicksort, heaps, shellsort, and binary trees topics. When using number of model answer views and use of reset button as measures of exercise difficulty, we found that the hardest exercises are related to the topics of heaps and quicksort. These exercises have higher use of model answers, higher exercise reset rates, and lower levels of post-proficiency attempts compared to other exercises. We note that proficiency exercises cover only algorithm mechanics, and so do not test students on more theoretical concepts. Thus, this analysis is only comparing the relative difficulty

of understanding the mechanics of various algorithms, and so does not address the question of the relative difficulty of algorithm analysis versus algorithm mechanics.

#### 4. INSTRUCTOR SURVEY RESULTS

To validate our process, we compared the results of automated analysis with opinions of course instructors. To that end, we distributed a survey to the CS education community via the SIGCSE mailing list. We asked respondents: (i) how long they have been teaching a post-CS2 course on Data Structures and Algorithms; (ii) what topics from such a course are the most difficult for students to understand; and (iii) what topics from such a course are the most difficult to teach. We received 23 responses with a mean teaching career of 16 years (median 15 years). Since a concept can be defined using different terms, we grouped answers that we considered to refer to the same topic. The result was 12 topics considered most difficult for students to understand, and 8 topics most difficult to teach. Table 2 shows the top 6 difficult topics to learn and to teach. Among the top topics considered hard for students, only trees and heaps are not also present in the list of hard topics to teach.

**Table 2: Summary of survey responses**

Topic	N	%
<b>Most difficult topics for students</b>		
Dynamic programming	7	18
Algorithm analysis	6	15
OOP & Design	6	15
Recursion	4	10
Trees, Heaps	3	7
Proofs	3	7
<b>Most difficult topics to teach</b>		
Complex algorithms	8	30
OOP & Design	4	15
Proofs	4	15
Algorithm analysis	3	11
Recursion	3	11
Dynamic programming	2	7

Dynamic programming had the most votes as difficult for students, but we note that most CS3 courses do not cover this in depth. Algorithm analysis received the next highest number of votes. Our IRT and log analyses also identify algorithm analysis as a hard topic for students. Instructors mentioned students' lack of proficiency in mathematics as a major reason why algorithm analysis proves hard. Instructors wrote "mathematical sophistication is the issue here", and "because students are afraid of math". Our analysis of use of trial-and-error also revealed that students are not at ease with mathematics topics. To explain why algorithms analysis is hard to teach, one instructor wrote "I still do not have good instructional material". That reason was also used for other topics like graphs and design. Heaps is another topic that was identified as hard both by our analysis and by instructors. In general, the survey responses correspond fairly well to our automated process.

#### 5. ALGORITHM ANALYSIS IS HARD

Our analysis shows that exercises related to algorithm analysis are harder than exercises covering algorithm mechanics. It also reveals that students might have some difficulty with

heaps and quicksort. Algorithm analysis is of particular interest since a main goal of CS3 is to teach students how to analyze algorithms, in order to design efficient software solutions. That is why algorithm analysis sections are present in almost all topics covered in the course. Careful analysis of the data logs reveals certain behaviors by students that could explain why students struggle with these concepts.

#### 5.1 Not spending enough time

We analyzed interaction logs from use of OpenDSA at three universities (Virginia Tech, University of Texas El Paso, and University of Florida). Table 3 shows estimated reading time for the algorithm analysis material from three sorting modules (Insertionsort, Mergesort, and Quicksort). More than 74% of students spent less than one minute on the analysis material for each of the three modules. Based on this result, we believe that most of the students are not reading the analysis material.

**Table 3: Time reading algorithm analysis material**

University	Module	N	$\mu(\text{sec})$	% < 1 min
VT	Insertionsort	98	63.57	74.48
	Mergesort	96	39.79	78.12
	Quicksort	92	64.71	73.91
UTEP	Insertionsort	26	49.84	80.76
	Mergesort	22	41.45	77.27
	Quicksort	16	16.18	93.75
Florida	Insertionsort	53	40.39	84.90
	Mergesort	44	18.63	95.45
	Quicksort	39	26.12	92.30
All	Insertionsort	177	54.6	78.52
	Quicksort	147	49.2	80.94

86% of students responding to a survey indicated that it is easier for them to understand how an algorithm works than to analyze the running time for that algorithm. Quotes include: "determining asymptotic running time because it is harder to visualize and less intuitive", "Complexities are confusing and math-like", "I think understanding how an algorithm work is easy. It is the style of presentation", "How the algs work. It is dependent on material, also abstract stuff is harder for me to understand".

78% of students who are more comfortable with dynamics attributed this to the material, as algorithm analysis is abstract and requires familiarity with mathematical notations. The other 22% attributed this to how concepts are presented in OpenDSA (dynamics are presented using visualizations, analysis is presented mostly through text). Quotes regarding the usefulness of OpenDSA's algorithm analysis content include: "Not any more useful than any other book", "Not as much as learning the algorithms themselves, but I felt it was as useful as any resource could be on the topic", "Yes, but not as much as understanding the algorithms", "It could have been more interactive with showing why the analysis was the way that it was", "I found it much more useful on Data structures. Algorithm analysis doesn't benefit quite as much from animations", "It was very detailed and kind of hard to follow", "I'd like there to be more visuals for analysis". Clearly respondents did not find the OpenDSA material on algorithm analysis different from other textbooks on that topic. This is not what they expected from OpenDSA, whose goal is to present content interactively.

## 5.2 Content presentation not engaging

When students were asked to provide suggestions for improving presentation of the analysis material in OpenDSA, most indicated they were expecting a more interactive presentation in the form of visualizations. Quotes include: “Visualizations definitely help.”, “I think making the clickthrough pictures into actual animations would be nice”, “more animation, the visualizations are great!”, “more visualizations is always good”, “Visualizations always help :)”, “visualizations showing each step of analysis would help”, “an animation will make a much bigger difference.”

## 6. CONCLUSION AND FUTURE WORK

Educational resources are rapidly moving online. As eTextbooks and interactive exercises become more prevalent, techniques to automatically discover the most difficult topics for students will become increasingly important. Doing so allows both instructors and designers of instructional content to focus their resources on the most difficult topics. Perhaps resolving the difficulty might be as simple as fixing a buggy exercise. But more generally, we find that specific concepts are truly hard. By examining the topic in detail, including its method of presentation, we might uncover better approaches to instruction, leading to better outcomes.

To illustrate, we are working on addressing the issues raised by students regarding the lack of visual presentation for algorithm analysis material in OpenDSA. Inspired by the concept of visual proofs [12], a set of Algorithm Analysis Visualizations (AAVs) were implemented for OpenDSA sorting modules [8]. We have collected preliminary data with two small classes using the sorting analysis visualizations. Summary results were collected for two modules teaching Insertion Sort and Quicksort. A Kruskal Wallis tests showed a significant difference ( $p < 0.01$ ) between the time spent for text versus visualizations for these two modules. This indicates that students spend more time on the material when presented as visualizations. Having proved the value of the concept, we will continue to expand on this approach.

## 7. ACKNOWLEDGMENTS

We gratefully acknowledge the support of the National Science Foundation under Grants DUE-1139861, IIS-1258571, and DUE-1432008.

## 8. REFERENCES

- [1] R. Baker, A. Corbett, and K. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, pages 531–540, 2004.
- [2] D. Barker-Plummer, R. Cox, and R. Dale. Student translations of natural language into logic: The grade grinder corpus release 1.0. In *Proceedings of the 4th international conference on educational data mining*, pages 51–60, 2011.
- [3] M. Berges and P. Hubwieser. Evaluation of source code with item response theory. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE ’15, pages 51–56, 2015.
- [4] P. Brusilovsky, J. Grady, M. Spring, and C.-H. Lee. What should be visualized?: Faculty perception of priority topics for program visualization. *SIGCSE Bulletin*, 38(2), June 2006.
- [5] N. Dale. Content and emphasis in CS1. *SIGCSE Bulletin*, 37(4):69–73, Dec. 2005.
- [6] N. B. Dale. Most difficult topics in CS1: Results of an online survey of educators. *SIGCSE Bulletin*, 38(2):49–53, June 2006.
- [7] F. Drasgow and C. L. Hulin. Item response theory. *Handbook of industrial and organizational psychology*, 1:577–636, 1990.
- [8] M. F. Farghally, E. Fouh, S. Hamouda, K. H. Koh, and C. A. Shaffer. Visualizing algorithm analysis topics. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, page 687, 2016.
- [9] E. Fouh, D. A. Breakiron, S. Hamouda, M. Farghally, and C. A. Shaffer. Exploring students learning behavior with an interactive etextbook in computer science courses. *Computers in Human Behavior*, pages 478–485, December 2014.
- [10] E. Fouh, V. Karavirta, D. A. Breakiron, S. Hamouda, S. Hall, T. L. Naps, and C. A. Shaffer. Design and architecture of an interactive etextbook—The OpenDSA system. *Science of Computer Programming*, 88:22–40, 2014.
- [11] K. Goldman, P. Gross, C. Heeren, G. L. Herman, L. Kaczmarczyk, M. C. Loui, and C. Zilles. Setting the scope of concept inventories for introductory computing subjects. *Transactions on Computing Education*, 10(2):5:1–5:29, June 2010.
- [12] M. T. Goodrich and R. Tamassia. Teaching the analysis of algorithms with visual proofs. In *SIGCSE Bulletin*, volume 30, pages 207–211, 1998.
- [13] R. K. Hambleton and L. L. Cook. Latent trait models and their use in the analysis of educational test data. *J. of Educational Measurement*, 14(2):75–96, 1977.
- [14] P. Jarušek and R. Pelánek. Analysis of a simple model of problem solving times. In S. Cerri, W. Clancey, G. Papadourakis, and K. Panourgia, editors, *Intelligent Tutoring Systems*, volume 7315 of *LNCS*, pages 379–388. Springer, 2012.
- [15] L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Seppälä, and P. Silvasti. Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. *Informatics in Education*, 3(2):267–288, September 2004.
- [16] G. Rasch. *Probabilistic models for some intelligence and attainment tests*. Danmarks Pædagogiske Institut, 1960.
- [17] G. Ravi and S. Sosnovsky. Exercise difficulty calibration based on student log mining. In *Proceedings of DAILE: Workshop on Data Analysis and Interpretation for Learning Environments*, 2013.
- [18] L. A. Sudol and C. Studer. Analyzing test items: Using item response theory to validate assessments. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE ’10, pages 436–440, 2010.
- [19] W. J. van der Linden and R. K. Hambleton. *Handbook of modern item response theory*. Springer, 2013.

# Acting the Same Differently: A Cross-Course Comparison of User Behavior in MOOCs

Ben Gelman  
Dept. of Computer Science  
George Mason University  
bgelman@gmu.edu

Matt Revelle  
Dept. of Computer Science  
George Mason University  
revelle@cs.gmu.edu

Carlotta Domeniconi  
Dept. of Computer Science  
George Mason University  
carlotta@cs.gmu.edu

Aditya Johri  
Dept. of Computer Science  
George Mason University  
johri@gmu.edu

Kalyan Veeramachaneni  
CSAIL, MIT  
Cambridge, MA, USA  
kalyan@csail.mit.edu

## ABSTRACT

Recent studies of MOOCs demonstrate their ability to reach a large number of users, but also caution against the high rate of dropout. Some have looked closely at MOOC participation in order to better understand how and when users start to disengage, and, if they remain engaged, in what activities they participate. Most of this prior work relies heavily on descriptive statistics or clustering methodologies to highlight basic user participation characteristics. In this paper, we adapt NMF to provide a multi-dimensional view of user participation. We use log data to create a bottom-up understanding of user participation, and identify five basic behaviors associated with participants' use of content and their engagement with assessment. Furthermore, we do a cross-course analysis across four courses and find that these five behaviors are present in all courses. Interestingly, users' participation patterns - how they engage in these five behaviors - vary across courses even when the course topics are similar. Our methodology can be applied to other datasets, and findings from this work can assist in interventions to help users successfully accomplish their learning goals.

## Keywords

MOOCs, Participant Behavior, NMF, Comparative Analysis

## 1. INTRODUCTION

As Massive Open Online Courses (MOOCs) grow in popularity, and offer an increasing variety of subjects across multiple platforms, there has been significant interest in MOOC users' participation patterns. Extremely low user completion rates [6] have motivated examinations and studies of MOOC behavior that aim to ascertain whether changes in pedagogy can improve completion outcomes, or if every incoming class contains a cohort of users that had no intention to complete.

We were motivated by this recent work to attempt to better understand MOOC users' behavioral patterns, and the evolution of participation over time and across courses. In this paper, we analyze data from four MOOC courses across three axes (*learners*, *time*, and *courses*), choosing methods that link behaviors and patterns across these three dimensions. Utilizing the rich features developed to characterize learners' weekly interactions, we adapt non-negative matrix factorization (NMF) [5] to study the importance of these features and the behavior of users over time [2].

Several factors make NMF particularly well-suited for this type of analysis. The non-negativity constraint helps to identify distinct but additive latent factors. In other words, we are able to learn user behaviors in terms of evolving parts due to NMF's additive latent factors and our temporal adaptation (linking behaviors across weeks). Through this study, we make the following unique contributions: 1) We identify behavioral patterns of users that are consistent across multiple MOOCs; 2) We demonstrate how these behaviors vary across different courses; and 3) We demonstrate the feasibility of a framework that can be applied across similar multi-dimensional datasets.

## 2. RELATED WORK

Several studies of MOOCs highlight low completion rates [13]. The University of Edinburgh launched six MOOCs on the Coursera platform in January 2013 [7]. Evaluations revealed that, of the 309,682 learners initially enrolled, 123,816 (about 40%) accessed the course sites during the first week ('active learners'), and 90,120 (about 29%) engaged with course content. Over the duration of the course, the number of active participants rose to 165,158 (53%). As a gauge of persistence, 36,266 learners (nearly 12%) engaged with week 5 assessments. This represented 29% of initial active learners (although individual numbers for each of the six courses ranged from 7% to 59%). In addition, 34,850 people (roughly 11% of those who enrolled) achieved a statement of accomplishment for reaching a percentage-based benchmark of course completion.

Similarly, when Duke University ran a Bioelectricity MOOC in 2012 [15], 12,175 students initially registered. Only 313 participants (2.6%) achieved a statement of accomplish-

ment. Learner feedback suggested three specific reasons for failure to complete [15]. [8] provides a compilation of available data on MOOC completion. Further analysis of the data shows that, of the 61 courses hosted by Coursera, the average completion rate was just over 6%. This combination of MOOCs' enormous popularity and extremely low completion rate has attracted significant interest.

[17] used a classification method that identifies a small number of longitudinal engagement trajectories in MOOCs. This classifier consistently identifies four prototypical trajectories of engagement: (1) *Completing*, (2) *Auditing*, (3) *Disengaging*, (4) *Sampling*. To decide these engagement patterns, the authors used a number of *binary* variables to determine whether a student accessed a resource or attempted a problem. In contrast, we begin to extract a number of richer descriptors about the students' interaction with the online learning platform.

[9] divides participants into five profiles: no-shows (those who register but never log in); observers (those who log in but do not take assessments); drop-ins (those who participate but do not attempt to complete the entire course); passive (those seeing the course as content to consume); and active (those participating in all the activities and enriching the course). Similarly, [16] distinguishes five groups of people depending on their level of participation in the MOOC forum: inactive (those that do not visit the forum); passive (those that just consume information); reacting (those that add further aspects to existing questions); acting (those that post questions and lead discussions); and supervising/supporting (those that lead discussions and summarize gained insights).

### 3. DATA

Our study utilizes four courses, including 6.002x (Fall 2012 and Spring 2013): Circuits and Electronics, 2.01x (Spring 2013): Elements of Structures, 3.091x (Spring 2013): Introduction to Solid State Chemistry. After filtering out learners who had no browsing events for the duration of the courses, the course sizes are 17379, 6339, 5597 and 8870 users, respectively. The course durations are all set to 14 weeks. Using the scripts from the MOOCdb project, we are able to extract 21 features. Table 1 shows the feature numbers and descriptions.

Figure 1 presents the course sizes dynamically. The count of active users for any week is given by the sum of users that have at least one non-zero feature in that week. The count of inactive users is the sum of users that have all-zero feature values in the current week, but had been active in a prior week. New users are those whose first non-zero feature is in the current week. The dropout value is the number of students who are inactive this week and will be inactive for all future weeks.

Because some features are complex and not fully explained by their feature names, we will expand their definitions here. Each feature is computed using the data collected in a week, and generates a single value, so if there are 14 weeks in a course, a user's feature vector will contain 14 values per feature.

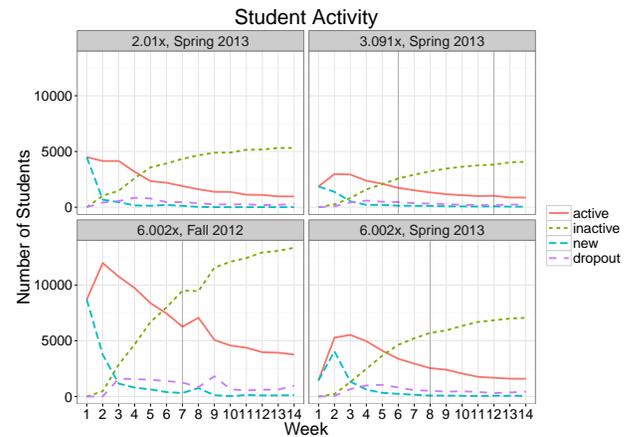


Figure 1: Student activity statuses over time for each class. Vertical lines denote midterm exams and quizzes.

**Time spent:** Feature 1 sums a user's total time spent on any and all events in the course. Feature 11 is the single longest time spent on any single resource (book, wiki, lecture videos, etc). Feature 12 is the time specifically spent on lectures, and feature 13 is the time spent on the course wiki.

**Homework participation:** Feature 4 is the count of all unique problems a learner attempted [1]. Feature 5 is the count of all attempts, including multiple tries at the same problem. Feature 6 is the count of all problems that the learner got correct (grade 1). Feature 7 is the average number of attempts per problem. Feature 18 counts all correct attempts, in order to identify users that correctly solve the same problem multiple times.

**Ratio-based features:** Feature 8 measures the total time spent on the course per correct problem by dividing features 1 and 6. Feature 9 divides the number of attempts (feature 5) by the number of correct problems (feature 6). Feature 19 divides total attempts (feature 5) by non-distinct correct attempts (feature 18).

**Difference-based features:** Features 14-17 represent the change in features 2, 7, 8, and 9, respectively. This is computed by taking the respective feature's value for the current week, subtracting the previous week, and then normalizing the result.

**Regularity and procrastination:** Feature 10 tells us how spread out a student's schedule is over the week by presenting the variance of his or her event timestamps. Feature 20 computes the average amount of time the user submits before the deadline (a zero value means an on-time submission, while a higher value means the work was submitted earlier). Finally, feature 21 calculates the standard deviation in working hours throughout the day—if the student starts work around the same time every day, the feature value will be low.

Feature extraction allows us to represent learners as a set of multiple time series. A learner's basic actions are collected and summarized into the 21 interpretive features on a weekly

Table 1: Students’ features.

Features’ Names	
1	sum_observed_events_duration
2	number_of_forum_posts
3	average_length_of_forum_posts
4	distinct_attempts
5	number_of_attempts
6	distinct_problems_correct
7	average_number_of_attempts
8	sum_observed_events_duration_per_correct_problem
9	number_problem_attempted_per_correct_problem
10	observed_event_timestamp_variance
11	max_duration_resources
12	sum_observed_events_lecture
13	sum_observed_events_wiki
14	difference_feature_2
15	difference_feature_7
16	difference_feature_8
17	difference_feature_9
18	attempts_correct
19	percent_correct_submissions
20	average_predeadline_submission_time
21	std_hours_working

basis. Because learners are represented as a set of features with per-week, aggregate values, time is a dimension of our data set.

#### 4. METHODOLOGY

Uncovering the behaviors of MOOC students requires simultaneously finding interaction patterns (behaviors) across a large number of students and permitting individual students to exhibit multiple behaviors. Since we assume student interactions may be the result of multiple behaviors, we choose to use a decomposition method (NMF) which results in a parts-based representation of student interactions. Students may exhibit multiple behaviors and their behaviors may change over time.

**Step 1: Apply NMF** Given a three dimensional vector representation of the student feature data with  $w$  weeks,  $f$  features, and  $n$  users, we construct the tensor  $A_{ijk}$ . We begin by applying non-negative matrix factorization to each feature-user matrix  $A_i$  for  $i = [1..w]$ . We use a standard implementation [14] with NNDSVD [3] for initialization of the basis matrix and Frobenius cost function. The rank parameter,  $r$ , is set to six, which is selected through approximation.

$$\mathbf{A}_i = \mathbf{B}_i \mathbf{C}_i \quad (1)$$

The results of factorizing  $A_i$  are  $B_i$  and  $C_i$ , the *basis* and *coefficient* matrices, respectively. The dimensions of  $B_i$  are  $f \times r$  and the dimensions of  $C_i$  are  $r \times n$ .

Each of the  $r$  column vectors in  $B_i$  contain  $f$  values that essentially describe the importance of each feature to the given column vector. In our data, we use the set of important features in each basis vector to describe a behavior. In matrix  $C_i^T$ , there are  $r$  column

vectors that contain  $n$  coefficient values, one for each user. The  $m^{th}$  column vector’s coefficient values in  $C_i^T$  describes how closely users associate with the  $m^{th}$  basis vector in  $B_i$ . Because every user has  $r$  coefficient values, it is possible for a user to identify with multiple basis vectors. This is significantly different than hard clustering approaches such as K-means, where groups are mutually exclusive.

**Step 2: Alignment** After performing the matrix factorization on each week, we have  $w$  basis matrices and  $w$  coefficient matrices. To identify persistent basis vectors and patterns, we must connect the results over time. There is no guarantee the order of the basis vectors is consistent over all weeks because the basis matrices are produced by independent executions of NMF. To achieve this, we first compute the cosine similarity using Equation (2) between two consecutive basis vectors. In other words, for each of the  $r$  basis vectors in week  $i$ , we compute the cosine similarity to all basis vectors in week  $i + 1$ , resulting in  $r^2$  computations. Ultimately, there are  $(w - 1)r^2$  similarity computations.<sup>1</sup>

$$\text{Sim}(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} \quad (2)$$

By examining the distribution of cosine similarity values, an alignment threshold may be selected. For our data, a threshold value of 0.95 was chosen to identify matching basis vectors between weeks. We found that after the first week, all basis vectors uniquely match *one and only one* basis vector in the consecutive week when a threshold of  $\geq 0.95$  is used. This phenomenon occurred for all four courses we used in our experiments. Although basis matrices for each week are estimated independently, we find five basis vectors which persist over time and occur in all the classes.

**Step 3: Normalize and define behaviors** The aligned, per-week basis vectors are normalized. We then average these aligned-normalized vectors into a single, representative *behavioral* vector. Having a single, normalized vector permits a semantic interpretation of the behavior based on relative feature values. By identifying the most important features (the ones with the largest values) in each *behavioral* vector, we are able to label the vectors by the interaction pattern they best represent.

#### Step 4: Coefficient analysis

Every student’s interaction attributes may be approximated using a weighted mixture of the discovered behavior vectors. These weights (coefficients) can be considered to define a soft-membership of a student to a behavior.

In order to decide if a user belongs to a behavior, we threshold the distribution of the coefficient values per

<sup>1</sup>We choose cosine similarity because it is a measure of angular similarity between two vectors. Thus, two basis vectors whose only nonzero entry is feature  $j$  will be extremely similar. This is valuable for aligning basis vectors whose distributions of features are similar.

week and per behavioral vector (or basis). This means that the algorithm will generate  $r \times w$  thresholds. The thresholding algorithm takes the entire range of coefficient values per vector and limits the range of values to the top  $x\%$ . The threshold (top  $x\%$ ) is a parameter. This means that if the range of coefficient values for a behavior is 0-100, then selecting a threshold of 0.85 will only consider users with coefficient values of 85-100 to be exhibiting that behavior. There is an additional minimum size parameter  $s$  that adjusts for a skewed distribution where a few users have significantly higher coefficient values than any other users. This skewed distribution causes the top  $x\%$  of coefficient values to only include these few users. If the number of users within the top  $x\%$  is less than the  $s$ , then the users will be saved, and the threshold computation will be repeated without them. For our data, we use a threshold of 0.85 with a minimum size parameter of 30.

We assign behaviors to students for each week using the data-derived thresholds. By tracking the set of behaviors across weeks, we generate a transition diagram that presents the number of students exhibiting each behavior over each week and the migration of users between various behaviors. The transition diagram allows us to understand the evolution of user behavior as a course progresses.

## 5. BASIS MATRIX RESULTS

The resulting basis matrices for 6.002x (Fall 2012) exhibit eight unique behaviors. Tables 2 and 3 numerically summarize behaviors for week one and the average of the other weeks, respectively. Because the first week manifests two unique behaviors, namely *introduction* and *sampling*, it is kept separate. From the second week onwards, all behaviors are persistent (at least 95% cosine similarity). This allows us to average weeks two through 14 in Table 3.

Basis vector one is dominated by feature 11 (*max\_duration\_resources*), which is the duration of the longest observed event this week. This vector represents a *deep* behavior, because the associated students must have spent a long time on a single resource.

Basis vector two is primarily decided by feature 10 (*observed\_event\_timestamp\_variance*). Because this feature tells us how spread out the student's schedule is over the week, this vector describes a *consistent* behavior. Having a high timestamp variance requires users to log in multiple times a week.

Basis vector three is primarily decided by feature 21 (*std\_hours\_working*), which is the standard deviation in working hours over the day. This could represent a *bursty* behavior—because a user must be active during different times in a day to obtain a high feature value, this could mean that the user has a single prolonged session or multiple, separate sessions.

Two basis vectors exist only in the first week of the course. Basis vector four in Table 2 is decided by feature three (*average\_length\_of\_form\_posts*) and feature two (*number\_of\_form\_posts*). This supports the idea that users inter-

Table 2: Matrix of normalized basis vectors (behaviors) for week 1 (course 6.002x fall 2012). The behaviors *Introduction* and *Sampling* are unique to week 1. Dominant feature values are shown in boldface.

Feature	Deep	Consistent	Bursty	Introduction	Sampling
1	0.012	0.000	0.001	0.000	0.088
2	0.000	0.000	0.000	<b>0.137</b>	0.000
3	0.000	0.000	0.000	<b>0.862</b>	0.000
4	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000
10	0.000	<b>0.988</b>	0.000	0.000	0.000
11	<b>0.981</b>	0.011	0.000	0.001	0.000
12	0.000	0.000	0.000	0.000	<b>0.665</b>
13	0.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000
21	0.008	0.000	<b>0.999</b>	0.000	<b>0.248</b>

acted heavily during the opening week of the course. The disappearance of this basis vector, however, tells us that forum interaction in later parts of the course was insignificant in 6.002x fall 2012. For this reason, this basis vector characterizes an *introduction* behavior.

Basis vector five in Table 2 is defined by features 12 (*sum\_observed\_events\_lecture*), 21 (*std\_hours\_working*), and 1 (*sum\_observed\_events\_duration*). This group of features supports the hypothesis that users are browsing through a lot of content during the first week of the course. This may be because users are interested in seeing what lies ahead in the course, or because some users may have joined only to gather information on one particular topic. Thus, basis vector five during the first week expresses a *probing* behavior.

After the first week, two more basis vectors persist. At this point, basis vector four is primarily characterized by feature 19 (*percent\_correct\_submissions*). By turning in assignments with high correctness, the corresponding students can be associated with a *performance* behavior. Basis vector five is strongly defined by feature 20 (*average\_predeadline\_submission\_time*). By turning in assignments long before their deadlines, these students can be associated with an *response* behavior.

When we apply the same analysis to other courses, we see similar behaviors. The average basis matrix tables for 2.01x, 3.091x, and 6.002x are not displayed because they exhibit the same behaviors as table 3 with 95% cosine similarity. It appears that each of these five behaviors—deep, consistent, bursty, performance, and response—appear in all of the courses. The key difference is that 6.002x has two additional behaviors that occur only in the first week. The introduction and sampling behaviors do not appear to be prevalent in the other courses. This could be due to course

Table 3: Average matrix of normalized basis vectors for weeks 2 through 14 (Course 6.002x, Fall 2012). Dominant feature values are shown in boldface.

Feature	<i>Deep</i>	<i>Consistent</i>	<i>Bursty</i>	<i>Performance</i>	<i>Response</i>
1	0.031	0.002	0.007	0.000	0.000
2	0.001	0.000	0.001	0.000	0.000
3	0.004	0.001	0.003	0.000	0.000
4	0.005	0.000	0.000	0.000	0.029
5	0.003	0.000	0.000	0.001	0.012
6	0.000	0.000	0.000	0.052	0.000
7	0.001	0.000	0.000	0.003	0.003
8	0.000	0.000	0.000	0.001	0.001
9	0.000	0.000	0.000	0.001	0.001
10	0.001	<b>0.993</b>	0.000	0.000	0.000
11	<b>0.922</b>	0.000	0.005	0.007	0.028
12	0.010	0.000	0.002	0.000	0.000
13	0.000	0.000	0.000	0.000	0.000
14	0.001	0.000	0.000	0.000	0.000
15	0.001	0.001	0.000	0.002	0.002
16	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.015	0.000
19	0.002	0.000	0.000	<b>0.743</b>	0.004
20	0.000	0.000	0.000	0.174	<b>0.920</b>
21	0.017	0.000	<b>0.980</b>	0.000	0.000

sizes, and the fact that 6.002x was the first edX course ever released. Users may have been encouraged to communicate in the forums early on (introduction), or there may have been users testing the waters of this new online course platform (sampling).

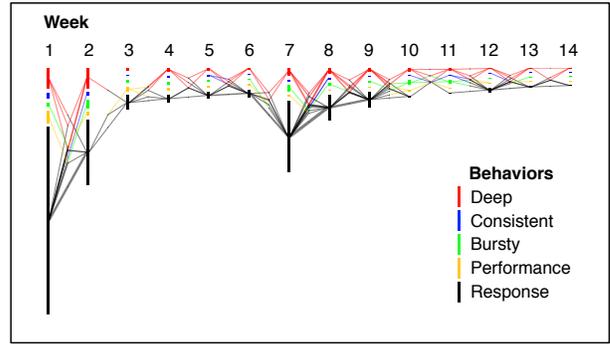
## 6. STUDENT TRANSITIONS

After applying the thresholding algorithm, we generate user behavior transition diagrams for each course. The size of each colored bar is scaled according to the amount of students exhibiting the behavior. The transition lines in between the bars are sized and directed based on user migration between sets of behaviors.

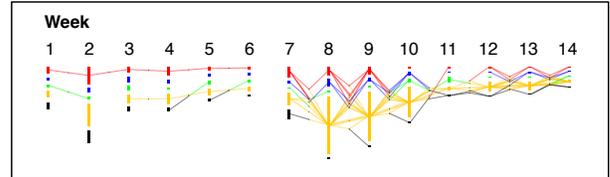
Using these diagrams, we can observe changes in the behaviors themselves, and the transitional motifs that occur due to user migration. After the first week or two, a single behavior persists as the largest. Additionally, this behavior tends to act as a hub for user migration. This phenomenon significantly highlights the fact that the behaviors may manifest differently despite the existence of the same five behaviors among all five courses.

In 2.01x, most user migration occurs into and out of the response behavior, with a secondary focus on the deep behavior. Notable moments occur in week 5 and weeks 10 to 12, where migration between consistent and deep occur. Otherwise, there are several recurrent transitions. These motifs include each permutation of deep and/or response migrating to deep and/or response.

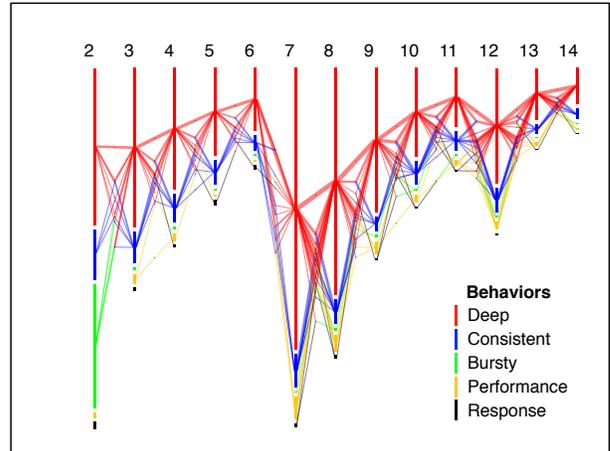
In 3.091x, most user migration occurs into and out of the performance behavior. Most unusually, there is very little migration in the entire first half of the course. Only in the second half does migration pick up to levels we would have expected given the results of the other courses. Although some migration patterns through the performance behavior



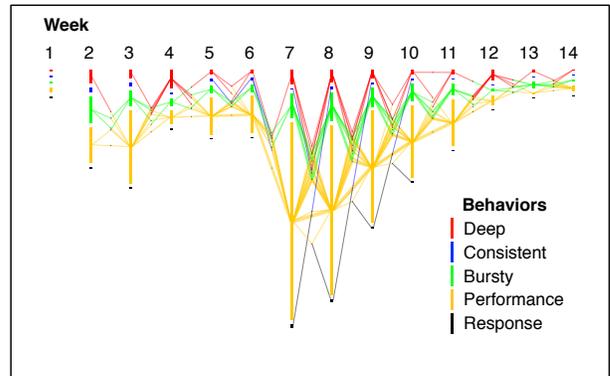
(a) 2.01x, Spring 2013



(b) 3.091x, Spring 2013



(c) 6.002x, Fall 2012



(d) 6.002x, Spring 2013

Figure 2: User behavior transitions over time. Vertical bars are numbers of students performing each behavior. Diagonal groups indicate transitions: for example, the transition  $\blacktriangleright$  indicates students who were **Deep** and **Bursty** and have transitioned to **Consistent**. Transition thickness is the log of the number of students involved.

repeat occasionally, they only occur for two to three weeks at a time. Thus, we do not infer any transitional motifs from this course.

In 6.002x fall, most user migration occurs through the deep behavior, with a secondary focus on the consistent behavior. A unique circumstance occurs between weeks one and two with the migration of the initially enormous bursty behavior. Besides this, the transitional motifs include each permutation of deep and/or consistent migrating to deep and/or consistent.

In 6.002x spring, most user migration occurs through the performance behavior. Unlike the other courses, there are two more behaviors through which there is significant migration: the deep and bursty behaviors. As a result, we see many more motifs than simply the permutations of the top two behaviors. In the early weeks, migration is heaviest through deep and performance. This means that early on, users are both engaged and performing well. In the middle weeks, during and after the midterm, there is a chaotic shuffle between behaviors as users deal with the course differently. In the later weeks, however, deep migration falls off and users mostly move between bursty and performance. This may suggest that users are capable of finishing their work in a single day or two and achieving high correctness simultaneously. This result could perhaps reflect a decreased difficulty in the later weeks of the course. The occurrence of multiple large behaviors appears to tell us more about the evolution of user behavior.

## 7. CONCLUSION

In this comparative study of four MOOC courses, we show how users follow five specific behaviors across the courses. We found that although these behaviors are common, their patterns of occurrence vary across courses. Through our multi-dimensional data and our adaptation of NMF, the results reveal in great detail the differences in behavior over time between the courses. Because our method analyzes behavior at every step of the MOOC experience, our work can improve the learning experience for all users, not just those that plan to finish the course. For future work, we can expand the purposes of user behavior trajectories by using Markov modeling for prediction. We can add newer, more descriptive features in addition to running the analysis with a higher rank in order to discover possible alternative behaviors. If course outcomes and assessment information are available, we can combine these with the dynamic behavioral motifs to better understand the underlying processes that fuel behavioral changes.

## 8. REFERENCES

- [1] Veeramachaneni, Kalyan, Halawa, Sherif, Dernoncourt, Franck, O'Reilly, Una-May, Taylor, Colin, and Do, Chuong. Moocdb: Developing standards and systems to support mooc data science. arXiv preprint arXiv:1406.2015, 2014a.
- [2] Sra, Suvrit, and Inderjit S. Dhillon. "Generalized nonnegative matrix approximations with Bregman divergences." *Advances in neural information processing systems*. 2005.
- [3] Boutsidis, Christos, and Efstratios Gallopoulos. "SVD based initialization: A head start for nonnegative matrix factorization." *Pattern Recognition* 41.4 (2008): 1350-1362.
- [4] Swinson, Christina J. "Mathematica." *CJs Blog of Miscellanies and Accelerator Physics*. N.p., 21 Jan. 2011. Web. 05 Mar. 2016.
- [5] Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." *Nature* 401.6755 (1999): 788-791.
- [6] Breslow, Lori, et al. "Studying learning in the worldwide classroom: Research into edX's first MOOC." *Research & Practice in Assessment* 8 (2013).
- [7] MOOCs@Edinburgh Group. MOOCs@Edinburgh (2013): Report#1, Available at: <http://hdl.handle.net/1842/6683> [Accessed: 20/01/14].
- [8] Jordan, K. (2013). MOOC Completion Rates: The Data, Available at: <http://www.katyjordan.com/MOOCproject.html> [Accessed: 18/02/14].
- [9] Hill, P. (2013). The Most Thorough Summary (to date) of MOOC Completion Rates|e-Literate. e-Literate blog. Retrieved June 10, 2013, from <http://mfeldstein.com/the-most-thorough-summary-to-date-of-mooc-completion-rates/>
- [10] Liyanagunawardena, T. R., Adams, A. A. & Williams, S. A. (2013). MOOCs: a systematic study of the published literature 2008–2012. *The International Review of Research in Open and Distance Learning*, 14, 3, 202–227.
- [11] Ebben, M. & Murphy, J. S. (2014). Unpacking MOOC scholarly discourse: a review of nascent MOOC scholarship. *Learning, Media and Technology*, 39, 3, 1–18. doi: 10.1080/17439884.2013.878352
- [12] EDUCAUSE (2012). What Campus Leaders Need to Know About MOOCs. EDUCAUSE BRIEFS. Retrieved June 10, 2013, from <http://www.educause.edu/library/resources/what-campus-leaders-need-know-about-moocs>
- [13] Onah, Daniel F. O., Sinclair, Jane and Boyatt, Russell (2014) Dropout rates of massive open online courses : behavioural patterns. In: 6th International Conference on
- [14] Pedregosa et al., Scikit-learn: Machine Learning in Python, *JMLR* 12, pp. 2825-2830, 2011. Education and New Learning Technologies, Barcelona, Spain, 7-9 Jul 2014. Published in: *EDULEARN14 Proceedings* pp. 5825-5834.
- [15] Belanger, Y. (2013). Bioelectricity : A Quantitative Approach, Available at [http://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/6216/Duke\\_Bioelectricity\\_MOO\\_C-Fall2012.pdf](http://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/6216/Duke_Bioelectricity_MOO_C-Fall2012.pdf)
- [16] F. Gruenwald, E. Mazandarani, C. Meinel, R. Teusner, M. Totschnig, and C. Willems, "openHPI-a Case-Study on the emergence of two learning communities," in *Proc. IEEE Global Eng. Edu. Conf.*, 2013, pp. 13–15.
- [17] Kizilcec, R. F., Piech, C., & Schneider, E. (2013, April). 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.

# Collaborative Problem Solving Skills versus Collaboration Outcomes: Findings from Statistical Analysis and Data Mining

Jiangang Hao  
Educational Testing Service  
ETS Rosedale Road, MS 02-T  
Princeton, NJ 08541  
jhao@ets.org

Lei Liu  
Educational Testing Service  
ETS Rosedale Road  
Princeton, NJ 08541  
lliu001@ets.org

Alina A von Davier  
Educational Testing Service  
ETS Rosedale Road  
Princeton, NJ 08541  
avondavier@ets.org

Patrick Kyllonen  
Educational Testing Service  
ETS Rosedale Road  
Princeton, NJ 08541  
pkyllonen@ets.org

Christopher Kitchen  
Educational Testing Service  
ETS Rosedale Road  
Princeton, NJ 08541  
ckitchen@ets.org

## ABSTRACT

With the aid of educational data mining and statistical analysis, we investigate the relationship between collaboration outcomes and collaborative problem solving (CPS) skills exhibited during the collaboration process. We found that negotiation skill contributes positively to the collaboration outcomes while purely sharing information does the opposite.

## Keywords

collaborative problem solving, simulation-based assessment, random forest

## 1. INTRODUCTION

Collaborative problem solving (CPS) is widely considered as one of the critical skills for academic and career success in the 21st century [9]. However, assessing CPS, particularly in a large-scale and standardized way, is very challenging, as one must take into account the forms of collaboration, the size of teams, and assessment contexts. Among the existing studies on assessing CPS, most of them are not designed from the perspective of a standardized assessment, but more from the perspective of revealing some important aspects of CPS [6, 16, 5, 22]. A recent review can be found in [21]. The first large-scale and standardized assessment for CPS was the international Assessment and Teaching of 21st century skills project (ATC21S) carried out by Griffin and colleagues [9, 4]. In this assessment, two students collaborate via text chat to solve computer-based CPS tasks and their communications as well as some other features (such as the response time) were coded automatically according

to a CPS framework [1]. Another large-scale assessment for CPS was carried out by the Programme for International Student Assessment (PISA) in its sixth survey in 2015 [17]. In this assessment, students collaborate with different number of virtual partners (avatars) on a set of computer-based collaborative tasks and they communicate with their virtual partners by choosing from a list of predefined texts. Both ATC21S and PISA 2015 consider the CPS as skills across different domains and the tasks used in their assessments are not confined into a specific domain.

In this paper, we report our findings on the relationship between the CPS skills and the collaboration outcomes in the domain of science, as we think CPS is more likely to be domain dependent. We developed a simulation-based task, in which two participants collaborate via text chat to complete a set of questions and activities on volcanoes [10]. We choose a simulation-based task because it provides students with opportunities to demonstrate proficiencies in complex interactive environments that traditional assessment formats cannot afford [14], which is especially suitable for measuring the complex skills such as CPS.

In the simulation task, for each item, we ask each member of a dyadic team to respond individually first (initial response). Then, after collaboration, each of them will be given a chance to submit a revised response. The difference between the initial and revised responses directly encodes the effect due to collaboration. Based on the data collected using Amazon Mechanical Turk, we introduce two variables, “number of changes” and “score change”, to characterize the collaboration outcomes. The “number of changes” is the total number of attempts by the team members to change the initial responses after the collaboration. Some of the attempts change the responses from correct to incorrect while some change the responses from incorrect to correct. This number reflects the willingness to make a change after the collaboration. On the other hand, the “score change” is the sum of the score changes between the initial and revised responses, which quantifies the results of the changes. Based on these two variables, we classify the teams into “effective

collaboration” (e.g., teams that have positive “score change”) and “ineffective collaboration” (e.g., teams that have negative “score change” or zero “number of changes”).

In addition to quantifying the collaboration outcomes, we introduced a “CPS profile” to characterize the CPS skills exhibited by each team during the collaboration process. The CPS profile is defined as the frequency distribution of CPS skills (unigram) and the consecutive CPS skill pairs (bigram). Random forest classification analysis [12, 3] is used to analyze the relationship between collaboration outcomes and the CPS skills. Random forest is a decision tree-based binary classifier, with increased robustness by using multiple trees rather than a single tree. It is mainly used as a classifier to map the features (independent variables) to labels (dependent variables). When training a random forest classifier, the relative importance of the feature variables for determining the labels can be obtained as a by-product. In our case, the feature variables are the CPS profile and the labels are the two classes of collaboration outcomes, e.g., effective and ineffective collaborations. By training a random forest classifier on the data, we found that negotiation skill is more important for a successful collaboration outcome.

## 2. METHOD

### 2.1 Assessment Instruments

We designed a research study to explore the relationship between CPS skills and the collaboration outcomes. In this large-scale study, we focused on the domain of science and limited the number of members of each team to two. We used text chat as the collaboration medium. There were two major assessment instruments: 1) A standalone test for general science knowledge consisting of 37 multiple-choice items adapted from the Scientific Literacy Measurement (SLiM) instrument [18]; 2) A web-based collaborative simulation task on volcanoes that require two participants collaborate to complete.

The simulation task was modified from an existing simulation, Volcano Trialogue [23]. In this simulation task, two participants worked together via text chat to complete the tasks. All of the turn-by-turn conversations and time-stamped responses to the questions were recorded in a carefully designed log file [11]. These conversations were used to measure CPS skills, while the responses to the in-simulation science items were used to measure science inquiry skills [23]. Figure 1 shows screenshot of the simulation task.

To capture the evidence for the outcomes of the collaboration, we designed a four-step response procedure for each item in the task: 1) Each participant was prompted to respond to the item individually before any collaboration; 2) Each participant was prompted to discuss the item with her partner; 3) Each participant was prompted to revise her initial response if she wanted; 4) A representative was randomly chosen to submit a team answer.

In this way, the changes in the responses before and after the collaboration reflect how effective the collaborations were and allow us to probe directly what CPS skills are more important for better collaboration outcomes.

### 2.2 Participants and Data



**Figure 1: Screenshots from the collaborative simulation task.**

We collected data through Amazon Mechanical Turk, a crowdsourcing data collection platform [13]. We recruited 1,000 participants with at least one year of college education to take the general science test. Then, they were teamed randomly into dyads to take the collaborative simulation task.

After removing incomplete responses, we had complete responses from 493 dyads. However, a further scrutiny of the data showed that many of the teams started some conversations even before the system prompted them to discuss. This means that they started conversations before or during the period that they are supposed to make initial responses individually. Different teams had nonprompted conversations for a different subset of the items, which complicates the analysis. Of the teams, 82 did not have nonprompted conversations while the other teams had nonprompted discussions for a varying number of items. We compared the scores of the general science knowledge test for participants from the 82 teams with the scores for the rest of the teams via a two-tailed t-test for independent samples, and the resulting p-value is 0.38. This indicates that participants from the 82 teams are not different in a statistically significant way from the rest of the participants in terms of the general science knowledge. To make our analysis clean, we will stick to the data from this 82 teams throughout this paper.

The data from the simulation task for each team include the responses to the items in the simulation and the text chat communications between the dyads around each item. There are 7 multiple-choice equivalent items. Around each item, there are about 5 turns of conversations.

### 2.3 Analysis

The focus of this paper is to investigate the relationship between the CPS skills and the collaboration outcomes. As such, our analysis focuses on the responses and communications in the collaborative simulation task.

#### 2.3.1 Scoring and Annotating

Students’ responses to the seven multiple-choice equivalent items were scored based on the corresponding scoring rubrics as presentend in [23]. In addition to the outcome response

data, we also applied a CPS framework to annotate the chat communications during the collaboration [15]. This CPS framework was developed based on the findings from computer-supported collaborative learning (CSCL) research [2, 7, 9, 21] and the PISA 2015 Collaborative Problem Solving Framework [17].

The framework outlines the four specific categories of the CPS construct (skills) we would like to focus on: *sharing ideas*, *negotiating ideas*, *regulating problem-solving activities*, and *maintaining communication*. Each of these major categories had some subcategories and the total number of subcategories amounted to 33 and a summary of the coding rubrics can be found in Table 1. All the coding was done at the subcategory level, based on which of the four major categories were assigned at a later point.

Two human raters were trained on the CPS framework, and they double-coded a subset of the discourse data (15% of the data). The unit of analysis was each turn of a conversation, or each conversational utterance. The raters had two training sessions before they started independent coding. In the first session, the author of the CPS framework (the second author) trained both raters on the 33 subcategories of CPS skills using the skills definitions and coding examples for each subcategory. In the second training session, the trainer and two raters coded data from one dyad together to practice the application of specific codes and address issues specific to classifying utterances using the CPS framework. After the training sessions, the two raters independently coded discourse data from about 80 dyads.

We used the unweighted kappa statistic to measure the degree of agreement between the human raters' coding. The unweighted kappa was 0.61 for all 33 subcategories and 0.65 for the four major categories. According to Fleiss and Cohen [8], a kappa value of 0.4 is an acceptable level of agreement for social science experiments.

### 2.3.2 Quantifying the Collaboration Outcomes

The difference between the revised response and initial response is a direct measure of the collaboration outcomes. If we treat each dyad as the unit of analysis, we need to define variables to quantify the answer changes for each item. We first introduce the "number of changes" (denoted as  $n$ ) to quantify how many revised responses are different from initial responses from both members of each dyad for each item. The possible values for  $n$  are  $\{0, 1, 2\}$ :  $n$  is zero when nobody makes any changes, one when only one person makes changes, and two when both members make changes. Next, we introduce "score change" (denoted as  $s$ ) to quantify the total score changes between the revised response and the initial response from both members of each dyad for each item. The definition of  $s$  is the sum of the score difference between initial responses and revised responses for the two members of each dyad. The possible states for  $s$  are  $\{-2, -1, 0, 1, 2\}$ . One should note that for the state  $s = 0$ , there are two different possibilities. The first is that both members do not change their responses. The second is that one member changes a response from incorrect to correct and the other changes from correct to incorrect. Therefore, to have a complete description of the changes at a dyadic level, we introduce the vector "item collaboration effect" for each

item,  $\delta_k = (s_k, n_k)$ , with  $\delta_k$  defined at the item level and subscript  $k$  denoting the item number. At the task level, we simply sum all items, which gives  $\Delta = (S, N)$ , where  $S = \sum_k s_k$  and  $N = \sum_k n_k$ . By convention, we use the lowercase  $n$  and  $s$  to denote the item level changes and the uppercase  $N$  and  $S$  to denote the task-level changes.

### 2.3.3 Quantifying the CPS Skills

Each turn-by-turn conversations was classified in one of the four categories of CPS skills (e.g., share ideas, negotiate ideas, regulate problem solving, and maintain communication). We introduce a "CPS profile" as a quantitative representation of the CPS skills of each dyad. The profile was defined by the frequency counts of each of the four CPS-skill categories or their combinations and had two levels, unigram and bigram. The unigram, bigram, or even ngram levels are used in natural language processing to represent text. We borrow this idea here to represent CPS skills and limit us to the unigram and bigram as the frequency count is too low for other ngram. The frequency counts of the different CPS skills were used at the unigram level, while the frequency counts of consecutive pairs of CPS skills in the conversations were used at the bigram level. As such, each dyadic team's communications can be represented by the corresponding CPS profile.

It is worth noting that though we consider only unigram and bigram of the CPS skills, other collaboration-related information can also be appended to the profile. For example, the number of turns, the total number of words, etc. Such a profile is essentially a vector representation of collaboration skills exhibited by each team. The vector nature of this representation allows us to easily calculate "similarity" or "dissimilarity" among the teams, which is the foundation of cluster analysis.

## 3. FINDINGS

We have introduced two variables,  $N$  and  $S$ , to quantify the collaboration outcomes. We also introduced the CPS profile to quantify the CPS skills. Now, we investigate the relationship between the CPS skills and the collaboration outcomes.

### 3.1 Effective versus Ineffective Collaboration

Based on the  $N$  and  $S$  variables, we define the effective collaboration and ineffective collaboration as follows

- Effective collaboration:  $N > 0 \cap S > 0$ .
- Ineffective collaboration:  $(N > 0 \cap S \leq 0) \cup N = 0$ .

We need to point out that the criteria for effective collaboration is not necessarily a fixed one. In the current study, we considered the collaboration as effective as long as at least one member made at least a total net change from incorrect to correct. If nobody in the team made at least one total net correct change, we thought of the collaboration as ineffective. Figure 2 shows how the 82 teams were distributed in the space spanned by  $S$  and  $N$ .

**Table 1: Coding rubric of CPS skills used in this paper was developed based on a review of CSCL research findings [2, 7, 9], and the PISA 2015 Collaborative Problem Solving Framework [17], with a focus on CPS in the domain of science. More details about the CPS framework can be found in [15].**

CPS skills	Student performance (subcategories)
Sharing ideas	<ol style="list-style-type: none"> <li>1. Student gives task-relevant information (e.g., individual response) to the teammate.</li> <li>2. Student points out a resource to retrieve task-relevant information.</li> <li>3. Student responds to the teammate's request for task-relevant information.</li> </ol>
Negotiating ideas	<ol style="list-style-type: none"> <li>4. Student expresses agreement with the teammates.</li> <li>5. Student expresses disagreement with teammates.</li> <li>6. Student expresses uncertainty of agree or disagree.</li> <li>7. Student asks the teammate to repeat a statement.</li> <li>8. Student asks the teammate to clarify a statement.</li> <li>9. Student rephrases/complete the teammate's statement.</li> <li>10. Student identifies a conflict in his or her own idea and the teammate's idea.</li> <li>11. Student uses relevant evidence to point out some gap in the teammate's statement.</li> <li>12. Student elaborates on his or her own statement.</li> <li>13. Student changes his or her own idea after listening to the teammate's reasoning</li> </ol>
Regulating problem solving	<ol style="list-style-type: none"> <li>14. Student identify the goal of the conversation.</li> <li>15. Student suggests the next step for the group to take.</li> <li>16. Student expresses confusion/frustration or lack of understanding.</li> <li>17. Student expresses progress in understanding.</li> <li>18. Student reflects on what the group did.</li> <li>19. Student expresses what is missing in the teamwork to solve the problem.</li> <li>20. Student checks on understanding.</li> <li>21. Student evaluates whether certain group contribution is useful or not for the problem solving.</li> <li>22. Student shows satisfaction with the group performance.</li> <li>23. Student points out some gap in a group decision.</li> <li>24. Student identifies a problem in problem solving.</li> </ol>
Maintaining communication	<ol style="list-style-type: none"> <li>25. Student responds to the teammate's question (using texts and text symbols).</li> <li>26. Student manages to make the conversation alive (using texts and text symbols, using socially appropriate language).</li> <li>27. Student waits for the teammate to finish his/her statement before taking turns.</li> <li>28. Student uses socially appropriate language (e.g., greeting).</li> <li>29. Student offers help.</li> <li>30. Student apologizes for unintentional interruption.</li> <li>31. Student rejects the teammate's suggestions without an accountable reason.</li> <li>32. Student inputs something that does not make sense.</li> <li>33. Student shows understanding of the teammate's frustration.</li> </ol>

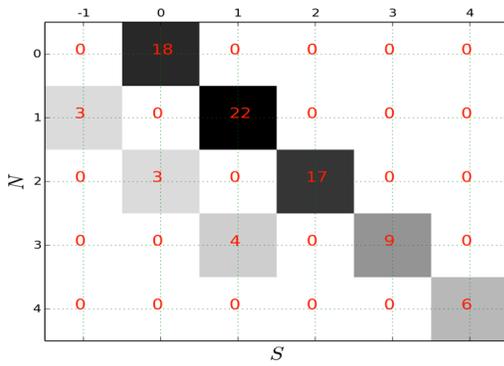


Figure 2: The distribution of the teams in space spanned by N and S.

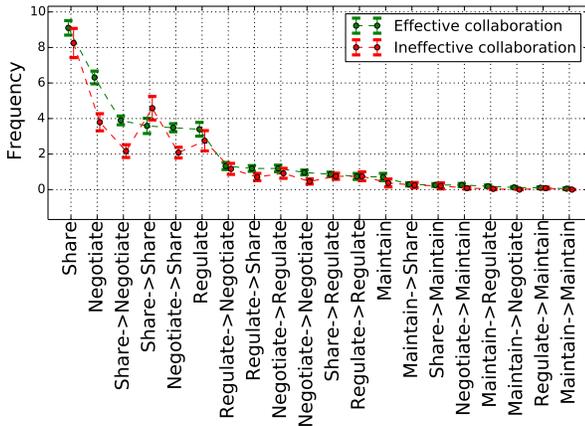


Figure 3: Unigram and bigram profile of CPS skills for the teams corresponding to effective and ineffective collaborations.

Next, we compare the mean CPS profiles of the teams from the effective and ineffective collaborations and the results are shown in Figure 3.

From these results, one can readily see that at the unigram level, the teams with effective collaboration show statistically significantly more negotiating skills than the teams with ineffective collaboration. At the bigram level, teams with effective collaboration exhibited statistically significantly more of the following consecutive CPS skill pairs: share-negotiate, negotiate-share, regulate-share, and negotiate-negotiate. However, the teams with ineffective collaboration showed many more share-share skill pairs.

### 3.2 Relative Importance of CPS Skills

Figure 3 shows certain CPS skills exhibit more different frequency for effective and ineffective collaborations, which means they have more weight in determining the collaboration outcomes. To get a more quantitative measure of the relative importance of each CPS skills (or skill pairs), we used two methods as follows.

First, we perform a t-test for each of the CPS skills (or skill

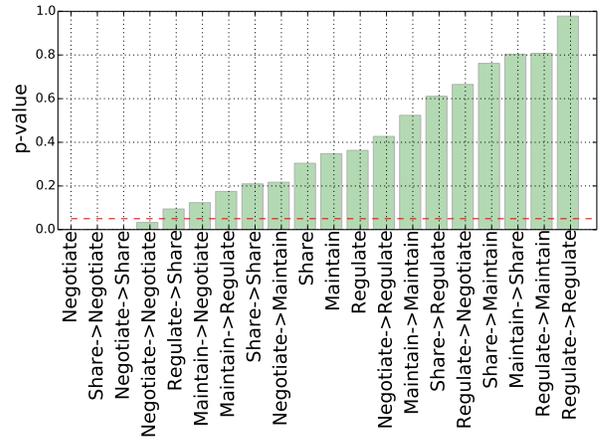


Figure 4: P-value of t-test on the frequency of different CPS skills corresponding to effective and ineffective collaborations. The red horizontal dashed line corresponds to a significant level of 0.05.

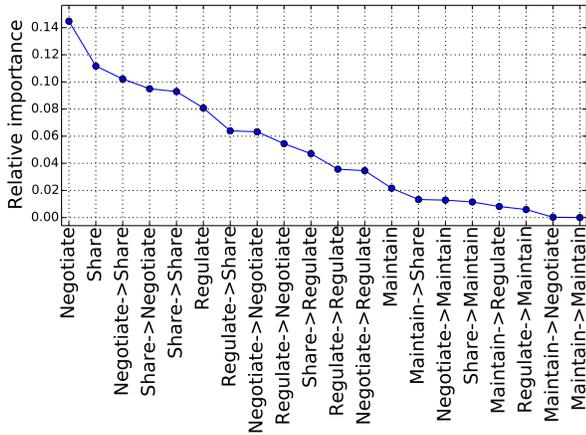
pairs) for the effective collaboration and ineffective collaboration groups. We use the corresponding p-value to tell which skills or skill pairs show more distinction. The p-value for each component of the CPS profile was shown in Figure 4. If we choose 0.05 as the significance level, negotiate, share-negotiate, negotiate-share and negotiate-negotiate stand out immediately.

A second method we used to find out the relative importance of the CPS skills or skill pairs (feature variables) is random forest classifier [12, 3]. We choose the collaboration outcomes as label variables. During the training of the classifier, a set of decision cuts were made on each feature variable. The relative depth of a feature used as a decision node in a decision tree represents the relative importance of that feature with respect to the predictability of the target labels. Generally speaking, features used at the top level of the decision tree will affect a larger fraction of the sample in terms of the final prediction. Therefore, the expected fraction over the trees in the forest can be used as an estimate of the relative importance of the features. Figure 5 shows the relative importance of the CPS skills and skill pairs based on such an analysis. The results show that negotiation-related skills top the ranking.

The results from these two different analyses converge nicely on that negotiation is a very critical skill for successful collaboration. This finding is consistent with the findings in the literature on knowledge-building discourse [19, 20], as knowledge is often built upon its use and negotiation includes interpretive process of making meaning of exchanged ideas.

## 4. CONCLUSIONS AND IMPLICATIONS

In this paper, we introduced a CPS profile approach to quantify the CPS skills of each team and found that the negotiation skill at the unigram level is important for better collaboration outcomes. At the bigram level, we found that more negotiation-related skill pairs, such as share-negotiate,



**Figure 5: Relative feature importance based on a random forest classifier.**

negotiate-share, regulate-share, and negotiate-negotiate, leads to more effective collaboration outcomes. However, purely sharing information with each other (share-share) is associated with poorer collaboration outcomes. This empirical finding may also inform the development of an outcome-oriented scale for CPS skills.

The current study also has limitations. For example, the items in the task are all relatively easy so that there are few turns for each item. There are not many items in the task, which limits the effect of the collaboration outcomes. All these issues will be resolved in our next round of data collection and analysis.

## 5. ACKNOWLEDGMENTS

Funding for this project is provided by Educational Testing Service through the game, simulation and collaboration initiative.

## 6. REFERENCES

- [1] R. Adams, A. Vista, C. Scoular, N. Awwal, P. Griffin, and E. Care. Automatic coding procedures. *Assessment and teaching of 21st century skills*, 2, 2015.
- [2] B. Barron. When smart groups fail. *The journal of the learning sciences*, 12(3):307–359, 2003.
- [3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [4] E. Care and P. Griffin. An approach to assessment of collaborative problem solving?. *Research & Practice in Technology Enhanced Learning*, 9(3):367–388, 2014.
- [5] E. G. Cohen, R. A. Lotan, B. A. Scarloss, and A. R. Arellano. Complex instruction: Equity in cooperative learning classrooms. *Theory into practice*, 38(2):80–86, 1999.
- [6] L. A. DeChurch and J. R. Mesmer-Magnus. The cognitive underpinnings of effective teamwork: a meta-analysis. *Journal of Applied Psychology*, 95(1):32, 2010.
- [7] P. Dillenbourg and D. Traum. Sharing solutions: Persistence and grounding in multimodal collaborative problem solving. *The Journal of the Learning*

- Sciences*, 15(1):121–151, 2006.
- [8] J. L. Fleiss and J. Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 1973.
- [9] P. Griffin, B. McGaw, and E. Care. *Assessment and teaching of 21st century skills*. Springer, 2012.
- [10] J. Hao, L. Liu, A. von Davier, and P. Kyllonen. Assessing collaborative problem solving with simulation based tasks. *proceeding of 11th international conference on computer supported collaborative learning*, 2015.
- [11] J. Hao, L. Smith, R. Mislavy, A. von Davier, and M. Bauer. Taming log files from game and simulation based assessment: data model and data analysis tool. *ETS Research Report*, in press.
- [12] T. K. Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.
- [13] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.
- [14] E. Klopfer, S. Osterweil, J. Groff, and J. Haas. Using the technology of today, in the classroom today. *The Education arcade*, 2009.
- [15] L. Liu, J. Hao, A. A. von Davier, P. Kyllonen, and D. Zapata-Rivera. A tough nut to crack: Measuring collaborative problem solving. *Handbook of Research on Technology Tools for Real-World Skill Development*, page 344, 2015.
- [16] H. F. O’Neil. *Workforce readiness: Competencies and assessment*. Psychology Press, 2014.
- [17] Organization for Economic Co-operation and Development [OECD]. Pisa 2015 draft collaborative problem solving assessment framework. *OECD Publishing*, 2013.
- [18] C.-J. Rundgren, S.-N. C. Rundgren, Y.-H. Tseng, P.-L. Lin, and C.-Y. Chang. Are you slim? developing an instrument for civic scientific literacy measurement (slim) based on media coverage. *Public Understanding of Science*, 21(6):759–773, 2012.
- [19] M. Scardamalia and C. Bereiter. Computer support for knowledge-building communities. *The journal of the learning sciences*, 3(3):265–283, 1994.
- [20] G. Stahl. *Group Cognition: Computer Support for Building Collaborative Knowledge (Acting with Technology)*. The MIT Press, 2006.
- [21] A. A. Von Davier and P. F. Halpin. Collaborative problem solving and the assessment of cognitive skills: Psychometric considerations. *ETS Research Report Series*, 2013(2):i–36, 2013.
- [22] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688, 2010.
- [23] D. Zapata-Rivera, T. Jackson, L. Liu, M. Bertling, M. Vezzu, and I. R. Katz. Assessing science inquiry skills using dialogues. In *Intelligent Tutoring Systems*, pages 625–626. Springer, 2014.

# Hint Availability Slows Completion Times in Summer Work

Paul Salvador Inventado,  
Peter Scupelli  
School of Design  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA, USA  
paulsb@andrew.cmu.edu,  
pgs@andrew.cmu.edu

Eric G. Van Inwegen,  
Korinn S. Ostrow,  
Neil Heffernan III  
Worcester Polytechnic Institute  
100 Institute Road  
Worcester, MA, USA  
egvaninwegen@wpi.edu,  
ksostrow@wpi.edu,  
nth@wpi.edu

Jaclyn Ocumpaugh,  
Ryan S. Baker,  
Stefan Slater,  
Mia Almeda  
Teachers College, Columbia University  
625 W. 120th Street,  
New York, NY, USA  
jo2424@tc.columbia.edu,  
baker2@exchange.tc.columbia.edu,  
slater.research@gmail.com,  
victoriaalmeda@gmail.com

## ABSTRACT

On-demand help in intelligent learning environments is typically linked to better learning, but may lead to longer completion times. This present work provides an analysis of how students interacted with a summer learning assignment when on-demand help was available, compared to when it was not. When hints were available from the start, students were more likely to delay work, compared to students for whom step-wise hints were only available after the third problem. When hints were always available, participants took significantly more time to complete a mastery learning assignment. We interpret this difference in time to complete the assignment as an opportunity to re-engage in productive math learning.

## Categories and Subject Descriptors

H1.2 [Information Systems]: User/Machine Systems – human factors

## General Terms

Measurement, Design, Experimentation, Human Factors

## Keywords

Hints, completion time, randomized controlled trial, ASSISTments

## 1. INTRODUCTION

Help-functions—including on-demand help, contextualized hints, or supplementary learning materials [2]—are a major asset of modern intelligent learning environments. These functions have often been associated with better student learning outcomes ([1][9][25]), but not all help has proven equally effective, and even well-crafted hints may be used ineffectively by students who do not actually need them ([2][20]). Research has shown cases in which help functions fail [1] and has sought to identify the contexts in which different types of help strategies are most effective ([12][22]).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EDM '16, June 29–July 2, 2016, Raleigh, NC, USA.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

Analysis of hint use serves many purposes and may be an obvious answer to *wheel-spinning*, where a student persists long past the point of productive effort [6]. It is also feasible to predict the problematic behaviors of hint misuse or hint abuse. Previous research has analyzed relationships between problem-related features (e.g., problem length, number of hints available, hint length) and student affect, behavior, and learning ([3][11][13][19]). Among other findings, hint length has been positively correlated with *gaming the system* [3], a behavior incorporating help abuse that is associated with poorer learning outcomes ([21][23]). Other research has indicated problems unrelated to the deliberate behavior of students. For example, poorly designed hints may lead to ineffective hint usage ([4][15]). Research also suggests that low-knowledge students, or those that need the most help, are the least likely to use it effectively ([2][3][18]).

In this paper, we present results from a randomized controlled trial (RCT) that examined how hint availability effected other aspects of student learning, including the time required for students to complete the assignment, presented using the ASSISTments online learning system [11]. To our surprise, we found that students who were given the option to request on-demand hints appeared to spend more time on tasks unrelated to the completion of the problem set (e.g., solve other problem sets, work on learning activities outside of ASSISTments, or engaged in activities external to the learning system). Specifically, these students took more time to complete the assignment even though they did not (a) spend significantly more time on task, (b) answer significantly more problems, or (c) make significantly more attempts per problem as compared to the control condition. The analyses presented herein explore this pattern more thoroughly, in order to contribute to the growing literature on help systems in online learning.

## 2. ASSISTMENTS

ASSISTments is an online learning system designed primarily for middle school mathematics. The platform allows teachers to easily create and assign their own problem sets (including questions, associated solutions, mistake messages, feedback) or to select from a set of *ASSISTments Certified Problems* (vetted by ASSISTment's expert team) ([11][22]). These problem sets simultaneously support student learning and serve as automated formative assessments that provide real-time data to teachers [11]. The platform is also used as a research tool to conduct RCTs

([8][16][26]). ASSISTments logs learning-related features at multiple granularities (e.g., problem text, problem type, student actions, timestamps, etc.). Figures 1 and 2 show screenshots of the types of ASSISTments problems used in the present work. Based on experimental condition, students were able to request hints, receive feedback messages, or simply answer the question.

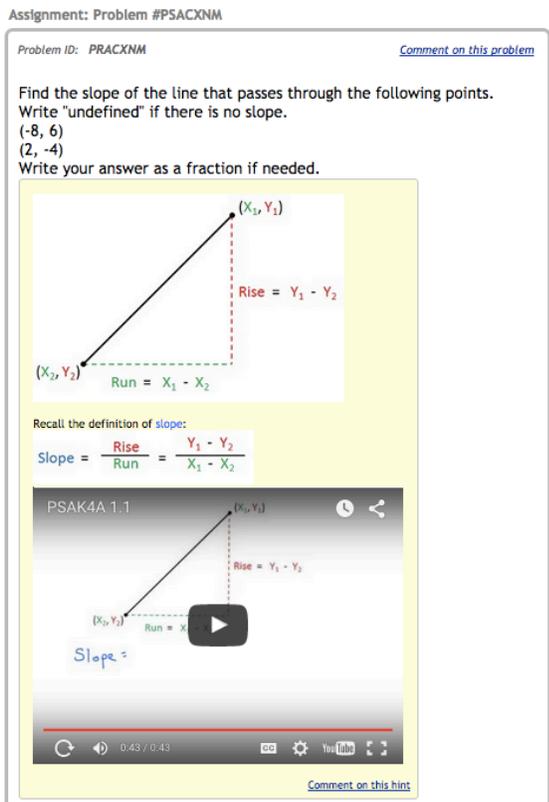


Figure 1. An example question from the hints-early condition, presented with its associated hints.

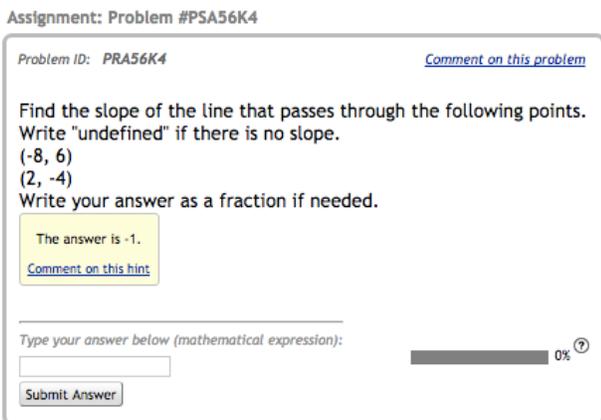


Figure 2. The same example question as presented in the no-hints-early condition.

### 3. METHODOLOGY

This study used an RCT design in which several linear presentations of a problem set were embedded within two conditions: a control condition with on-demand hints (*hints-early*, HE) or an experimental condition with on-demand hints only after the third problem (*no-hints-early*, NHE). The problem set for this study (available at [14]) was chosen from ASSISTments Certified

content and was designed to address the 8<sup>th</sup> grade Common Core State Standard, “Finding Slope from Ordered Pairs,” [17]. It was deployed within ASSISTments as a *Skill Builder*, a type of problem set requiring students to accurately answer three consecutive problems in order to complete the assignment.

Students were randomly assigned into one of 12 groups (6 control and 6 experimental) when they began the problem set. As depicted in Figure 3, students in each group saw the same 3 problems, but presentation order was randomized to minimize cheating (i.e., A-B-C, A-C-B, B-A-C, etc.). All students, regardless of condition, received immediate correctness feedback (e.g., “Sorry try again: ‘2’ is not correct”).

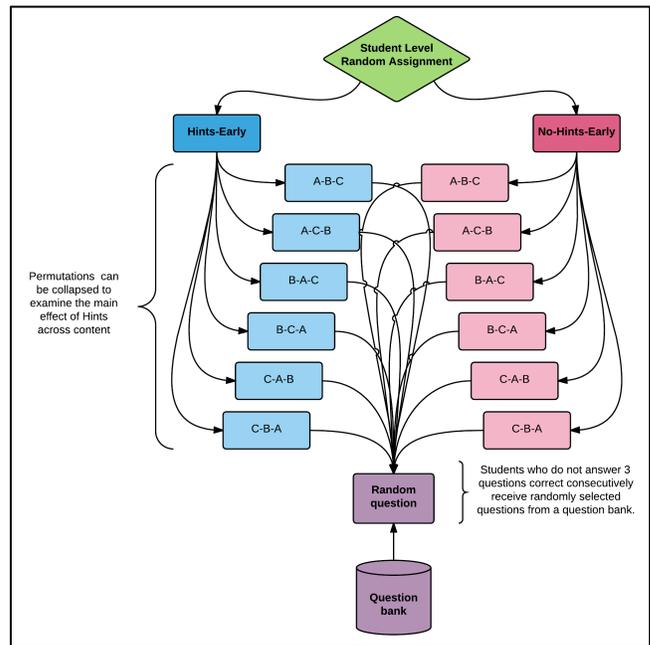


Figure 3. Research Design depicted as a flow chart.

In a Skill Builder, students are able to attempt each problem multiple times, but (in line with common practice) problem accuracy is calculated using binary correctness on the student’s first attempt (1=Right, 0=Wrong) [10]. Students who did not answer the first three problems correctly were assigned additional problems randomly selected from a skill bank. In order to provide all students with adequate learning support, all students were permitted on-demand hints—regardless of condition—upon reaching these additional problems.

Figures 1 and 2 demonstrate how the interface differed by condition. In the HE condition, students could access hints at any time by clicking on a button in the lower right corner of their screen. The problem remained on the screen while video tutorials and text-based hints were simultaneously delivered (text-based hints ensured access when school firewalls or connectivity issues may have limited access to YouTube). In contrast, the NHE condition only offered a *Show Answer* button in the lower right corner of the screen during the first three problems (a design seen in early intelligent tutors [24]) allowing students who were stuck to move on to the next problem and eventually complete the assignment.

#### 3.1 Student Populations

To help retain students’ math skills, the Skill Builder in this study was one of many assigned as summer work at two suburban high

schools (henceforth Schools A and B) in the Northeastern U.S. **School A** was an agricultural/vocational high school that assigned this Skill Builder to 113 9<sup>th</sup> graders and 95 10<sup>th</sup> graders, along with numerous other Skill Builders (32 in 9<sup>th</sup> grade, 36 in 10<sup>th</sup>). **School B** was a high school without a known specialization; it assigned this Skill Builder (as well as 45 others) to 204 9<sup>th</sup> graders. Students worked on these assessments throughout the summer (Jun-Sept 2015) and data was harvested six months later.

Condition distributions were well matched for student gender (HE: 101 f., 86 m., 29 unknown vs. NHE: 93 f., 89 m., 14 unknown), school, grade level, and classroom section. Students in both conditions had the same **prior Skill Builder completion rate** (HE:  $M=0.91$ ,  $Mdn=1.0$ ; NHE:  $M=0.91$ ,  $Mdn=1.0$ ,  $p=.463$ ), which was computed by dividing the sum of prior Skill Builders started by the number of prior Skill Builders completed (amongst all ASSISTments assignments experienced by students in the sample). Analysis using Mann-Whitney U tests (which are robust to skew) with a Benjamini-Hochberg false-discovery rate post-hoc correction for multiple tests ( $p<.05$ ) [7], yielded no significant differences between the two conditions on several measures including total number of problems solved, time per problem, and number of attempts.

### 3.2 Measures Considered

This study considered several measures pertaining to students' answers and hint patterns. As noted above, students only completed the Skill Builder when they correctly answered three consecutive questions using first attempts. However, students were able to attempt problems multiple times. Students wishing to advance to the next problem but unable to generate the correct answer were able to request a bottom-out hint. When hints were available, students had to view between 1 and 3 regular hints before they were able to obtain the bottom-out hint, which provided the correct answer. In the first three problems of NHE condition, students could select *Show Answer*, which displayed only the bottom-out hint, but no additional assistance.

Several measures based on these behavioral patterns were considered, including: **number of problems solved (PS)**, **mean answer-attempts per problem (MAA)**, **total answer attempts (TAA)**, **total hint requests (THR)** and **mean hint requests per problem (MHR)**. Spanning conditions, participants required 9.12 problems on average ( $Mdn=9.0$ ,  $SD=3.32$ ) to complete the assignment. Spanning conditions and problems, students averaged 16.14 total answer attempts ( $Mdn=14.0$ ,  $SD=10.72$ ), or 1.72 answer attempts ( $Mdn=1.71$ ,  $SD=0.78$ ) per problem. On average, students requested approximately one hint per nine problems ( $Mdn=0.0$ ,  $SD=2.23$ ) throughout the Skill Builder. There were no significant differences in the aforementioned measures by condition according to Mann-Whitney U tests conducted with false discovery rate post-hoc corrections.

Next, we assessed several time-based measures to determine how hints were affecting students' completion rates. Basic measures including the number of **days** and **weeks** it took for a student to finish the Skill Builder were considered. These measures were analyzed both by completion time and by week of completion. As the data was heavily skewed (most students finished in week 1), a Mann-Whitney U test was used to analyze completion time. Six months after beginning the study, when data was harvested, seventy-two students (18%) had not completed the Skill Builder. Students who completed the Skill Builder were grouped according to whether it had taken them 1, 2, 3, or 4 or more weeks to complete, while those who never finished the Skill Builder were labeled as *incomplete*. We also considered, **Completion time**

(CT, in seconds), or the total time it took students to complete the assignment, which was calculated by subtracting the start time of the first problem from the end time of the last problem solved.

Because the time students spent solving a problem was skewed, with a median of 1.1 minutes ( $M=16.22$  hr,  $SD=4.69$  days,  $Min=2$  sec,  $Max=74.96$  days), this value was *winsorized* to 15 minutes (900 sec) in order to exclude irrelevant conditions (e.g., disconnection from the network, shifts between learning activities, off-task behavior). The fifteen-minute time frame accounted for 93% of the data.

The winsorized measures were used to calculate **time-on-problem (TOP, in seconds)** for each problem in the Skill Builder that the student attempted to solve (i.e., end time minus start time for each problem). This measure was subsequently used to generate several others, including **mean time-per-problem (MTPP)**, which showed a mean of 2.62 min ( $Mdn=2.35$  min,  $SD=1.78$  min) across all students. For each student, TOP was also **totalled** across all attempted problems (**TOP-total**), resulting in a mean of 23.42 minutes ( $Mdn=20.72$  min,  $SD=16.93$  min) across all students. Finally, **total time-between-problems (TTBP)**, was calculated by subtracting TOP-total from each students' completion time. Readers should note that because students were allowed to return to this assignment over the course of the summer, these values were comparatively large ( $M=6.73$  days,  $Mdn=43$  sec,  $SD=14.49$  days). However, as Table 1 shows, variation among students who took more than one week was minimal at the problem level.

**Table 1. Mean values of time-based measures according to completion-time categories (weeks).**

Week	PS	TOP-total	MTPP	TTBP	CT
1	9.15	20.2 m	2.2 m	0.48 d	0.49 d
2	10.04	35.9 m	3.7 m	10.1 d	10.1 d
3	9.00	38.2 m	4.4 m	18.5 d	18.5 d
≥ 4	11.81	38.6 m	3.3 m	40.8 d	40.8 d
Incomplete	5.55	16.9 m	3.6 m	5.4 d	N/A

*Note.* PS – problems solved; TOP-total – total time on problem; MTPP – mean time per problem; TTBP – total time between problems; CT – completion time, m = minutes, d = days

## 4. RESULTS

ASSISTments automatically logged data in analyzable form. The following subsections present the results on hint usage, problem attempts, skill builder completion, and time-on-problem.

### 4.1 Hint Usage and Problem Attempts

This study used four primary measures of student actions, including total answer attempts, mean answer attempts, total hint requests, and mean hint requests per problem. Because the two conditions in this study only applied to the first three problems (after which, students in the no-early-hints condition also had access to regular hints), we report on values for the first three problems and those that follow separately.

Table 2 presents significant differences both between and within-conditions. There were no significant differences between conditions with respect to the number of attempts per problem or the total number of attempts used in solving the first three problems of the Skill Builder. That is, the availability of hints in the first three problems did not effect the number of attempts used or the number of hints requested over the course of the experiment. Likewise, the significant differences observed within condition all trended in the same direction, suggesting little to no effect.

**Table 2. Significant differences in answer attempts and hint requests by condition and within condition ( $p < .05$ ).**

Measure	HE vs. NHE		1st 3 vs. Other problems	
	1st 3	Others	HE	NHE
TAA	NS	NS	Others > 1st3	Others > 1st3
MAA	NS	NS	NS	Others > 1st3
THR	N/A	NS	1st3 > Others	N/A
MHR	N/A	NS	1st3 > Others	N/A

Note. TAA – total answer attempts; MAA – mean answer attempts; THR – total hints requests; MHR – mean hint requests; HE – hints-early; NHE – no-hints-early; NS – not significant

## 4.2 Hint Usage and Skill Builder Completion

One of the most important measures in this study was whether or not students were eventually able to demonstrate skill mastery by consecutively answering three of the Skill Builder questions accurately. Chi Squared tests revealed no significant difference between conditions in the proportion of students who did not complete the Skill Builder ( $\chi^2(1, N=412)=0.714, p=.398$ ).

Non-completion in both conditions was associated with lower prior Skill Builder completion rates, suggesting that students' inability to master this Skill Builder was indicative of larger issues in completing their mathematics assignments (HE:  $U=1115.5, p < .001$ , NHE:  $U=471, p < .001$ ). Non-completion was also associated with higher numbers of hint requests and answer attempts, both of which occurred across significantly fewer problems than worked by students who were able to complete the Skill Builder. Finally, non-completion was associated with significantly longer time worked across problems (**TOP-total**).

Despite nearly identical Skill Builder completion rates, the two conditions differed significantly in the time it took students to complete the problem set (HE:  $M=208.23$  hrs,  $Mdn=38.55$  min, NHE:  $M=67.52$  hrs,  $Mdn=20.9$  min,  $U=16835, p=.008$ ). Specifically, as shown in Table 3, students in the no-hints-early condition completed the Skill Builder faster than those in the hints-early condition. These results were complemented by Chi Squared results that analyzed the distribution of students completing the assignment over several weeks,  $\chi^2(4, N=411)=8.981, p=.062$ . Again, this might seem obvious, as students who access hints tend to take longer to digest problem and feedback content, but further analysis suggests other factors should also be considered.

**Table 4. Time-on-problem comparison by condition (in minutes)**

Condition	Mean (SD)							Median				
	Regular Hints Requested						Bottom-out Hint	Regular Hints Requested			Bottom-out Hint	
	N	0 Hints	N	1 Hint	N	2 Hints		0 Hints	1 Hint	2 Hints		
<b>First 3 Problems</b>	373	1.78 (1.15)	103	3.62 (1.33)	0	N/A	167	2.98 (1.45)	1.48	3.85	N/A	2.95
HE	191	1.65* (1.17)	103	3.62 (1.33)	0	N/A	81	3.47* (1.33)	1.37*	3.85	N/A	3.43*
NHE	182	1.92* (1.13)	0	N/A	0	N/A	86	2.55* (1.43)	1.80*	N/A	N/A	2.53*
<b>Other Problems</b>	366	1.52 (0.92)	22	2.52 (1.93)	59	3.27 (1.23)	56	3.27 (1.23)	1.33	1.78	3.02	2.98
HE	190	1.50 (0.87)	13	2.02 (1.92)	30	3.20 (1.33)	29	3.23 (1.33)	1.33	1.65	2.92	2.93
NHE	176	1.53 (0.98)	9	3.25 (1.82)	29	3.37 (1.15)	27	3.28 (1.15)	1.42	3.65	3.47	3.02
<b>All Problems</b>	377	1.58 (0.78)	113	3.50 (1.37)	58	3.32 (1.17)	174	3.05 (1.35)	1.53	3.65	3.22	3.03
HE	195	1.50 (0.73)	104	3.53 (1.33)	29	3.28 (1.20)	87	3.45* (1.27)	1.52	3.63	2.93	3.40*
NHE	182	1.67 (0.83)	9	3.25 (1.82)	29	3.37 (1.15)	87	2.65* (1.33)	1.57	3.65	3.47	2.70

Note. Units are in minutes. \* $p < .05$ . N – number of students; HE – hints-early; NHE – no-hints-early.

**Table 3. Number of students per condition who completed the Skill Builder each week**

Weeks	HE (N=215)	NHE (N=196)
1	125 (58%)	137 (70%)
2	15 (7%)	13 (7%)
3	5 (2%)	3 (1%)
≥ 4	30 (14%)	13 (7%)
Incomplete	40 (19%)	31 (16%)

Note. HE – hints-early; NHE – no-hints-early

## 4.3 Hint Usage and Time-on-Problem

Hint availability could effect time-on-problem (**TOP**) in more than one way, even when students use hints effectively. Students who need hints may be expected to answer more slowly than their peers, but powerful hints may actually reduce the time that a struggling student takes to complete a problem (compared to a situation in which the same student did not have access to hints).

Table 4 (calculated with the Benjamini-Hochberg correction) shows a complex interaction between time-per-problem and hint use, but overall there were few differences between conditions. On the whole, the use of (regular) hints lead to longer time on problem (**TOP**) measures, but the effect of bottom-out hints differed by condition. In both conditions, students who used bottom out hints took longer to complete problems than those who did not use them. However, those who used bottom-out hints in the HE condition took less time per problem than those who only requested one (regular) hint. The latter pattern could be indicative of *gaming* behavior, and this warrants further investigation, but it is also possible that students who quickly realized their mistakes clicked through to the bottom-out hint in order to start work on the next problem.

Results further indicated that differences were driven by hint use effects in the first three problems, where students who did not have access to hints (the NHE condition) were significantly slower at answering than those who did (HE) ( $M=1.92$  min,  $Mdn=1.80$  min vs  $M=1.65$  min,  $Mdn=1.37$  min). This was a predictable difference, as struggling students in the HE condition could ask for hints, thereby removing themselves from this calculation, while struggling students in the NHE condition could only remove themselves from this calculation by requesting a bottom-out hint.

Significant differences within and between conditions (summarized in Table 5) showed trends that suggested that behavior in the first three problems was driving the differences between the two conditions, where hint-access was restricted to the students in the HE condition. Interestingly, in the first three problems the mean time per problem was statistically similar. That is, for the first three problems, the HE and NHE condition did not differ overall, which suggests the need for understanding individual differences, such as those highlighted in Table 4. The significant differences between conditions emerged primarily in total time between problems (**TTBP**) and in the total completion time (**CT**), with students in the hints-early condition showing larger values for both measures.

**Table 5. Time Measures per Condition ( $p < .05$ ).**

	HE vs. NHE		1st 3 vs. Other problems	
	1st 3	Others	HE	NHE
MTPP	NS	NS	1st3 > Others	NS
TTBP	HE > NHE	NS	NS	Others > 1st3
CT	HE > NHE	NS	NS	Others > 1st3

*Note.* MTPP – mean time-per-problem; TTBP – total time between problems; CT – completion time; HE – hints-early; NHE – no-hints-early; NS – not significant

Further analyses revealed complementary patterns in within-condition differences. Students in the hints-early condition had significantly higher mean time-per-problem (MTPP) on the first three problems than they did on later problems ( $M=3.67$  min,  $Mdn=2.63$  min vs.  $M=2.17$  min,  $Mdn=1.98$  min,  $U=13281$ ,  $p < .001$ ), suggesting that those who effectively used these hints in the first three problems were learning the material well enough to complete later problems more efficiently. There were no significant differences in this group for other time-based measures (**TTBP** or **CT**). In contrast, students in the no-hints-early condition showed no significant differences for **MTPP**, but had longer **TTBP** and **CT** patterns for later problems than for the first three problems.

## 5. DISCUSSION

The present experiment was designed to explore the effects of ASSISTments' on-demand hints system. For ethical reasons, we limited differences between the control condition (providing hints) and the experimental condition (withholding hints) to the first three problems. All students had access to hints following the third problem to retain overall learning. However, effects could be seen even after students had moved past these first three problems.

The data used in the study was collected from one of many Skill Builders assigned to students for summer work. We explored the data using several different measures, extracting information about the number of attempts each student made, the number of hints (regular or bottom-out) they requested, and the length of time needed to complete the assignment.

Some findings were quite predictable, as reading hints would take more time than simply answering problems, assuming students were assigned problems that matched their current ability. However, other findings were more surprising. Even though students made the same number of attempts per problem and per assignment, those in the HE condition took significantly longer to complete the Skill Builder.

Students in the HE condition also spent relatively more time between problems compared to those in the no-hints-early condition, but only during the first three problems, where conditions were truly distinct. One interpretation of this finding is

that students in the HE condition were taking more time between problems to process the new material they were learning. An alternative explanation is that students were procrastinating—deliberately putting off working on the Skill Builder out of difficulty or apathy (as summer work is highly self-regulated). These students could have been seeking out an easier Skill Builder to work on or may have spent their time doing something completely unrelated. Still, this latter interpretation may not be detrimental if students were using the time to work on other assignments. As Baker and colleagues have suggested [5], a student that goes off task and is able to re-engage afterwards may be more productive in the long run than those who persist at all costs.

## 6. CONCLUSION

This work presented an investigation of how students completing summer work responded to having or not having hints available on the first three problems of a Skill Builder assignment within the ASSISTments online learning system. When hints were available from the start, students were more likely to delay work in comparison to students for whom step-wise hints were only available after the third problem. When hints were always available, participants took significantly more time to complete the Skill Builder. We interpreted the difference in completion times as an opportunity to re-engage towards more productive math learning. In future work, we plan to conduct a similar study during the school year to examine how results differ in a more controlled and less self-regulated learning environment.

## 7. ACKNOWLEDGEMENTS

We gratefully acknowledge funding from the NSF (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428 & DRL-1031398), the US Dept. of Ed (IES R305A120125 & R305C100024 and GAANN), and the ONR.

## 8. REFERENCES

- [1] Aleven, V., McLaren, B., Roll, I., & Koedinger, K. 2006. Toward Meta-Cognitive Tutoring: A Model of Help-Seeking with a Cognitive Tutor. *Int J Artif Int in Ed*, 16, 101-130.
- [2] Aleven, V., Stahl, E., Schworm, S., Fischer, F., Wallace, R. 2003. Help seeking and help design in interactive learning environments. *Rev Educ Res*, 73(3), 277-320.
- [3] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. 2004. Off-Task Behavior in the Cognitive Tutor Classroom: When Students Game The System. *Proc ACM CHI*, 383-390.
- [4] Baker, R.S.J.d., de Carvalho, A.M.J.A., Raspat, J., Aleven, V., Corbett, A.T., Koedinger, K.R. 2009. Educational Software Features that Encourage and Discourage "Gaming the System". *Proc 14<sup>th</sup> Int Conf Artif Int in Ed*, 475-482.
- [5] Baker, R.S.J.d., Moore, G., Wagner, A., Kalka, J., Karabinos, M., Ashe, C., Yaron, D. 2011. The Dynamics Between Student Affect and Behavior Occuring Outside of Educational Software. *Proceedings of the 4th bi-annual International Conference on Affective Computing and Intelligent Interaction*.
- [6] Beck, J., & Gong, Y. 2013. Wheel-spinning: Students who fail to master a skill. *Arti Int in Ed*. Berlin: Springer.
- [7] 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.

- [8] Broderick, Z., O'Connor, C., Mulcahy, C., Heffernan, N. & Heffernan, C. 2011. Increasing Parent Engagement in Student Learning Using an Intelligent Tutoring System. *J Int Learn Res*, 22(4):523-550.
- [9] Campuzano, L., Dynarski, M., Agodini, R., & Rall, K. 2009. Effectiveness of reading and mathematics software products: Findings from two student cohorts. Washington, DC: U.S. Dept Ed, Inst Ed Sci.
- [10] Corbett, A. T., & Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model User-Adap Intra*, 4(4), 253-278.
- [11] Heffernan, N., & Heffernan, C. 2014. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning & Teaching. *Int J AIED* 24(4),470-97.
- [12] Heiner, C., Beck, J., & Mostow, J. 2004. Improving the help selection policy in a Reading Tutor that listens. *InSTILL/ICALL Symposium*.
- [13] Inventado, P.S. & Scupelli, P. 2015. Data-Driven Design Pattern Production: A Case Study on the ASSISTments Online Learning System. *Proc 20<sup>th</sup> Euro Conf Pattern Languages of Programs*.
- [14] Inventado, P.S., Scupelli, P., Van Inwegen, E.G., Ostrow, K.S., Heffernan, N., Baker, R.S., Slater, S., & Ocumpaugh, J. 2015. Materials for *Hint Availability Slows Completion Times in Summer Work*. Retrieved from <https://goo.gl/xyli5h>
- [15] Koedinger, K.R., & Aleven, V. 2007. Exploring the assistance dilemma in experiments with Cognitive Tutors. *Educ Psychol Rev* 19.3: 239-264.
- [16] Li, S., Xiong, X., & Beck, J. 2013. Modeling student retention in an environment with delayed testing. *Int Educ Data Mining Society*, 328-329
- [17] Natl Gov Assoc Ctr Best Practices, Council of Chief State School Officers. 2010. Common Core State Stds. Washington D.C.
- [18] Nelson-Le Gall, S. 1987. Necessary and unnecessary help-seeking in children. *J Genetic Psychol*, 148, 53-62.
- [19] Ocumpaugh, J., Baker, R.S, Rodrigo, M.M.T. 2015. *Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) 2.0 Technical & Training Manual*. NY, NY: Teachers College, Columbia U. Manila, Philippines: Ateneo Laboratory for the Learn Sciences.
- [20] Pane, J.F., McCaffrey, D.F., Slaughter, M.E., Steele, J.L., & Ikemoto, G.S. 2010. An experiment to evaluate the efficacy of Cognitive Tutor geometry. *J Res Educ Eff*, 3(3), 254-281.
- [21] Pardos, Z., Baker, R.S., San Pedro, M.O.C.Z., Gowda, S., Gowda, S. 2014. Affective states & state tests: Investigating how affect and engagement during the school year predict end of year learning outcomes. *J Learn Analytc*, 1(1), 107-28.
- [22] Razzaq, L.M., & Heffernan, N.T. 2009. To Tutor or Not to Tutor: That is the Question. *AIED*, 457-464.
- [23] San Pedro, M.O., Baker, R., Heffernan, N., Ocumpaugh, J. 2015. Exploring College Major Choice and Middle School Student Behavior, Affect and Learning: What Happens to Students Who Game the System? *Proc 5<sup>th</sup> Int Learn Analytc Know*, 36-40.
- [24] Schofield, J. W. 1995. *Computers and Classroom Culture*. Cambridge University Press.
- [25] VanLehn, K. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ Psychol*, 46(4), 197-221.
- [26] Whorton, S. 2013. Can a computer adaptive assessment system determine, better than traditional methods, whether students know mathematics skills? MA thesis, Computer Science Department, Worcester Polytechnic Institute.
- [27] Wijekumar, K., Meyer, B., & Lei, P. 2012. Large-scale RCT with 4<sup>th</sup> graders using intelligent tutoring of the structure strategy to improve nonfiction reading comprehension. *Educ Tech Res Dev*, 60(6), 987-1013.

# On Competition for Undergraduate Co-op Placements: A Graph Mining Approach

Yuheng Jiang and Lukasz Golab  
University of Waterloo, Canada  
{y29jiang,lgolab}@uwaterloo.ca

## ABSTRACT

We propose a graph mining methodology to analyze the relationships among academic programs from the point of view of co-operative education. The input consists of student - job interview pairs, with each student labelled with his or her academic program. From this input, we build a weighted directed graph, which we refer to as a program graph, in which vertices correspond to academic programs and edge weights denote the percentage of jobs that interviewed at least one student from both programs. We show that various properties of this graph have natural interpretations in terms of the relationships among academic programs and competition for co-op jobs. We also present a case study that illustrates the utility of the proposed methodology.

## 1. INTRODUCTION

According to the World Association for Cooperative and Work-integrated Education, 275 institutions from 37 countries have implemented cooperative education (co-op) programs [17]. Co-op experiences are vital because they supplement students' classroom skills and help them to gain practical experience.

We propose a graph mining methodology to analyze the relationships and competition among academic programs in the context of co-op. Our motivation is threefold. First, with academic institutions introducing new programs in recent years [6, 15], it is often unclear how one program differs from another. As a result, employers may not know which programs to advertise their jobs to and students may not realize that they qualify for a job targeted to a related program (e.g., Computer Science vs. Software Engineering). Understanding similarities among programs can lead to more effective job and academic classification schemes and therefore can help match job opportunities with qualified students. This analysis can also help students choose programs of study that correspond to their desired careers. Second, data from the co-op system may be used to identify multi-disciplinary programs that enable their students to obtain various types of jobs. This issue is becoming increasingly important given the recent rise in popularity of multi-disciplinary and well-rounded education [1, 2, 5, 10, 16]. Third, analyzing co-op job data can reveal jobs that are exclusive to par-

ticular departments, and, conversely, departments whose students compete for jobs with students from other departments. The university can choose to attract more employers that offer jobs to programs facing strong competition. Thus, the problems we study in this paper are critical to co-operative education from the student's, employer's and institution's perspective.

While some of these questions have been raised in prior work (details in Section 2), we propose a data-driven technique for answering them. Our input consists of student - job interview pairs, with each student labelled with his or her academic program. We transform this input to a graph, which we refer to as a *program graph*, in which vertices correspond to academic programs and edge weights denote the percentage of jobs that interviewed at least one student from both programs. Thus, the larger the edge weight, the stronger the relationship and competition between two programs.

Within the program graph, we are interested in vertices forming clusters or communities, vertices that are connected to many such clusters, and vertices that are strongly connected to their neighbours. As we will show, these graph properties have natural interpretations in the context of co-op. Graph clustering and community detection determine groups of related programs whose students interview for the same types of jobs; programs with connections to multiple clusters are likely to be multi-disciplinary; and programs with strong connections to their immediate neighbours face strong competition for jobs.

## 2. RELATED WORK

The majority of related work qualitatively or statistically analyzed co-op education through survey data with fewer than 100 entries. To the best of our knowledge, the first research work that used a large-scale data-driven methodology was our previous work [9]. We analyzed satisfaction with the co-op process using three years of evaluation data (students' evaluations of their employers and employers' evaluations of students). We found that students received better evaluations in their senior years, but they rated their first employer the highest. We also found that senior students outperformed junior students in work placements abroad, and extended work terms at the same employer (spanning more than one academic term) did not increase student satisfaction. In this paper, we target a different problem of understanding the relationships among academic programs.

In the context of academic programs, Wilson and other researchers urged traditional academic disciplines to be updated to better reflect reality [6, 15]. Furthermore, Hesketh found that employers have trouble advertising to specific programs and instead they ad-

vertise based on desired skillsets [8]. As we will show, clusters in the program graph indicate similar programs and suggest related programs that employers can advertise their jobs to. Additionally, it was suggested that programs can be evaluated based on their students' ability to obtain jobs [7, 14], which is a question that can be answered with the help of our methodology. Also, while the importance of multi-disciplinary education has been widely recognized [1, 2, 5, 10, 16], we propose a data-driven methodology for analyzing whether students from a particular academic program qualify for different types of jobs.

### 3. METHODOLOGY

We are given a dataset corresponding to student - job interview pairs, with each student labeled with his or her academic program and each interview associated with a job ID. We propose a methodology that relies on transforming the student-job interview pairs to an edge-weighted directed graph  $G = (V, E)$ , with a set of vertices  $V$  and a set of edges  $E$ . Vertices correspond to academic programs and edges represent relationships among programs. Let  $e_{ij}$  be the weight of the edge  $E_{ij}$  from vertex  $v_i$  to  $v_j$ , and let  $J_i$  be the list of distinct jobs that interviewed students from program  $v_i$ . We define  $e_{ij}$  as the fraction of jobs that interviewed at least one student from both programs; i.e., the fraction of jobs in  $J_i$  that also appear in  $J_j$ :

$$e_{ij} = \frac{|J_i \cap J_j|}{|J_i|} \quad (1)$$

This can also be interpreted as a conditional probability that a job interviewed at least one student from program  $v_j$  given that it interviewed at least one student from program  $v_i$ .

The direction of edges is important. For a program node  $v_i$ , an incoming edge weight from  $v_j$  measures the fraction of jobs in  $J_j$  that also interviewed at least one student from  $v_i$ . Thus, a large incoming edge weight of  $v_i$  from  $v_j$  means that most jobs interviewing at least one student from  $v_j$  also interviewed at least one student from  $v_i$ . Conversely, a large outgoing edge weight from  $v_i$  to  $v_j$  means that most jobs interviewing at least one student from  $v_i$  also interviewed at least one student from the other program.

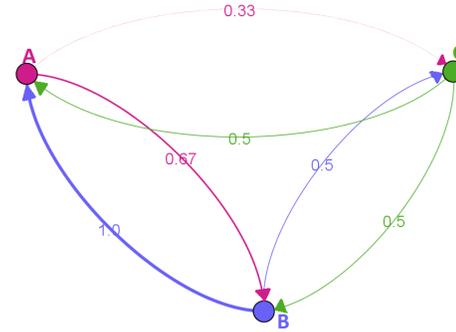
We give an example in Table 1, which corresponds to 4 jobs, 9 interviews and 8 students from three programs (A, B and C). The job lists for each program are:  $J_A = \{1, 2, 3\}$ ,  $J_B = \{1, 2\}$ , and  $J_C = \{2, 4\}$ . The corresponding program graph is shown in Figure 1, and the edges are colour-coded by the source vertex. The edge weight from Program A to Program B is  $|\{1, 2\}|/|\{1, 2, 3\}| = 2/3 = 0.67$ , meaning that 67 percent of jobs that interviewed at least one student from Program A also interviewed at least one student from Program B. The edge weight from Program B to Program A is  $|\{1, 2\}|/|\{1, 2\}| = 2/2 = 1$ , meaning that every job which interviewed a student from program B also interviewed a student from program A. Thus, the larger the edge weight, the stronger the relationship and competition between two programs.

Our definition of edge weights assumes that a relationship between two programs exists if at least one student from both programs *interviewed* for the same job; if there are many such jobs, then the edge weight will be larger.

Having explained how the program graph is constructed, we now clarify how properties of the program graph are related to the types and extent of relationships among academic programs in the context of co-op jobs:

**Table 1: Sample interview data**

Student ID	Program Name	Job ID
1	A	1
2	C	2
3	B	1
3	B	2
4	B	1
5	A	2
6	A	3
7	C	2
8	C	4



**Figure 1: An example of a program graph**

- **Clusters:** Clusters in a graph represent closely connected vertices. In our context, clusters represent related programs whose students interview for (mostly) the same jobs.
- **Outliers:** Given a graph clustering, we define outliers as vertices that have strong connections to other vertices from multiple clusters (as opposed to “normal” vertices connected mostly to other vertices within the same cluster). In our analysis, outliers correspond to multi-disciplinary programs: students from those programs have interviews in common with students from several different program clusters.
- **Fan-out:** (Weighted) fan-out measures the (weighted) number of outgoing edges of a vertex. In our context, weighted fan-out corresponds to the competition that a program faces from other programs. High weighted fan-out means that most jobs interviewing at least one student from the given program also interviewed students from other programs. As we will explain shortly, we use a modified version of standard weighted fan-out that takes into account the fact that our edge weights are defined in terms of set intersections (of the job sets of different programs).

In the remainder of this section, we describe the graph algorithms that may be used to identify program clusters, multi-disciplinary programs and programs facing strong competition.

#### 3.1 Finding Clusters of Similar Programs

We use two techniques to find clusters of similar programs: near-clique finding and community detection.

The density of a graph (or subgraph) is the number of edges divided by the maximum possible number of edges, i.e.,  $\frac{|E|}{|V|*(|V|-1)}$ . A clique is a group of vertices that are fully connected and therefore have a density of one. A near-clique is a group of vertices where

the subgraph consisting of them and their edges has a density of nearly one, i.e., a group of vertices that is nearly fully connected. However, since our program graph is weighted and directed, we want to find near-cliques with large edge weights. To do this, we first remove all edges from the program graph except the five percent with the largest edge weights. The resulting graph may leave some vertices disconnected, while other pairs of vertices may only have an incoming or an outgoing edge. Then, we remove edge directions and simply retain an edge between two programs if there is either an incoming or an outgoing edge. Finally, we return all near-cliques from the resulting graph with density of at least 0.8.

In addition to identifying densely connected subgraphs via near-clique finding, we use the Louvain Modularity algorithm [4] to partition the vertices into disjoint clusters (communities), such that vertices with the same cluster are densely connected and vertices in different clusters are sparsely connected. This algorithm is included in many graph mining tools such as Gephi [3] and aims to maximize *modularity*, which compares the sum of the weights of intra-cluster edges resulting from given clustering with that of a randomly connected graph with the same number of edges [13].

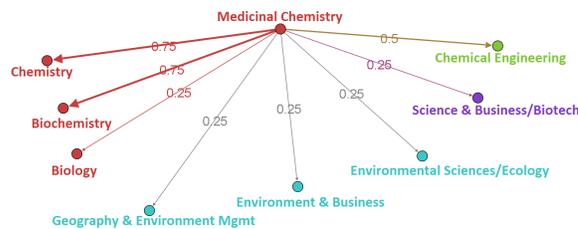
Newman [12] introduced modularity for weighted undirected graphs. We translate this metric to weighted directed graphs as follows. Let  $c_i$  be the community that a vertex  $v_i$  belongs to, and  $m = \sum_{ij} e_{ij}$ , i.e., the sum of all the edge weights in the graph. The fraction of the edge weights that are intra-cluster is  $\frac{1}{m} \sum_j e_{ij} \delta(c_i, c_j)$ , where  $\delta(c_i, c_j)$  is equal to 1 if  $c_i = c_j$  (i.e. vertices  $v_i$  and  $v_j$  belong to the same cluster) and 0 otherwise.

Let  $k_i = \sum_j e_{ij}$  (i.e., the sum of the weights of the edges that connect to vertex  $v_i$ ). Consider another graph in which the fan-outs of all the vertices are the same but the edges are randomly connected. In such a graph, the probability of an edge existing between vertices  $v_i$  and  $v_j$  is  $\frac{k_i k_j}{2m}$ . The modularity of a graph clustering is defined as:

$$Q = \frac{1}{m} \sum_{i,j} (e_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \quad (2)$$

$Q = 0$  means that the community detection result is no better than random. The maximum value for  $Q$  is 1. Higher modularity indicates more effective partitioning with more intra-cluster edges and fewer inter-cluster edges.

The Louvain Modularity method is iterative and includes two phases. In the first phase, each vertex starts in a different community. Then, for each vertex  $v_i$ , we compute the gain in modularity if  $v_i$  is moved to the community that its neighbour ( $v_j$ ) belongs to. If the gain is positive, the change happens; otherwise  $v_i$  remains in its original community. This process is repeated iteratively and sequentially until no further improvements can be made. The outcome of the first phase is only a local optimum of modularity since the order of processing of the vertices will affect the result. In the second phase, a new graph is created such that the vertices are the communities obtained in the first phase, and edge weights are the sums of edge weights between vertices in the two communities. We reapply the process in the first phase on this new graph. The algorithm stops when maximum modularity is reached. To account for the effect of order, we run this algorithm multiple times and keep the result with the highest modularity.



**Figure 2: Direct competitors of Medicinal Chemistry, colour-coded by clusters**

One characteristic of this algorithm is that it avoids creating small clusters. Lambiotte et al. [11] add a *resolution parameter*  $t$  to control the number of clusters. The new modularity definition is shown in Equation 3. The default  $t$  value is 1; smaller values of  $t$  lead to more and smaller communities.

$$Q_{new}(t) = (1 - t) + \frac{1}{m} \sum_{i,j} (e_{ij} t - \frac{k_i k_j}{m}) \delta(c_i, c_j) \quad (3)$$

### 3.2 Finding Multi-Disciplinary Programs

To find multi-disciplinary programs, we start with the clusters/communities obtained by the Louvain Modularity algorithm. Intuitively, if an academic program has strong connections to other programs from multiple clusters (each of which corresponds to different types of jobs), it may be multi-disciplinary.

For each program, we propose a multi-disciplinary score as follows. For each cluster  $c_i$  identified by the Louvain Modularity algorithm, let  $p_i$  be the fraction of the total weight of the outgoing edges from the given program to the programs only in  $c_i$ . Then, for a given program, we compute the entropy of the distribution of edge weights among different communities simply as  $\sum_i -p_i \log_2 p_i$ . High entropy means that the given program has strong links to programs in multiple clusters and therefore may be multi-disciplinary.

We illustrate this concept with an example. Suppose that students in the Medicinal Chemistry program had interviews in common with students from eight other programs belonging to four clusters, labeled red, blue, purple, and green, as shown in Figure 2, with vertices colour-coded by their clusters. Only the outgoing edges from Medicinal Chemistry are relevant since they represent the percentage of jobs from  $J_{MedicinalChemistry}$  that also interviews students from its neighbour programs. The sum of all out-going edge weights of Medicinal Chemistry is 3.25.  $p_{red} = (\sum_{i \in red \text{ cluster}} e_{MedicinalChemistry,i}) / 3.25 = (0.75 + 0.75 + 0.25) / 3.25 = 0.54$ , which is the sum of weights of edges from Medicinal Chemistry to the programs in the red cluster. Similarly,  $p_{blue} = 0.23$ ,  $p_{green} = 0.15$ , and  $p_{purple} = 0.08$ . Thus, the multidisciplinary score of Medicinal Chemistry is  $-p_{red} \log_2 p_{red} - p_{blue} \log_2 p_{blue} - p_{purple} \log_2 p_{purple} - p_{green} \log_2 p_{green} = 1.67$ .

### 3.3 Finding Programs Facing Competition

We define the extent of competition that a program faces using a “set fan-out” metric. We want to compute the fraction of jobs that interviewed students from the given program which also interviewed at least one student from another program. For a given vertex (program)  $v_i$ , we define:

$$\text{Set Fan Out}_i = \frac{|\cup_{j \neq i} (J_i \cap J_j)|}{|J_i|} \quad (4)$$

A set fan-out of zero means that all the jobs that interviewed at least one student from program  $v_i$  only interviewed students from  $v_i$  and no other program. Students from such a program may have specialized skills that students from other programs do not have. A set fan-out of one means that every job that interviewed at least one student from program  $v_i$  also interviewed at least one student from another program. In other words, there were no jobs that exclusively interviewed students from  $v_i$  and therefore students from  $v_i$  may be facing strong competition for jobs.

Returning to Table 1,  $J_A = \{1, 2, 3\}$ ,  $J_B = \{1, 2\}$ , and  $J_C = \{2, 4\}$ . For Program A, its set fan-out is  $\frac{|(J_A \cap J_B) \cup (J_A \cap J_C)|}{|J_A|} = \frac{|1, 2\}}{|1, 2, 3\}} = \frac{2}{3} = 0.67$ . It means that students from Program A competed with students from other programs in 67 percent of their jobs. 33 percent of jobs that interviewed students from Program A did not interview students from other programs. The set fan-out for Program B is 1 and for Program C it is 0.5.

## 4. CASE STUDY

We now describe a case study that illustrates the utility of the proposed methodology. To carry out the analysis, we used the Gephi toolkit [3] which includes the Louvain Modularity algorithm. We used data from a large Canadian university including all interviews taking place in summer 2014, for co-op jobs taking place in Fall 2014. For each student - interview pair, the dataset includes the student's academic program and year, and job information such as the company name, job title, and targeted programs and academic years. The dataset consists of 4,194 students from 93 academic programs, 2,890 jobs and 16,855 interviews. On average, each job interviewed 5.8 students and each student had 4 interviews.

This academic institution has six faculties, each comprised of a number of academic programs: Science (programs include Physics and Earth Sciences), Mathematics (programs include Computer Science and Actuarial Science), Engineering (programs include Electrical, Mechanical, Civil, etc.), Arts (programs include Economics, Psychology and Sociology), Environment (programs include Planning and Geomatics) and Applied Health Science (AHS) (programs include Kinesiology and Recreation and Leisure Studies). All Engineering programs and several programs from other faculties (mainly Mathematics) have mandatory co-op education; other programs have optional co-op. As a result, most of the students and jobs in our dataset are from Engineering and Mathematics.

Rather than using all available data, we build the program graph using only the interviews of *senior* students (in their third and fourth academic years). Junior-level jobs tend to be less specialized, meaning that (junior) students from many different departments may qualify for an interview. In particular, we noticed that entry-level computer programming jobs interview students from many programs, including those outside computing. By focusing on senior students, we avoid generating edges in the program graph that correspond to junior-level jobs and may not truly indicate a relationship between programs. The resulting program graph contains 88 vertices (corresponding to programs that have at least two senior students in co-op) and 1,315 pairs of directed edges.

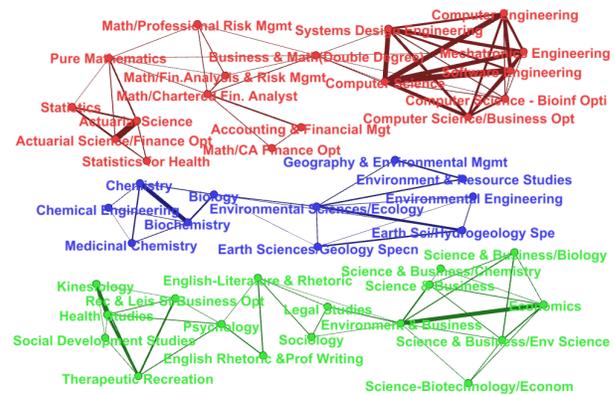


Figure 3: Vertices and edges participating in near-cliques

The program graph is a single connected component, i.e., there exists a path from every vertex to another. Its density is 0.34, meaning that one third of all possible program pairs had at least one interview in common. On average, the length of the shortest path between any two vertices is 1.7 and the diameter of the graph (i.e., the maximum length of any shortest path between two vertices) is three. The number of edges per vertex ranges from 4 to 66, with an average of 30.

## 4.1 Finding Clusters of Similar Programs

### 4.1.1 Near-Clique Finding

We begin by identifying near-cliques in the program graph (but considering only the five percent of edges with the largest weights, as described in Section 3). Figure 3 plots a subgraph of the program graph containing only the 46 vertices and 104 edges (in the top 5 percent of edge weights) that participate in the 25 near-cliques that we found. Three groups of programs appear to participate in the near-cliques, and we use a different colour for each. The larger the edge weight, the thicker the edge.

The red group at the top contains programs related to computing and maths. There is one near-clique with Software Engineering, Computer Engineering, Computer Science, Systems Design Engineering and Mechatronics Engineering. This suggests that Systems Design and Mechatronics students compete (interview) for software and programming jobs with students from core computing programs such as Computer Science. There are also two smaller near-cliques corresponding to Statistics/Actuarial Science and Accounting/Financial Analysis. Additionally, Pure Mathematics is connected to both of these; in fact Pure Mathematics students had interviews in common with students from 18 other programs. This suggests that Pure Mathematics students also interview for jobs in statistics, finance and business. Upon further inspection, we found that most such jobs were in financial trading.

The blue group of vertices in the middle includes two near-cliques: one with Chemistry-related programs and one with Earth Science and Environment-related programs. Based on these observations, the university may choose to either merge some of these related programs or redesign them to remove some of the overlap.

The green group at the bottom shows interesting connections. For instance, Economics seems strongly connected to Science & Business and Environment & Business, suggesting that these joint pro-



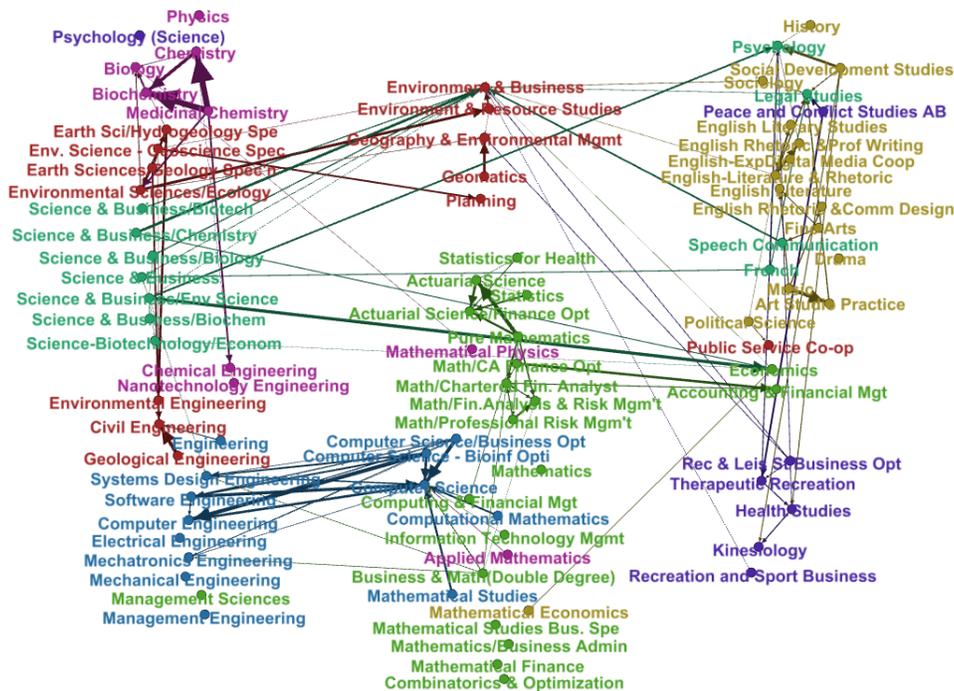


Figure 4: Clustering of the program graph into seven communities

We applied the proposed methodology on a large co-op data set from a major Canadian university. Our findings and their significance may be summarized as follows.

The clustering and community detection results (Section 4.1) correspond to job categories and academic specializations, which are not always evident from the University's academic structure. This suggests a job classification hierarchy to help advertise jobs to groups of related programs. Our results can also help students plan their academic and employment careers.

In Section 4.2, we identified multi-disciplinary programs which have strong connections to multiple clusters. These results can help students select programs that will give them broad skills and job qualifications, and can help institutions confirm that programs designed to be multi-disciplinary are producing students who qualify (i.e., are able to obtain interviews) for various types of jobs.

In Section 4.3, we identified programs where there were no jobs that only interviewed students from that particular program. That is, students from that program always competed for jobs with students from other programs. The university may wish to attract more employers that offer jobs to these under-represented programs.

## 6. REFERENCES

- [1] R. Barnett. Supercomplexity and the curriculum. *Studies in Higher Education*, 25(3):255–265, 2000.
- [2] R. Barnett. Learning for an unknown future. *Higher Education Research & Development*, 31(1):65–77, 2012.
- [3] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In Proc. of the International AAAI Conference on Weblogs and Social Media, 2009.
- [4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [5] M. Borrego and J. Bernhard. The emergence of engineering education research as an internationally connected field of inquiry. *Journal of Engineering Education*, 100(1):14–47, 2011.
- [6] E. El-Khawas. Higher education re-formed: Peter scott (ed.): Falmer press, London, 2000. *Higher Education Policy*, 14(1):93–95, 2001.
- [7] Z. Fadeeva, Y. Mochizuki, K. Brundiers, A. Wiek, and C. L. Redman. Real-world learning opportunities in sustainability: from classroom into the real world. *International Journal of Sustainability in Higher Education*, 11(4):308–324, 2010.
- [8] A. J. Hesketh. Recruiting an elite? employers' perceptions of graduate education and training. *Journal of Education and Work*, 13(3):245–271, 2000.
- [9] Y. Jiang, W. Y. S. Lee, and L. Golab. Analyzing student and employer satisfaction with cooperative education through multiple data sources. *Asia-Pacific Journal of Cooperative Education*, 16(4):225–240, 2015.
- [10] D. Kember, A. Ho, and C. Hong. The importance of establishing relevance in motivating student learning. *Active Learning in Higher Ed.*, 9(3):249–263, 2008.
- [11] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint 0812.1770*, 2008.
- [12] M. E. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.
- [13] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, Feb 2004.
- [14] A. Wiek, L. Withycombe, and C. L. Redman. Key competencies in sustainability: a reference framework for academic program development. *Sustainability Science*, 6(2):203–218, 2011.
- [15] A. Wilson. Strategy and management for university development. In *Higher Education Re-Formed*, Falmer Press, pp. 29–44, 2000.
- [16] A. Wilson. *Knowledge power: interdisciplinary education for a complex world*. Routledge, 2010.
- [17] World Association for Cooperative & Work-integrated Education (WACE). Accessed on 25 Feb 2016, at [www.waceinc.org/global\\_institutions.html](http://www.waceinc.org/global_institutions.html).

# Expediting Support for Social Learning with Behavior Modeling

Yohan Jo<sup>†</sup>, Gaurav Tomar<sup>†</sup>, Oliver Ferschke<sup>†</sup>, Carolyn P. Rosé<sup>†</sup>, Dragan Gašević<sup>‡</sup>

<sup>†</sup>School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

{yohanj, gtomar, ferschke, cprose}@cs.cmu.edu

<sup>‡</sup>Schools of Education and Informatics, The University of Edinburgh, Edinburgh, UK  
dgasevic@acm.org

## ABSTRACT

An important research problem for Educational Data Mining is to expedite the cycle of data leading to the analysis of student learning processes and the improvement of support for those processes. For this goal in the context of social interaction in learning, we propose a three-part pipeline that includes data infrastructure, learning process analysis with behavior modeling, and intervention for support. We also describe an application of the pipeline to data from a social learning platform to investigate appropriate goal-setting behavior as a qualification of role models. Students following appropriate goal setters persisted longer in the course, showed increased engagement in hands-on course activities, and were more likely to review previously covered materials as they continued through the course. To foster this beneficial social interaction among students, we propose a social recommender system and show potential for assisting students in interacting with qualified goal setters as role models. We discuss how this generalizable pipeline can be adapted for other support needs in online learning settings.

## 1. INTRODUCTION

More and more recent work in educational data mining and learning analytics refers to a “virtuous cycle” of data leading to insight on what students need and then improvements in support for learning [17]. An important goal is tightening this cycle to improve learning experience. We are interested especially in social learning, drawing from a Vygotskian theoretical frame where learning practices begin within a social space and become internalized through social interaction. This may involve limited interaction, such as observation, or more intensive interaction through feedback, help exchange, sharing of resources, and discussion.

There are two main contributions of this paper. The first is to propose a pipeline that can expedite the cycle of data infrastructure, learning process analysis, and intervention (Figure 1). Data infrastructure provides a uniform inter-

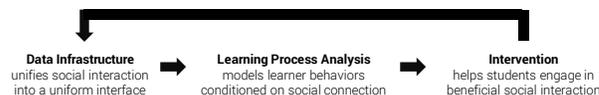


Figure 1: Pipeline for educational data mining in social learning.

face for heterogeneous data from social interaction in various platforms, such as connectivist Massive Open Online Courses (cMOOCs) [15], hobby communities, and Reddit communities, where people engage in follower-followee relations, post updates to their account, engage in threaded discussions, and also optionally link in blogs, YouTube videos, and other websites. Learning process analysis aims to analyze students’ processes depending on their social network configurations and to identify beneficial kinds of social connections. We developed a probabilistic graphical model that analyzes sequences of behaviors in terms of topics expressed and social media types that students actively engage in over time. Finally, intervention is introduced to foster beneficial social connections among students. We developed a recommender system that matches qualified students to discussions to increase opportunities for them to interact with other peers. The pipeline is iterative such that data from participation is used to create models that trigger interventions in subsequent runs of the course. Data from those later runs can be used to train new and better models in order to improve the interventions, and so on.

Our second contribution is to present findings from an application of the proposed pipeline to data from a social learning environment called ProSolo [12], in order to investigate the positive influence of observing goal-setting behavior. While goal-setting has been intensively researched and proven to be an important self-regulated learning (SRL) practice that often leads to success in learning, the influence of a student’s goal-setting behavior on observers has little been investigated empirically. If goal-setting students turn out to be good role models, that is, beneficial to their social peers, we can encourage and help students to make such social connections with goal setters to enhance their learning experience. The usefulness of this effect may be especially desirable in online courses where the number of instructors is limited, or online communities that are not structured like courses, where students are required to take more agency in forging a learning path for themselves within an ecology of resources.

In the remainder of this paper, we first motivate the specifics of our pipeline as situated within the literature. Next, we present our pipeline and its application, along with findings.

## 2. RELATED WORK

Vygotsky’s view of social interaction as a key to learning and Bandura’s social learning theory [1] emphasize the importance of interaction to learning. In social contexts, by vicarious learning, students observe external models and learn from those observations even when not actively engaged in interaction [19]. Observation of role models facilitates motivation and self-efficacy for a task [14] and may be associated with positive changes in the observer’s behavior [9]. Drawing on this theoretical foundation, the positive impact of social interaction has been investigated in collaborative work [8] and in online courses [11]. Yet, to our knowledge, our work is the first to investigate goal-setting behavior specifically as a qualification of a role model in online learning.

Several data infrastructures have been introduced to aid educational data mining for Massive Open Online Courses (MOOCs). For instance, MOOCdb [18] and DataStage<sup>1</sup>, designed to store raw data from MOOCs, consolidate click-stream data from different MOOC platforms in a single, standardized database schema. This allows for developing platform-independent analysis tools, thus enabling analyses that span multiple courses hosted by different MOOC providers with reduced development effort. While these infrastructures focus on behavior data represented by click-stream logs, our proposed infrastructure deeply represents other aspects of student interactions, such as discussion behavior and social relationships, which require the natural language exchange between students.

Analysis of students’ learning processes has been a critical topic in education. Our method contributes to the literature on time series behavior modeling. Approaches to learning process analysis differ in the definition of the basic building block, often conceived of as states within a graph. Common building blocks for tutoring systems and educational games include knowledge components [22] and actions [13]. In dialogue settings, it is common to code each utterance according to a coding scheme and analyze the sequence of codes [4]. In a MOOC context, states are often defined as course units [3], course materials [3], or discussions [2]. Such predefined states, however, may not be the ideal units of states, especially in online courses where students can selectively engage in learning resources. Therefore, unsupervised modeling approaches are appealing for the purpose of identifying states that are meaningful indications of student interests obtained in a data-driven way. Our model belongs to the class of Markov models, which have been proposed to learn latent states and state transitions [6, 21].

In MOOCs, a student’s learning process is affected by other peers especially through interaction in forums, which offer opportunities to develop communication and community. Hence, social recommendation algorithms can introduce appropriate students to certain discussions for productive interaction. Suggested matches should be appropriate when viewed either from the discussion or student side [16], for

<sup>1</sup><http://datastage.stanford.edu/>

example by suggesting a student to participate in discussions based on both the potential benefit of the student’s expertise as an asset to the discussions while respecting the limitations of a student’s resources for participation in more than a limited number of discussions [20]. Our model can recommend discussions to a student by balancing the benefit of the student’s qualification to discussions, her relevance to discussions, and required effort.

## 3. THREE-PART ANALYTICS PIPELINE

Our pipeline is designed to expedite the process of exploiting student data leading to data-driven decision-making for enhancing student learning (Figure 1).

In this pipeline for social learning, the first component is a data infrastructure that maps diverse forms of social interaction into a common structure. This uniform interface allows the subsequent components—learning process analysis and intervention—to apply the same tools to different data, even from distinctly different discourse types, with little modification. Our development of this infrastructure, DiscourseDB<sup>2</sup>, represents discourse-centered social interaction as an entity-relation model. Discourses (e.g., forums or social media) and individual contributions in a discourse (e.g., posts, comments, and utterances) are represented as generic containers generalizable to diverse social platforms. DiscourseDB also allows for defining arbitrary relations between contributions, e.g., a “reply-to” relation derived from the explicit reply structure of the platform versus one inferred through some automated analysis process. This flexibility helps the subsequent components of the pipeline avoid data-specific processing. DiscourseDB can store both active and passive activities of individuals, such as creating, revising, accessing, and following contributions, as well as forming social connections with other individuals. DiscourseDB is the key component of our pipeline, based on which the next components perform integrated analyses of discourses and social networking on multiple platforms with reusability.

The second component of our pipeline is analysis of students’ learning processes depending on their social connections. The goal is to assess students’ needs of support by understanding how learning processes are affected by social interaction and what types of social interactions are helpful to students. Just as Bayesian knowledge tracing enables modeling the learning process from a cognitive perspective and then supporting a student’s progress through a curriculum, Bayesian approaches can model learning processes at other levels, including supportive social processes. And similarly, these models can then be used to trigger support for the learning processes in productive ways. Hence, the third component of our pipeline draws upon insights obtained from the analysis to introduce interventions that can help students make beneficial social connections with other peers. We will propose two concrete examples of machine learning techniques for these two components in Section 5 and Section 6 respectively.

## 4. APPLICATION OF PIPELINE

The remainder of the paper presents an example application of our general pipeline to a specific problem. We propose ex-

<sup>2</sup><http://discoursedb.github.io>

ample models for learning process analysis and intervention that can build upon DiscourseDB. After this description we discuss our findings. This section introduces the data set for that exploration.

#### 4.1 Problem and Data

We examine goal-setting behavior as a potential qualification of good role models via learning process analysis and foster social connections with goal setters via recommendation support. Since most MOOCs and informal learning communities lack a measure to identify potentially good role models (e.g., a pretest), increased frequency of effective goal-setting behaviors may serve as an indirect indicator of success, as previous studies showed positive relationships between goal-setting behavior and learning outcomes [5, 23].

The data was collected from an edX MOOC entitled *Data, Analytics, and Learning* (DALMOOC) [12], which ran from October to December 2014. This course covered theoretical principles about learning analytics as well as tutorials on social network analysis, text mining, and data visualization. This MOOC was termed a *dual layer* MOOC because students had the option of choosing a more standard path through the course within the edX platform or to follow a more self-regulated and social path in an external environment called ProSolo. The ProSolo layer allowed students to set their own learning goals and follow other students so that they could view activities and documents that offered clues about how to approach the course productively. While a huge literature on analysis of MOOC data focuses on Coursera, edX, and Udacity MOOCs, other platforms with more social affordances are growing in popularity. In order to serve the goal of identifying support needs and automating support that may be triggered in a social context, it is advantageous to work with data from socially-oriented platforms. We used the log data from ProSolo as our object of analysis, which include students’ discussions on ProSolo and their own blogs and Twitter that they identified on their ProSolo profile pages, evidence of students’ social connection with each other, and “goal notes,” which students can use to set their learning goals in their own words.

We preprocessed discussion data before running our model. First, we filtered course-relevant tweets using the hashtags #prosolo, #dalmooc, and #learninganalytics. We confirmed that the tweets identified as irrelevant by this process have little to do with course activity. Because we are not interested in irrelevant content, we replaced such content with a tag to indicate irrelevant content. In order to prevent topics from being defined in terms of document types, we removed Twitter mentions and “RT” from tweets as well as other function words including URLs from all documents. Descriptive statistics for the data set are listed in Table 1.

#### 4.2 Goal Quality and Social Connection

To categorize the quality of goal-setting behavior of each student, we first annotated each goal note written by students indicating whether it indeed contains a goal or not. 58% of goal notes contained goals. An example goal note is as follows: “*to understand learning analytics and see how these may be useful for my teaching and in particular, my learning resource design/development.*” On the basis of this annotation, we categorized students into three classes: (1) goal

Goal notes	62	Tweets (relevant)	715
ProSolo posts	318	Tweets (irrelevant)	25,461
Blog posts	359		
Users	1,729	Social connections	814

Table 1: Descriptive statistics for ProSolo data.

setters, (2) goal participants, and (3) goal bystanders. Goal setters have goal notes that mention their distal or/and proximal goals. Goal participants have goal notes, all of which are about something other than goals, e.g., experiences or questions. Goal bystanders have no goal notes. Note that the category of a student can change over time. All students start as goal bystanders and may become a goal participant or a goal setter as time passes. A student’s *social connection* is then categorized into seven classes: (S1) has already been following a goal setter, (S2) started to follow a goal setter at the current time point (S3) has been following a goal participant (but no goal setter), (S4) started to follow a goal participant at the current time point, (S5) has been following a goal bystander (at best), (S6) started to follow a goal bystander at the current time point, and (S7) follows no one. S2, S4, and S6 mean that a student’s social connection improved at the current time point, whereas S1, S3, and S5 indicate that a student remained in the same social connection category as in the previous time point.

### 5. LEARNING PROCESS ANALYSIS

Learning process analysis aims to assess students’ needs of support. Hence, we model students’ behavior and analyze their learning processes as they experience changes in their social connections throughout the course.

#### 5.1 Model

Our model automatically extracts a representation of students’ learning processes based on their discussions in a course and their social connections, which may reveal the influence of different configurations within the social space (see our technical report [7] for details). We define the building blocks of learning processes, i.e., states, in terms of discussed topics and the document types used for discussions (e.g. Twitter, blog). Given the sequences of timestamped documents and social connection types for students, our latent Markov model infers a set of states, along with the main topics and document types for each state. The learned topics reflect students’ interests, and the document types show how students use different media for different interests. The model also learns transition probabilities between states, conditioned on the social connection category in the source state. This discloses how learning processes differ depending on students’ social connection types.

#### 5.2 Findings

We applied the model to the ProSolo data and examined the correlation between the categories of social connection and learning behaviors. We ran our model with the number of states set to 10 and the number of topics set to 20. We defined the unit of a time point as one week, and if a student had no activity in a certain week, that week was omitted from her sequence.

State	Topics	RelGoalNote	IrGoalNote	Post	Blog	RelTweet	IrTweet
0	Course-irrelevant tweets	0.00	0.00	0.00	0.00	0.00	1.00
1	Concept map, network analysis (Week 9)	0.00	0.00	0.02	0.01	0.18	0.78
2	Social capital (Week 3)	0.04	0.01	0.19	0.30	0.18	0.27
3	Tableau (Week 2), Gephi (Week 3), Lightside (Week 7)	0.01	0.03	0.10	0.28	0.24	0.34
4	Prediction models (Week 5)	0.01	0.02	0.29	0.22	0.10	0.36
5	Data wrangling (Week 2)	0.01	0.01	0.12	0.08	0.26	0.52
6	Visualization (Week 3)	0.05	0.02	0.24	0.47	0.08	0.15
7	Epistemology, assessment, pedagogy (Week 4)	0.05	0.00	0.18	0.22	0.30	0.25
8	Prediction, decision trees (Week 5)	0.02	0.02	0.19	0.40	0.09	0.28
9	Share, creativity (mixed topics)	0.00	0.02	0.12	0.13	0.21	0.52

Table 2: Learned states with their topics and document type distribution (each row sums to 1). (RelGoalNote: goal notes containing a goal, IrGoalNote: goal notes without a goal, Post: posts on ProSolo, Blog: personal blog posts, RelTweet: course-relevant tweets, IrTweet: course-irrelevant tweets)

	Social Connection			
	GS $S_1+S_2$	GP $S_3+S_4$	GB $S_5+S_6$	NO $S_7$
# Time Points	139	315	265	821
% Time Points				
State 0	<b>0.59**</b>	<b>0.75</b>	<b>0.75</b>	<b>0.71</b>
State 1	<b>0.17*</b>	<b>0.10</b>	<b>0.03</b>	<b>0.04</b>
State 2	0.05	0.02	0.02	0.04
State 3	<b>0.04*</b>	<b>0.00</b>	0.01	<b>0.01</b>
State 4	0.01	0.02	0.03	0.06
State 5	0.05	0.03	0.06	0.05
State 6	0.05	0.02	0.02	0.02
State 7	0.03	0.01	0.03	0.02
State 8	0.00	0.03	0.02	0.02
State 9	0.01	0.04	0.03	0.04

Table 3: Proportion of the time points students stay in each state depending on the social connection (each column sums to 1). “\*\*” and “\*” indicate that GS is significantly different from other categories in bold with  $p < 0.01$  and  $p < 0.05$ , respectively, by Pearson’s  $\chi^2$  test. GS, GP, and GB each represent either “has been following” or “started to follow” a goal setter, a goal participant, and a goal bystander, respectively. NO means to follow no one.

### 5.2.1 Learned States

The model learns states with their topics and document type distributions (Table 2). Most states are aligned well with course units covering important course topics. However, State 0 is where students do not participate in course discussion but post course-irrelevant tweets. State 3 is about hands-on practice of software tools across the course, and State 9 covers many side topics. Tweets tend to take a large proportion and goal notes a small proportion in every state due to their relative volumes. Blog posts are actively used for summarizing readings and tutorials, and tweets are used as a means of communicating with lecturers (e.g., State 5). ProSolo posts are most accessible to ProSolo users, so students use them to reveal their opinions and questions.

### 5.2.2 Students Following Goal Setters

According to the investigation of students’ learning processes, based on the number of weeks they spent in each state (Table 3) and state transition patterns (Figure 2), students who follow goal setters show the following positive learning behavior:

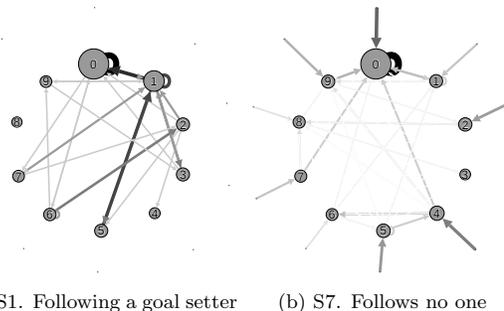


Figure 2: State transition patterns. Nodes are states whose size reflects the number of weeks students visit the states. Edges are transitions whose thickness and darkness reflect transition frequency. Edges without a source node represent the probability of being the first state in a learning path.

**Twitter usage:** The students following goal setters spend noticeably fewer weeks on irrelevant tweets (State 0).

**Participation duration:** The topics of the states in which students stay reveal how long they persist in the course. The students following goal setters are more likely to discuss the material taught in the last week (State 1), that is, they are active in the last phase of the course.

**Activities of interest:** The number of weeks students spend in each state reflects the activities students are interested in. The students following goal setters were more active in hands-on practice (State 3) than other students. Hands-on practice requires higher motivation than merely watching lectures, so these students might have been helped by observation of role models as discussed in the literature [14]. This trend would have not been as clear using predefined states based on course units [3].

**Study habits or challenges:** Transition patterns may reveal students’ study habits or challenges. Figure 2a shows frequent transitions between three states (States 1, 3, and 5) that are associated with materials taught in different weeks. Such transitions may reflect the SRL strategy of activating and applying prior knowledge to the current situation [10].

These positive effects associated with following goal setters are not apparent with other social connection types, e.g., following no one (Figure 2b). This indicates that “who to follow” is more important than simply following someone.

## 6. INTERVENTION FOR SUPPORT

On the basis of the insights obtained from the previous component, the third component of our pipeline is to offer appropriate support, especially towards fostering beneficial social connections between students. We argue that a recommender system can serve this purpose, by presenting its potential positive impact as assessed on the corpus.

### 6.1 Model

Our recommender system aims to match qualified students (e.g., goal setters) to discussions so that they can interact with and benefit the discussants through discussions (see our technical report [7] for details). Our model has two steps: relevance prediction and constraint filtering. The relevance prediction step learns the relevance between students and discussions using student- and discussion-related features. The learned relevance reflects students' preferences and tendencies, but may not reflect the ideal matches for fostering learning. The constraint filtering step thus combines the relevance scores with some constraints that foster interaction between qualified students and other students, and finalizes recommendations.

### 6.2 Findings

Since we have identified positive learning behaviors of students who follow goal setters, we may want to support students by fostering interaction with goal setters. Instead of recommending direct following relations, which are not supported by many learning platforms, we recommend discussions to qualified students so that they can interact with the discussants. We first assess the extent to which students are sensitive to qualified students prior to explicit intervention, and then present the potential added value of our recommendation model.

#### 6.2.1 Students' Awareness of Role Models

Our first step is to assess whether students can identify effective role models in discussion activities (ProSolo posts), by measuring the impact of the information about students' qualifications on the prediction of discussion participation. This task is to infer links between students and discussions that we hid from an observed static snapshot of a network of discussion participation based on observable data. A measured positive impact here would indicate some sensitivity on the part of students to interact with qualified students naturally. We train a predictive model of students' participation in discussions on two thirds of student-discussion pairs. We then predict the discussion participation of the remaining pairs. Our evaluation metric is mean average precision (MAP).

We compared four configurations by varying the information about students' qualifications that is used as feature for relevance prediction. In particular, *CAMF* uses only basic features, such as the numbers of discussions each student initiated and participated in and each discussion's length, number of replies, and participants. *CAMF\_G* and *CAMF\_C* add information about goal quality and degree centrality, respectively, and *CAMF\_GC* adds both. The evaluation was conducted as a link prediction task, based on the relevance scores predicted in the relevance prediction step. Students' qualification information did not improve link prediction ac-

Configuration	MAP	Configuration	MAP
<i>CAMF</i>	0.465	<i>CAMF_C</i>	0.455
<i>CAMF_G</i>	0.438	<i>CAMF_GC</i>	0.439

Table 4: MAP for link prediction.

Configuration	OB	Configuration	OB
<i>GoalPart</i>	1.888	<i>MCCF_G</i>	3.683
<i>HighCent</i>	1.943	<i>MCCF_C</i>	3.770
<i>GoalPart_HighCent</i>	1.873	<i>MCCF_GC</i>	3.656

Table 5: Overall Community Benefit for recommendation.

curacy (Table 4). This means that students are not proactively sensitive to peers' qualifications while participating in discussions, which supports our view that explicit recommendation could be valuable for encouraging students to interact with qualified peers through discussions.

#### 6.2.2 Recommendation Quality

The recommendation of discussions should be consistent with both the relevance between students and discussions (the relevance prediction step) and constraints for beneficial social connection (the constraint filtering step). To this end, we evaluated recommendation quality on Overall Community Benefit (OB) [7]: the relevance of our recommendations penalized by the burden on the students induced by the recommendations. The higher OB the better.

We tested three configurations by varying the constraints incorporated into the constraint filtering step. *MCCF\_G* requires that every discussion have at least one goal participant or goal setter. *MCCF\_C* requires that every discussion have at least one student whose degree centrality is higher than 0.1. *MCCF\_GC* requires both. In addition, the following configurations were tested as baseline without incorporation into the model. *GoalPart* filters goal participants or goal setters after making recommendations based on predicted relevance. Similarly, *HighCent* filters students with degree centrality higher than 0.1. *GoalPart\_HighCent* filters goal participants or goal setters with degree centrality higher than 0.1. Incorporating the constraints about students' goal quality and degree centrality into the model (*MCCF\_G*, *MCCF\_C*, and *MCCF\_GC*) achieved higher OB than the simple filtering approaches (Table 5). That is, our algorithm effectively matches qualified models to relevant discussions in such a way that students in every discussion can interact with qualified models while balancing the load of the models.

## 7. DISCUSSION

According to our learning process analysis, students benefit from social connections with effective goal setters through ProSolo's follower-followee functionality. They stay longer in the course, engage in hands-on practices, and link materials across the course. This supports the view that goal-setting behavior is a useful qualification for potential role models. According to the discussion participation prediction task, explicit intervention is important for helping students be aware of qualified students and interact with them via discussions. Therefore, we incorporated the information about students' qualifications into our recommendation model as

constraints, successfully matching qualified learning partners to relevant discussions.

This work started from the need for expediting data analysis and analysis-informed support in social learning where students interact with one another via various social media in order to pursue their own learning goals. This expedition builds on DiscourseDB, data infrastructure for complex interaction data from heterogeneous platforms. We proposed a probabilistic graphical model to analyze students' learning processes depending on the state of their social connections, and proposed a recommender system that can improve student support on the basis of the insights obtained from the analysis. This pipeline arguably should allow us to apply the techniques to different learning communities with little effort.

Goal-setting behavior is an important practice in SRL and is known to be difficult for students, so an analysis towards improvement of this skill is arguably valuable. Nevertheless, in this study we have not examined how this behavior influences the domain learning of students. This is due both to the limited data size for our first trial to use ProSolo in MOOCs as well as a lack of learning gain measures. However, the modeling techniques proposed in this paper can readily be applied to other data sets if the requisite data become available. We are also interested in investigating different SRL strategies besides goal-setting in social learning, and how social interaction influences the SRL behaviors of the students. Ultimately, the real value of the work will be demonstrated not with a corpus analysis, as for our proposed recommendation approach, but with an intervention study in a real MOOC. We are working towards incorporating this approach in a planned rerun of DALMOOC.

## 8. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under grants ACI-1443068 and IIS-1320064, and by the Naval Research Laboratory and Google.

## 9. REFERENCES

- [1] A. Bandura. *Social Learning Theory*. Morristown, N. J.: General Learning Press, 1971.
- [2] A. Bogarín and R. Cerezo. Discovering students' navigation paths in Moodle. In *EDM '15*, pages 556–557, 2015.
- [3] C. Coffrin, L. Corrin, P. de Barba, and G. Kennedy. Visualizing patterns of student engagement and performance in MOOCs. In *LAK '14*, pages 83–92, Mar. 2014.
- [4] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth. Unsupervised modeling for understanding MOOC discussion forums. *LAK '15*, pages 146–150, 2015.
- [5] J. Husman and D. F. Shell. Beliefs and perceptions about the future: A measurement of future time perspective. *Learning and Individual Differences*, 18(2):166–175, Apr. 2008.
- [6] Y. Jo and C. P. Rosé. Time Series Analysis of Nursing Notes for Mortality Prediction via a State Transition Topic Model. In *CIKM '15*, 2015.
- [7] Y. Jo, G. Tomar, O. Ferschke, C. P. Rosé, and D. Gašević. Expediting support for social learning with behavior modeling. *arXiv:1605.02836*, 2016.
- [8] I. Molenaar and M. M. Chiu. Effects of sequences of socially regulated learning on group performance. In *LAK '15*, pages 236–240, Mar. 2015.
- [9] E. L. Paluck, H. Shepherd, and P. M. Aronow. Changing climates of conflict: A social network experiment in 56 schools. *Proceedings of the National Academy of Sciences*, 113(3):566–571, Jan. 2016.
- [10] P. R. Pintrich. A Conceptual Framework for Assessing Motivation and Self-Regulated Learning in College Students. *Educational Psychology Review*, 16(4):385–407, Dec. 2004.
- [11] C. P. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer. Social factors that contribute to attrition in MOOCs. In *L@S '14*, pages 197–198, Mar. 2014.
- [12] C. P. Rosé, O. Ferschke, G. Tomar, D. Yang, I. Howley, V. Aleven, G. Siemens, M. Crosslin, D. Gasevic, and R. Baker. Challenges and Opportunities of Dual-Layer MOOCs: Reflections from an edX Deployment Study. In *CSCL '15*, pages 848–851, 2015.
- [13] E. Rowe, R. S. Baker, and J. Asbell-Clarke. Strategic game moves mediate implicit science learning. In *EDM '15*, pages 432–436, 2015.
- [14] D. H. Schunk and A. R. Hanson. Peer models: Influence on children's self-efficacy and achievement. *Journal of educational psychology*, 77(3):313–322, 1985.
- [15] G. Siemens. Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*, 2014.
- [16] L. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM transactions on computer-human interaction*, 12(3):401–434, 2005.
- [17] C. Thille. Education Technology as a Transformational Innovation. *White House Summit on Community Colleges: Conference Papers*, pages 73–78, 2010.
- [18] K. Veeramachaneni, S. Halawa, F. Deroncourt, U. O'Reilly, C. Taylor, and C. Do. Moocdb: Developing standards and systems to support MOOC data science. *CoRR*, abs/1406.2015, 2014.
- [19] P. H. Winne and a. F. Hadwin. Self-regulated learning and socio-cognitive theory. *International Encyclopedia of Education*, pages 503–508, 2010.
- [20] D. Yang, D. Adamson, and C. P. Rosé. Question recommendation with constraints for massive open online courses. In *RecSys '14*, pages 49–56, 2014.
- [21] J. Yang, J. McAuley, J. Leskovec, P. LePendou, N. Shah, and B. Informatics. Finding Progression Stages in Time-evolving Event Sequences. In *WWW '14*, pages 783–793, Apr. 2014.
- [22] C. Zhao and L. Wan. A shortest learning path selection algorithm in e-learning. *Int'l Conference on Advanced Learning Technologies*, pages 94–95, 2006.
- [23] B. J. Zimmerman. Goal setting: A key proactive source of academic self-regulation. In *Motivation and self-regulated learning: Theory, research, and applications*, pages 267–295. Erlbaum, 2008.

# On generalizability of MOOC models

Łukasz Kidziński, Kshitij Sharma, Mina Shirvani Boroujeni, Pierre Dillenbourg  
Computer Human Interaction in Learning and Instruction  
École polytechnique fédérale de Lausanne  
{lukasz.kidzinski,kshitij.sharma,mina.shirvaniboroujeni,pierre.dillenbourg}@epfl.ch

## ABSTRACT

The big data imposes the key problem of generalizability of the results. In the present contribution, we discuss statistical tools which can help to select variables adequate for target level of abstraction. We show that a model considered as over-fitted in one context can be accurate in another. We illustrate this notion with an example analysis experiment on the data from 13 university Massive Online Open Courses (MOOCs). We discuss statistical tools which can be helpful in the analysis of generalizability of MOOC models.

## Keywords

Massive open online courses, MOOCs, bias-variance trade-off, generalizability

## 1. INTRODUCTION

The rapid growth of Massive Online Open Courses (MOOCs) has shown significant impact not only on the education but also on educational research. Over 100 world class universities partner with MOOC platforms to provide free education. Many of these universities, use data analytics to provide indicators to the policy makers, and valuable insights to the teachers and producers.

Researchers from emerging educational fields, such as learning analytics and educational data mining, attempt to make sense from the huge datasets from the MOOC providers (for example Coursera, Edx). These large datasets provide an opportunity to detect the slightest differences in the behaviour which are correlated to the students' performance.

However, the big data involves the risk of misinterpreting the results. The misinterpretations could surface mainly because of two reasons. First, the effect sizes are few orders of magnitude smaller than we used to expect in classical educational psychology studies; and the results are still significant due to the large sample. Second, "black-box" approaches like Support Vector Machines or Neural Networks give us great

predictive power of models but do not explain the underlying processes.

Both of these reasons can lead to "overfitting" a model for a given context. Still, the same model can be accurate in another context as illustrated in Figure 2. Choosing too specific descriptors could lead to the models which precisely describe one student but fail to generalize to new concepts. Too vague descriptors tend to generalize better but inform less about the specifics of the underlying processes. In statistical terminology this is often referred to as the "bias-variance trade-off".

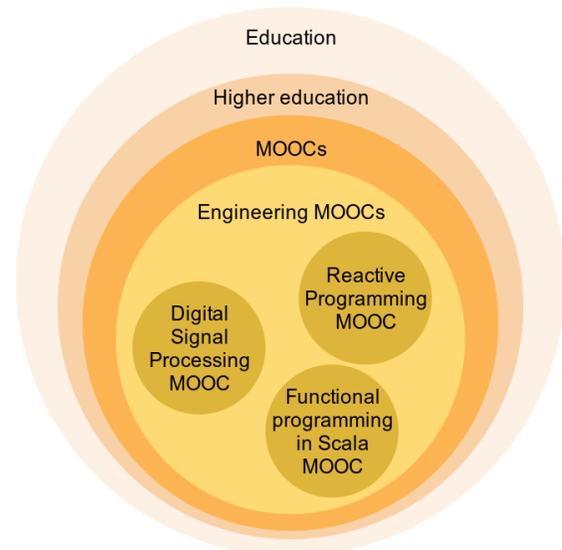


Figure 1: Example of layers to which we can draw conclusions from instances of MOOCs if the generalizability issues are addressed correctly.

The bias-variance trade-off is the central problem in statistical learning. It corresponds to the fact that one cannot minimize both quantities, "bias" and "variance", at the same time. A model with large bias is a smooth model not meant to fit sample points very closely but still captures the general trend in the data. Conversely, a model with large variance (not smooth) varies a lot for similar input parameters in order to fit well to each point in the dataset, often causing the so-called "overfitting".

The objective of this paper is to highlight the potential problem of closed-world context of MOOC research. We discuss techniques for leveraging existing models to more general context. We argue that designing context independent features is crucial for building generalizable models and we illustrate how variable selection process can be enhanced with statistical techniques. We illustrate a statistical technique which can be helpful in the choice of the important variables.

We address the following three research questions:

1. How to measure the extent to which the MOOC research as generalizable?
2. How to leverage predictive models in a MOOC to a broader context?
3. How to improve model's accuracy by restraining the scope of the variables used for prediction purposes?

## 2. RELATED WORK

### 2.1 Student Categorization

The common approach for finding generalizable patterns is to classify students into groups. To the best of our knowledge, there exist only a few categorisation schemes, mostly based on what emerges as a pattern of behaviour from MOOC students. These categories are based on the students' motivation [20], engagement patterns [10, 14, 16, 7] or demographics [5, 4].

There are many categorisation schemes depending on the engagement patterns. [10] categorised the students in Completing, Auditing, Disengaging and Sampling students based on their activities which range from watching majority of lectures and submitting all the assignments (Completing) to watching only one or two lectures and no assignment submissions (Sampling). In a connectivist MOOC setting, [14] categorised students into Active (students who adapt well to the connectivist pedagogy), Passive (frustrated ones) and Lurkers (who actively follow the course but do not interact with anyone). Phil Hill first categorised MOOC students into Lurkers (ones who only enrol or sample the course), Active (fully engaged with the course material, quizzes and forums), Passive (only consume the content, did not participate in forums) and Drop-ins (consumed only a part of the course as an Active student) [8]. Later he revised his categories and divided the Lurkers into No-shows and Observers [7].

These schemes are either defined by hard-coded thresholds or by unsupervised learning techniques. For that reason, they remain robust in terms of generalizability within the MOOC's context, but they are hard to generalize outside of it. In this study, we will rather discuss regression than classification/clustering, keeping in mind that similar observations can be done in both contexts.

### 2.2 Performance and engagement prediction

Student's performance is one of the key metrics analyzed in MOOCs. Many studies chose performance as an indicator for showing the value of the categorization methods. Massive datasets allow us to discover relation between performance and even the smallest factors like the number of

pauses during watching a MOOC video or ratio of a video re-played [12]. Performance is also a crucial indicator for policy makers and MOOC practitioners. Reports focus on performance of MOOCs as a function of performance of students [13].

Previous studies on performance often concern a small set of MOOCs [1, 17, 9]. These studies provide insights about a large cohort of students and generalize to another cohorts, however the studies encounter lack of generalizability due to a small sample in the sense of course variability. In other studies, authors used time spent on lecture video, lecture quiz, homework, forum, quiz, assignments to predict students' learning gain [3, 11, 21, 3]. Lauria et al. [11] used the amount of content viewed, forum read, number of posts, assignments and quizzes submitted, to predict the performance and the engagement of the students. Wolff et al. [21] used the temporal clickstream data to predict students' performance.

These studies risk having high bias towards the courses in context and thus might lack the generalizability to be extended to courses with different content and/or courses from different domain. However in the aforementioned works, it is difficult to confirm our claim due to small number of MOOCs being analyzed. An example with generalizable set is shown by [2], where authors used the weekly time series data with 2-, 3-, 4-, and 5- grams to predict the final grades of the students. They experienced issues with the predictive models being generalisable - the model accuracy decreases as the authors used the same course session, to a different session from the same course, to a different course.

## 3. PROBLEM STATEMENT

In the MOOC context, models with large variance might correspond to the cases where one includes specific information about users, which are characterising only the sample at hand. For example, a model which includes exact timing of actions into account, could fit precisely to the data, since it identifies the user by the time of his actions, but it provides no generalizability to new samples. Conversely, models with high bias correspond to situations when one considers general indicators like only the number of forum activities in a MOOC - thus, the model will fit worse to specific users but is more likely to generalize.

In practice, it is impossible to make both variables small, i.e. to retain both good fitness and smoothness. We need to choose the complexity of the model such that the sum of these two quantities is minimised. One could show that for any statistical learning method, the error can be decomposed to variance and bias terms. For a given target value  $y$ , predictors  $x$  and the estimator  $\hat{f}$ , the error of the model can be depicted as:

$$E[(y - \hat{f}(x))^2] = \text{Bias}[\hat{f}(x)]^2 + \text{Var}[\hat{f}(x)] + \sigma^2, \quad (1)$$

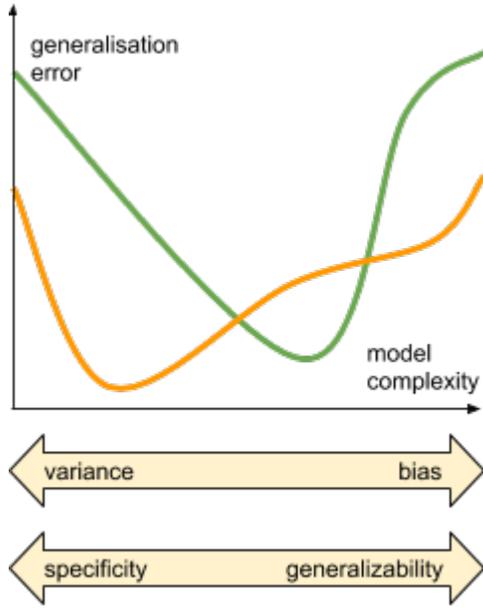
where  $\sigma$  is the standard deviation of the residuals,

$$\text{Bias}[\hat{f}(x)] = E[\hat{f}(x)] - f(x)$$

and

$$\text{Var}[\hat{f}(x)] = \text{E}\left[(\hat{f}(x) - \text{E}[\hat{f}(x)])^2\right].$$

In other terms, bias is the squared distance between the real output  $f(x)$  and the average prediction for given  $x$ , i.e. the  $\text{E}[\hat{f}(x)]$ . The bias gets large whenever the average of predictions  $x$  differs highly from  $\text{E}[\hat{f}(x)]$ . Conversely, the variance, expressing how do prediction vary from average around  $x$ , gets large whenever the variability is high.



**Figure 2: Influence of bias-variance trade-off on the generalization error - illustrative conceptual drawing.**

The ideal model would have both quantities  $\text{Bias}[\hat{f}(x)]^2$  and  $\text{Var}[\hat{f}(x)]$  equal to zero, but, as we mentioned before, it is not practically possible. However, we can control this error, as both quantities depend on the complexity of the model. For example, a linear model with large number of parameters has high variance and thus the error term increases. On the contrary, if one chooses low complexity (small number of variables), the model might have high error due to the high bias. The “best” model is somewhere in the middle, as illustrated by the green curve in Figure 2.

What is often missed in the analysis of the bias-variance plot, is that the error depends also on the context in which we generalize. Particularly in the MOOC context, in Figure 2 the green curve corresponds to generalization to another instance of the same MOOC, whereas the error follows a different pattern (orange curve) if we change the context to another MOOC.

## 4. MATERIALS AND METHODS

As we focus on the concept of generalizability of models and robustness of variables, we investigate our approach on

several different MOOCs. We used data from 13 MOOCs, from EPFL, from both coursera and edX platforms. The dataset contains 1 MOOC which had 3 sessions in 3 consecutive semesters and 2 MOOCs which had 2 sessions in 2 consecutive semesters, as indicated in Table 4.

This setup allows us to investigate several aspects of generalizability. We investigated the fit of a model in correspondence to: 1) the course itself; 2) another instance of the same course; 3) another engineering course.

### 4.1 Setup

In order to attain a generalizable model, the setup must be consistent between the training data and the test data. Thus, we use the variables which could be defined for all the courses. Additionally, all the scores are normalized to the same range (0 - 100). Since courses have different lengths, we focus only on student activities in the first week. Finally, since 95% of the students did not submit any assignments and significantly bias linear models, we analysed only those students who got at least 1 point as their final grades. Note, that the context we are defining serves mainly as an illustration, thus we choose a relatively simple setup for transparency.

As the measure of performance of a model we take the Normalized Mean Squared Error (NMSE), defined as

$$NMSE = \text{Var}(y - \hat{f}(x)) / \text{Var}(y),$$

where  $y$  is the dependent variable to predict,  $\hat{f}$  is the estimator of the relation between  $y$  and independent variable  $x$  and  $\text{Var}$  corresponds to the sample variance.

### 4.2 Example method

In the linear regression, the main source of complexity is due to the number of variables in the model. Classical statistics provide us with robust tools for variable selection, such as ANOVA, Akaike Information Criteria. These techniques are useful for their inferential value, however, they do not guarantee the best generalizability in terms of prediction.

One of the techniques, where the complexity is controlled using a parameter that also affects the performance of the model, is regularized linear regression. In classical statistics, called ridge regression, the standard linear model is extended with an additional, regularizing term. This regularising term controls the parameters of the model with respect to the performance measure based on the prediction, by decreasing the importance of variables which do not account for the prediction.

In particular, given the independent variables  $X_1, X_2, \dots, X_d$  and the dependent variable  $Y$  we build a model minimizing

$$\text{E}\|Y - \beta_1 X_1 - \beta_2 X_2 - \dots - \beta_n X_d\|^2 + \lambda \sum_{i=1}^k \beta_i^p, \quad (2)$$

where  $d$  is the number of variables,  $\beta_1, \beta_2, \dots, \beta_d$  are the parameters of the model and  $p = 2$ .

If  $\lambda$  is large, we put more weight to the sum of  $\beta$ s. Therefore, the number of parameters will be reduced and the model will have a low bias. On the other hand, if  $\lambda$  is small, the model corresponds to linear regression and the variance is high since we use all the variables.

We chose this model for our analysis since it allowed us to control both the bias and the variance with a single parameter  $\lambda$ . Moreover, changing the value of  $p$  from 2 to 1 is (2), gives better results in many setups. Hence, we choose to use  $p = 1$ . The model is known in the machine learning literature as LASSO [19]. The complete algorithm, for those interested, can be found in [19]. Here we are refraining ourselves to the basic description as this is not the main focus of the paper.

### 4.3 Variables

For illustrating the problem, we chose the students' final grade in the course as the dependent variable. Following are the features that we extracted from the data for modeling this value.

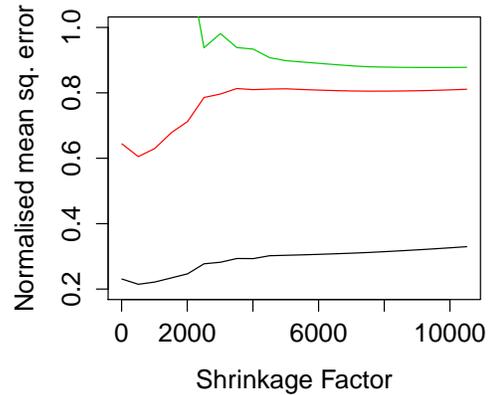
1. **Counts:** We counted different online activities exhibited by the students. 1) *Lectures*: lecture view, lecture re-view, lecture download and lecture re-download; 2) *Quiz*: quiz submission, quiz re-submission, here we differentiated between the quizzes as an exercise, in-video quizzes and the surveys; 3) *Assignments*: assignment submission and assignment re-submission; 4) *Forums*: thread launches, upvotes, downvotes, subscriptions, views, comments and posts.
2. **Delays:** We computed the time difference between the different events in the MOOC structure and students' activities. 1) *First View Delay*: the time difference between the first view or first download of the lecture and the time when the lecture was online; 2) *Overall View Delay*: the average first view delay for all the lecture views and downloads; 3) *Between Lecture Delay*: the time difference between the views or downloads of two different lectures; 4) *Within Lecture Delay*: the average time difference between two views and/or downloads of the same lecture; 5) *First Quiz Attempt Delay*: the time difference between the first submission for a quiz and the time when the quiz was online; 6) *Within Quiz Time*: the time difference between two attempts for the same quiz; 7) *Overall Quiz Attempt Delay*: the average first quiz attempt delays for all the quizzes.
3. **Progress:** We computed the score difference between the two consecutive attempts to the same quiz or the same assignment.
4. **2-way Transitions:** We labeled the different activities as L, A, Q and F for lectures, assignments, quizzes, and forums respectively. Further, we constructed a time-series of the actions and counted how many times the action pairs (for example, AA, AL, AF, LQ, FL, 16 pairs) occur in the time series for each student.

5. **3-way Transitions:** using the same time series, as to compute the 2-way transitions, we counted how many times the action triples (for example, AAA, FAL, QAF, LLQ, FLL, 64 triples) occur in the time series for each student.

## 5. RESULTS

Using the variables, defined in the Section 4.3, we illustrate the setup for modelling the data. As we mentioned in the Section 4.1, we considered only the activities from the first week of the courses and from those students who scored at least 1. We would also like to emphasize here that the main aim of this contribution is not to present a model that has the least error, but to show how we can build generalizable models taking into account the bias-variance trade off.

In the proposed setup, we demonstrate how generalizable a model is to: 1) the students from the same course (separate test set of 20% of observations), 2) the students from another instance of the same course, 3) to a different course.



**Figure 3: Prediction error (NMSE) for the test samples, for the different values of the shrinkage factor  $\lambda$  in (2), using all the variables.**

First, we analyze the model fit to the first session of the *Numerical Analysis* course and test it on: itself, another session of *Numerical Analysis* and *House Water Treatment Systems* a course from a different domain. We illustrate the results in Figure 3. We observed that the model which had highest predictive power on the test set in the session 1 (black curve) has the worse predictive power for another instance of the same course (red curve), but still performs well. The optimal shrinkage factor ( $\lambda$  in equation (2)) turns out to be close to 0 in both cases. This shows that almost all the variables we introduced are included in the model. We could conclude that the model generalizes to another instances of the same course.

However, as we hypothesized, the full model did not fit at all to a course from a different domain. Only with a large value of the shrinkage factor, which removed 97 variables

from the model, we obtain a model with some informative value for a course from another domain. Furthermore, the errors become similar for all the courses, illustrating that the model has lower variance. It generalizes better to another course but it lost its fit to the Numerical Analysis course.

We conducted the identical analysis (see Table 1) on all the courses mentioned in the dataset. In all the cases, generalizability to another course required significant decrease in the complexity, using the shrinkage factor. Removing certain variables from the model turns out to be crucial for the performance. Since we started with 134 variables, to further analyze the ability to generalize, we restricted ourselves to a simpler case with the first three (counts, delays and progress) groups of variables introduced in Section 4.3.

The same patterns were observed in this simpler case. The optimal model for prediction in the same instance and in another instance of the course have the lowest error if the complexity (variance) is high. However, the model with such a high complexity exhibits poor performance in another course, from the same domain, i.e. the linear optimization.

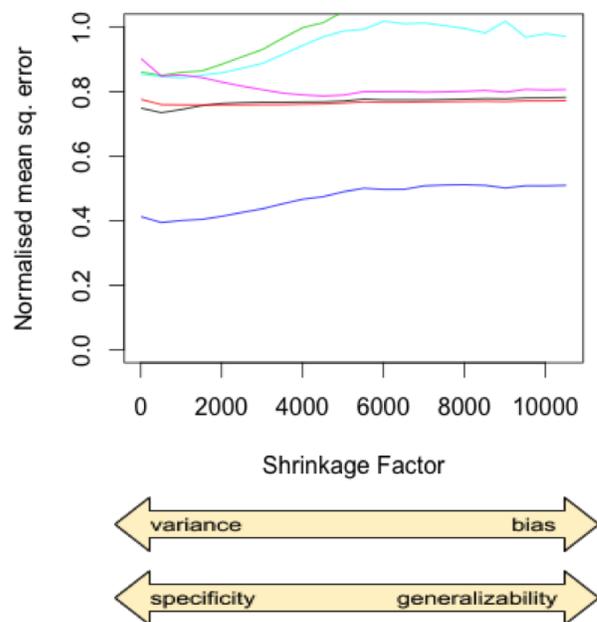
As hypothesized, variables which were removed by LASSO, are course-structure dependent. The most generalizable models contain the variables related to the lecture, forum and quiz activities. These variables provide the required generalizability to the model and hence we observe that as we increase the shrinkage factor, the predictive power of the model increasingly became similar for the different courses.

## 6. CONCLUSION

We demonstrated through examples that in the terms of bias-variance trade-off, achieving both the specificity and generalizability is not possible while modelling student behaviour. Through the statistical methods available, one can only achieve one of the two goals, or find an optimum solution that is specific to one course and only reveals the surface learning behaviour of the students from a course from another domain, or vice versa.

Similar validation framework, analysing fitness in the same course, another session of the same course and another course was previously introduced [2] in literature. Results from this work are equivalent to ours with some predefined and fixed complexity parameter. In our work we show that practitioners can modulate the complexity and generalizability by selecting a subset of variables.

Previous works, have small sample size in terms of number of MOOCs. It is therefore difficult to assess their generalizability. For example, Social Network Analysis (as shown by [18, 15, 6]) is based on the motivation of the student - if the students are sharing the exact answers (or revealing them in some other ways) forum view can play a big role in achievement. Clickstreams (as shown by [21]) in a video are highly dependent on the content. Finally, from the methodological perspective generalizability is also a design choice - for example - if we choose a smaller number of clusters in unsupervised learning, we may obtain more robust results (smaller variance higher bias).



**Figure 4: Illustration of bias-variance trade-off from engineering courses. Prediction error (NMSE) for the test samples, for the different values of the shrinkage factor  $\lambda$  in (2)**

## 7. DISCUSSION

Our goal was to illustrate the generalizability issue which we encounter in any machine learning or learning analytics setups. We did not compare multiple algorithms, but we used a simple one to support our claims. It is worth mentioning that the same phenomenon is encountered in any other machine learning method.

Moreover, the same analysis can be performed with any regularized regression algorithms, i.e., consisting a parameter to control the complexity of the model, like SVM, logistic regression, neural networks, etc. In each of these methods regularization selects the optimal sets of parameters.

Finally, the choice of the feature set should be based on the desired outcome of modelling student behaviour in a MOOC. If the goal is to attain high predictability in a small variety of courses, one could choose to include course-structure related variables. On the other hand, if the modelling requirement is to have a decent generalizability over a wide variety of courses, one has to compromise the predictability over a set of courses and select only the course-structure-independent variables.

## 8. REFERENCES

- [1] 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.
- [2] C. Brooks, C. Thompson, and S. Teasley. A time series

**Table 1: Results from the identical analysis done on all the other courses as shown in Figures 3. The courses with N/A in the second column had only one session. The errors reported are NMSE. The values in the perenthesis are the optimal shrinkage factors in given context.**

Course Name	Testing on the same course	Testing on other session	Testing on different course
Digital signal processing	0.76 (10)	0.99 (10)	0.35 (2010)
Geomatics	0.67 (10)	N/A	0.35 (2010)
House water treatment systems	0.58 (10)	N/A	0.48 (510)
Linear optimisation	0.67 (10)	N/A	0.36 (3010)
Mechanics	0.68 (10)	N/A	0.59 (1010)
Sanitation	0.80 (510)	N/A	0.73 (1010)
Structures	0.95 (10)	0.93 (2010)	0.84 (4510)
Micro-controllers	0.35 (2010)	0.35 (2010)	0.35 (2010)

- interaction analysis method for building predictive models of learners using log data. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 126–135. ACM, 2015.
- [3] J. Champaign, K. F. Colvin, A. Liu, C. Fredericks, D. Seaton, and D. E. Pritchard. Correlating skill and improvement in 2 moocs with a student’s time on tasks. In *Proceedings of the first ACM conference on Learning@Scale conference*, pages 11–20. ACM, 2014.
- [4] G. Christensen, A. Steinmetz, B. Alcorn, A. Bennett, D. Woods, and E. J. Emanuel. The mooc phenomenon: who takes massive open online courses and why? Available at SSRN 2350964, 2013.
- [5] J. DeBoer, G. S. Stump, D. Seaton, and L. Breslow. Diversity in mooc students? backgrounds and behaviors in relationship to performance in 6.002 x. In *Proceedings of the Sixth Learning International Networks Consortium Conference*, 2013.
- [6] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth. Unsupervised modeling for understanding mooc discussion forums: a learning analytics approach. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 146–150. ACM, 2015.
- [7] P. Hill. Emerging student patterns in moocs: A graphical view, 2013.
- [8] P. Hill. The four student archetypes emerging in moocs. *E-Literate*. March, 10:2013, 2013.
- [9] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O’ Dowd. Predicting mooc performance with week 1 behavior. In *Educational Data Mining 2014*, 2014.
- [10] 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.
- [11] E. J. Lauría, J. D. Baron, M. Deviredy, V. Sundararaju, and S. M. Jayaprakash. Mining academic data to improve college student retention: An open source perspective. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 139–142. ACM, 2012.
- [12] N. Li, L. Kidziński, P. Jermann, and P. Dillenbourg. Mooc video interaction patterns: What do they tell us? In *Design for Teaching and Learning in a Networked World*, pages 197–210. Springer International Publishing, 2015.
- [13] A. McAuley, B. Stewart, G. Siemens, and D. Cormier. The mooc model for digital practice. 2010.
- [14] C. Milligan, A. Littlejohn, and A. Margaryan. Patterns of engagement in connectivist moocs. *MERLOT Journal of Online Learning and Teaching*, 9(2), 2013.
- [15] W. C. Paredes and K. S. K. Chung. Modelling learning & performance: a social networks perspective. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 34–42. ACM, 2012.
- [16] T. Petty and A. Farinde. Investigating student engagement in an online mathematics course through windows into teaching and learning. *Journal of Online Learning and Teaching*, 9(2):261–270, 2013.
- [17] S. Rayyan, D. T. Seaton, J. Belcher, D. E. Pritchard, and I. Chuang. Participation and performance in 8.02 x electricity and magnetism: The first physics mooc from mitx. *arXiv preprint arXiv:1310.3173*, 2013.
- [18] D. Rosen, V. Miagkikh, and D. Suthers. Social and semantic network analysis of chat logs. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pages 134–139. ACM, 2011.
- [19] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- [20] J. Wilkowski, A. Deutsch, and D. M. Russell. Student skill and goal achievement in the mapping with google mooc. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 3–10. ACM, 2014.
- [21] A. Wolff, Z. Zdrahal, A. Nikolov, and M. Pantucek. Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 145–149. ACM, 2013.

# Closing the Loop with Quantitative Cognitive Task Analysis

Kenneth R. Koedinger  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15201  
koedinger@cmu.edu

Elizabeth A. McLaughlin  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15201  
mimim@cs.cmu.edu

## ABSTRACT

Many educational data mining studies have explored methods for discovering cognitive models and have emphasized improving prediction accuracy. Too few studies have “closed the loop” by applying discovered models toward improving instruction and testing whether proposed improvements achieve higher student outcomes. We claim that such application is more effective when interpretable, explanatory models are produced. One class of such models involves a matrix mapping hypothesized (and typically human labeled) latent knowledge components (KCs) to the instructional practice tasks that require them. An under-investigated assumption in these models is that both task difficulty and learning transfer are modeled and predicted by the same latent KCs. We provide evidence for this assumption. More specifically, we investigate the data-driven hypothesis that competence with Algebra story problems may be better enhanced not through story problem practice but through, apparently task irrelevant, practice with symbolic expressions. We present new data and analytics that extend a prior close-the-loop study to 711 middle school math students. The results provide evidence that *quantitative cognitive task analysis* can use data from task difficulty differences to aid discovery of cognitive models that include non-obvious or hidden skills. In turn, student learning and transfer can be improved by closing the loop through instructional design of novel tasks to practice those hidden skills.

## Keywords

Cognitive task analysis, cognitive model, transfer, knowledge components, close-the-loop experiment

## 1. INTRODUCTION

As the field of Educational Data Mining (EDM) strives for technical innovation, there is risk of losing the “E” in “EDM”, that is, of not making a clear link to the “Educational” in “Educational Data Mining”. Connected with this concern is

the temptation to evaluate EDM research only in terms of predictive accuracy and not place value on interpreting the resulting models for plausibility and generalizable insights. While it is possible to use uninterpretable or “black box” predictive models in educational applications (e.g., [1]), interpreting model results is an important step toward improving educational theory and practice for three reasons: 1) for advancing scientific understanding of learning or educational domain content, 2) for generalization of models to new data sets (cf., [19]), and 3) for gaining insights that lead to improved educational technology design.

Whether an educational application of EDM is through a black box model or mediated by data interpretation, the most important, rigorous, and firmly grounded evaluation of an EDM result is whether an educational system based on it produces better student learning. Such an evaluation has been referred to as “closing the loop” (e.g., [16]) as it completes a “4d cycle” of system **d**esign, **d**eployment, **d**ata analysis, and **d**iscovery leading back to design. The loop is closed through an experimental comparison of a system redesign with the original system design.

Use of the “close the loop” phrase, in our writing, goes back at least to [12]. Early examples of data-driven tutor designs, that is, of a close-the-loop experiment, can be found in [13] which tested a tutor redesign based on discoveries from data originally published in [17] and in [4], which was based on data analysis [5]. It is notable that a systematic process for going from data to system redesign was not articulated in this early work, but has been increasingly elaborated in more recent writings [especially 16].

This paper further specifies a particular class of analytic methods, namely *quantitative cognitive task analysis* methods, and how to use them to close the loop. The output of a cognitive task analysis (CTA) is a model of the underlying cognitive processing components (so-called knowledge components or KCs) that need to be learned to perform well in a task domain. Quantitative CTA uses data on task difficulty and task-to-task learning transfer to make inferences about underlying components.

### 1.1 Cognitive Task Analysis

In general, Cognitive Task Analysis (CTA) uses various empirical methods (e.g., interviews, think alouds, observations) to uncover and make explicit cognitive processes experts use and novices need to acquire to complete complex tasks [3]. Various representations of the resulting cognitive model (e.g., goal trees, task hierarchies, if-then procedure descriptions) are used to design or redesign

instruction. Close-the-loop experiments in different domains demonstrate that students learn more from instruction based on CTA than from previously existing instruction (e.g., medicine [23]; biology [8]; aviation [20]). These results come from CTAs using qualitative research methods that are costly and substantially subjective.

Quantitative CTA methods provide greater reliability and are less costly (though ideally used as a complement to qualitative CTA). An early close-the-loop study [13] based from a Difficulty Factors Assessment (DFA) showed that algebra students are better at solving beginning algebra story problems than matched equations. In a controlled random assignment experiment, the newly designed instructional strategy was shown to enhance student learning beyond the original tutor. Besides DFA, automated techniques can further reduce human effort and can be used on large data sets. An early example used learning curve analysis to identify hidden planning skills in geometry area [16] that resulted in tutor redesign. In a close-the-loop experiment comparing the original tutor to the redesigned tutor, students reached mastery in 25% less time and performed better on complex planning problems on the post-test. Further research [15] has shown how a search algorithm (e.g., Learning Factors Analysis) can generate better alternative cognitive models.

A key assumption behind DFA is that significant differences in task difficulty can be used to make non-obvious (sometimes counter-intuitive) inferences about underlying cognitive components and, in turn, these components help predict learning transfer and guide better instructional design. Similarly, statistical models of learning, including both logistic regression and Bayesian Knowledge Tracing variations, also tend to assume that both task difficulty and learning transfer can be predicted using the same KC matrix.

Recent work explored this connection [18] and found, across 8 datasets, that statistical models that use the *same* KC matrix to predict task difficulty *and* learning transfer produce better results than models that use *separate* matrices (item vs. KC). A key goal of this paper is to further investigate this difficulty-transfer linkage claim by extending evaluation of it through close-the-loop experimentation.

## 1.2 Illustrating Quantitative CTA

Consider the problems in Table 1 and try to answer the following question before reading on. Assuming the goal of instruction is to improve students' skill at translating story problems into algebraic symbols (e.g., translating the 2\_step story in the first column of Table 1 into "62+62-f"), which will yield better transfer of learning: practice on 1\_step story problems (columns 2 and 3) or practice on substitution problems (column 4)? Note that in the close-the-loop experiment we ran, similar multiple matched problem sets were created. A different problem set was used for practice than was used for transfer. For example, students who saw the 2-step problem in Table 1 as a transfer post-test item would not see the associated 1\_step or substitution problems from Table 1 as practice problems. So, again, which yields better transfer to 2\_step problems, practice on 1\_step or substitution?

If you answered that practice on the 1\_step story problems will better transfer to 2\_step story problems, you are in good company as learning commonalities underlying problem formats (i.e., deep features) is a known factor in aiding

analogy and transfer [9; 10]. But, the following quantitative analogy cognitive task analysis suggests a different answer.

**Table 1. Examples of problem variations and their solutions.**

2_step	1_step	1_step	substitution
Ms. Lindquist teaches 62 girls. Ms. Lindquist teaches <i>ffewer boys</i> than girls. Write an expression for how many students Ms. Lindquist teaches.	Ms. Lindquist teaches 62 girls. Ms. Lindquist teaches <i>b boys</i> . Write an expression for how many students Ms. Lindquist teaches.	Ms. Lindquist teaches 62 girls. Ms. Lindquist teaches <i>ffewer boys</i> than girls. Write an expression for how many boys Ms. Lindquist teaches.	Substitute 62-f for b in 62+b Write the resulting expression.
62+62-f	62+b	62-f	62+62-f

Using DFA, [11] explored the struggle beginning algebra students have with translating story problems into symbolic algebra expressions. A common belief is that story problems are hard due to comprehending the story content. However, two results indicate that comprehension is not a major roadblock. First, students are better able to solve 2\_step problems when given a value (e.g., answering 116 when f is given as 8 in the 2\_step story shown in Table 1) than when asked to write the symbolic expression (e.g., 62+62-f or even 62+62-8) [11]. Second, students do not do better when given explicit comprehension hints of the needed arithmetic operations than they do on 2\_step symbolization problems without hints [11]. If comprehension is not the key challenge, perhaps production of the target algebraic symbols is. Their results show students perform consistently better (62% vs. 40%) symbolizing both 1\_step problems (e.g., producing 62+b and 62-f for the 1\_step problems in Table 1) than on 2\_step problems (e.g., producing 62+62-f for the 2\_step story problem in Table 1).

These results suggest inferences about unobserved or "hidden skills" that are needed to translate 2\_step stories into symbolic expressions such as learning how to put one algebraic expression inside another (e.g., as the one- operator expression 40m is inside the two-operator expression 800- 40m). The results are consistent with a need for skills that extend the implicit grammar for generating expressions for 1\_step symbolization to recursive structures (e.g., "expression => expression operator quantity" and "expression => quantity operator expression"). Furthermore, they suggested that practicing non-contextual substitution problems (see last column of Table 1) should help students (implicitly) learn the desired recursive grammar structures and the corresponding production skills for constructing more complex expressions.

## 1.3 Analysis Methods

Our first analysis explores how much substitution practice transfers to story symbolization. We pursue this question with respect to broad outcomes and learning processes. This analysis replicates the high level analysis of the prior study (2008-09) [14] with a full dataset accumulated across four school years (2008-12). Our second analysis probes, more specifically, the question of the cognitive model link between task difficulty and learning transfer that underlies quantitative cognitive task analysis and, more generally, adaptive tutoring models like Bayesian Knowledge Tracing. Practically, the theoretical claim that learning transfer can be inferred from task difficulty data suggests that we can design instruction that produces better transfer of learning using models built from difficulty data (which is easier to collect).

Our third analysis examines whether statistical models of the learning process data support conclusions drawn from the outcome data. Does learning curve analysis indicate whether and how tasks (e.g., substitution problems) designed to isolate practice of CTA-discovered hidden skills (e.g., recursive grammar) transfer to complex tasks that theoretically require these skills (e.g., 2\_step story problems)?

## 2. METHOD

The original 2008-09 study [14] and current close-the-loop study were run with middle school students as part of a math course. In the original study, students were randomly assigned to either a substitution practice condition (N=149) or 1\_step story practice condition (N=154). Since then, additional data with random student assignment was collected over three school years from 2009-12 (N=234 for substitution practice, N=174 for 1\_step story practice) using the same problem set in ASSISTments. As previously described [14], the study involved a pre-test, instruction, and post-test. For the substitution condition, substitution problems were embedded as instruction interleaved with 2\_step story problems (posttest). For the 1\_step condition, 1\_step problems were used as instruction interleaved with the same 2\_step story problems. The pretest for a given version and order was the same for both conditions. Order was determined by difficulty of 2\_step problems from a pilot study and included a sequence of 2\_step problems from easy to hard or hard to easy.

Small changes were made to the automated scoring to give better feedback on unusual but arguably correct answers (e.g., d60 instead of 60d). For consistency in scoring, manual corrections made to the 0809 dataset [14] were combined with the corrections to the 0912 dataset and automatically applied to every answer in the combined dataset (0812).

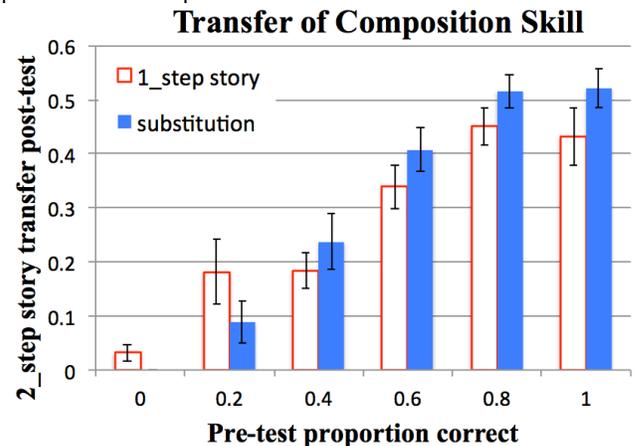
## 3. RESULTS AND DISCUSSION

### 3.1 High Level Transfer

In our first study [14], we reported significant main effects for condition and order while controlling for pretest, and no significant two-way or three way interaction effects when version was added as an independent variable. In the new study, we add a fifth factor for when the data was collected (i.e., from 0809 or from later years 0912). Most importantly, in a full five-factorial ANCOVA (in R with pretest as the covariate), we found a main effect for condition ( $F(1,679) = 4.5, p < 0.05, d = .21$ ). Main effects were also found for pre-test ( $F(1,679) = 235.3, p < 0.001$ ), order ( $F(1,679) = 117.8, p < 0.001$ ), and version ( $F(1,679) = 19.8, p < 0.001$ ), but study year was insignificant. Significant two-way interactions were found for pre-test and condition ( $F(1,679) = 4.05, p < 0.05$ ), pre-test and order ( $F(1,679) = 18.69, p < 0.001$ ), and order and year ( $F(1,679) = 10.77, p < 0.01$ ). No other higher-level interactions were significant (all  $p > 0.05$ ).

The pre-test by condition interaction is a consequence of the substitution treatment having a greater effect for students with higher pre-tests. Based on a median split of pre-test scores, students with a higher pre-test, showed greater benefits of substitution practice (52% posttest) over 1\_step practice (44%). In contrast, students with a lower pre-test show less benefit of substitution practice (24% posttest) over 1\_step practice (20%). This interaction is theoretically consistent with the cognitive task analysis in that students who cannot generate symbolizations for 1\_step problems (e.g., 800-y and 40x) will

not have the raw material they need to compose 2\_step expressions (e.g., 800-40x). Figure 1 illustrates the interaction. Substitution practice produces transfer to story problem symbolization for the 82% of students (580 of 711) with pre-tests of at least 40%. For the 18% of students without 1\_step story skills (below 40% on the pre-test), substitution practice does not provide a benefit.



**Figure 1. The benefit of substitution practice for symbolizing 2\_step story problems is present for the 82% of students with some incoming competence in 1\_step story symbolization (at least 40% correct).**

The two other reliable interactions in the ANCOVA are not of theoretical significance, but we report them for completeness. The pre-test by order interaction is manifest in that the difference between high and low pre-test students is bigger on the easier post-test problems (63% - 31% = 32%), which appear in the hard-to-easy order, than on the harder post-problems (38% - 10% = 28%), which appear in the easy-to-hard order. The order by year interaction is a consequence of students in the 0912 school years showing more sensitivity to the order manipulation than students in the 0809 school year, such that they do relatively better on the easy problems (46% vs. 41%), but worse on the hard problems (24% vs. 30%).

### 3.2 Difficulty Reliably Predicts Transfer

In this analysis, we more precisely test the following general logic: If difficulty data indicates a hidden skill that makes an important task hard, then inventing new practice tasks to isolate practice of that hidden skill will transfer to better learning of that hard task. The specific version of the logic in this domain is: If the hard part of symbolizing a two operator story problem is in composing symbolic expressions, then practice on substitution problems should transfer to better performance on story problem symbolization. Our data set affords an interesting opportunity to more precisely test this logic because the difficulty data we have indicates hidden skills for some problem types, but not others. A precise application of the “hidden-skill-transfer” logic stated above is that we should see the predicted transfer for those problem types in which the hidden skill is indicated by the difficulty data. For the other problem types, there should be no reliable transfer.

We used the current data to reevaluate the “composition effect” [11]. This analysis is shown in Table 2 where task difficulty and transfer results are shown for each of the eight problems. Consider the row for the *class* problem (referred

to as “students” in the data file), which is illustrated in Table 1. The answer for the 2\_step story and substitution problems, namely 62+62-f, is shown in the second column. The third and fourth columns show the proportion correct on the 1\_step story problems, (.75 for the “a” step with the answer 62+b and .70 for the “b” step with the answer 62-f). The fifth column (labeled a\*b) shows the probability of getting both of these steps correct, computed here as the product of the proportion correct on each step,  $.53 = .75 * .70$ . This value is the baseline for the composition effect.

The sixth column is the proportion correct on the 2\_step story problem, 0.13. This value was computed from student performance on the pre-test for both conditions and the post-test for the 1\_step practice condition. We did not use the post-test for the substitution practice condition to estimate the composition effect as the theory predicts that substitution practice should reduce that effect.

**Table 2. Composition effects are found for all but the bottom two problems**

Problem name	2_step solution	1_step (a)	1_step (b)	a*b	2_step	Composition Effect		Subst transfer
						a*b - 2_step	2_step/(a*b)	
trip	550/(h-2)	0.65	0.78	0.51	0.11	0.40	0.22	0.08
class	62+62-f	0.75	0.70	0.53	0.13	0.40	0.25	0.12
jackets	d-1/3*d	0.58	0.54	0.29	0.16	0.13	0.56	-0.02
sisters	(72-m)/4	0.71	0.63	0.45	0.32	0.13	0.72	0.15
rowboat	800-40m	0.75	0.55	0.38	0.28	0.10	0.73	0.07
children	(972+b)/5	0.66	0.75	0.5	0.38	0.12	0.76	0.09
cds	5*12*c	0.71	0.74	0.52	0.52	0.00	1.00	0.14
mcdonalds	5*h-7	0.66	0.85	0.56	0.72	-0.16	1.29	-0.06

mx+b) and the cds form 5\*12\*c involves a repetition of the same operator which can be treated as a 1-operator solution, namely, 60c (as 17% of students did). Students may have specialized knowledge for producing these forms that do not require general recursive grammar knowledge.

The final column (Subst transfer) shows how much substitution practice transferred to 2\_step symbolization as computed by the difference in post-test scores on each problem for the two experimental groups.

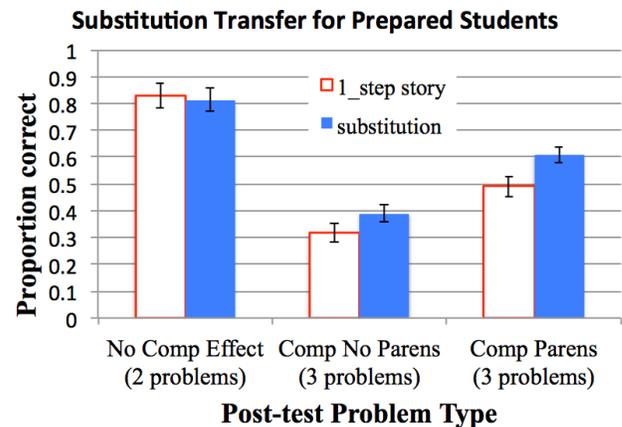
To test the hidden-skill-transfer hypothesis, we expect the cds and mcdonalds problems to show less transfer and the other problems to show more. While this is not strictly the case (cds shows transfer and jackets does not), there is a trend here that is illustrated in Figure 2. It shows the relationship between difficulty variation in the composition process and variation in the amount of transfer produced by substitution practice in the close-the-loop experiment. To better highlight the point, the graph shows the data from the 353 students at or above the median on the pre-test -- the ones for which improvement in composition skills should produce better post-test performance on 2\_step story problems requiring such skills.

Consistent with the hidden-skill-transfer hypothesis, there is no transfer benefit (first two bars in Figure 2) for the two problem forms with no composition effect (mcdonalds and cds). There is large transfer effect for the three problems (trip, sisters, and children) involving parentheses (last two bars), which present greater challenges for composing expressions and the need for

A composition effect is indicated when students are less likely to correctly symbolize a two operator story than to correctly symbolize both of the matched one operator stories. The seventh column displays this difference ( $.40 = .53 - .13$  for the class problem). The eighth column shows the estimated conditional probability that students can compose a single two-operator expression (e.g., 62+62-f) given they have correctly formulated the two source one-operator expressions (e.g., 62+b and 62-f). Since  $p(2\_step) = p(a*b) * p(2\_step | a*b)$ , we get  $p(2\_step | a*b) = p(2\_step)/p(a*b)$ , thus for the class problem  $p(2\_step | a*b) = .13/.53 = .25$ . The lower this value, the bigger the composition effect.

The important feature to note about values in the composition effect columns is that they indicate there is no composition effect for the cds and mcdonalds problems (see the last two rows). Both are relatively well-practiced forms, the 5h-7 for mcdonalds is a high frequency linear form (i.e.,

students to acquire more complex implicit grammar structures for generating correct parenthetic expressions. There is an immediate transfer effect for the three problems (class, jackets, and rowboat) not involving parentheses (middle bars), consistent with the fewer composition skills required. Note that success on these problems is oddly lower overall. We return to this point in the learning curve analysis where we do some search for new difficulty factors



**Figure 2. Transfer is limited to the problems that show a composition effect in task difficulty comparisons.**

and hypothesize a new hidden skill that could be pursued in future close-the-loop instructional design. These results add to prior evidence [18] supporting the hypothesis that differences in task difficulty and transfer effects are observable manifestations of the same underlying KCs.

### 3.3 Learning Curve Analysis

As a visual representation of student performance data over time (i.e., as opportunity increases, error rates are expected to decrease), learning curves can be used to explore areas of student difficulty and transfer of learning [21]. Following this prior work, we used the statistical model for learning curve prediction built into DataShop (see PSLCDataShop.org): The Additive Factors Model is a logistic regression model that generalizes Item Response Theory by having latent variables for knowledge component difficulty in place of item difficulty and by adding a third growth term, a knowledge component learning rate, in addition to the student proficiency and knowledge component difficulty terms. We evaluate four different knowledge component models in terms of their prediction fit to all of the test and instructional items each student experienced. For our metrics, we use root mean squared error (RMSE) averaged over 20 runs of 3-fold item-stratified and student-stratified cross validation. Given the focus on understanding the difficulty and transfer characteristics of the task environment, we put particular value on predictive generalization across items (as item stratification achieves by randomly putting all data on each item in the same fold) but also report the predictive generalization across students (as student stratification achieves by randomly putting all data on each student in the same fold).

The results of a learning curve analysis are shown in Table 3. The first row displays a simple baseline no-transfer model that treats each problem type (2\_step, 1\_step, and substitution) as requiring a different knowledge component (KC). The second row displays a substitution transfer model that introduces transfer between substitution problems and 2\_step problems by having a recursive grammar KC common to both problems. The 2\_step problems have an additional KC for comprehending the story and the 1\_step problems have a different unique KC. As shown in the last columns, this substitution transfer model produces a reduction in RMSE on the item stratified cross validation, down to 0.426 from 0.429. This small change is associated with a small change in the models and changes at this level (in the thousandths) have proven meaningful in producing a prior close-the-loop improvement [16]. This close-the-loop study provides further evidence that small prediction differences can be associated with significant learning gains.

Corresponding with the discussion above regarding the unique challenges of solutions requiring parentheses, the paren-enhanced model (third row in Table 3) adds a parenthesis KC to the 2\_step and substitution versions of the *trip*, *sisters*, and *children* problems. Surprisingly, this model does not improve the item generalization ( $0.428 > 0.426$ ), though it does improve student generalization ( $0.473 < 0.477$ ). The predictions of this model fail to account for the variance in difficulty of the non-parentheses problems.

As mentioned above, we were surprised that a couple of the non-parentheses problems posed great difficulty. In particular, the *class* (62+62-f) and *jackets* (d-1/3d) problems were quite hard (13% and 16% correct before substitution instruction). We hypothesized the difficulty of these problems was due to a quantity being referenced twice in the solution expression (i.e., 62 in the *class* problem and d in the *jackets* problem). To test this hypothesis we built the double-ref-enhanced model (fourth row in Table 3) by adding a double-ref KC to the paren-enhanced model on both of the 2\_step and substitution versions

of the *class* and *jackets* problems. The result is a substantially better prediction than the prior model on both item generalization ( $0.416 < 0.428$ ) and student generalization ( $0.468 < 0.473$ ).

**Table 3. Knowledge component learning curve model comparison.**

	KCs	Recursive grammar skill for 2_step & substitution	Paren skill	Double-ref skill	Item stratified CV (RMSE)	Student stratified CV (RMSE)
No-transfer	3	0	0	0	0.429	0.478
Substitution transfer	3	1	0	0	0.426	0.477
Paren-enhanced	4	1	1	0	0.428	0.473
Double-ref-enhanced	5	1	1	1	0.416	0.468

We have not yet modeled, but have recognized an alternative or additional explanation for the difficulty of the *class* and *jackets* problems. Right expanding forms, which require the “expression => quantity operator expression” rule, may be harder than left expanding forms, which require the “expression => expression operator quantity” rule. This idea garners plausibility from cognitive theory given that right expanding forms may require more cognitive load to maintain the subexpression to be written (e.g., 62-f) while the first part is planned and written (e.g., “62 +”). This analysis predicts that the *trip*, *class*, *jackets*, and *rowboat* problems should be more difficult and they are the most difficult 2\_step problems.

Future analytic and modeling efforts should pursue these plausible new hidden skills hypotheses and, if confirmed, a close-the-loop study should test whether focused instruction on double reference problems and/or more practice on right expanding expressions yields better learning transfer.

## 4. SUMMARY AND CONCLUSION

It is worth noting that the control condition in this study is highly similar to the treatment. Many might say, if you practice algebra, you learn algebra. Under that simple analysis, no differences should be expected between the conditions. Further, this control condition is a highly plausible instructional approach supported by a straightforward rational task analysis and by many colleagues who predict it should work: To prepare for story problems involving two operators, practice story problems involving one operator. The detailed data-driven quantitative cognitive task analysis suggested otherwise, in particular, that an inherent difficulty for algebra students learning to symbolize complex story problems is not in the story problem comprehension but in the production of more complex symbolic forms. Isolated practice in producing such forms, as the substitution problems provide, should enhance this hidden cognitive skill and yield better transfer. In a large data pool (711 students) collected in middle school math classes across four school years, our close-the-loop experiment demonstrated strong support for this data-driven prediction.

Our analysis also provides support for cognitive and statistical models that use the same underlying latent constructs (e.g., knowledge components) to predict both task difficulty and task-to-task transfer. This result is not only important to the science of learning, but it has practical relevance to the goal of using data-driven discoveries about domain learning challenges to design instruction for learning transfer. Task

difficulty data can be more easily collected than task-to-task transfer data. Ideal transfer data (i.e., comparing performance on task B when task A is or is not practiced before it) requires giving students curriculum sequences that may harm their learning, therefore, it is costly and ethically challenging. Task difficulty data, when appropriately modeled, provides promise that these cost and ethical challenges can be minimized.

Although this paper does not present new data mining methods, it does indicate that attempts to automatically discover cognitive models, such as LFA [2] and others like it (e.g., Rule Space [22], Knowledge Spaces [24], and matrix factorization [6; 7]) can be used to generate instructional designs that improve student learning and transfer. While innovation in data mining methods is a crucial part of EDM research, it is important to the health of the field and its relevance to society that we pursue more close-the-loop studies and keep the E in EDM!

## 5. ACKNOWLEDGEMENTS

This work was supported in part by IES award #R305C100024 and NSF award #SBE-0836012.

## 6. REFERENCES

- [1] 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., Beck, J. (2006). Adapting to When Students Game an Intelligent Tutoring System. In *Proc Int Conf Intelligent Tutoring Systems*, 392-401. Jhongli, Taiwan.
- [2] Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, T.-W. Chan (Eds.) *Proc 8th Int Conf ITS*, 164-175.
- [3] Clark, R.E., Feldon, D., van Merriënboer, J., Yates, K., & Early, S. (2007). Cognitive task analysis. In J.M. Spector, M.D. Merrill, J.J.G. van Merriënboer, & M.P. Driscoll (Eds.), *Handbook of research on educational communications and technology* (3rd ed., pp. 577–593).
- [4] Corbett, A.T. and Anderson, J.R. (1995). Knowledge decomposition and subgoal reification in the ACT programming tutor. In *Proc Artificial Intelligence and Education, 1995*. Charlottesville, VA: AACE.
- [5] Corbett, A.T., Anderson, J.R., Carver, V.H. and Brancolini, S.A. (1994). Individual differences and predictive validity in student modeling. In A. Ram & K. Eiselt (eds.) In *Proc Sixteenth Annual Conference of the Cog Sci Soc*.
- [6] Desmarais MC. (2011). Mapping question items to skills with non-negative matrix factorization. *SIGKDD Explor*, 13, 30–36.
- [7] Desmarais M.C. & Naceur, R. (2013). A matrix factorization method for mapping items to skills and for enhancing expert- based Q-matrices. In *Proc Artificial Intelligence and Education, 2013*. Memphis, TN, 441–450.
- [8] Feldon, D. F., Timmerman, B. C., Stowe, K. A., & Showman, R. (2010). Translating expertise into effective instruction: The impacts of cognitive task analysis (CTA) on lab report quality and student retention in the biological sciences. *J Research in Sci Teaching*, 47(10), 1165–1185.
- [9] Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy, *Cognitive Science*, 7, 155- 170.
- [10] Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15, 1-38.
- [11] Heffernan, N. & Koedinger, K. R. (1997). The composition effect in symbolizing: The role of symbol production vs. text comprehension. In Shafto, M. G. & Langley, P. (Eds.) *Proc of the 19<sup>th</sup> Annual Conf Cog Sci Soc*, (pp. 307-312).
- [12] Koedinger, K. R. (2002). Toward evidence for instructional design principles: Examples from Cognitive Tutor Math 6. In *Proceedings of PME-NA*, pp. 21-49.
- [13] Koedinger, K. R., & Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *Interactive Learning Environments*, 5, 161-180.
- [14] Koedinger, K.R. & McLaughlin, E.A. (2010). Seeing language learning inside the math: Cognitive analysis yields transfer. In S. Ohlsson & R. Catrambone (Eds.). *Proc 32nd Annual Conf Cog Sci Soc* (pp. 471-476.)
- [15] Koedinger, K. R., McLaughlin, E. A., & Stamper, J. C. (2012). Automated Student Model Improvement. In Yacef, K., Zaïane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (Eds.) *Proc 5th Int Conf on EDM*. (pp. 17-24)
- [16] Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013). Using data-driven discovery of better student models to improve student learning. In *Proc Int Conf on Artificial Intelligence in Education*, pp 421-430.
- [17] Koedinger, K.R., & Tabachneck, H.J.M. (1995). Verbal reasoning as a critical component in early algebra. Paper presented at the annual meeting of the *American Educational Research Association*, San Francisco, CA.
- [18] Koedinger, K. R., Yudelson, M., & Pavlik, P.I. (in press). Testing Theories of Transfer Using Error Rate Learning Curves. *Topics in Cognitive Science Special Issue*.
- [19] Liu, R., Koedinger, K. R., & McLaughlin, E. A. (2014). Interpreting Model Discovery and Testing Generalization to a New Dataset. In Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (eds.) *Proc 7th International Conference on Educational Data Mining* (pp.107-113).
- [20] Seamster, T.L., Redding, R.E., Cannon, J.R., Ryder, J.M., & Purcell, J.A. (1993). Cognitive task analysis of expertise in air traffic control. *Int J Aviat Psy*, 3, 257–283.
- [21] Stamper, J. & Koedinger, K.R. (2011). Human-machine student model discovery and improvement using data. In Biswas, G., Bull, S., Kay, J. & Mitrovic, A. (Eds) *Proc 15th Int Conf, AIED 2011* (pp.353-360).
- [22] Tatsuoka KK. (1983).Rule space: an approach for dealing with misconceptions based on item response theory. *J Educ Meas*, 20, 345–354.
- [23] Velmahos, G. C., Toutouzas, K. G., Sillin, L. F., Chan, L., Clark, R. E., Theodorou, D., & Maupin, F. (2004). Cognitive task analysis for teaching technical skills in an inanimate surgical skills laboratory. *The American Journal of Surgery*, 18, 114-119
- [24] Villano M. (1992). Probabilistic student models: Bayesian belief networks and knowledge space theory. *Proc 2nd Int Conf Intelligent Tutoring Systems*, NewYork: Springer-Verlag.

# Does a Peer Recommender Foster Students' Engagement in MOOCs?

Hugues Labarthe  
LAMOP  
University of Paris I  
France (75005)  
+33 649487203  
hugues.labarthe@ac-creteil.fr

François Bouchet  
Sorbonne Universités,  
UPMC Univ Paris 06  
CNRS, LIP6 UMR 7606  
75005 Paris, France  
+33 144277135  
francois.bouchet@lip6.fr

Rémi Bachelet  
Centrale Lille  
University of Lille  
Lille - France  
+33 320335466  
remi.bachelet@ec-lille.fr

Kalina Yacef  
School of Information  
Technologies  
The University of Sydney  
Australia  
+61 2 9351 6098  
kalina.yacef@sydney.edu.au

## ABSTRACT

Overall the social capital of MOOCs is under-exploited. For most students in MOOCs, autonomous learning often means learning alone. Students interested in adding a social dimension to their learning can browse discussion threads, join social medias and may message other students but usually in a blind and somehow random way, only hoping to find someone relevant, available and also willing to interact. This common isolation might be a contributing factor on student attrition rate and on their general learning experience. To foster learners' persistence in MOOCs, we propose to enhance the MOOC experience with a recommender which provides each student with an individual list of rich-potential contacts, created in real-time on the basis of their own profile and activities. This paper describes a controlled study conducted from Sept. to Nov. 2015 during a MOOC on Project Management. A recommender panel was integrated to the experimental users' interface and allowed them to manage contacts, send them an instant message or consult their profile. The population ( $N = 8,673$ ) was randomly split into two: a control group, without any recommendations, and an experimental group in which students could choose to activate and use the recommender. After having demonstrated that these populations were similar up to the activation of the recommender, we evaluate the effect of the recommender on the basis of four factors of learners' persistence: attendance, completion, success and participation. Results show the recommender improved all these 4 factors: students were much more likely to persist and engage in the MOOC if they received recommendations than if they did not.

## Keywords

Recommender system, MOOC, persistence, social learning.

## 1. INTRODUCTION

Understanding and reducing the attrition rate in Massive Open Online Courses is still a concern for many scientists, measuring and predicting attrition [2, 10], and trying to uncover its factors [6, 8]. There is a common assumption that students doing well by themselves are more likely to get involved in the learning community. But the paradox is that students do not necessarily know how to initiate and have meaningful conversations within this community, may feel shy or inhibited in such crowded places, which results in further isolation.

Therefore, while learning is above all a social undertaking [1], it turns out that most MOOCs students learn on their own. Far behind the connectivist model, transmissive MOOCs have been implementing functionalities such as synchronous or asynchronous discussions [4], peer grading, potential team mates' geolocation, groups, etc. In such systems, students find others to connect with either in a blind manner or through user-defined filters. Most importantly, contacts are initiated by the students themselves, who need to actively search for others. So it remains extremely difficult to find the right person to interact with in a newly-formed and distance learning MOOC community. This feeling of isolation hinders the learning experience and is a major factor of student attrition [7, 11]. Indeed, the size of students' cohorts and the fact that they usually work at home, at various times and pace, cultivates isolation rather than connection with other students for learning [5], a problem already well-noted before the MOOC era and which led to attempts to reinforce the sense of community [3, 9]. Numerous works have emphasized the need to help people socialising, on the basis that social learning might foster persistence. It requires not only helping students to know how to work with others (and thus to plan tasks for students to perform in a cooperative way), but also in the first place to find relevant potential learning mates one would want to interact with.

In this paper, we address this issue: to foster learners' persistence in MOOCs, we have designed, implemented and tested a recommender system. Our recommender provides each student with a list of high-potential social contacts, on the basis of their own profile and activities. We hypothesise that offering integrated personal data-driven recommendations may increase the students' persistence and success in the MOOC. We chose to consider four key categories of indicators of persistence: attendance, completion, scores and participation.

This paper is organized as follows: in section 2, we present the experiment with our peer recommender, its context and design, the different groups of students considered, the data collected and its preprocessing. In section 3, we analyse the differences in terms of persistence between the experimental groups, and in section 4, we check whether these differences are related to our recommender system. We then conclude the paper with a discussion on limits and on some perspectives of future work.

## 2. EXPERIMENT WITH A PEER RECOMMENDER

### 2.1 Context of the experiment

We built a peer recommender system and deployed it during the 6<sup>th</sup> session of a French Project Management MOOC<sup>1</sup>, powered by Unow<sup>2</sup> using a customised version of the Canvas platform [7]. The course lasted 9 weeks, from September to November 2015 and had a total of 24,980 students enrolled. Chronologically, it started with a 4 week long pre-MOOC period (week -3 to -1), where students could perform some self-assessment, introduce themselves on the discussion threads, explore the platform and so on. Then the 4 week-long core part of the MOOC (week 1 to 4 included) took place, with lecture videos, assignments, quizzes and so on. During the remaining 5 weeks (week 5 to 9), students followed their specialisation modules and took their final exam. In parallel to the main MOOC, students could additionally register to two possible streams: (i) an Advanced Certification stream where, in the first four weeks (1 to 4), learners also had to submit three assignments and perform peer-reviews; (ii) a Team track, where students also had to join a team and practice on a real project. The topic of the MOOC being Project Management, this MOOC assumes that learners, in addition to working individually and autonomously to obtain their certification, should also get involved as much as they can in the community. Figure 1 shows the overall MOOC timeline as well as the number of students who reached various checkpoints in the MOOC [e.g. 7716 students took quiz 1 between week 1 (release time) and week 9 (end of the MOOC)].

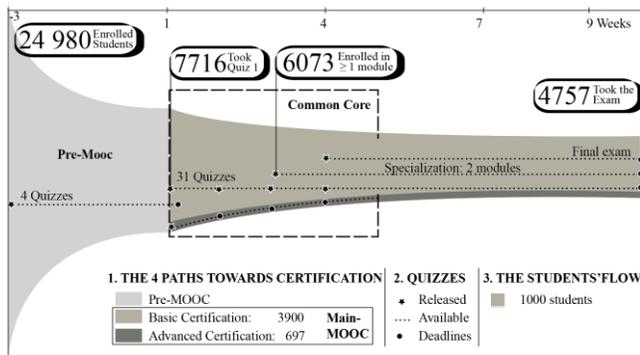


Figure 1. The 6<sup>th</sup> edition of the Project Management MOOC: a chronological overview

### 2.2 The peer recommender widget

The recommendation widget is displayed on the navigation bar on the left side of the screen in a space normally empty (cf. Figure 2). It displays 3 lists: a list of suggested contacts in green, a list of contacts marked as favorite in orange and a list of ignored contacts in grey (A). In each list, other students are represented as a thumbnail showing their name and photo (if any). When bringing the mouse pointer over a thumbnail, it also displays the beginning of their biography (if any) as well as 4 icons: one to send a private message, one to contact them through the chat, one to add them as a favorite and one to ignore them (B). The chat widget is shown on the bottom right-hand corner of the interface and minimised by default. When a message is received, an icon is added and a sound played (C). Bringing the mouse pointer over the widget expands it, giving access to two tabs: in the first tab, the favorite contacts appear and a chat can be initiated with up to 6 of them at the same

time. The second tab gives access to a list of previous chats, and one can reopen them to keep interacting with the student(s) associated to that chat (D).

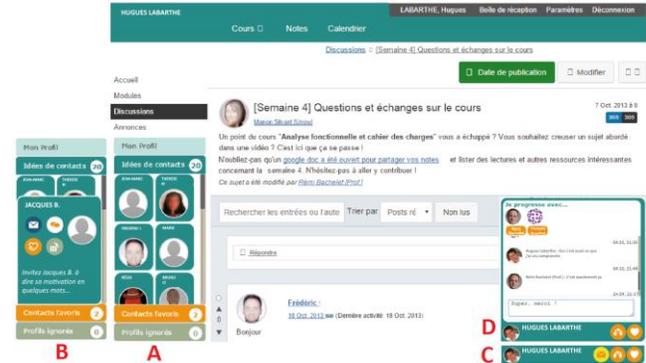


Figure 2. Recommendations and chat widgets

### 2.3 Experimental Design

In order to evaluate the effect of the recommender system (RS), we performed a controlled study. A set of experimental groups was offered access to the recommender whilst the control group (*Ctrl*) was not. Among the experimental groups, some students accepted the use of the recommender (*ToU*) and others did not. Then among those who accepted it, some interacted with it (*Int*) — i.e. managed contacts, consulted profiles and attempted to write messages— and others did not (*No\_Int*) — i.e. had the RS widget visible but did not interact at all with it (an interaction being defined as a click on the interface, as mouse-overs were not recorded). The experimental group was also split in three, each subgroup using a different recommendation algorithm (contact suggestions could be either random, based on social features only, or on a combination of social and advancement features). We shall not compare in this paper the efficiency of these algorithms but focus only on the RS' effect.

### 2.4 Deployment of the Recommender

The recommender was progressively deployed at the beginning of the 4-week core period (week 1 onwards): 100 students on day 1, 4,500 on day 5, 10,000 on day 10. Overall,  $N = 8673$  students visiting the platform during this period of time were randomly split between the control group ( $N_{Ctrl} = 1792$ ) and the experimental ones ( $N_{exp} = 6881$ ). The experimental group had roughly 3 times more students than the control one because of the aforementioned three subgroups, which will not be considered here. Among students in the experimental groups,  $N_{ToU} = 2025$  accepted the recommender Terms of Use (allowing data collection for research purpose) and thus had access to recommendations. Among those students,  $N_{Int} = 271$  interacted with the recommendations panel and the chat associated with it (i.e.  $N_{No\_Int} = N_{ToU} - N_{Int} = 1754$ ). Those figures are summarised on Figure 3.

### 2.5 Data Collection and Pre-processing

We extracted two types of data from the MOOC: learning traces as interaction logs, and demographic information coming from students' answers to a demographic questionnaire they could fill during the Pre-MOOC period, or as they started the MOOC for students arriving late on the platform.

One main way to understand how learners behave is by looking at the interaction logs and the learning records. Overall, 3.95 million

<sup>1</sup> MOOC Project Management, <http://mooc.gestiondeprojet.pm/>

<sup>2</sup> Unow, <http://www.unow.fr/>

pages were displayed from Sept. 1<sup>st</sup> to Nov. 22<sup>nd</sup> (week -3 to 9) for 373,937 different URLs. We classified them into semantic categories consisting of an action and an area of the website. The URLs combine references to 3 main actions: browsing, viewing content, and downloading resources. Students performed these actions on 12 areas as shown in Table 1. In total, students browsed pages with references to 357 different resources: 8.5% are the homepage, 8.3% lesson pages and 43% quizzes. Many students in developing countries download videos on a third-party website, so these figures should only be used to differentiate students' profiles.

We created 10 variables from this learning dataset to capture students' persistence in the MOOC, which could be grouped into four broad categories: attendance, completion, score and participation. These indicators are shown in Table 2.

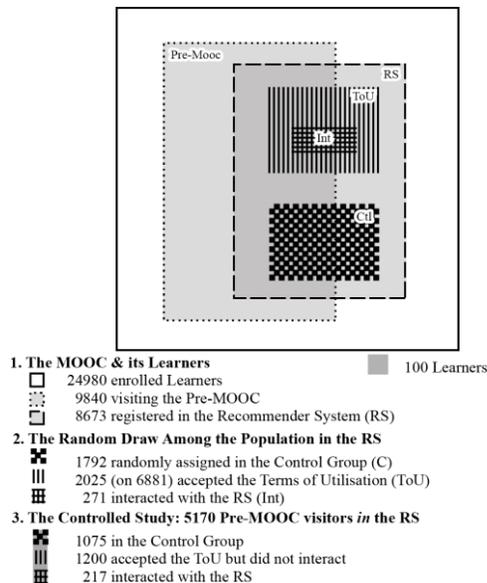


Figure 3. MOOC cohort sizes and overlaps (to scale)

Table 1. Tagging logs towards actions and areas

Categories • subcategories	Brow-sing	View-ing	Down-loading	Total (%)
[homepage]	336,941			8.5
Announcements	27,768			0.7
Assignments	6,602	68,591		1.9
• Syllabus	64,611			1.6
• Corrected assignments		77,270		2.0
• Peer-reviewing materials		59,865		1.5
• Downloaded assignments		69,510	23,606	2.4
Calendar		2,214		0.1
Discussions	35,763	119,777		3.9
Grades	42,961	27,655		1.8
Modules	489,325			12.4
• Badges		80,834		2.0
Others	440			0.0
Pages	7,761			0.2
• Lessons		327,882		8.3
• Other Contents		323,469		8.2
• Downloads	58,981			1.5
Quizzes	11,713	1,686,448		43.0
Profiles		2,678		0.1
TOTAL %	27.4	72.0	0.6	100

Finally, in addition to these learning related variables, we extracted the social features from one of three research surveys filled by participants before Nov. 11th. 10,331 learners completed this survey, from which 1,454 were enrolled in the control group and 5,397 in the experimental groups. 6 variables were considered: student's gender, country, year of birth, their level of study (coded as follow: 0, without A-Level; from 1 to 3: years of university course; 4: master degree; 5: PhD), the previous experiences of MOOCs (0 for newcomers, 2 for experienced with MOOCs; 4 for recurring Project Management MOOC students) and the participation to the Pre-MOOC (0 or 1).

Table 2. Retrieving data related to persistence

Category	Indicators
Attendance	1. Number of days the student visited the platform 2. Number of pages the student accessed 3. Time spent on these pages [max = 600 s]
Comple-tion	4. Number of attempts to complete a quiz 5. Number of quizzes completed
Scores	6. Final score [31 compulsory quizzes + exam]
Participa-tion	7. Number of posts on discussions (forums) 8. Average length of discussion posts 9. Number of messages sent via the Conversations (private messages) 10. Average length of private messages

## 2.6 Were groups similar before treatment?

In order to assess the similarity between the control group and the experimental ones before the experiment started, we compared their social and behavioral features (cf. Table 3). The data analysis indicates no significant differences between the two groups in terms of gender, countries, year of birth, level of study, previous MOOC experience and attendance on the platform. We can therefore consider the groups were similar before the experiment.

Table 3. Variation between Groups (ANOVA)

Features (number of values)	F	P-value
Gender (2)	0.573348	0.448958
Countries (91)	2.14E-06	0.998834
Year of Birth (59)	3.266974	0.070732
Level of Study (6)	1.195992	0.274163
Previous experiences of MOOCs (3)	0.009721	0.921462
Participation to the Pre-MOOC (2)	0.586452	0.443815

## 3. GROUP BEHAVIOUR ANALYSIS

Table 4 shows the comparison between 3 groups: the control group (Ct), and among the experimental one, the ones which accepted and did (resp. didn't) use the recommender (No\_Int - resp. Int). Figures show the students who experienced RS were those that displayed the strongest values for the 10 indicators of persistence considered. In particular, the average number of daily visits, pages viewed and duration increase from Ct to No\_Int and Int. The standard deviation increases too, revealing that the highest variation of behavior is observed among those who interacted with the RS. In terms of quizzes, the learners who experienced the RS completed 2 more quizzes than the others and scored on average 17 points higher with a smaller standard deviation. Finally, their participation in discussions and conversations are also higher. Reading these figures, it appears that students who experienced the recommender were also more engaged with the course and its community: even though the 271 students in the Int group did not spend so much time online overall, they have managed to obtain higher scores in terms of completion, quiz scores and participation.

However, the fact that students who used the recommender were also more engaged is not sufficient to express causality between the two. The uncertainty resides in the fact that in the experimental group, students could *choose whether or not* to have a recommender widget, and *whether or not* to actually make use of it. It could be the case that, in fact, students who are very engaged are more likely to use the recommender.

**Table 4. Average and standard deviation (in italics) of persistence indicators for experimental versus control groups**

Indicators	Attendance from W1 to W4			Completion Nov. 22 <sup>nd</sup>		Scores /100	Participation from W4 to W9			
	1	2	3	4	5	6	7	8	9	10
Ctl N=1792	10 7	323 285	1h38 1h57	26.4 22.5	20 14	32.2 28.7	0.7 3.2	69 137	0.3 2.1	31 127
No_Int N=1754	12 7.5	411 373	2h08 2h23	30.5 24	21.6 13.3	36.1 30.1	1.4 5.6	111 190	0.6 2.1	52 177
Int N=271	16.1 6.9	616 405	3h46 3h07	43.2 24.7	26.9 10	49.1 27.8	2.7 6.1	154 186	1.6 3.8	107 212

## 4. EFFECTS OF THE RECOMMENDER

To determine the RS' real effect on learners' persistence, we need to compare cohorts that were similar in terms of persistence before the experiment started and see how they evolve during the course of the MOOC. For example, we want to find out whether, among students who were very passive before the recommender was made available, a larger proportion of those who used the recommender persisted in the MOOC. To do so, we first clustered students during the Pre-MOOC period (i.e. before they were allocated to a group, and before the RS was made available) based on their level of engagement (section 4.1). We then, in each cluster, analysed the control and experimental groups according to each dimension of persistence at the end of the main MOOC period.

### 4.1 Pre-MOOC activity clusters

To cluster students in the Pre-MOOC period, we used as features the times spent on 18 of the actions in areas shown in Table 1 (i.e. excluding those related to material not yet available). During the Pre-MOOC, 294,209 pages were accessed by the 9,840 students who were present in the Pre-MOOC period. We used the k-means algorithm to extract clusters and found the best solution involved 4 groups, shown in Table 5 and called A, B, C D on the basis of their time spent (A being the most active and D the least). Students in cluster A spent over 1h40 on the website viewing lessons, quizzes and discussions (sum of the mean values). The second cluster (B) spent less than 40 minutes, essentially in the quizzes area; in the third cluster, C, the time is even shorter and those in the last one, D, stayed less than 2 min on the website in total.

Table 6 shows the distribution across the 4 Pre-MOOC clusters of students who would later belong to groups *Ctl*, *No\_Int* and *Int*. Since we want to follow the evolution of the students who were present in the Pre-MOOC period, we must only consider the intersecting population. The populations of the various groups are now:  $N_{Pre\&Int} = 217$  students who interacted with the recommender (vs.  $N_{int} = 271$ );  $N_{Pre\&No\_Int} = 1,200$  (vs.  $N_{No\_Int} = 1,754$ ) who accepted its ToU without using it;  $N_{Pre\&Ctl} = 1,075$  (vs.  $N_{Ctl} = 1,792$ ) who were randomly enrolled in the control group.

To deal with the sample size difference and compare the features of students in *Int* with students in *Ctl* and *No\_Int*, a subsample was ten times randomly drawn for each cluster – e.g. in the PreMOOC\_D cluster, 77 persons out of 551 were ten times randomly drawn. The percentage averages in tables 8, 10 and 12

are computed only on the basis of features of students from these subsamples. We will now exclusively focus on the last 3 Pre-MOOC clusters since the most active group (PreMOOC\_A) is very small (8) and already very engaged.

**Table 5. Interactions and clusters during the Pre-MOOC**

Features (in seconds)	PreMoo	PreMoo	PreMoo	PreMoo
	c _D	c _C	c _B	c _A
browsing_homepage	21	48	149	411
browsing_announcements	1	4	15	81
browsing_assignment	4	14	48	210
browsing_discuss_topics	2	8	26	190
browsing_grades	1	3	11	30
browsing_modules	7	43	140	428
browsing_pages	0	1	6	8
browsing_quizzes	0	1	2	2
downloading_assignment	0	0	0	2
viewing_assignment	1	11	49	208
viewing_calendar_events	0	0	0	7
viewing_discuss_topics	13	82	226	857
viewing_grades	0	0	1	1
viewing_modules	0	7	24	65
viewing_pages	25	163	550	1472
viewing_profiles	0	1	2	37
viewing_quizzes	33	768	1167	1965

**Table 6. Clusters and Groups during the Pre-MOOC**

	N (%)	N	Ctl	No_Int	Int
PreMoo_c_D	66	6,386	551	578	77
PreMoo_c_C	26	2,534	393	404	78
PreMoo_c_B	7	658	118	190	54
PreMoo_c_A	1	62	13	28	8
Total	100	9,640	1,075	1,200	217

### 4.2 Attendance during the Common Core

We clustered all enrolled students ( $N=24,980$ ) using the full set of features in Table 4 for a total of 3,110,321 pages seen during the Common Core. We obtained 4 clusters, shown in Table 7, named according to their attendance quality (A the best, D the worst). Cluster Att\_D, with 77% students, has the poorest overall mean in regards to all the features, not exceeding 6 minutes spent interacting with all pages. The mean values of the second cluster, Att\_C (with 17% students), total around 1h30min. The two last clusters, Att\_B and Att\_A, contain 3% each of the population: the main difference is the time spent by Att\_A in the assignments area.

We then explored how the pre-MOOC students evolve into these attendance clusters, according to their activities during the Common Core (cf. Table 8, where figures in a row represent 100% of the mentioned *Ctl*, *No\_Int* and *Int*). Considering the lower clusters D to B, these figures suggest that the recommender system played a significant role on the duration of the visits of the learners from clusters D, C and B, that is to say 99% of the Pre-MOOC population. Indeed, one can see that students who used to be in D, having the RS marginally increased their persistence, but significantly increased the persistence of students who used it (32% of them now being in cluster B vs. 8% for students of the control group). For students in clusters C and B during the pre-MOOC, we observe a similar pattern: simply having access to the RS tended to increase their persistence, and actually using the RS tended to significantly decrease their chance of dropping out (i.e. ending up in cluster D, the least active students).

**Table 7. Interactions and clusters during the Common Core**

Features (in seconds)	At_D	At_C	At_B	At_A
_others	0	0	1	2
browsing_	15	214	554	856
browsing_announcements	1	12	53	61
browsing_assignments	5	48	181	155
browsing_discussion_topics	2	23	90	315
browsing_grades	1	32	160	276
browsing_modules	22	430	1022	1249
browsing_pages	0	4	4	6
browsing_quizzes	0	7	7	6
downloading_assignments	0	3	5	144
viewing_assignments	7	248	636	9334
viewing_calendar_events	0	1	11	5
viewing_discussion_topics	14	127	467	1477
viewing_grades	0	10	48	216
viewing_modules	3	57	169	177
viewing_pages	67	1025	2766	2398
viewing_profiles	0	1	4	18
viewing_quizzes	180	3257	8286	5165
% students	77	17	3	3

**Table 8. Attendance: Evolution of the learners from the Pre-MOOC to the Core-MOOC periods**

↓From To→	At_D	At_C	At_B	At_A	Group
PreMooc_D 66%	39	49	8	4	Ctl
	33	49	12	7	No_Int
	9	39	32	19	Int
PreMooc_C 26%	26	50	9	16	Ctl
	24	43	12	20	No_Int
	17	45	12	27	Int
PreMooc_B 7%	16	48	12	24	Ctl
	16	38	15	31	No_Int
	2	37	20	41	Int

### 4.3 Completion and final scores

We clustered again the student population, using scores and activity in the examination points (i.e. scores obtained at the 31 quizzes and the final exam by the end of the MOOC). Each score is standardised to marks out of 100. We obtained again 4 clusters, which centroids are shown in Table 9. The values of the centroid of the first cluster indicates a large part of students (71%) who participated in the first 2 quizzes but obtained a very low score on them and then did not participate again in any assessment. The centroid of the second cluster (4% of learners) corresponds to students who easily passed the quizzes of the first week but dropped out on the second. The third cluster (4%) has similar students, but who gave up in week 3. Finally, the last cluster (21%) contains all the students who completed all the quizzes and final exam with high scores in each.

Once again figures in Table 10 show that, by accepting the recommendations and, even more, interacting with its panel, the learners went closer to completion and obtained better scores. In particular, we observe as before for students in clusters D and B that the mere presence of the RS has a small positive impact on their chances to complete (or at least to stay longer on the MOOC before giving up), but that students who use the RS benefit the most from an increased chance to complete. For students in cluster C, the use of the RS seems to have made some of them drop out overall a bit later (week 2 instead of week 1) but did not increase their chance to complete the MOOC.

**Table 9. Completion and score clusters during whole MOOC**

Week	Quiz	D	C	B	A	Week	Quiz	D	C	B	A
1	1	3	92	92	96	2	17	0	1	67	92
	2	1	82	82	87		18	0	0	48	83
	3	0	92	92	96		19	0	1	57	95
	4	0	82	89	95	3	20	0	1	39	92
	5	0	76	93	98		21	0	1	40	96
	6	0	54	78	87		22	0	1	36	95
	7	0	63	92	98		23	0	1	33	91
2	8	0	26	93	96	4	24	0	1	31	94
	9	0	18	94	97		25	0	1	29	89
	10	0	10	92	95		26	0	1	10	91
	11	0	7	88	93		27	0	1	5	93
	12	0	4	85	93		28	0	0	2	90
	13	0	2	83	93		29	0	1	1	96
	14	0	2	86	95	30	0	0	1	95	
	15	0	1	76	89	31	0	0	1	86	
	16	0	1	75	93	EXAM	0	1	3	78	
N (%)		71	4	4	21	N (%)		71	4	4	21

**Table 10. Completion and final scores: Evolution of the learners from the Pre-MOOC to the Core-MOOC periods**

↓From To→	Co_D	Co_C	Co_B	Co_A	Group
PreMooc_D 66%	32	5	13	49	Ctl
	27	6	14	53	No_Int
	10	5	4	81	Int
PreMooc_C 26%	15	9	11	65	Ctl
	9	9	14	69	No_Int
	8	14	13	65	Int
PreMooc_B 7%	8	5	8	79	Ctl
	5	9	14	73	No_Int
	4	2	11	83	Int

### 4.4 Participation to the Common Core

The total number and average length of the messages sent by each student were retrieved from the Canvas database (discussions and conversations). Using k-means with features from the participation section of Table 2, we obtained once again 4 clusters, shown in Table 11: a first cluster, Pa-D (89% of 24,980 enrolled learners) did not interact at all with others. The centroid of the second one indicates 2 posts of an average of 237 characters on the discussion topics (9%). The third cluster (2%) seems to have a similar activity but slightly stronger in term of number of posts (2.7) and average post length (599 characters). The last 1% is highly committed to the course and its community: most of them correspond to students who were part of the advanced certification stream.

Table 12 shows how students in the Pre-MOOC clusters are distributed over the 4 participation clusters at the end of the MOOC. Figures reveal a consistent positive effect of the mere presence of the RS across the initial Pre-MOOC clusters: there are always less students in cluster Pa\_D in the *No\_Int* group than in the control group. Less surprisingly, students who interacted with the RS generally did so to send a message to someone, so they overall also ended up less often being in a situation where they do not interact at all with anyone else (complete isolation). Finally, we can see that merely giving students access to a recommender panel does not prevent them from being social-lazy: a majority (82%, 88% 69% respectively in clusters D, C and B) of the students who interacted with the RS did not attempt to directly contact anyone else. These figures are however probably lower than they would be if every student had access to the associated direct chat module, and still better than in the Control group (96%, 91% and 80% respectively

in clusters D, C and B) who could only contact others in a blind way through the forum or private messages.

**Table 11. Participation Clusters of all enrolled students**

Attribute	Pa-D	Pa-C	Pa-B	Pa-A
Nb** of discussions	0	2	2	9
Discussions length*	2	237	599	264
Nb** of conversations	0	0	0	7
Conversations length*	1	9	19	542
N%	89	9	2	1

\*: average number of characters; \*\*: number of posts/messages sent

**Table 12. Participation: Evolution of the learners from the Pre-MOOC to the Core-MOOC periods**

↓From To→	Pa_D	Pa_C	Pa_B	Pa_A	Group
PreMooc_D 66%	78	18	2	2	Ctl
	67	25	4	4	No_Int
	47	35	6	12	Int
PreMooc_C 26%	76	15	4	5	Ctl
	69	18	4	9	No_Int
	62	26	4	9	Int
PreMooc_B 7%	66	14	4	15	Ctl
	53	25	6	15	No_Int
	39	30	7	24	Int

## 5. Discussion, conclusion and perspectives

We conducted a controlled study during a Project Management MOOC, in which a recommender panel integrated to the user interface provided suggestions and allowed contact management, instant messaging and profile consultation. Students were randomly split into a control group (without any recommendations), and an experimental group (in which they could activate and use our recommender). The number of the students involved in this experience was relatively high: among 6881 selected students, 2025 accepted the Term of Use of the recommender and 279 accessed its functionalities. We have shown that these populations were similar before the activation of the recommender, and evaluated its effect according to four categories of indicators relative to learners' persistence: attendance, completion, success and participation. Results suggested that our recommender improved these four categories of indicators: students are much more likely to persist and engage in the MOOC if they receive recommendations than if they do not.

The main interest was then to evaluate the effect the recommendations might have played in such increased rates of engagement. To do so, we focused on clustering similar learners according to their activities before the beginning of the course, leading to four groups from the least (D) to the most (A) active students. We analysed the way 3 of these 4 groups (representing 99% of the students) were evolving in terms of attendance, completion and score, participation. We observed overall a significant improvement of students' engagement, not only for those who interacted with the recommendations, but, more largely, for all of those accepted using the recommendation system.

This study presented several limitations: (1) for experimental purposes, we restricted the access to the direct communication tool; (2) since not all students had access to the RS and the chat, the teaching team could not use them for pedagogical activities, which could have boosted the effect of the RS; (3) students in the control group were not asked to accept the RS Terms of Use, since they would not be given access to it – however, while it is thus possible that students who accepted the ToU were more motivated, the analysis presented in section 2.6 shows that students in the control

and experimental groups were similar in terms of participation before the beginning of the core MOOC and demographics.. Furthermore, the most significant results were obtained comparing students who interacted vs. those who did not interact with the RS, and these results are not affected.

Overall, this controlled study is highly supporting the idea that recommending learners to learners, in such crowded places as MOOC platforms, is an effective way to get them more involved in terms of attendance, completion, scores and participation. In the future, we intend to look into more details the impact of the different recommendation strategies, and the different ways students interacted with the recommendation system.

## 6. ACKNOWLEDGMENTS

This work was funded by the French Educational Board and by the Human-Centred Technology Cluster of the University of Sydney. We thank Unow for letting us deploy our RS on their MOOC.

## 7. REFERENCES

- [1] Bandura, A. 1971. *Social Learning Theory*. General Learning Corporation.
- [2] Bouchet, F. and Bachelet, R. 2015. Do MOOC students come back for more? Recurring Students in the GdP MOOC. *Proc. of the European MOOCs Stakeholders Summit 2015* (Mons, Belgium), 174–182.
- [3] Croft, N., Dalton, A. and Grant, M. 2010. Overcoming Isolation in Distance Learning: Building a Learning Community through Time and Space. *Journal for Education in the Built Environment*. 5, 1, 27–64.
- [4] Ferschke, O., Yang, D., Tomar, G. and Rosé, C.P. 2015. Positive Impact of Collaborative Chat Participation in an edX MOOC. *Proc. of Artificial Intelligence in Education: 17th International Conference, AIED 2015, Madrid, Spain, June, 2015*. Springer. 115–124.
- [5] Gütl, C., Rizzardini, R.H., Chang, V. and Morales, M. 2014. Attrition in MOOC: Lessons Learned from Drop-Out Students. *Proc. of Learning Technology for Education in Cloud. MOOC and Big Data*. Springer. 37–48.
- [6] Kizilcec, R.F., Piech, C. and Schneider, E. 2013. Deconstructing Disengagement: Analyzing Learner Subpopulations in Massive Open Online Courses. *Proc. of the Third International Conference on Learning Analytics and Knowledge* (New York, NY, USA), 170–179.
- [7] Labarthe, H., Bachelet, R., Bouchet, F. and Yacef, K. 2016. Towards increasing completion rates through social interactions with a recommending system. *Proc. of the European MOOCs Stakeholders Summit 2016* (Graz, Austria), 471-480.
- [8] Rosé, C.P., Carlson, R., Yang, D., Wen, M., Resnick, L., Goldman, P. and Sherer, J. 2014. Social Factors That Contribute to Attrition in MOOCs. *Proc. of the First ACM Conference on Learning @ Scale Conference* (New York, NY), 197–198.
- [9] Rovai, A.P. 2002. Building Sense of Community at a Distance. *The International Review of Research in Open and Distributed Learning*. 3, 1.
- [10] Yang, D., Sinha, T., Adamson, D. and Rosé, C.P. 2013. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. *Proc. of the NIPS Data-Driven Education Workshop*.
- [11] Yang, D., Wen, M. and Rosé, C.P. 2014. Peer Influence on Attrition in Massive Open Online Courses. *Proc. of the 7th International Conference on Educational Data Mining* (London, UK), 405–406.

# A Contextual Bandits Framework for Personalized Learning Action Selection

Andrew S. Lan  
Rice University  
mr.lan@sparfa.com

Richard G. Baraniuk  
Rice University  
richb@sparfa.com

## ABSTRACT

Recent developments in machine learning have the potential to revolutionize education by providing an optimized, personalized learning experience for each student. We study the problem of selecting the best personalized learning action that each student should take next given their learning history; possible actions could include reading a textbook section, watching a lecture video, interacting with a simulation or lab, solving a practice question, and so on. We first estimate each student’s knowledge profile from their binary-valued graded responses to questions in their previous assessments using the SPARFA framework. We then employ these knowledge profiles as contexts in the contextual (multi-armed) bandits framework to learn a policy that selects the personalized learning actions that maximize each student’s immediate success, i.e., their performance on their next assessment. We develop two algorithms for personalized learning action selection. While one is mainly of theoretical interest, we experimentally validate the other using a real-world educational dataset. Our experimental results demonstrate that our approach achieves superior or comparable performance as compared to existing algorithms in terms of maximizing the students’ immediate success.

## 1. INTRODUCTION

In traditional classrooms, learning has largely remained a “one-size-fits-all” experience in which the instructor selects a single learning action for all students in their class, regardless of their diversity in backgrounds, learning goals, and abilities. The quest for a fully personalized learning experience began with the development of intelligent tutoring systems (ITSs) [6, 19, 38, 40]. However, to date, ITSs are primarily *rules-based*, meaning that building an ITS requires domain experts to consider every possible learning scenario that students can encounter and then manually specify the corresponding learning actions in each case. This approach is not scalable, since it is both labor-intensive and domain-specific.

Machine learning-based personalized learning systems [30] have shown great promise in reaching beyond ITS to scale to large numbers of subjects and students. These systems automatically create *personalized learning schedules*, a series of *personalized learning actions* (PLAs) for each individual student to take that maximizes their learning. Examples of PLAs include reading a textbook section, watching a lecture video, interacting with a simulation or lab, solving a practice question, etc. Instead of domain-specific rules, machine learning algorithms are used to select PLAs automatically by

analyzing the data students generate as they interact with learning resources.

The general problem of creating a fully personalized learning schedule for each student can be formulated using the partially observed Markov decision process (POMDP) framework [31]. POMDPs utilize models on the students’ latent knowledge states [23, 28] and their transitions [8, 11, 18, 22] to learn a PLA selection policy (a mapping from the knowledge state space to the set of learning actions) that maximizes a reward received in the possibly distant future (long-term learning outcome). Previous work applying POMDPs to personalized learning have achieved some degree of success [4, 9, 32, 33]. However, learning a personalized learning schedule using a POMDP is greatly complicated by the curse of dimensionality; the solution quickly becomes intractable as the dimensions of the state and action spaces grow [31]. Consequently, POMDPs have made only a limited impact in large-scale personalized learning applications involving large numbers of students and learning actions.

A more scalable approach to personalized learning is to learn a PLA selection policy using the *multi-armed bandits* (MAB) framework [10, 27], which is more suitable to optimizing students’ success on immediate follow-up assessments (short-term learning outcome). The simplicity of the MAB framework makes it more practical than the POMDP framework in real-world educational applications, since it requires far less training data.

### 1.1 Contributions

In this paper, we study the problem of selecting PLAs for each student given their learning history using MABs. We first estimate each student’s latent concept knowledge profile from their learning history (specifically, their binary-valued graded responses to questions in previous assessments) using the sparse factor analysis (SPARFA) framework [23]. Then, we use these concept knowledge profiles as contexts in the contextual (multi-armed) bandits framework to learn a policy to select PLAs for each student that maximize their performance on the follow-up assessment.

We develop two algorithms for PLA selection. The first algorithm, CLUB, has theoretical guarantees on its ability to identify the optimal PLA for each student. The second algorithm, A-CLUB, is more intuitive and practical; we experimentally validate its performance using a real-world educational dataset. Our experimental results demonstrate

that A-CLUB achieves superior or comparable performance to existing algorithms in terms of maximizing students’ immediate success.

## 1.2 Related work

The work in [27] applies an MAB algorithm to educational games in order to trade off scientific discovery (learning about the effect of each learning resource) and student learning. Their approach is context-free and thus not ideally suited for applications with significant variation among the knowledge states of individual students. Indeed, it can be seen as a special case of our work in this paper when there is no context information available.

The work in [36] applies a contextual bandits algorithm to the problem of selecting the optimal PLA for each student given their previous exposure to learning resources. In their approach, each dimension of the context vector corresponds to the students’ exposure to one learning resource. Thus, the context space quickly grows large as the number of learning resources increases. Our approach, in contrast, performs dimensionality reduction on student learning histories using the SPARFA framework and uses the resulting student concept knowledge profiles as contexts. This feature enables our approach to be applied to datasets where student learning histories contain a large number of learning resources.

The work in [29] collects high-dimensional student–computer interaction features as they play an educational game and uses them to search for a good teaching policy. We emphasize that our approach can be applied to almost all educational applications, not just computerized educational games, since it only requires graded response data of some kind.

The works in [10] and [20] both use some form of expert knowledge to learn a teaching policy. The approach of [10], in particular, uses expert knowledge to narrow down the set of possible PLAs a student can take. Our approach, in contrast, requires no expert knowledge and is therefore fully data-driven and domain-agnostic.

The work in [26] fuses MAB algorithms with Gaussian process regression in order to reduce the amount of training data required to search for a good teaching policy. Their work requires the policy to be parameterized by a few parameters, while our framework does not and can thus learn more complicated policies using only reward observations.

The work in [35] found that various student response models, including knowledge tracing (KT) [11], IRT models [28, 34, 5], additive factor models (AFM) [8], and performance factor models (PFM) [16] can have similar predictive performance yet lead to very different teaching policies. While these results are indeed interesting, we emphasize that the focus of the current work is to develop policy learning algorithms rather than comparing student models.

## 2. PROBLEM FORMULATION

We study the problem of creating a personalized learning schedule for each student by selecting the PLA they should take based on their prior learning experience. We assume that a student’s learning schedule consists of a series of assessments with PLAs embedded in between, a setting that is

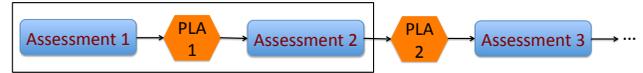


Figure 1: A personalized learning schedule.

typical in traditional classrooms, blended learning environments, and online courses like MOOCs [12, 13]. Each PLA can correspond to studying a learning resource, e.g., reading a textbook section, watching a lecture video, conducting an interactive simulation, solving a practice question, etc., or a combination of several learning resources.<sup>1</sup> Assessment could be a pop-quiz with a single question, a homework set with multiple questions, or a longer exam. Each student’s personalized learning schedule can be visualized as in Figure 1, where a PLA is taken between consecutive assessments (starting after Assessment 1).

*The goal of this work is to select the optimal PLA for each student given their learning history (their graded responses to previous assessments) that maximizes their immediate success, i.e., the credit they receive on the following assessment.* We aim to learn this learning action selection rule from data. For simplicity of exposition, we will place PLA 1 between Assessment 1 and Assessment 2 (as encased in the box in Figure 1) as a running example throughout the paper.

Let  $A$  denote the total number of PLAs available, let  $K$  denote the number of latent concepts covered up to Assessment 1, and let  $Q$  denote the number of questions in Assessment 2, with  $s_i, i = 1, \dots, Q$  the maximum credit of each question. Let  $Y_{i,j}$  denote the binary-valued graded response of student  $j$  to question  $i$ , with  $Y_{i,j} = 1$  denoting a correct response and  $Y_{i,j} = 0$  an incorrect response. In order to pin down a feasible PLA selection algorithm, we make two simplifying assumptions: i) We assume that a reliable estimate of each student’s latent concept knowledge vector (estimated from their graded responses to Assessment 1), denoted by  $\mathbf{c}_j \in \mathbb{R}^K$ , is available to the PLA selection algorithm. Such an estimate can be obtained using any IRT-like method, e.g., SPARFA [23]. ii) We assume that the PLA selected for each student will directly affect their performance on Assessment 2.

With this notation in place, we can restate our goal as selecting a PLA for student  $j$ , given their current concept knowledge<sup>2</sup>  $\mathbf{c}_j$  in order to maximize their performance (i.e., their expected credit  $\sum_{i=1}^Q s_i \mathbb{E}[Y_{i,j}]$ ) on Assessment 2.

### 2.1 Background on bandits

The multi-armed bandit (MAB) framework [3] studies the problem of a player trying to learn a policy that maximizes the total expected reward by playing (pulling the arms of) a collection of slot machines with a fixed number of trials and no prior information about each machine. Each machine has a fixed reward distribution that is unknown to the player. The key to maximizing the total expected reward is to find the right balance between exploration (playing

<sup>1</sup>Our notion of PLA is very general, and we do not restrict ourselves to studying a single learning resource.

<sup>2</sup>In practice, we augment  $\mathbf{c}_j$  as  $[\mathbf{c}_j^T \mathbf{1}]^T$  to add an “offset” parameter to each arm.

machines that might yield high rewards) and exploitation (repeatedly playing the machine with the highest observed reward). Analogously, a personalized learning system must strike a balance between testing the efficacy of every learning action (exploration) and maximizing the students' learning outcomes using observations on the actions (exploitation) [27].

*Contextual (multi-armed) bandits* [1, 2, 15, 24, 37] extend the MAB framework by accounting for the existence of additional information on the player and/or the machines, referred to as "contexts", in order to improve the policy. Our PLA selection problem fits squarely the contextual bandits framework, where the current estimates of students' concept knowledge correspond to the contexts and each PLA corresponds to an arm. Pulling an arm corresponds simply to selecting a PLA. In this paper, the context will include only information on the students. See Sec. 5 for a discussion on extending our framework to incorporate information on the learning resources into the contexts.

### 3. ALGORITHMS

The two algorithms we develop in this section are so-called upper confidence bound (UCB)-based algorithms [3]. These algorithms maintain estimates of the expected reward of each arm together with confidence intervals around these estimates, and iteratively update them as each new pull and its corresponding reward is observed. They then pull the arm with the highest UCB on the reward, which is equal to the expected reward plus the width of the confidence interval.

#### 3.1 CLUB: An algorithm in theory

We first develop the *contextual logistic upper confidence bound* (CLUB) algorithm in order to provide theoretical guarantees for the PLA selection problem. We assume that the binary-valued student responses to the questions in Assessment 2 are Bernoulli random variables with success probabilities following a logistic model

$$p(Y_{i,j_{a_s}} = 1) = \Phi_{\log}(\mathbf{c}_{j_{a_s}}^T \mathbf{w}_i^a) = \frac{1}{1 + e^{-\mathbf{c}_{j_{a_s}}^T \mathbf{w}_i^a}}, \quad s = 1, \dots, n_a,$$

where  $\mathbf{w}_i^a \in \mathbb{R}^K$  is the parameter vector that characterizes the students' responses to question  $i$  after taking PLA  $a$ . Also,  $j_{a_s}$  denotes the index of the  $s^{\text{th}}$  student taking PLA  $a$ , and  $n_a$  denotes the total number of students taking PLA  $a$ .  $\Phi_{\log}(\cdot)$  denotes the inverse logit link function.

The maximum-likelihood estimate (MLE) of  $\mathbf{w}_i^a$  is

$$\hat{\mathbf{w}}_i^a = \arg \min_{\mathbf{w}} - \sum_{s=1}^{n_a} \log p(Y_{i,j_{a_s}} | \mathbf{c}_{j_{a_s}}, \mathbf{w}), \quad (1)$$

which can be computed using standard logistic regression algorithms [17] whenever the MLE exists (see [39, Sec. 5.1] for a detailed discussion on the conditions under which the MLE exists).

As detailed in Algorithm 1, CLUB maintains MLEs of the parameter vector  $\mathbf{w}_i^a$  of each PLA together with a confidence interval around it. Then, after receiving a student's concept knowledge vector  $\mathbf{c}_j$ , CLUB selects the PLA with the highest UCB on the expected credit on the student's following assessment.

---

#### Algorithm 1: CLUB

---

**Input:** A set of student concept knowledge state estimates

$\mathbf{c}_j, j = 1, 2, \dots$ , and parameters  $\lambda_0, \delta, \eta, \epsilon$

**Output:** PLA  $a_j$  for each student,  $j = 1, 2, \dots$

MLE<sub>all exist</sub>  $\leftarrow$  False,  $n_a \leftarrow 0, \forall a$

**for**  $j \leftarrow 1$  **to**  $\infty$  **do**

**if** MLE<sub>all exist</sub> **then**

        Estimate  $\hat{\mathbf{w}}_i^a, \forall i, a$  according to (1)

$\Sigma_a \leftarrow \lambda_0 \mathbf{I}_K + \sum_{s=1}^{n_a} \mathbf{c}_{j_{a_s}} \mathbf{c}_{j_{a_s}}^T, \forall a$

$a_j \leftarrow$

$\arg \max_a \sum_{i=1}^Q s_i (\Phi_{\log}(\mathbf{c}_j^T \hat{\mathbf{w}}_i^a) + c_i(n_a) \sqrt{\mathbf{c}_j^T \Sigma_a^{-1} \mathbf{c}_j})$

**else**

        Randomly select  $a_j$  among PLAs where  $\exists i$  s.t.  $\hat{\mathbf{w}}_i^a$  does not exist

$n_{a_j} \leftarrow n_{a_j} + 1$

    MLE<sub>all exist</sub>  $\leftarrow$  True

**for**  $a \leftarrow 1$  **to**  $A$  **do**

**for**  $i \leftarrow 1$  **to**  $Q$  **do**

**if**  $\hat{\mathbf{w}}_i^a$  does not exist (verified via [39, Thm. 2])

**then**

                    MLE<sub>all exist</sub>  $\leftarrow$  False

The constants in Algorithm 1 are given by  $c_i(n_a) = \sqrt{2K(3 + 2 \log(1 + 2a_m^2/\lambda_0)) \log n_a K / \delta / b_{i,a}}$ , where  $a_m = \sqrt{K + 2\sqrt{K \log(1/\eta)} + 2 \log(1/\eta)}$  and  $b_{i,a} = 1/(2 + e^{\|\mathbf{w}_i^a\|_{2a_m}} + e^{-\|\mathbf{w}_i^a\|_{2a_m}})$ , and  $0 < \delta, \eta \ll 1$ . Algorithm 1 exhibits theoretical optimality guarantees (omitted due to space constraints and available at [www.sparfa.com](http://www.sparfa.com) [21]).

#### 3.2 A-CLUB: An algorithm in practice

Since in practice we do not know the values of the constants  $\Delta_{a,j}$  and also need to set the parameters  $\epsilon, \delta$ , and  $\eta$ , Algorithm 1 and its theoretical guarantees are not directly applicable. Furthermore, as the number of students grows, the confidence bounds around the estimates of each PLA's parameters might become overly pessimistic, causing the algorithm to over-explore [15]. Therefore, we now develop a second CLUB-like algorithm that leverages the asymptotic normality of the MLEs of the PLA parameters [14]. The asymptotic normality property states that, as the number of students grows large, the estimation error of the parameter  $\mathbf{w}_i^a$  for each PLA converges to a normally distributed random vector with zero mean and a covariance matrix that is a scaled inverse of the Fisher information matrix

$$\mathbf{F}_a := \sum_{s=1}^{n_a} \frac{\mathbf{c}_{j_{a_s}} \mathbf{c}_{j_{a_s}}^T}{2 + e^{\mathbf{c}_{j_{a_s}}^T \mathbf{w}_i^a} + e^{-\mathbf{c}_{j_{a_s}}^T \mathbf{w}_i^a}}.$$

Thus, we can build a confidence ellipsoid around the point estimate generated by (1), albeit asymptotically. In practice, since the true values of the parameters  $\mathbf{w}_i^a \forall i, a$  are unknown, we will use their estimates  $\hat{\mathbf{w}}_i^a$  to approximate the Fisher information matrix.

Armed with the confidence ellipsoid, we can now compute the upper bound of the expected response of student  $j$  on each question in Assessment 2 after taking PLA  $a$ . This cor-

---

**Algorithm 2: A-CLUB**

---

**Input:** A set of student concept knowledge state estimates,

$$\mathbf{c}_j, j = 1, 2, \dots, \text{parameter } \alpha$$

**Output:** PLA  $a_j$  for each student

$\text{MLE}_{\text{all exist}} \leftarrow \text{False}, n_a \leftarrow 0, \forall a$

**for**  $j \leftarrow 1$  **to**  $\infty$  **do**

**if**  $\text{MLE}_{\text{all exist}}$  **then**

    Estimate  $\widehat{\mathbf{w}}_i^a, \forall i, a$  according to (1)

$$\mathbf{F}_a \leftarrow \lambda_0 \mathbf{I}_K + \sum_{s=1}^{n_a} \frac{\mathbf{c}_j \mathbf{c}_j^T}{2 + e^{\mathbf{c}_j^T \mathbf{w}_i^a} + e^{-\mathbf{c}_j^T \mathbf{w}_i^a}}, \forall a$$

$a_j \leftarrow$

$$\arg \max_a \sum_{i=1}^Q s_i \Phi_{\log}(\mathbf{c}_j^T \widehat{\mathbf{w}}_i^a + \sqrt{\alpha(\mathbf{c}_j^T \mathbf{F}_a^{-1} \mathbf{c}_j)/n_a})$$

**else**

    Randomly select  $a_j$  among PLAs where  $\exists i$  s.t. MLE of  $\mathbf{w}_i^a$  does not exist

$n_{a_j} \leftarrow n_{a_j} + 1$

$\text{MLE}_{\text{all exist}} \leftarrow \text{True}$

**for**  $a \leftarrow 1$  **to**  $A$  **do**

**for**  $i \leftarrow 1$  **to**  $Q$  **do**

**if** MLE does not exist for  $\mathbf{w}_i^a$  (verified via [39, Thm. 2]) **then**

$\text{MLE}_{\text{all exist}} \leftarrow \text{False}$

responds to the following constrained optimization problem<sup>3</sup>

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && -\frac{1}{1 + e^{-\mathbf{c}_j^T \mathbf{w}}} \\ & \text{subject to} && (\mathbf{w} - \widehat{\mathbf{w}}_i^a)^T \mathbf{F}_a (\mathbf{w} - \widehat{\mathbf{w}}_i^a) \leq \alpha/n_a, \end{aligned}$$

where  $\alpha$  is a parameter controlling the size of the confidence ellipsoid and thus the amount of exploration. The solution to this problem is given by  $\mathbf{w} = \widehat{\mathbf{w}}_i^a + \sqrt{\frac{\alpha}{n_a \mathbf{c}_j^T \mathbf{F}_a^{-1} \mathbf{c}_j}} \mathbf{F}_a^{-1} \mathbf{c}_j$ .

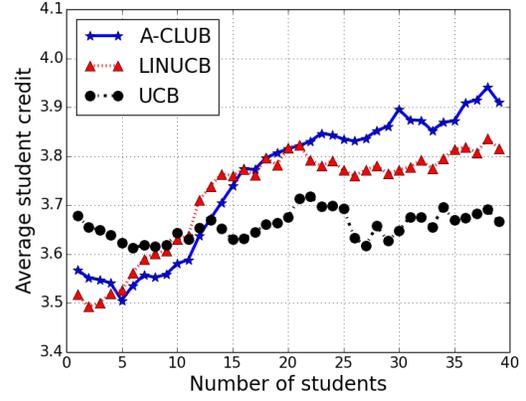
Therefore, we obtain an upper bound for the expected grade for student  $j$  on question  $i$  after taking PLA  $a$  as  $\Phi_{\log}(\mathbf{c}_j^T \widehat{\mathbf{w}}_i^a + \sqrt{\alpha \mathbf{c}_j^T \mathbf{F}_a^{-1} \mathbf{c}_j / n_a})$ . We thus arrive at Algorithm 2, which we dub asymptotic CLUB (A-CLUB).

## 4. EXPERIMENTS

In this section, we validate our algorithms experimentally on personalized cohort selection using a college physics course dataset. We will compare the performance of Algorithm 2 against other baseline (contextual) MAB algorithms. We do not compare Algorithm 1, since its theoretical bounds are usually too pessimistic in practice [15]. For comparisons using two additional datasets, see [21].

**Dataset.** The dataset consists of the binary-valued graded responses in a semester-long physics course administered on OpenStax Tutor [30] with  $N = 39$  students answering 286 questions. Cognitive science experiments were conducted in this course to test the effect of spacing versus massed practice on the students' long-term retrieval performance of knowledge [7]. For this purpose, the students were randomly divided into two cohorts containing 19 and 20 students. There are

<sup>3</sup>We assume  $\mathbf{c}_j$  is non-zero; otherwise we would simply select a PLA at random.



**Figure 2: Average student credit on Assessment 5 vs. number of students used by three algorithms. Student performance on the follow-up assessment increases as the algorithms have access to more training data. Concretely, using data from 38 students, A-CLUB finds a PLA selection policy whereby students perform approximately 10% better than selecting randomly.**

a total of 11 weekly assessments and 3 review assessments throughout the course. In the first three assessments, both cohorts received the same set of assessment questions. Starting from Assessment 4, apart from the same set of assessment questions both cohorts received on the concepts covered in the current week, each cohort also received additional, different questions. One cohort received spaced practice questions related to the concepts they learned several weeks earlier, while the other cohort received massed practice questions related to the concepts they learned in the current week. Each cohort received some spaced practices and some massed practices throughout the semester so that the sets of questions assigned to each cohort were identical in the end.

**Experimental setup.** Since the students in Cohorts 1 and 2 receive different sets of questions on Assessment 4, we investigate how this difference affects their learning on the concepts they learn next, i.e., their performance on Assessment 5. Treating each cohort as a PLA, our goal is to maximize the students' performance on Assessment 5 by assigning them to the cohort (selecting the PLA) that benefits them the most. Therefore, in our setting the number of PLAs is  $A = 2$ . We take the students' graded responses to questions in Assessments 1–3 and apply SPARFA to estimate each student's  $K$ -dimensional concept knowledge vector  $\mathbf{c}_j$ , which we use as the context. We set the number of concepts to  $K = 3$ .<sup>4</sup> Since Cohorts 1 and 2 also receive different questions for Assessment 5 as part of the spacing vs. mass retrieval practice experiment on new concepts covered in Week 5, we take the set of  $Q = 5$  questions shared between the two cohorts to evaluate their performance. Since MAB algorithms analyze students sequentially, we randomly permute the order of the students and average our results over 2000 random permutations.

<sup>4</sup>In our experiments, we have found that the performance of SPARFA and A-CLUB is robust to the number of concepts  $K$  as long as  $K \ll Q$ .

	A-CLUB	LINUCB	UCB
Training set	<b>3.69</b>	3.68	3.65
Test set	<b>3.89</b>	3.77	3.70

**Table 1: Performance comparison of A-CLUB against two baseline algorithms on personalized cohort selection on the physics course dataset. A-CLUB outperforms the other algorithms in terms of average student credit on the follow-up assessment (out of a full credit of 5) on both the training and test sets.**

*Evaluation method.* We use the unbiased offline evaluation approach in [24, 25] to evaluate our algorithms. We use only the students that were actually assigned to the same cohort as chosen by our algorithms and ignore the other students. This approach evaluates the decision making algorithms under the scenario where the data is collected in a specific “off-line, off-policy” manner, i.e., the data is collected by selecting PLAs for each student uniformly at random across every PLA, as opposed to a more typical MAB setting where PLAs are chosen for students sequentially given the observed follow-up assessment performance of previous students. Such a scenario fits our experimental setup well and yields an unbiased estimate of the expected reward for each student [25]. We use the students’ total credit on Assessment 5, i.e.,  $\sum_{i=1}^Q s_i Y_{i,j}$ , as the metric to evaluate the performance of the algorithms.

*Results and discussion.* Figure 2 shows the students’ average credit (out of a full credit of 5) on Assessment 5 vs. the number of students the algorithms use for the algorithms A-CLUB, LINUCB [24], and UCB [3]. The parameters in every algorithm were tuned for best performance. We see that the average student credit increases as the number of students the algorithms observe increases, i.e., the algorithms improve their PLA selection policy as they see more training data. As a concrete example, by comparing the average student credit at the first and last points on the curves, we see that A-CLUB has found a policy that yields students approximately 10% more credit than a policy that selects PLAs randomly.

Following the approach in [24], we also conduct an experiment by separating the dataset into a training set with 80% of the students and a test set with 20% of the students, to validate both the efficiency (performance on the training set) and efficacy (performance on the test set) of A-CLUB. We train the above three algorithms on the training set and apply the learned PLA selection policy to the test set, and report the average student credit obtained on both sets. Table 1 indicates that A-CLUB outperforms the other algorithms on both the training set and the test set. Better performance on the test set means that A-CLUB learns a better policy than the other algorithms, while better performance on the training set means that it learns this policy very quickly as the amount of training data increases.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a contextual (multi-armed) bandits framework for PLA selection that maximizes students’ immediate success on a follow-up assessment, given the latent concept knowledge estimated from their binary-valued graded responses to questions in previous assessments. Our contextual logistic upper confidence bound (CLUB) algorithms learn such a policy and achieve better or comparable performance than baseline algorithms.

There are a number of avenues for future work. First, our context vectors are indexed by student features only, while in the general contextual bandits setting the contexts can be indexed by both student features and features of the learning resources. SPARFA-Trace [22], a recently developed framework for time-varying learning and content analytics, features a mechanism to analyze the content, quality, and difficulty of all kinds of learning resources (i.e., textbook sections, lecture videos, practice questions, etc.). We can apply this approach to extract features from the learning resources that we can integrate into the contexts in our algorithms. Second, we can incorporate an additional PLA that corresponds to “no action”, due to the cost of taking actions, as considered in [36]. This extension would enable students with high knowledge on the concepts covered to avoid repeated practice and advance more quickly to new concepts. Third, we are interested in integrating our approach into more sophisticated contextual bandit algorithms, e.g., [37] to reap further performance improvements.

## 6. ACKNOWLEDGEMENTS

Thanks to Phillip Grimaldi, former pinball champion of Indiana, for collecting the physics course dataset and Mihaela van der Schaar for insightful suggestions. Visit our website [www.sparfa.com](http://www.sparfa.com), where you can learn more about the SPARFA project and purchase SPARFA t-shirts and other merchandise.

## 7. REFERENCES

- [1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, Dec. 2011.
- [2] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Machine Learning Research*, 3:397–422, Mar. 2003.
- [3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, May 2002.
- [4] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *Proc. 9th Intl. Conf. on Intelligent Tutoring Systems*, pages 373–382, June 2008.
- [5] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and D. Pritchard. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. In *Proc. 5th Intl. Conf. on Educational Data Mining*, pages 95–102, June 2012.
- [6] P. Brusilovsky and C. Peylo. Adaptive and intelligent web-based educational systems. *Intl. J. Artificial Intelligence in Education*, 13(2-4):159–172, Apr. 2003.

- [7] A. C. Butler, E. J. Marsh, J. Slavinsky, and R. G. Baraniuk. Integrating cognitive science and technology improves learning in a STEM classroom. *Educational Psychology Review*, 26(2):331–340, June 2014.
- [8] H. Cen, K. R. Koedinger, and B. Junker. Learning factors analysis – A general method for cognitive model evaluation and improvement. In *Proc. 8th Intl. Conf. on Intelligent Tutoring Systems*, pages 164–175, June 2006.
- [9] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180, Jan. 2011.
- [10] B. Clement, D. Roy, P. Oudeyer, and M. Lopes. Multi-armed bandits for intelligent tutoring systems. *J. Educational Data Mining*, 7(2):20–48, 2015.
- [11] 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, Dec. 1994.
- [12] Coursera. <https://www.coursera.org/>, 2016.
- [13] edX. <https://www.edx.org/>, 2016.
- [14] L. Fahrmeir and H. Kaufmann. Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, 13(1):342–368, Mar. 1985.
- [15] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, Dec. 2010.
- [16] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proc. 10th Intl. Conf. on Intelligent Tutoring Systems*, pages 35–44, June 2010.
- [17] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2010.
- [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. 7th Intl. Conf. on Educational Data Mining*, pages 99–106, July 2014.
- [19] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. *Intl. J. Artificial Intelligence in Education*, 8(1):30–43, 1997.
- [20] 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.
- [21] A. S. Lan and R. G. Baraniuk. A contextual bandits framework for personalized learning action selection – Extended version. Technical report, Rice University, 2016.
- [22] A. S. Lan, C. Studer, and R. G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proc. 20th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 452–461, Aug. 2014.
- [23] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *J. Machine Learning Research*, 15:1959–2008, June 2014.
- [24] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. 19th Intl. Conf. on World Wide Web*, pages 661–670, Apr. 2010.
- [25] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proc. 4th ACM Intl. Conf. on Web Search and Data Mining*, pages 297–306, Feb. 2011.
- [26] R. Lindsey, M. Mozer, W. Huggins, and H. Pashler. Optimizing instructional policies. In *Advances in Neural Information Processing Systems*, pages 2778–2786, Dec. 2013.
- [27] Y. Liu, T. Mandel, E. Brunskill, and Z. Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In *Proc. 7th Intl. Conf. on Educational Data Mining*, pages 161–168, July 2014.
- [28] F. Lord. *Applications of Item Response Theory to Practical Testing Problems*. Erlbaum Associates, 1980.
- [29] T. Mandel, Y. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. In *Proc. Intl. Conf. on Autonomous Agents and Multi-agent Systems*, pages 1077–1084, May 2014.
- [30] OpenStaxTutor. <https://openstaxtutor.org/>, 2016.
- [31] W. Powell. *Approximate Dynamic Programming: Solving The Curses of Dimensionality*. John Wiley & Sons, 2007.
- [32] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto. Faster teaching by POMDP planning. In *Proc. 15th Intl. Conf. on Artificial Intelligence in Education*, pages 280–287, June 2011.
- [33] A. N. Rafferty, M. M. LaMar, and T. L. Griffiths. Inferring learners’ knowledge from their actions. *Cognitive Science*, 39(3):584–618, Apr. 2015.
- [34] M. D. Reckase. *Multidimensional Item Response Theory*. Springer, 2009.
- [35] J. Rollinson and E. Brunskill. From predictive models to instructional policies. In *Proc. 8th Intl. Conf. on Educational Data Mining*, pages 179–186, June 2015.
- [36] C. Tekin, J. Braun, and M. van der Schaar. eTutor: Online learning for personalized education. In *Proc. 40th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 5545–5549, April 2015.
- [37] C. Tekin and M. van der Schaar. RELEAF: An algorithm for learning and exploiting relevance. *IEEE J. Selected Topics in Signal Processing*, 9(4):716–727, June 2015.
- [38] 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. *Intl. J. Artificial Intelligence in Education*, 15(3):147–204, Aug. 2005.
- [39] D. Vats, C. Studer, A. S. Lan, L. Carin, and R. G. Baraniuk. Test size reduction via sparse factor analysis. *Preprint*, June 2014.
- [40] B. P. Woolf. *Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing E-learning*. Morgan Kaufman Publishers, 2008.

# How Good Is Popularity?

## Summary Grading in Crowdsourcing

Haiying Li  
Rutgers University  
10 Seminary Place  
New Brunswick, NJ 08901  
1-848-932-0868  
haiying.li@gse.rutgers.edu

Zhiqiang Cai  
University of Memphis  
365 Innovation Dr.  
Memphis, TN 38152  
1-901-678-2364  
zca@memphis.edu

Arthur C. Graesser  
University of Memphis  
365 Innovation Dr.  
Memphis, TN 38152  
1-901-678-2364  
grasser@memphis.edu

### ABSTRACT

In this paper, we applied the crowdsourcing approach to develop an automated popularity summary scoring, called wild summaries. In contrast, the golden standard summaries generated by one or more experts are called expert summaries. The innovation of our study is to compute LSA (Latent Semantic Analysis) similarities between target summary and wild summaries rather than expert summaries. We called this method CLSAS, i.e., crowdsourcing-based LSA similarity. We evaluated CLSAS by comparing it with other approaches, Coh-Metrix language and discourse features and LIWC psychometric word measures. Results showed that CLSAS alone could explain 19% of human summary score, which was equivalent to the variance explained by dozens of language and discourse features and/or the word features. Results also showed that adding language and/or word features to CLSAS increased small additional correlations. Findings imply that crowdsourcing-based LSA similarity approach is a promising method for automated summary assessment.

### Keywords

Summary grading, Crowdsourcing, LSA Similarity, Coh-Metrix, LIWC

### 1. INTRODUCTION

The use of the summarization strategy enables to improve reading comprehension and production of expository texts for both L1 learners [1] and L2 learners [2]. Summarizing involves reading processes and reproducing processes. Reading process requires the learners to identify the main ideas and distinguish the important points from the unimportant points. Reproducing process requires the learners to restate the important ideas in a coherent, precise and accurate manner in their own words [3]. Learners' summarizing skill depends on the ability to construct a coherent mental model of the text, which is aligned with text discourse [4]. This ability consists of three knowledge components: rhetorical text structures and genres, propositional text content, and a coherent mental model for a variety of genres [4], which are important for reading comprehension [5]. Summarization strategy is an effective instructional strategy [6] to help students improve these abilities [7] and summary writing is therefore considered as a good measure of reading comprehension at a deep level.

Grading summaries are time-consuming and costly for teachers, so it is impossible for teachers to provide a real-time and instant summary score, let alone provide the instant feedback on the quality of summaries. Researchers thereby have developed the

automated summary assessments with the techniques of natural language processing and machine learning [4,8]. These assessments are not practical for teachers because they require model building based on human expert summaries as the reference summaries and a large amount of human summary grading. Thus, model rebuilding is time-consuming and costly for teachers. Each time teachers need to repeat such complex steps as expert-written summaries as reference, human-scored summary as the training set, model training, and model evaluation. As summary writing is a weekly assignment for middle school and high school students, summary grading will be a common task for teachers. The present automated summary assessments will not reduce but increase the teachers' workload. These methods are impractical for teachers to use. Teachers need a more efficient and effective summary assessment with least efforts.

In this paper, we applied the crowdsourcing approach to develop an automated "popularity" summary scoring. Crowdsourcing enables a diverse and a large amount of population to generate abundant summaries, which are called "popularized summaries" or "wild summaries." In contrast, the golden standard summaries generated by one or more experts are called "expert summaries." The innovation of our study is to compute LSA (Latent Semantic Analysis) similarities between the target summary and the wild summaries instead of expert summaries. We called it CLSAS, namely, crowdsourcing-based LSA similarity. We proposed CLSAS was a robust measure for summary grading.

This study makes innovative contributions to the automated summary assessment for three reasons. First, it is efficient and effective, because the model was built based on one feature rather than dozens of features. Second, it is unnecessary for human experts to generate the golden summaries on each quality level. The model was built based on the wild summaries generated by all of the summary writers. Third, it is unnecessary for human experts to manually grade summaries for the model training.

The next section briefly reviews research on automated summary assessment, crowdsourcing approach, and three advanced text analysis tools, LSA similarity [9], Coh-Metrix [10] and LIWC (Linguistic Inquiry and Word Count) [11].

#### 1.1 Automated Summary Assessment

Techniques of natural language processing and machine learning have been used to develop the automated summary assessment [4,8]. Diverse features used in the assessment range from semantic features measured by LSA [8] to language features exacted by BLEU (Bilingual Evaluation Understudy) [4], ROUGE (Recall-

Oriented Understudy for Gisting Evaluation) [12], TERp (Translation Error Rate Plus) [4], and N-gram [12]. Some features were used to detect plagiarism in summary (e.g., N-gram [4]), assess coherence of the summary (e.g., LSA [8] and N-gram [12]), evaluate content unit (e.g., unigram overlap [8]), or examine the length of summary [4]. These assessments were proved to robustly predict human summary grading [4,8] but had the following limitations.

First, all of these assessments need reference summaries that are generated by one or more human experts [4,8]. The reference summaries have different qualities, ranging from good to poor on multiple-point scales [4]. The student's summary is graded by comparing with the reference summaries. The similarities could be computed by similarities of LSA [8], a lexical and phrasal overlap (e.g., ROUGE) [8], N-gram overlap (e.g., BLEU) [4,8], summary length [4], or token count [4]. Second, the sufficient amount of human-graded summaries at each quality level is required to build the model for the supervised learning. Third, different language and discourse features and algorithms are tested in order to build a better fit model. As these assessments are not content independent, these three cycles are repeated if summaries' source text changes. These tasks definitely increase extra workload for teachers, so it is hard and impractical to spread these approaches. It is necessary to develop a summary assessment without expert reference summaries, human grading, and model rebuilding for a new source text. This study aims to explore a real-time and efficient summary assessment that requires the least efforts so that teachers can easily use it by themselves.

## 1.2 Crowdsourcing

Crowdsourcing refers to a process that mobilizes a huge amount of population (called crowd workers) to accomplish the complex, collaborative, and sustainable tasks on demand and at large scale, especially from an online community rather than traditional employees or suppliers [13]. Crowd workers can either be volunteers for collective projects such as Wikipedia or paid via platform such as Amazon's Mechanical Turk, one popular crowdsourcing platform [13]. Crowdsourcing is frequently used to generate ideas and break down creative tasks into smaller pieces [13-17]. The application of crowdsourcing is an emerging approach in research. For example, some researchers asked crowd workers to create or retrieve content for new stories [16,17], to generate a story [14] or summaries of social media events [15]. This collaborative work provides an author diverse ideas or contents quickly [13-17].

## 1.3 LSA Similarity

LSA [18] is a mathematical and statistical technique that represents knowledge about words, sentences, paragraphs, and documents on the basis of a large corpus of texts. LSA reduces a large corpus of texts to 100-300 dimensions using singular value decomposition technique. The conceptual similarity between two texts is computed as the geometric cosine between the vectors representing two texts. The cosine value varies from -1 to 1 [18,19], with the higher score representing higher similarity.

LSA is used to assess coherence in Coh-Metrix [10] and quality of essays [8, 20-22]. In addition, LSA has been utilized in the intelligent tutoring system (ITS) to assess the constructed response or the open response, such as AutoTutor [19]. These assessment systems for essay, summary, or open response requires expert reference summaries and human-graded summaries generated by human experts. Few studies do not use expert

summaries as reference. Summarization in machine translation develops a fully automated approach to evaluate ranking systems that requires no expert summaries [8]. However, it requires a large amount of content annotations and is restricted to the ranking system, which it is not appropriate for teachers to use for summary grading. Cai et al. [9] explored the LSA similarity model without the golden standard reference for the open response assessment. Instead, the reference was all the responses written by students except the target response. We borrowed this approach in this study and use the learners' summaries as the reference summaries.

## 1.4 Coh-Metrix

Coh-Metrix (cohmetrix.com) is a computer-based tool that automates many language- and text-processing mechanisms over hundreds of measures of cohesion, language, and readability [10]. Coh-Metrix is developed based on a multilevel theoretical framework [23]. This framework specifies six theoretical levels: words, syntax, explicit textbase (e.g., explicit propositions, referential cohesion), situation model (also called mental model), discourse genre and rhetorical structure (the type of discourse and its composition), and the pragmatic communication level. The first five of these six levels have metrics captured in the Coh-Metrix automated text analysis tool [10].

The current version of Coh-Metrix [10] extracts 110 measures, which are categorized into genre (narrative versus informational), LSA space (e.g., text cohesion), word information (e.g., familiarity, concreteness, imageability, meaningfulness, age of acquisition), word frequency, part of speech, density score (e.g., density of pronouns), logic operators (e.g., *if-then*), connectives (e.g., *therefore*), type/token ratio, polysemy and hypernym, syntactic complexity (e.g., noun phrase density), readability (e.g., Flesch-Kincaid grade level), co-reference cohesion (e.g., noun overlap, argument overlap), along with five primary components extracted based on these features (e.g., narrativity, word concreteness, syntactic simplicity, referential cohesion, and deep cohesion).

## 1.5 LIWC

LIWC (Linguistic and Inquiry Word Count) [11] computes the percentage of words in a text that fit into the linguistic or psychological categories. The 2015 LIWC dictionary contains 6,400 words, word stems, and select emoticons. It generates 93 measures that are categorized into the following categories: word count, summary language variables (e.g., analytical thinking, authentic, emotional tone), linguistic dimensions (e.g., functional words, pronouns, conjunctions), other grammar (e.g., common verbs, interrogatives), psychological processes (e.g., affective, social, cognitive, informal language). The word count function of LIWC attempts to match each word in a given text to a word in the various categories.

The LIWC categories have been confirmed as valid and reliable markers of a variety of psychologically meaningful constructs [11]. The different categories of words would be expected to predict psychological dimensions. For example, negative emotion words would be diagnostic of gloomy texts. The function words (particularly pronouns) are diagnostic of social status, personality, and various psychological states. Differences in function word use can be reflected by gender, age, and social class. LIWC is used to measure the formal versus informal language formality [24,25].

This paper combined the crowdsourcing approach with the LSA similarity to assess summaries. This approach was evaluated by comparing the Coh-Metrix language and discourse features and

the LIWC word features with the human-graded summary scores as the criteria. Specially, seven models were trained and compared their predictability for the human summary scores: (1) CLSAS, (2) Coh-Metrix language features (94), (3) LIWC word features (93), (4) Coh-Metrix + LIWC, (5) CLSAS + Coh-Metrix, (6) CLSAS + LIWC, and (7) CLSAS + Coh-Metrix + LIWC. It is necessary to clarify that the human-graded summary scores were only used to evaluate but not build the model. We hypothesize that crowdsourcing-based LSA similarity is an efficient, effective, and reliable measure for summary grading for the following two reasons. First, LSA is a most robust feature for semantic meaning [11] than the language and word features. Second, the wild summaries as reference maximally represent diversity of students' summaries as compared with expert summaries.

## 2. METHOD

### 2.1 Participants

Crowd workers ( $N = 201$ ) volunteered for 3-hour monetary compensation (\$30) on Amazon Mechanical Turk (AMT), a trusted and commonly used data collection service [21]. The basic requirement for participation is that they have the goal to improve English summary writing. Participants were required to complete writing 8 summaries, but only 1,481 summaries were collected due to the technical issues. 71% participants were Asian, 16% white or Caucasian, 7% African American, 5% Hispanic, 2% other. Their average age was 33.50 ( $SD = 8.79$ ), 57% were male, and 81% with bachelor degree or above.

### 2.2 Materials

Participants read 8 expository texts with different topics and text difficulties in the AutoTutor CSAL. CSAL is an intelligent tutoring system that teaches adult learners the summarization strategies in order to improve their reading comprehension [19]. Participants were required to write a summary with 50-100 words for each text. Four texts are on comparison-contrast text structure and another four on cause-effect text structure (See Table 1). The text difficulty was measured with the Coh-Metrix formality ( $z$ -score) at the multiple textural levels and Flesch-Kincaid grade level, sensitive to word length and sentence length [24]. These 8 texts were formal and above grade 8 to early college grades [24]. The balanced Latin-square designs were applied to control for order effects in terms of text difficulty, topics and text structures.

### 2.3 Summary Grading

The summaries were graded based on four components: topic sentence, content, grammar and mechanics, and signal words. Table 2 lists the detailed descriptions for three scales of each component, from 0 (minimum) to 2 (maximum) points. Thus, the total score ranged from 0 to 8. Four English native researchers graded summaries, 1 male and 3 female. There were three rounds of training for summary grading and after each grading, and then the disagreements were discussed. Before grading, they got familiar rubrics and then they started the three-round grading with one per week. Each round included 32 randomly-selected summaries (4 from each text and 8 texts in total). Inter-rater reliability was assessed by the intraclass correlation coefficient with a two-way random model and absolute agreement type. The average inter-rater reliability reached the threshold: Cronbach's  $\alpha = .82$ , intraclass correlation coefficient = .80. As the average of reliabilities for three training sets were high, each grader graded summaries for two texts in the same text structure.

**Table 1. Source Texts and the Number of Summaries ( $N$ ).**

Structure	Topics	Formality	FKGL	Words	$N$
Comparison	Butterfly and Moth	.12	8.6	255	183
	Hurricane	.20	9.4	222	185
	Walking and Running	.18	8.9	399	187
	Kobe and Jordan	.14	9.2	299	187
Causation	Floods	.47	9.2	230	186
	Job Market	.62	10.9	240	181
	Effects of Exercising	.28	9.1	195	189
	Diabetes	.64	11.7	241	182

**Table 2. Rubrics for Scoring Summary**

Categories	2 points	1 points	0 point
Topic Sentence	A clear topic sentence that states the main idea.	A topic sentence that touches upon the main idea.	The summary does not state the main idea.
Content	Major details stated economically and arranged in a logical order. No minor or unimportant details or reflections.	Some but not all major details stated and not necessarily in a logical order. Some minor or unimportant details or reflections.	Few major details stated and not necessarily in a logical order. Many minor or unimportant details or reflections.
Mechanics and Grammar	Few or no errors in mechanics, usage, grammar or spelling.	Some errors in mechanics, usage, grammar or spelling that to some extent interfere with meaning.	Serious errors in mechanics, usage, grammar or spelling, which make the summary difficult to understand.
Signal Words	Uses the clear and accurate signal words to connect information.	Uses several clear and accurate signal words to connect information.	Uses several clear signal words to connect information.

### 2.4 Measures

In this study, we employed three approaches to assess summaries: semantic meaning measured by LSA similarity, Coh-Metrix, and LIWC. The crowdsourcing-based LSA similarity score was the LSA cosine between a target summary and all the wild summaries from the corresponding source text. 94 language and discourse features were utilized to train and build the Coh-Metrix summary assessment model. All of 93 psychometric word features were utilized to train and build the LIWC summary assessment model.

## 2.5 Procedure

Participants took a demographic survey, a pretest (1 comparison and 1 causation), training (2 comparisons and 2 causations), and a posttest (1 comparison and 1 causation). On tests, participants wrote summaries by themselves. During training, two agents first interactively presented the importance of signal words for two text structures (comparison and causation) and how to use signal words to identify the corresponding text structure. Then participants interacted with the conversational agents to learn a summarizing strategy with adaptive scaffolding. Participants were required to write a summary with 50 to 100 words for each text. If the amount of words was beyond the range, the agents reminded the participants of the required length. If the participants copied the original sentences with 10 consecutive words, the agents reminded them of using their own words. Agents did not provide the adaptive feedback for their summary writing, but commented on three summary examples with good, medium, and bad qualities for each source text. The primary interface during training was shown in Figure 1.

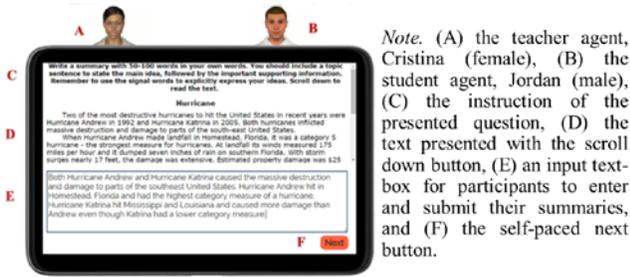


Figure 1. Screenshot of Learning Interface.

## 3. RESULTS

A series of linear regressions with 10-fold cross-validation in WEKA was performed on 7 models, respectively. Fisher z was used to compare the difference between two pairs of correlations (see Table 3). Results revealed that crowdsourcing-based LSA similarity robustly predicted human summary grading ( $r = .44$ ;  $R^2 = .19$ ), as well as 55 Coh-Matrix measures ( $r = .43$ ;  $R^2 = .18$ ), 57 LIWC measures ( $r = .47$ ;  $R^2 = .22$ ), and 108 measures by Coh-Matrix (57) and LIWC (51) jointly ( $r = .46$ ;  $R^2 = .21$ ). This indicates that the variance explained by one LSA similarity measure is equivalent to the variance explained by more than 55 language features or word features, and more than 100 language and word features jointly.

Adding 94 Coh-Matrix features to CLSAS added an additional variance ( $r = .51$ ;  $R^2 = .26$ ) in explaining human grading scores. Adding 93 LIWC features also added an additional variance ( $r = .55$ ;  $R^2 = .30$ ). Adding both Coh-Matrix and LIWC feature added an additional variance ( $r = .49$ ;  $R^2 = .24$ ), but the increased variance was significantly lower than by adding either Coh-Matrix or LIWC features. Due to the limited pages and the significant predictors in the Coh-Matrix + LIWC model overlapped with those in the Coh-Matrix model or the LIWC model, we only reported the predominant predictors in the Coh-Matrix model and LIWC model as below.

The 55 Coh-Matrix measures consisted of 9 descriptive (e.g., word count, sentence length), 4 referential cohesions (e.g., noun overlap, argument overlap), 5 LSA overlap (e.g., adjacent sentences, LSA given, LSA new), 3 lexical diversity (e.g., type-token ratio), 5 connectives (e.g., logical, additive), 3 situation

model (e.g., causal verbs and particles, LSA verb overlap), 5 syntactic complexity (e.g., minimal edit distance, sentence syntax similarity), 4 syntactic pattern density (e.g., noun phrase density, verb phrase density), 16 word information (e.g., noun, adjective, hypernymy for nouns), and 1 readability (e.g., Flesch Kincaid Grade Level).

The 57 LIWC features consisted of 3 summary variables (e.g., analytical thinking, authentic), 3 language metrics (e.g., sentence length, words with more than 6 letters), 11 function words (e.g., personal pronouns), 4 grammar other (e.g., regular verb, quantifiers), 4 affect words (e.g., emotion words, anger), 3 social words (e.g., friend, gender referents), 3 cognitive processes (e.g., tentativeness, certainty), 3 perceptual (e.g., seeing, hearing), 3 biological processes (e.g., body, health), 2 core drives and needs (affiliation and risk focus), 1 relativity, 4 personal concerns (e.g., religion, home), 2 informal speech (swear and filler), and 3 all punctuations (e.g., apostrophes, comma).

Table 3. Fisher's z: Comparisons of Correlations

Models	1	2	3	2+3	1+2	1+3
1 ( $r=.44$ )	---					
2 ( $r=.43$ )	-0.34	---				
3 ( $r=.47$ )	1.03	1.36	---			
2+3 ( $r=.46$ )	0.68	1.02	-0.35	---		
1+2 ( $r=.51$ )	2.46**	2.80**	1.43	1.78*	---	
1+3 ( $r=.55$ )	3.97***	4.31***	2.94**	3.29**	1.51	---
1+2+3 ( $r=.49$ )	1.74*	2.07*	0.71	1.05	-0.73	-2.24*

Note. 1 = LSA similarity; 2 = Coh-Matrix features; 3 = LIWC features. \*  $p < .05$ . \*\*  $p < .01$ . \*\*\*  $p < .001$ .

## 4. DISCUSSION

This paper developed an effective and efficient automated summary assessment, called crowdsourcing-based LSA similarity (CLSAS). Crowdsourcing enables a diverse and a mass of people to produce abundant wild summaries. CLSAS used the wild summaries rather than the human expert summaries as the reference when computing LSA similarities. The CLSAS was validated by comparing with Coh-Matrix language features, LIWC word features, and both language and word measures together with human-scored summaries as the criteria. Results indicated that CLSAS measure predicted human summary grading as well as over 55 language measures, 57 word measures, and 108 language and word measures, respectively. Even though adding language features, word features, or both to CLSAS improved the predictability, the predictability of CLSAS alone is most robust with correlation coefficient above 6.74 in each model. Findings imply that crowdsourcing-based LSA similarity approach is a promising method and will have good popularity in automated summary assessment.

One possible explanation for the significant predictability of CLSAS is that the wild summaries generated by diverse populations display diverse qualities as compared with few expert summaries. These wild summaries maximally represent the target summary. On the hand, the wild summaries represent neutralized or averaged semantic meaning, which is called *centroid*. The centroid might better capture the semantic meaning represented in

the target summary. For example, the CLSAS model showed that LSA similarity had a very high coefficients,  $\beta = 8.60$ , which was substantially higher than other measures' in other models.

The Coh-Matrix measures are different from the crowdsourcing-based LSA similarity due to its nature on measuring cohesion, language, and readability rather than semantic meaning [10]. One semantic measure of LSA similarity between the target summary and the crowdsourcing-based summaries is equivalent to 55 Coh-Matrix language measures. Among these language measures, LSA overlap among all sentences in paragraph reached 5.43 for mean and 2.07 for standard deviation; LSA given/new -3.60 for mean and -2.39 for standard deviation; and LSA overlap between adjacent sentences, -1.20 for mean. The other measures showed very low coefficients, generally below 1.00. This implies that a range of language measures jointly plays a role in assessing summaries, but LSA measures are attributed more than others.

Besides the predominant role of LSA measures, other important Coh-Matrix measures included lexical diversity ( $\beta = 3.92$ ) measured by type-token ratio. Type-token ratio is widely used for both automated essay assessment [19] and automated summary assessment [4]. When the type-token ratio is high, namely, more unique words are used, the lexical diversity is high and the text is likely to be either very low in cohesion or very short. Oppositely, when the type-token ratio is low, namely, more words are repeatedly used, the lexical diversity is low, but cohesion is high. Summarizing requires conciseness and brevity, so in one summary, repeatedly using the same word will lower the quality of summary. Another two crucial measures are sentence syntax similarity between adjacent sentences ( $\beta = 4.49$ ) and across paragraphs ( $\beta = -4.71$ ). The high syntax similarity between adjacent sentences suggests the uniformity and consistency of the syntactic construction. This implies that the whole summary is consistent in syntactic construction. However, the low syntax similarity across paragraphs results in greater syntactic variety.

Another two most robust predictors are paragraph count ( $\beta = -12.74$ ) and word length (number of syllables;  $\beta = 4.32$ ). These two measures are frequently used in the automated summary [4] and essay assessment [19]. Our study controlled the number of words of summaries, which explains why word count is not a robust predictor, as compared with the previous studies [9]. As the summary should be brief and concise, more paragraphs demonstrate the poor quality in conciseness. However, the high word length increases difficult to read and represents an academic or formal language style [25] in the summary.

The phenomena that the Coh-Matrix features were unevenly weighted did not occur in the LIWC features. Specifically, among Coh-Matrix measures, the measures such as cohesion, syntactic and lexical complexity are more robust than measures at the word level. LIWC measures are all at the word level, but go beyond the linguistic words. They expand to diverse psychometric words, such as analytical thinking, emotion, and social. All the LIWC measures are evenly weighted to predict human summary scores. This pattern occurs in the Coh-Matrix and LIWC joint model as well. These findings suggest that each type of words plays a small piece of role, as compared to language and semantic measures.

Fisher's  $z$  comparisons CLSAS with Coh-Matrix measures, LIWC measures, and Coh-Matrix + LIWC measures demonstrated no differences in explained variance in human summary grading between CLSAS and Coh-Matrix, CLSAS and LIWC, and CLSAS and Coh-Matrix + LIWC. The findings supported our

hypothesis that CLSAS could predict human summary grading as well as dozens of language measures and/or LIWC measures.

To further evaluate the validity of CLSAS, we added Coh-Matrix, LIWC, and Coh-Matrix + LIWC measures to CLSAS model with different combinations. Results showed adding each of these features increased the predictability. It is easier to explain the incremented model because the language and word features represent different aspects of summary assessment and enable to compensate the semantic feature. No matter what features were added to CLSAS, CLSAS is consistently the most significant feature in the models. Specifically, the correlation coefficient of LSA was 7.49, 6.74, and 6.80 when adding the Coh-Matrix language features, the LIWC word features, and both, respectively. Therefore, LSA similarity was a robust feature for summary assessment, no matter when it is used alone or jointly with other features.

## 5. CONCLUSION

These findings suggest that crowdsourcing-based LSA similarity (CLSAS) is a robust predictor of human summary grading and it is a reliable measure for the automated summary assessment, as compared with a range of language and word measures. As CLSAS has a powerful predictability for human summary score, the wild summaries are assumed as a promising and encouraging approach to replace the expert summaries for its time-saving and efficient. Opposed to the tedious and time-consuming manual summary grading, the wild summaries have no doubt for its popularity and practicability for teachers. This efficient and effective summary grading could dramatically encourage and motivate the teachers to instruct the summarization strategy. Consequently, this will enhance the students' summarization skills, especially summary writing. For example, when teachers need to grade the students' summaries, they could use all of the summaries that the students wrote as the reference. These summaries wildly generated by the students represent diverse qualities. For a particular target summary, the teacher only clicks the target summary and its CLSAS will be automatically computed with all of the summaries. Each time teachers need summary grading, they could repeat this cycle, no any human grading is needed. Based on the LSA similarity score, the summary score could be generated.

This crowdsourcing approach could be popularized and applied to the ITS learning and assessment environment as well. The current ITS assessment assesses the open response with a list of stored expectations and misconceptions [19]. Unfortunately, students' answers could not be assessed accurately due to the unmatched "golden" reference. To address this issue, the crowdsourcing generated responses could be adopted as the reference to replace the limited number of responses that the human expert generates. However, the reliability and validity of the wild open responses need to be evaluated in the future research.

The future study should concentrate on scaling crowdsourcing-based LSA similarity score into 3- or 5-point scales that teachers usually use for a better interpretation. The present study only showed its predominant role in summary assessment without specifying the extent to which LSA similarity score represents the different levels of summaries. The present study compared the CLSAS approach with dozens of measures, which may have an overfitting problem. The future study could select the most popular features that are used in the automated summary assessment and compared them with the CLSAS approach.

To sum up, this study proposed an innovative approach, crowdsourcing-based summary assessment, to the summary assessment from two perspectives. First, the summary reference could be a range of summaries that are wildly generated by a lot of population who are not necessary to be experts. Second, LSA similarity between the target summary and the wildly-generated summaries is a powerful predictor for human summary grading. This innovation will advance the development of automated assessment, especially automated assessment in the ITS.

## 6. ACKNOWLEDGMENTS

The research reported in this paper was supported by the National Science Foundation (0325428, 633918, 0834847, 0918409, 1108845) and the Institute of Education Sciences (R305A080594, R305G020018, R305C120001, R305A130030).

## 7. REFERENCES

- [1] McNamara, D. S., Crossley, S. A., & McCarthy, P. M. (2010). The linguistic features of quality writing. *Written Communication, 27*, 57–86. DOI=[10.1177/0741088309351547](https://doi.org/10.1177/0741088309351547).
- [2] Crossley, S. A., Salsbury, T., McNamara, D. S., and Jarvis, S. 2010. Predicting lexical proficiency in language learner texts using computational indices. *Language Testing, 18*, 561-580. DOI=[10.1177/0265532210378031](https://doi.org/10.1177/0265532210378031).
- [3] Kintsch, E. (1990). Macroprocesses and microprocesses in the development of summarization skill. *Cognition and Instruction, 7*(3), 161-195. DOI=[10.1207/s1532690xci0703\\_1](https://doi.org/10.1207/s1532690xci0703_1).
- [4] Madnani, N., Burstein, J., Sabatini, J. and O'Reilly, T., 2013. Automated scoring of a summary writing task designed to measure reading comprehension. NAACL/HLT 2013, 163.
- [5] Kintsch, W., 1998. Comprehension: A paradigm for cognition. Cambridge university press.
- [6] Friend, R., 2001. Effects of strategy instruction on summary writing of college students. *Contemporary Educational Psychology, 26*(1), 3-24. DOI=[10.1006/ceps.1999.1022](https://doi.org/10.1006/ceps.1999.1022).
- [7] G. Yu. 2003. Reading for summarization as reading comprehension test method: Promises and problems. *Language Testing Update, 32*:44–47.
- [8] Passonneau, R. J., Chen, E., Guo, W. and Perin, D. 2013. Automated pyramid scoring of summaries using distributional semantics. In *ACL* (Sofia, Bulgaria, August 4-9, 2013), 143-147.
- [9] Cai, Z., Graesser, A.C., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D. and Butler, H., 2011. Dialog in ARIES: User input assessment in an intelligent tutoring system. In *Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems* (San Francisco, C.A., August 4-6, 2011), 429-433
- [10] McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). Automated evaluation of text and discourse with Coh-Metrix. New York: Cambridge University Press.
- [11] Pennebaker, J.W., Boyd, R.L., Jordan, K. and Blackburn, K., 2015. *The development and psychometric properties of LIWC2015*. UT Faculty/Researcher Works.
- [12] Lin, C.Y. and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (Edmonton, Canada, May 27-June 1, 2003). Association for Computational Linguistics, 71-78
- [13] Kittur, A., Nickerson, J.V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M. and Horton, J. 2013. . The future of crowd work. In *Proceedings of the Conference on Computer Supported Cooperative work* (Antonio, TX, February23-27, 2013), ACM, 1301-1318
- [14] Kim, J., Cheng, J. and Bernstein, M.S. 2014. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative work & Social Computing* (Vancouver, BC, March 14-18, 2014). ACM, 745-755
- [15] Kim, J. and Monroy-Hernandez, A., 2015. *Storia: Summarizing social media content based on narrative theory using crowdsourcing*. arXiv preprint arXiv:1509.03026.
- [16] Matias, J.N. and Monroy-Hernandez, A., 2014, . NewsPad: Designing for collaborative storytelling in neighborhoods. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems* (Totonto, Canada, April 26-May 01, 2014). ACM, 1987-1992.
- [17] Agapie, E. and Monroy-Hernandez, A., 2015. Eventful: Crowdsourcing Local News Reporting. arXiv preprint arXiv:1507.01300.
- [18] Landauer, T. K., McNamara, D., Dennis, S., and Kintsch, W. (Eds.). (2007). Handbook of latent semantic analysis. Mahwah, NJ: Erlbaum.
- [19] Li, H., Shubeck, K., and Graesser, A. C. (2016). Using technology in language assessment. In D. Tsagari and J. Banerjee (Eds.), Contemporary second language assessment. London, UK: Bloomsbury Academic.
- [20] Landauer, T.K., Laham, D. and Foltz, P.W. 2003. Automatic essay assessment. *Assessment in Education: Principles, Policy & Practice, 10*(3), 295-308.
- [21] Burstein, J. 2003. The E-rater scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis and J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective*. Mahwah, NJ: Erlbaum, 113-122.
- [22] Nenkova, A. and Passonneau, R., 2004. Evaluating content selection in summarization: The pyramid method.
- [23] Graesser, A. C., & McNamara, D. S. (2011). Computational analyses of multilevel discourse comprehension. *Topics in Cognitive Science, 3*, 371-398. DOI=[10.1111/j.1756-8765.2010.01081.x](https://doi.org/10.1111/j.1756-8765.2010.01081.x).
- [24] Graesser, A.C., McNamara, D.S., Cai, Z., Conley, M., Li, H. and Pennebaker, J., 2014. Coh-Metrix measures text characteristics at multiple levels of language and discourse. *The Elementary School Journal, 115*(2), 210-229. DOI=[10.1086/678293](https://doi.org/10.1086/678293).
- [25] Li, H., Graesser, A.C., Conley, M., Cai, Z., Pavlik Jr, P.I. and Pennebaker, J.W., 2015. A New Measure of Text Formality: An Analysis of Discourse of Mao Zedong. *Discourse Processes, 1*-28. DOI=[10.1080/0163853X.2015.101011](https://doi.org/10.1080/0163853X.2015.101011).

# Beyond Log Files: Using Multi-Modal Data Streams Towards Data-Driven KC Model Improvement

Ran Liu  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
ranliu@cmu.edu

Jodi Davenport  
WestEd  
300 Lakeside Drive, 25<sup>th</sup> Floor  
Oakland, CA 94612  
jdavenport@wested.org

John Stamper  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
jstamper@cs.cmu.edu

## ABSTRACT

The increasing use of educational technologies in classrooms is producing vast amounts of process data that capture rich information about learning as it unfolds. The field of educational data mining has made great progress in using log data to build models that improve instruction and advance the science of learning. Thus far, however, the predictive and explanatory power of such models has often been limited to the actions that educational technologies can log. A major challenge in incorporating more contextually rich data streams into models of learning is collecting and integrating data from different sources and at different grain sizes. We present our methodological advances in automating the integration of log data with additional multi-modal (e.g., audio, screen video, webcam video) data streams. We also demonstrate several examples of how integrating multiple streams of data into the knowledge component (KC) model refinement process improves the predictive fit of student models and yields important pedagogical implications. This work represents an important advancement in facilitating the integration of rich qualitative details of students' learning contexts into the quantitative approaches characteristic of EDM research.

## Keywords

Multi-Modal Data Analytics, KC Model Improvement, Log Data, Structured Event Analysis of Multiple Streams (SEAMS)

## 1. INTRODUCTION

As student learning becomes increasingly conducted on computers and other digital devices, vast amounts of learning-related data are produced. Ideally, such data will provide a rich picture of student knowledge and behaviors (e.g., [8]). But predicting performance and generating pedagogical insight is limited, in the majority of cases, to the actions that digital systems can log. Computerized tutors are often used in a classroom context, and log data cannot capture all learning phenomena. A student working at a computer might be working independently with few outside influences. Alternatively, she might be in a lively classroom, with other students around her, talking and even offering suggestions. Data that capture the context surrounding educational technology use may add to and complement log data. In some cases, it may lead to critical insights.

Educational data mining analyses often omit additional contextual data for a number of reasons. Data on classroom context are difficult to collect. Data from different sources are often collected at different grain sizes, which are difficult to integrate. Here, we present work that extends educational data mining techniques to incorporate multiple modalities of data (computer log files, audio, screen videos, and webcam videos). We present methods we developed that help streamline both the collection of additional

streams of data and the linkage across multiple streams. In two experiments, we then demonstrate the value of incorporating multi-modal, contextually rich data streams into established educational data mining techniques. In the first experiment, students use a chemistry virtual lab tutor and, in the second, students use an intelligent tutoring system to collaborate on fraction arithmetic.

Specifically, we extend methods of data-driven knowledge component (KC) model refinement [17] by incorporating, into the process, multiple streams of data spanning different modalities. We show that KC model improvements uniquely derived from these additional data beyond log files led to improved predictive models of student learning and behavior. These improved models of learning, in turn, can generate actionable knowledge for systems, students, teachers, and researchers.

## 2. BACKGROUND

### 2.1 Related Work

Recent work reflects a growing interest in multi-modal data analytics, particularly surrounding project-based, constructionist, and/or informal learning contexts [4, 18]. These efforts have focused on capturing divergent student strategies [4] and interactions that happen outside of a traditional computer tutor environment (e.g., with peers and with the physical environment [16]). Their primary goal is to make technologies supporting open-ended learning environments more scalable and to develop assessments appropriate for this type of learning.

Areas of research within the EDM community have also focused on collecting sources of data computer logs cannot capture to serve as "ground truth" labels in training log-data based detectors. These efforts have largely focused on modeling and detecting students' motivational and affective states [2, 8, 15]. For example, models can detect patterns of log data activity that precede affective states like confusion, frustration, and boredom. Physiological data may also be collected and used to develop models that can detect affective states from machine-readable signals, such as facial features, body movements, and electrodermal activity [14].

Outside of these pockets of the community, though, the majority of EDM research has focused exclusively on using log data to model learning. Building statistical models to predict step-level performance and data-driven KC model (or Q-matrix [3]) discovery are examples of major branches of EDM research that are typically limited to computer-logged data. In the present work, we demonstrate the value of expanding EDM research to include additional data streams that convey important contextual information about students' learning. We also present methodological advancements that improve the ease with which

additional data streams can be collected and incorporated into educational data mining methods more broadly.

## 2.2 Data-Driven KC Model Improvement

Knowledge component models are an important basis for the instructional design of automated tutors and are important for accurate assessment of learning. Knowledge components (KCs) refer to units of knowledge representation (e.g., facts, concepts, or skills) that students need in order to solve problems. A KC Model maps a set of KCs mapped to a set of items or problem steps. Student models that are based on more accurate KC models produce better predictions of what a student knows based on their performance and, thus, result in better assessment and improved learning and instruction [11]. Cognitive Task Analysis is the traditional method for creating cognitive models of learning, but it requires subjective decisions and large amounts of human time and effort. Data-driven techniques of KC model discovery and refinement, when applied to large sets of educational data, can provide both more objectivity and reduce human effort.

A method developed by [17] leverages tools available in the PSLC DataShop [10] to identify potential improvements to a KC model in a data-driven manner. This method iterates through the following steps: (1) inspect learning curve visualizations and best-fitting statistical parameter estimates for the best existing KC model, (2) identify problematic KCs, (3) hypothesize changes to the KC model based on examining constituent problem content and applying domain expertise, and (4) re-fit the statistical model with the revised KC model and assess improvements in predictive accuracy. The premise for this method is that a hallmark of learning on a well-defined KC is a smooth learning curve that shows monotonic improvement in performance over time. KCs that lack these learning curve characteristics, but not because students are at ceiling performance, are likely to involve certain problem steps that require unlabeled difficulty factors or knowledge demands.

After a problematic KC is identified, its constituent problem steps must be examined in order to identify potential hidden difficulty factors. Thus far, this part of the process is limited to what computer log data. For example, a researcher might examine the error rates of the different constituent problem steps for the KC in question and the problem step names to gain clues about hidden difficulties. In the best-case scenario, the researcher might have access to the actual problem content for the dataset (as in [17]) and can apply domain knowledge to identify potential KC modifications. This step of content examination can be greatly enriched by additional streams of contextually rich data from the relevant moments of learning. To this end, we present a method of integrating streams of contextual audio and video data into the KC model refinement process. We show that such integration leads to insights that would not be derived by solely analyzing log data or curriculum content in isolation. We present several examples of how these insights lead to quantitative KC model improvements that improve the overall fit of student models to the data.

## 3. METHODS

We developed a method of semi-automatically extracting epochs, across multi-modal data streams, associated with the moments during which students engage with a particular KC of interest. This allows the content reviewer, after identifying a candidate KC, to not only view the curriculum content associated with a given KC but also to experience students engaging with that curriculum content through multiple modalities.

There are many ways to collect additional streams of contextually rich data (e.g., using video cameras, external microphones, eye-trackers, and sensors). We focused on a method that minimizes both deployment effort and interference with students' usage of educational technology to increase the likelihood that researchers would consider collecting, analyzing, and sharing such data. In the following experiments, we used Camtasia to simultaneously capture audio recordings, screen videos, and webcam videos of the students. Camtasia can be run in the background to collect all of these streams of data while a student engages with educational software. We installed Camtasia to all classroom laptops in advance of the two studies. On each day of the studies, we opened Camtasia and prepared recording settings before each class period so that all students needed to do was click a red "Record" button prior to logging into the tutors. At the end, students were led through a simple sequence of steps to ensure that their recordings were saved and named properly for easy post-hoc identification.

All recordings (audio, screen video and webcam video) for a single session are initially saved in a Camtasia-specific file format. We used the batch processing function to import and convert the original files to MP4 files that contained all data streams merged. We used timestamp information within the log files to map segments of log data to the appropriate corresponding multi-modal video stream. This step required human input, as Camtasia does not automatically log the system time (at millisecond level) that marks the start of the video recording. For each video file, someone must identify the offset between the beginning of the video and the time of some event in the log file. This offset can then be used to automatically align all remaining events between the log file and the corresponding video files.

We developed a tool called Structured Event Analysis of Multiple Streams (SEAMS) that builds upon the moviepy Python package in conjunction with the FFMPEG multimedia framework to automatically extract video epochs associated with specific events in the log data. The tool allows the user to indicate any event type that can be identified by labels within the log data and generates a folder of video clips that contain all epochs of the merged data streams pertaining to the particular event of interest (in this case, a specific KC at the specific opportunity count). With the relevant epochs grouped together in a manner that allows for quick and effortless analysis by a human examiner, it becomes much easier to quickly view multi-modal data streams to identify hidden knowledge demands towards KC refinement.

We applied our methods to examine the contributions of additional multi-modal data streams on KC model refinement across two classroom experiments. One experiment engaged students in a Chemistry Virtual Lab tutor for which we collected both screen videos and webcam data of learners' facial expressions in addition to traditional log data. The other experiment engaged students in a Collaborative (partner-based) Fraction tutor, and we collected screen videos and audio recordings of students' collaborative dialogue. Due to processor limitations of the school laptops that were available for the Collaborative Fraction tutor experiment, we were not able to collect webcam data. Using the data from both of these studies, we illustrate the application of our methods to leverage the additional multi-modal data streams to improve upon existing KC models. These KC model improvements, in turn, yielded insights about how to improve instruction within the respective tutors.

## 4. EXPERIMENTS

### 4.1 Chemistry

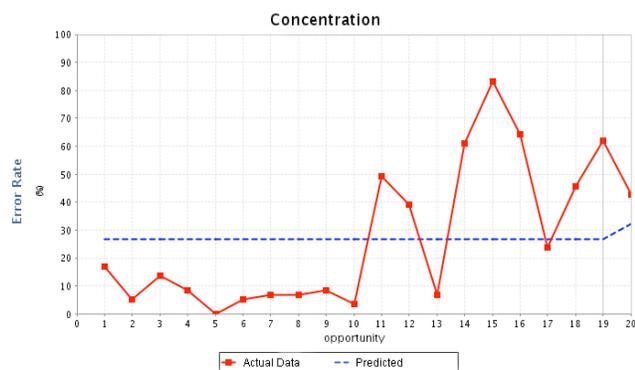
ChemVLab+ (chemvlab.org) provides a set of high school chemistry activities designed to build conceptual understanding and inquiry KCs [6]. In each activity, students work through a series of tasks to solve an authentic problem and receive immediate, individualized tutoring. As students work, teachers are able to track student progress throughout the activity and attend to students that may be lagging behind. Upon completion of the activities, students receive a report of their proficiency on targeted KCs, and teachers can view summary reports that show areas of mastery or difficulty for their students. In the current study, students completed four modules: PowerAde: Using Sports Drinks to Explore Concentration and Dilution, The Factory: Using a City Water System to Explore Dilution, Gravimetric Analysis, and Bioremediation of Oil Spills.

#### 4.1.1 Participants

Participants were 59 students at a high school in the greater Pittsburgh area enrolled in honors chemistry classes. They participated in four Stoichiometry modules of the ChemVLab+ educational tutor. They completed these modules across four 50-minute class periods spread over the course of 3 weeks. We collected, using Camtasia, audio recordings and screen video captures for 58 students and webcam recordings of facial expressions for a subset of 25 students who were comfortable with their face being recorded during tutor use.

#### 4.1.2 Results

The newly developed methods facilitated the identification of the way in which a problematic KC needed to be split as well as technical issues that impacted student learning. First, following methods described in [17], we identified a knowledge component called *Concentration* that seemed to have uncharacteristically high error rates on later practice opportunities (Figure 1). This KC represents understanding that the measure of concentration is the amount of substance (e.g., a sports drink powder) in a volume of substrate (e.g., water). It also represents being able to read, report, and compare concentrations of solutions.

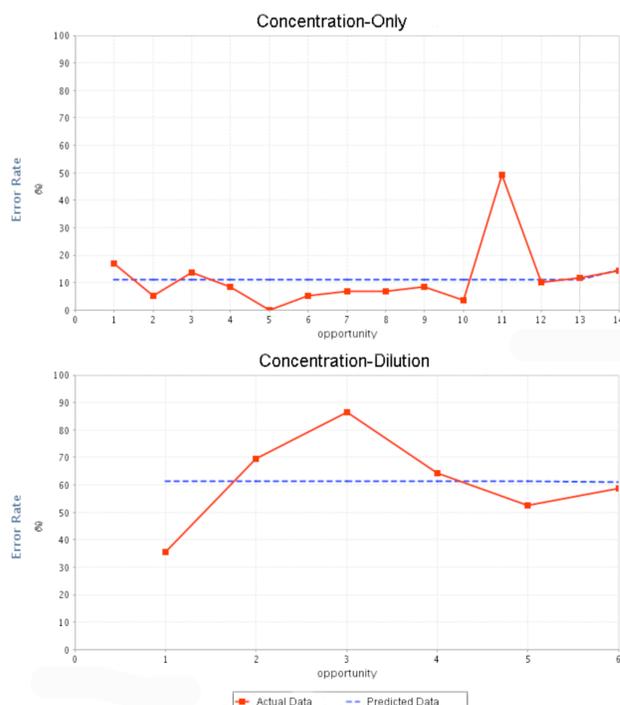


**Figure 1.** Aggregate learning curve for the *Concentration* KC as originally defined by the ChemVLab+ tutor.

We then used the methods described in Section 3 to automatically extract the screen and webcam videos of all epochs of students engaging with the Concentration KC on their 11<sup>th</sup>, 12<sup>th</sup>, 14<sup>th</sup>, 15<sup>th</sup>, 16<sup>th</sup>, and 19<sup>th</sup> practice opportunities. These were the opportunities on which the KC learning curve had unusually high error rates.

Qualitative analyses of these video stream epochs revealed that students were particularly confused by problems that involved

dilution in conjunction with concentration, particularly when a dilution ratio or “factor” is involved. Students demonstrated this confusion as they responded to prompts such as ‘Create a 1:2 dilution of the reported sample’ or ‘Add water to the sample until the concentration is diluted by a factor of 2’. The correct solution requires students to know that the amount of substance (e.g., the powder) takes up negligible volume, so to dilute the powder by 2x, the total amount of water needs to be doubled. Students demonstrated shallow knowledge by responding to prompts like these by adding two parts water to one part solution rather than adding one part water to one part solution, which halves the concentration. In another example, prompt ‘Dilute this sample by a ratio of 6:1’ student tended to add six parts water to one part of solution (making the resulting amount of powder to volume 1:7), part rather than adding five parts of water to one part of solution (making the resulting amount of powder to volume 1:6).

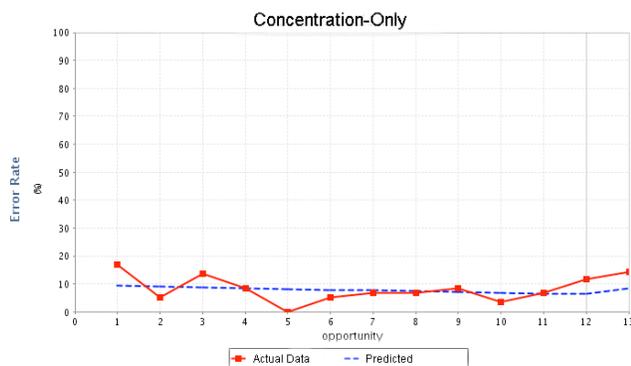


**Figure 2.** Aggregate learning curves for the two new KCs, *Concentration-Only* and *Concentration-Dilution*, resulting from the KC model refinement process.

Based on this insight, we split the Concentration KC into cases where the problem step required a conceptual understanding of dilution ratios/factors (Concentration-Dilution) and cases where it did not (Concentration-Only). The learning curves for the resulting two KCs are shown in Figure 2. The curves are much smoother than the original learning curve, with the exception of a particular opportunity count with unusually high error rate in the resulting ‘Concentration-Only’ KC at practice opportunity 11.

To further examine this unusual blip, we re-applied our method to automatically extract screen and webcam videos of all epochs of the 11<sup>th</sup> opportunity to practice the Concentration-Only KC. We noticed that the majority of problem steps experienced by students on this opportunity count were from a particular screen in the tutor in which the problem text was cut off in the interface. This resulted in students being confused about what they should be doing on this problem. Guessing the answer incorrectly was a common first attempt, as was clicking a hint button. Since the

problem text was fine when viewed on research computers, it did not appear to be a problem with the educational software itself. We hypothesize that the problem may have been due to a unique interaction between the software and the resolution of the computers that students were working on. This is a reality of educational technology deployment in classrooms, and it would have been impossible to know from strictly the log data file or even problem content records that this was the source of students' struggle. If we had only accessed the recorded (idealized) version of the problem content, we may have incorrectly attributed the high error rate on this problem step to intrinsic content present within the problem. After separating these problem steps out from the 'Concentration-Only' KC, the resulting learning curve was much smoother, with an overall low error rate (Figure 3).



**Figure 3. Resulting *Concentration-Only* learning curve, after separating out the problem step in which students experienced a technical difficulty during deployment.**

Student model predictive fit metrics are shown in Table 1 for the different KC models when used in conjunction with the Additive Factors Model [5] and reveal an improvement in predictive fit across all metrics (AIC, BIC, and 10-fold cross validation) after splitting the original Concentration KC based on our qualitative analysis of student behavior during epochs of that KC (Row 2). Further improvements in predictive fit across all metrics were observed after we separated out the problem step that contained missing problem text during implementation (Row 3).

**Table 1. Student model fit metrics comparing different models resulting from the KC model refinement process.**

	AIC	BIC	Cross Validation RMSE
Original KC model	6694.58	7196.59	0.3859
'Concentration'-Split KC Model	6388.35	6904.12	0.3838
'Concentration'-Split KC Model with text-error problem step separated	6318.95	6848.47	0.3819

Both of these KC model refinements, each of which resulted in a substantive and consistent improvement in predictive accuracy when used by the Additive Factors Model, were uniquely dependent on qualitative analyses of the video data we had collected using Camtasia. Although it may have been possible to recognize that the concept of dilution ratios was an additional difficulty factor by purely accessing problem content, there were many other differences between the high error-rate problem steps

and the low error-rate problem steps that constituted the original Concentration KC. For example, many of the higher error rate problem steps were part of a different activity (Activity 2, The Factory) than the lower error rate problem steps were (Activity 1, Powerade). Only by observing the students specifically exhibiting actions suggestive of possessing a shallow understanding of dilution ratios (via Camtasia screen videos) and affective states resembling frustration (via webcam videos) were we able to quickly identify the true hidden difficulty factor. Another benefit of this insight, perhaps even more significant than generating a better fitting KC model, is that there are clear implications for instructional redesign. That is, future iterations of the ChemVLab+ tutor might include instruction that more directly targets the misconceptions that students seem to have about the relationship between dilution ratios and existing solutions.

Discovering the high-error-rate problem step in which text was cut off would not have been possible without viewing the real context in which students experienced the problem. Since it was not a general problem with the ChemVLab+ tutor but, rather, an idiosyncrasy in that problem's display on the technology used in the classroom, the Camtasia screen videos were critical in correctly attributing the source of these errors.

## 4.2 Collaborative Fraction Tutor

The collaborative fraction tutor is online software developed by researchers at Carnegie Mellon University that helps students become better at understanding and working fractions. The tutor was created using Cognitive Tutor Authoring Tools, which allow for rapid development and easy deployment of intelligent tutors [1]. This particular fraction tutor supports collaboration between partners in order to learn fraction-solving KCs such as addition, subtraction, comparing fractions to determine which is larger or smaller, finding the least common denominator, and finding equivalent fractions. In the tutor, each student in a pair can control only part of the screen, so both partners must work together in order to finish the problem. One student cannot do the whole thing him or herself. Students work at the same time and can talk about what they are doing, ask for help from their partner, and generally collaborate to get the correct answer.

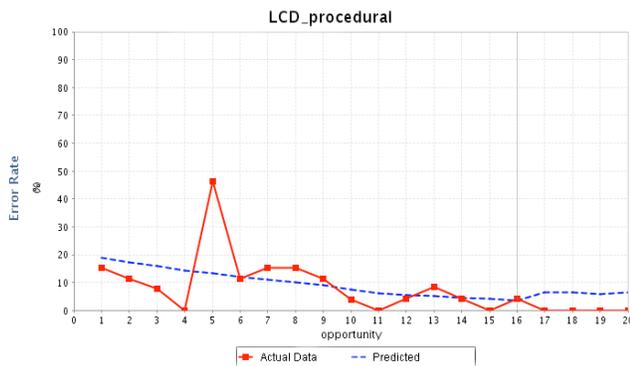
### 4.2.1 Participants

Participants were 26 fifth grade students at a middle school in the greater Pittsburgh area enrolled in an advanced math class. Students participated across five 45-minute class periods on consecutive days within a week. On the first and last days, students took a computerized pre- and post-test, respectively. They engaged in the Collaborative Fraction Tutor during the three consecutive days between the pre- and post-test days. Students spent half of each class period working individually and half collaborating with a partner. Students were paired with the same person for all partner activities throughout the experiment. We also collected audio and screen video captures for all students working both individually and in pairs on the three tutor use days.

### 4.2.2 Results

The newly developed methods facilitated the identification of KCs that needed to be split. First, as in [17], we identified a knowledge component called *LCD\_procedural* that was noisy, in particular due to an uncharacteristically high error rate on the 5<sup>th</sup> practice opportunity (Figure 4). We then used the methods described in Section 3 to automatically extract the combined audio and screen videos of all epochs of students engaging in their 5<sup>th</sup> opportunity of the *LCD\_procedural* KC. Based on qualitative analyses of the

video and audio streams, it was clear that the most common mistake that students were making on those practice opportunities was multiplying the two denominators but failing to reduce the product to find the least common multiple. This was particularly apparent in students' collaborative dialogue following their incorrect first attempts. Students often verbalized the realization that there must be a smaller common multiple. This verbalization did not occur on problems in which the product of denominators happened to be the correct solution. This suggests that there was a separate learning curve for the additional difficulty factor of cases where finding the least common denominator required reducing the product of the two original fractions' denominators to find a smaller common multiple. Based on this, we split the LCD\_procedural KC into cases where the LCD required reducing from the product of denominators (LCD\_procedural\_REDUCE) and cases where it simply was the product of denominators (LCD\_procedural\_PRODUCT). The resulting learning curves (Figure 5) are much smoother than the original learning curve.



**Figure 4.** Aggregate learning curve for the *LCD\_procedural* KC as originally defined by the Collaborative Fraction tutor.

The student model predictive fit metrics (Table 2) for the different KC models, when used in conjunction with the Additive Factors Model, reveal a substantial improvement in predictive fit across all metrics (AIC, BIC, and 10-fold cross validation) after splitting the original LCD\_procedural KC based on our qualitative analysis of student behavior during epochs of that KC.

Through the audio-video segments, we observed students make denominator-product-based errors on their incorrect first attempts and realize they needed to find a smaller common multiple on certain problem steps. This greatly streamlined our identification of the hidden difficulty factor. As a result, we were able to quickly

identify the appropriate KC split that led to much smoother learning curves and a better fitting student model.

This discovery also has important instructional implications: for example, the tutor might incorporate a bug message specific to students' inputting the product of the two denominators when the answer is a smaller multiple (i.e., "Can you find a smaller number that divides both denominators?"). A student model based on the revised KC model (with 'LCD\_procedural' split into two separate KCs) would also result in students receiving more practice on problems in which the correct answer is a smaller multiple than the product of the two denominators. These instructional changes, resulting from the audio dialogue and video driven insights, will give students better support to overcome this difficulty.

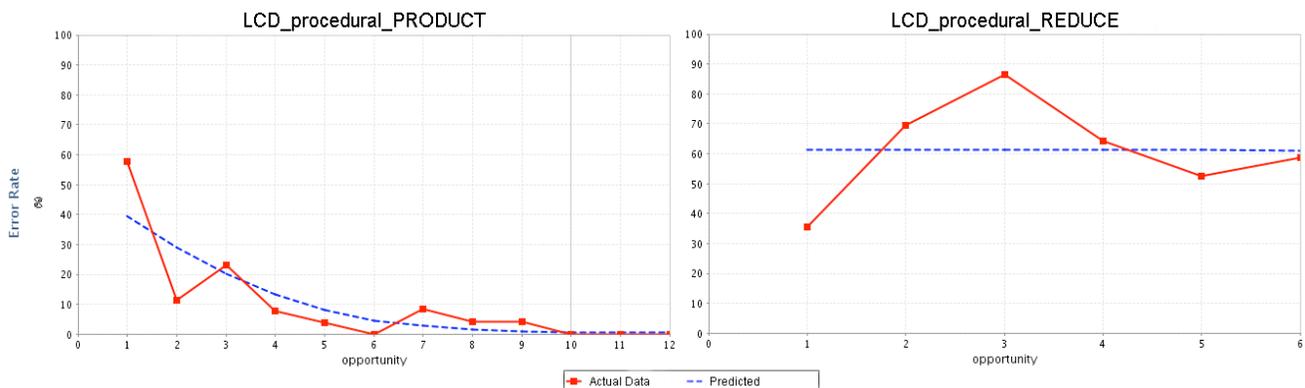
**Table 2.** Student model fit metrics compared between the original KC model and the improved KC model resulting from multi-modal data stream driven refinement process

	AIC	BIC	Cross Validation RMSE
Original KC model	3497.6	4156.3	0.2738
'LCD_procedural' split KC model	3462.2	4134.5	0.2734

## 5. DISCUSSION & FUTURE WORK

The vast majority of EDM research, especially research focused on predicting student performance and generating pedagogical insights, is limited to models based on computer-logged data. A recognized issue within the EDM community is that log data cannot capture all learning phenomena; it can miss important details of both learning processes and the learning context. Recent advances in DataShop [10] allow researchers to connect problem names in log data to screenshots of problem content and encourage inclusion of contextual details in custom fields of log data. Clearly, however, there are still instances where a better understanding of the implementation environment and students' experience working through certain problem steps is needed, as demonstrated here.

The main contributions of this work are (1) developing methodological advancements (e.g., the SEAMS tool) that facilitate the ease with which EDM researchers can incorporate context-rich data streams into quantitative modeling techniques, and (2) demonstrating the utility of doing so. Using a top-down,



**Figure 5.** Aggregate learning curves for the two new KCs, *LCD\_procedural\_PRODUCT* and *LCD\_procedural\_REDUCE*, resulting from our KC model refinement process5.

KC visualization driven method, we show that valuable qualitative insights can be obtained from targeted segments of audio and video data even without fully “coding” all of the multiple streams. We also show that these qualitative insights lead to quantitative model fit improvements and actionable pedagogical implications.

There are many promising areas for future work based on the methods we have developed here. The present work has focused on refining an existing KC model. Educational data does not always come with an existing expert-labeled KC model, and there have been recent efforts to automatically generate, or discover, KC models [9, 12, 13]. One concern about fully machine-discovered models is their interpretability. The ability to view contextually-rich audio and video segments corresponding to machine-discovered KCs will facilitate the interpretation of these KCs and, in turn, help researchers refine their methods to yield more interpretable or cognitively plausible KC models.

Another interesting issue that contextually-rich streams of data are uniquely suited to address is the attribution of pauses of activity in the log data. A pause in the data because a student is off-task has very different implications than a pause because the student is actively help-seeking outside of the educational technology interface. Being able to use detailed information about students’ learning context can help produce correct interpretations of log data activity and, in turn, more robust student models.

Finally, one of the interesting data streams we collected in the Chemistry dataset was student-facing webcam video. Aside from noticing the moments during which students seemed frustrated in the Chemistry tutor due to confused about dilution ratios, we have not yet fully explored the extent to which the webcam data could be used to improve KC models and student models. There is rich potential for our methods to facilitate connections between the cognitive (e.g., knowledge component modeling) and the affective [2, 8] branches of EDM research.

## 6. ACKNOWLEDGMENTS

We thank Jacklyn Powers and Jenny Olsen for help in collecting the Chemistry and Math tutor data used in our experiments here. This research was supported by the National Science Foundation (Grants #DRL-1418072, PI Davenport and #DRL-1418181, PI Stamper) and the Institute of Education Sciences (Training Grant R305B110003 to Liu). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF or IES.

## 7. REFERENCES

- [1] Aleven, V., Sewall, J., McLaren, B.M., and Koedinger, K.R. (2006). Rapid authoring of intelligent tutors for real-world and experimental use. In *Proceedings of the 6th ICALT*. IEEE, Los Alamitos, CA, pp. 847-851.
- [2] Baker, R.S., Corbett, A.T., Roll, I. and Koedinger, K.R. (2008). Developing a generalizable detector of when students game the system. *UMUAI*, 18(3), pp. 287-314.
- [3] Barnes, T. (2005). The q-matrix method: Mining student response data for knowledge. In *Proceedings of the AAAI-EDM Workshop*, pp. 978-980.
- [4] Blikstein, P. (2013). Multimodal learning analytics. In *Proceedings of the 3rd International Conf. on LAK*. ACM, New York, NY, pp. 102-106.
- [5] Cen, H., Koedinger, K.R., and Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In *Proceedings of the 8th International Conference on ITS*, pp. 164-175. Berlin: Springer-Verlag.
- [6] Davenport, J., Rafferty, A., Timms, M., Yaron, D., and Karabinos, M. (2012). ChemVLab+: Evaluating a virtual lab tutor for high school chemistry. In *International Conf. of the Learning Sciences*.
- [7] D’Mello, S.K., and Calvo, R. (2011). Significant Accomplishments, New Challenges, and New Perspectives. In R. A. Calvo and S. D’Mello (Eds.), *New Perspectives on Affect and Learning Technologies*. New York: Springer, pp. 255-272.
- [8] Graesser, A.C., Conley, M., and Olney, A. (2012). Intelligent tutoring systems. In K.R. Harris, S. Graham, and T. Urdan (Eds.), *APA Educational Psychology Handbook: Vol. 3. Applications to Learning and Teaching*. Washington, DC: American Psychological Association, pp. 451-473.
- [9] Gonzalez-Brenes, J.P., and Mostow, J. (2012). Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. In *Proceedings of the 5th International Conf. on EDM*.
- [10] Koedinger, K.R., Baker, R.S., Cunningham, K., Skogsholm, A., Leber, B., and Stamper J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In Romero C, Ventura S, Pechenizkiy M, Baker RSJd (Eds.), *Handbook of Educational Data Mining*. Boca Raton, FL: CRC Press.
- [11] Koedinger, K.R., Stamper, J.C., McLaughlin, E.A., and Nixon, T. (2013). Using data-driven discovery of better cognitive models to improve student learning. In *Proceedings of the 16th International Conf. on AIED*.
- [12] Lan, A.S., Studer, C., Waters, A.E., and Baraniuk, R.G. (2014). Sparse Factor Analysis for Learning and Content Analytics. *Journal of Machine Learning Research*, 15, pp. 1959-2008.
- [13] Lindsey RV, Khajah M, Mozer MC. (2014). Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in Neural Information Processing Systems*, 27, pp. 1386-1394.
- [14] Picard, R., Vyzas, E., and Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), 1175-1191.
- [15] Porayska-Pomsta, K, Mavrikis, M, D’Mello, S.K., Conati, C., and Baker, R. (2013). Knowledge elicitation methods for affect modelling in education. *IJAIED*, 22, 107-140.
- [16] Schneider, B., and Blikstein, P. (2014). Unraveling Students’ Interaction Around a Tangible Interface Using Gesture Recognition. In *Proceedings of the 7th Annual International Conf. on EDM*.
- [17] Stamper, J. and Koedinger, K.R. (2011). Human-machine student model discovery and improvement using DataShop. In *Proceedings of the 15th International Conf. on AIED*.
- [18] Worsley, M. (2012). Multimodal learning analytics: enabling the future of learning through multimodal data analysis and interfaces. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, pp. 353-356.

# Seeking Programming-related Information from Large Scaled Discussion Forums, Help or Harm?

Yihan Lu

School of Computing, Informatics & Decision Systems  
Engineering, Arizona State University,  
699 S. Mill Ave., Tempe AZ, USA  
lyihan@asu.edu

I-Han Hsiao

School of Computing, Informatics & Decision Systems  
Engineering, Arizona State University,  
699 S. Mill Ave., Tempe AZ, USA  
Sharon.Hsiao@asu.edu

## ABSTRACT

Online programming discussion forums have grown increasingly and have formed sizable repositories of problem solving-solutions. In this paper, we investigate programming learners' information seeking behaviors from online discussion forums. We design engines to collect students' information seeking processes, including query formulation, refinement, results examination, and reading processes. We model these behaviors and conduct sequence pattern mining. The results show that programming learners indeed seek for programming related information from discussion forums by actively searching on the site and reading posts progressively according to course schedule topics. Advanced students consistently perform query refinements, examine search results and commit to read, however, novices do not. In addition, advanced students commit to read posts, but novices only skim.

## Keywords

Programming; Information Seeking; Hidden Markov Model; Discussion Forums; Sequential pattern mining;

## 1. INTRODUCTION

In teaching and learning programming, students are typically asked to refer to API (Application Programming Interface) or programming textbooks for relevant information (i.e. code syntax or code examples). In recent years, open & free online communities (such as homework-help sites, discussion forums for MOOCs courses etc.) have grown increasingly and have formed sizable repositories of problem solving-solutions. They are filled with thousands of programming problem-solving tips, such as "how-to" questions [1], people-valued examples, and the examples' explanations [2] etc. On the other hand, from a constructive point of view, the action of articulating a problem and initiating search or referencing can also be a valuable learning activity as well as browsing the solution. In software engineering field, such programming information seeking has already been recognized as a core sub-task in software maintenance [3, 4]. Programmers are even being referred as task-oriented information seekers, which they focus on finding the answers they need to complete a task using a variety of information sources [5]. There are tools that have been built to make completing programming tasks easier, such as Mica [6]. However, none of these tools focuses on amplifying learning opportunities if any, rather, centers on task-oriented problem solving facilitation.

In addition, according to Information Foraging theory [7], finding information is human nature. To successfully form information seeking criteria for a given programming problem requires complex cognitive activities (i.e. defining and verbalizing the programming problem; refining query criteria and selecting

results; strategies application etc.) To better support information seeking and learning, we focus on learners' behaviors in seeking programming-related information. Specifically, we investigate in an online large-scale discussion forum, StackOverflow, which is one of the biggest online programming Q&A sites communities and currently hosts a massive amount of heterogeneous definitions, solutions and examples of programming languages. Are those assorted content in the forum helpful or harmful for programming learners?

Studies have shown that while there is a positive connection between the usage of StackOverflow and GitHub (open source code management service), StackOverflow's users consider the site to be more attractive and beneficial for learning programming [8]. In recent learning science literature, learning-from-observing paradigm appears to be a promising strategy, which passive participants (such as lurkers who consume content without contributions) can still learn by reading the postings-and-replies exchanges from others due to the constructive responses in the content [9]. Knowledgeable students can benefit from text with cohesive gaps by making active retrieval and inferences [10]. They can also benefit from building memory and fluency through the active retrieval opportunities and to refine the conditions of application through feedback on incorrect solution attempts in problem solving [11]. On the other hand, novices may benefit from seeing examples of solution steps and from seeing the entire solution structure to make sense of the role of each step in order to construct integrated knowledge components for generating plans and sub goals [12]. In this work, our goal is to investigate what are programming learners' tactics in searching for relevant information from online discussion forums and how do they look for relevant learning materials from massive forum posts.

In this paper, we design engines to capture programming learners' activities on StackOverflow site, such as problem verbalization in queries, query revision and other information seeking processes. We collect a semester long of *informal* programming learning activities from programming discussion forum. We model their information seeking activities by using Hidden Markov Model and data mine the post of their readings.

## 2. LITERATURE REVIEW

### 2.1 Modeling Information Seeking In Learning

Traditionally, information seeking is associated with behavioral science theories, which focus on seekers' information needs, searching strategies, and how they use the information. For example, self-awareness of one's information needs, self-regulated learning strategies, information searching experience and ability, etc.[13-15]. Puustinen and Rouet [13] further

classified help-seeking behavior into different types on a help-seeking continuum, a function of the helpers' capacity to adapt answers to their needs. In more recent information seeking literature, we see studies show that users commonly exhibit exploratory behavior in a great extent when performing searches [14]. Marchionini [15] identifies a range of search activities that differentiate exploratory search from look up search (i.e. fact-finding retrieval). Such behavior is especially pertinent to learning and investigating activities, which is the targeted area of interest in our research.

## 2.2 Modeling Learning From Discussion Forums

Over the decades, data mining on discussion forums has been carried out through various formats, network analyses, topical analyses, interactive explorers, knowledge extraction, etc. [16-18]. Due to calculation complexities (since linguistic features rely on computer processing power), most of these in-depth analyses were performed offline [19, 20]. As a result, the lesson learned could only be applied in the next iteration of system development. Recently, however, we begin to see some studies that focus on dynamic support for users [21]. With the rapid growth of free, open, and large user-based online discussion forums, it is essential, therefore, for education researchers to pay more attention to emerging technologies that facilitate learning in cyberspace. For instance, Wise, Speer, Marbouti, and Hsiao [22] studied an invisible behavior (listening behavior) in online discussions, where the participants are students in a classroom instructed to discuss tasks on the platform; van de Sande & Leinhard [23] investigated online tutoring forums for homework help, making observations on the participation patterns and the pedagogical quality of the content; Hanrahan, Convertino & Nelson [24] and Posnett, Warburg, Devanbu, & Filkov [25] studied expertise modeling in a similar sort of discussion environment.

## 3. METHODOLOGY

### 3.1 Research Platform & Data Collection

In this project, we deployed a Chrome browser plugin to track users' query, searching, and reading behaviors on StackOverflow (SO). User can search query on StackOverflow and identify their intention with this tool. The browser plugin has two main features. (1) It provides a direct search channel for users to issue queries on StackOverflow; (2) It displays users' search histories. We collect not only users' search queries, but also their search intentions, including "Knowledge seeking", "Method learning", "Problem solving", and "Other" (indicated by the user). Most importantly, we log all the users' behaviors, comprising of scrolls, clicks, selections, and corresponding actions' time. The behavior tracking function resides on StackOverflow site once initial log in via the SO search tool. In another word, all students' behaviors on StackOverflow site will be logged after at least one time log in via SO Search Tool. However, since they issue the queries directly from StackOverflow site, their intention will be marked as "not specified".

### 3.2 Study Setup

In order to understand the students' information seeking behaviors on discussion forums, we conducted a user study in a programming class in Arizona State University. Students were encouraged to install the browser plugin search tool. They were told that their search activities would be collected via the tool. All students' programming information seeking behavior was logged during the entire semester.

Additionally, we also conducted a controlled session of lab class during the semester. In the lab class, students were instructed to solve a complex task (implement a 3-way merge sort algorithm) by using the information-seeking tool within 75 minutes. All the students' searching and reading behaviors on StackOverflow were recorded.

Students were given a pretest to examine their pre knowledge about programming. In this study, the students are split into two groups (*Novice & Advanced*) based on their pretest median score, which is ranged from 0 to maximum score 20.

### 3.3 Data Descriptive

Among 86 students in the Object-Oriented Programming class, 71 students voluntarily installed our search plugin, whose operations on SO were automatically recorded, 55 of them also used the plugin to search queries. There were 44 of them took the pretest. According to their pretest score distribution, 24 of them were identified as novices, and 20 were classified as advanced students.

#### 3.3.1 Query data log

For these 55 students provided query information, the average query number is 9.55 (max 56, min 1, median 8), and the average number of operations is 7179 (min 1, median 2917, max 140300). In terms of the query content, the average number of words in each query is 3.76, and the number of distinct words is 573. The frequency distribution for each word approximately follows Zipf's law, which states that the relation between the word frequency and its rank is exponential in general. Considering the pre knowledge of students, queries are separate by whether the provider is novice or advanced student. The novices provided more query in average (13.2±11.7) than advanced students (8.9±9.0), but novices' length of each query (3.47±2.01) is shorter than advanced ones (4.62±2.61), which indicated a lower quality according to Belkin's research [28].

#### 3.3.2 Operation data log

There are 466,659 operations logged including *scroll up*, *scroll down*, *click* and *select* for both searching and reading phases. We found that for both groups of students, novices and advanced students, generated the majority of the operations in reading and in scrolling down. There were 19.3% operations are scrolling up in the searching phase in general, which was not a trivia finding. It showed that users were going back and forward to review the posts content before they decide to click in to proceed further reading in detail. However, ideally a successful search process is that after entering the query, the best item would be shown in the first place of the search result, so that the user would not even need to scroll before clicking to view a result. However in reality, users need to scroll down when they do not feel satisfied with the results provided in the first view, and this unsatisfying ratio is reflected by the scrolling back and forward operation percentage.

On the other hand, the time cost before each operation shows that when browsing search results, users appear to spend more time (37.8%) before clicking or selecting, while they are faster when reading a specific question-answer thread. This fact indicates that users would read more carefully, or be more serious when choosing a thread to read among the search results.

Considering pre knowledge difference, the ratio of scroll back for novices were lower in searching phase compared to the advanced students, but their scroll back ratio is higher in reading phase. This indicates that the novices were more likely to make a choice without browsing more search results, and they had to read the content for more times compare to advanced students.

### 3.4 Programming Information Seeking Actions

#### Actions

In order to analyze students programming information seeking behavior on discussion forums, we categorize their actions into 6 categories based on Marchionini’s [18] information seeking processes: formulate queries, query refinement, results examination, and reading. According to the amount of operations made on each single page, we further split search and reading (by median) in large-search (LS), small-search (SS), large-read (LR), small-read (SR). Table 1 describes detail of user search actions.

Based on the operation data collection and the above action definitions, 2681 actions were identified in total, and the distribution of action distribution is shown in Figure 2.

Table 1. Programming information seeking actions

Actions	Description
Query (Q)	a student issues an query to look for information from programming discussion forum
Refine query (q)	a student modifies the original Q and issues a similar query (word adjacent distance less than 0.3)
Large search (LS)	A student browses the search result page and did operations more than the median of all search pages (31 operations)
Small search (SS)	A student browses the search result page and did operations less than the median
Large read (LR)	A student reads a Q&A thread page, and did operations more than the median of all reading pages (64 operations)
Small read (SR)	A student reads a Q&A thread page, and did operations less than the median

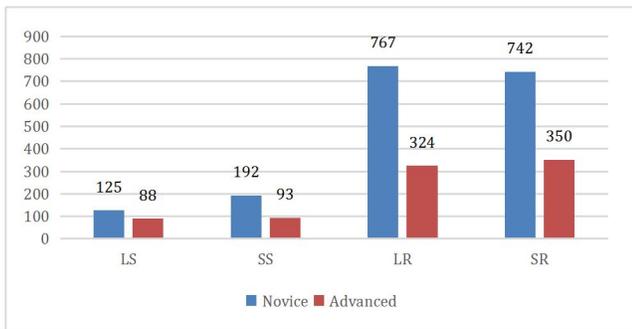


Figure 2. Number of actions identified for novices and advanced students

### 3.5 Modeling Programming Information Seeking From Discussion Forums Using HMM

The Hidden Markov Model (HMM) is a popular method for modeling sequential data. Previous studies have already shown its ability in modeling user information search process [26], survey design [27] and student learning process [28]. In this study, we employ the HMM to model users’ hidden tactics in searching for programming related information on discussion forums, and refer the actions on the site (e.g. query refinement, results examination, content reading, information extraction) as the generated hidden tactics. The hidden tactics can be explained as the strategy used as informal learning activities by looking for programming related information.

We have a sequence of information seeking behaviors from T1 to TM, and each state is one of those predefined information seeking actions:  $TS = \{Q, q, LS, SS, LR \text{ and } SR\}$ . HMM assumes that we also have a sequence of hidden states, from H1 to HM, and each answer type is generated by a corresponding hidden state, but different answer types can be generated by the same hidden state with different probabilities. A HMM model has several parameters: the number of hidden states HS, the start probability of each states  $\pi$ , the transition probabilities among any two hidden states  $A_{ij}$ , and the emission probability from each state to each action  $b_{ij}$ . By only defining the HS and  $\pi$ , a Baum-Welch algorithm [29] can be used to learn the emission and transition probabilities.

## 4. EVALUATION RESULTS

### 4.1 Mapping HMM Patterns to Information Seeking Processes

In this section HMM is used to detect the students’ information seeking behavior pattern. In order to identify the complete sequence of information seeking operations, we only included those operations following a query recorded. The web paged that the students searched from other search engines, where queries were not included, are excluded.

The first step of using HMM is to determine the number of hidden states. A larger number of states will help to describe the model more precisely, while the risk of over-fitting is also increased. In model selection, the information criterion such as the Akaike Information Criterion (AIC) or its variants Bayesian information criterion (BIC) [29] can be used to determining the optimal number of states. Based on models best performance by AIC, we choose HS=3 and HS=5 for *Advanced* and *Novice* groups accordingly (Figure 3).

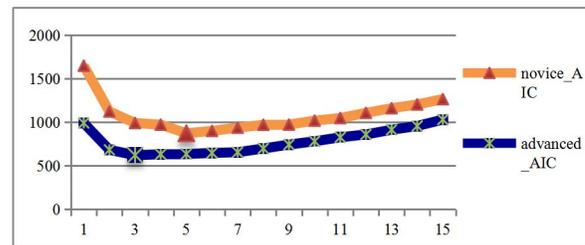


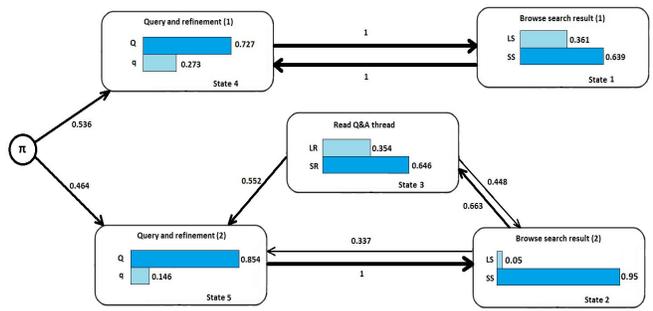
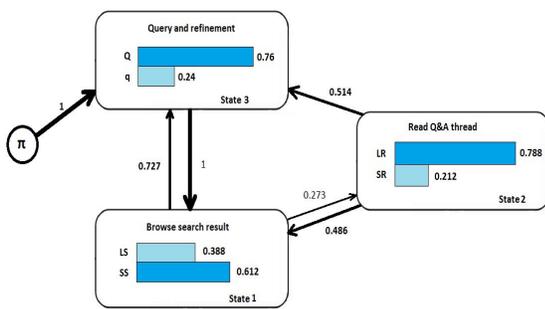
Figure 3. Choosing number of hidden state using AIC.

The emission probability of each hidden state to information seeking operations is shown in Table 2, in which the probabilities under 0.05 were removed for better presentation of the results. The hidden states can be treated as the underlying “tactics” or “principles” when students look for programming information from the discussion forum. For example, *Advanced* group HS2 demonstrates the stronger students’ reading behaviors, which they appear to do more careful readings and fast browsing; while in *Novice* group HS3, students tend to perform more superficial reading than careful reading. While *advanced* group shows more coherent searching, browsing and reading behaviors (each behavior is observed by single state), novices show duo searching and browsing behaviors. *Novice* HS4 and HS1 states seem to have similar searching and browsing behaviors as *advanced* group. However, *Novice* HS5 exhibits more distinct searches by issuing queries and lower probability in refining queries. In addition, *Novice* HS2 shows high probabilities in small search, which can be interpreted as careless results examination.

**Table 2. The hidden states of programming information seeking operations ( $b_{ij}$ )**

hidden states	Q	q	LS	SS	LR	SR	
<i>Advanced</i>	HS1	0	0	0.39	0.61	0	0
	HS2	0	0	0	0	0.79	0.22
	HS3	0.76	0.24	0	0	0	0
<i>Novice</i>	HS1	0	0	0.36	0.64	0	0
	HS2	0	0	0.05	0.95	0	0
	HS3	0	0	0	0	0.35	0.65
	HS4	0.73	0.27	0	0	0	0
	HS5	0.85	0.15	0	0	0	0

Figure 4 is plotted according to the transition probability, and the prior probability is shown in Table 3. The probabilities under 0.05



**Figure 4. Advanced (left) and Novice (right) students' information seeking transition probability diagrams**

#### 4.1.1 Advanced students refine query; novices don't

Advanced students consistently performed query refinements (3:1 ratio) before they examine the results (HS3  $\rightarrow$  HS1). Novices behaved differently. Part of them followed the similar pattern as *Advanced* students did, tuning the queries before examine the results (HS4  $\rightarrow$  HS1). However, when these novices refined queries, there were no consecutive actions followed in the next step (Figure 4 – right top), which indicated that they did not go to any reading page. On the other hand, when novices did minimum query refinements (HS5  $\rightarrow$  HS2), they did manage to proceed to next step, which was the reading phase (HS5  $\rightarrow$  HS2  $\rightarrow$  HS3). This fact suggested that novices may lack of query-results examination ability and lead to no reading (HS4  $\rightarrow$  HS1). In addition, as the HS2 of *Novice* group shows, 95% of the likelihood that the operations were small searches, which means that novices tended not to scrutinize the search results, they only examined the results minimally, even move on to read forum posts (HS5  $\rightarrow$  HS2  $\rightarrow$  HS3). They could read whatever the discussion forum has recommended (i.e. top returned items).

In fact, Table 4 shows the total amount of time that each student spent on searching or reading pages. It is surprising to see that novices spent more than 130 minutes on just reading, while advanced students spent about 40 minutes. Similarly, novices spent more time on searching compare to advanced students. The reason of the time difference is not only they browsed more pages, but also their time spent on each page is longer. These findings indicate that the novices' searching and browsing behaviors only consist of minimum query refinement so that they had to spend more time to read and understand search results, which can be due

are removed. HS3 has the highest prior probability (start probability) in *advanced* group, which means that advanced students always begin with issuing query and modifying the query. So do the majority of the weaker students. In addition, HS5 state is also another beginning state with high probability for novices. It shows that there is also a great probability that novices start issuing queries with minimal query refinement. However, what are the impacts of the amount of query refinement? We have to look at what is happening next. According to Figure 4, the *Advanced* & *Novice* state transition diagrams, there are several findings listed below:

**Table 3. The prior probability of each hidden state ( $\pi$ )**

	HS1	HS2	HS3	HS4	HS5
<i>Advanced</i>	0	0	$\frac{1}{3}$	-	-
<i>Novice</i>	0	0	0	<u>0.536</u>	<u>0.464</u>

to insufficiency of vocabulary in searching and lack of judgment in finding reading resources. We further looked into students' reading behavior and reading content in the following section. Despite the reading quality, novices' behaviors can also suggest the *hidden danger* of online large-scale discussion forums, where the existing filtering mechanisms (such as badges, acceptance, and votes) may not be enough, especially for novice learners.

**Table 4. Total time spent on searching and reading average per student**

total time (seconds) / student	Novice (N=24)	Advanced (N=20)
Search	340.5	146.4
Read	7870.3	2366.6

#### 4.1.2 Advanced students read and novices skim

When students eventually landed on forum post pages and read, we found that *advanced* students committed to careful reading, while novices did more skimming (*Advanced* HS2: 0.79 LR; *Novice* HS3: 0.65 SR). In fact, we found that novices cost more time in small reading than advanced students, while in large reading advanced students spent slightly more time, but there was no significant difference between groups. These results reveal that novices performed less reading in search results filtering, but once they did, they would spend time to read. Thus, it led us to examine their learning effect. Do novices and advanced students have similar effects after reading?

## 4.2 Reading and Learning Effects

### 4.2.1 Students read posts according to course schedule topics

In order to understand what content were students' reading, we crawled all the posts that students read from StackOverflow, and performed text mining with MALLET<sup>1</sup> LDA toolkit with default  $\alpha=30/N$ ,  $\beta=0.01$ ,  $itr=1000$ . We found students were reading the contents from discussion forums according to the course weekly topics, from week 1 *Java Basis* to week 9 *LinkedList*. We then used all the topic words generated from the LDA model to compute Shannon entropy score in estimating the topic focus (Figure 5). There are several interesting findings: *Advanced* students were generally more focused across all topics (smaller topic entropy), except week 4 and week 9. The effect was much more apparent in complex topics: *Recursive* (Table 5 shows the extracted topic words, which we found advanced students read posts regarding to a specific recursive implementation Fibonacci sequence, which novices did not). In week 4 and 9, advanced students were found to be less focused in terms of reading more diverse topics was due to those two weeks were exam periods. Therefore, it is understandable that students might read a wider range of topics that were covered over exam periods.

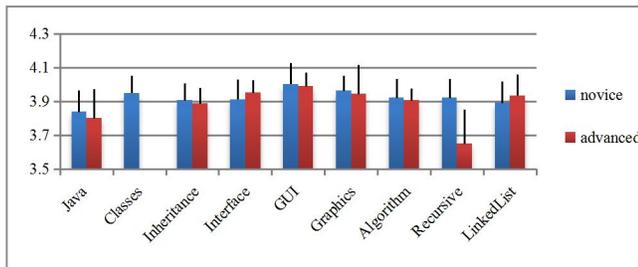


Figure 5. Weekly readings' keywords by novices and advanced students

Table 5. Recursive topic words by novices and advanced students

*Novice: {type, code, recursive, dynamic, void, write, result, example, loop, print, add, wikipedia, error, int, version, method, operator, pseudo, easy, program, static, mathematics, call, line, learn, number, work, value, function, undefined}*

*Advanced: {function, method, value, static, return, int, change, version, recursive, result, error, mathematics, program, line, number, fibonacci, sequence, fib, wikipedia, operator, pseudo, easy, type, print, example, code, learn, void, traverse, loop}*

### 4.2.2 Learning Effects

Based on the percentage of large read rate in reading pages, we found that the more students spending time in reading on StackOverflow, the higher final score they obtained ( $r=0.418$ ,  $p<0.01$ ). Additionally, we found that the slope of novices and advanced students had little difference, while the intercept of novices is higher. This fact indicates that novice and advanced students gained the same benefits from increasing large read rate, however, in order to achieve the same score, novices has to read more carefully. Figure 6 shows the connection between large read rate and final exam score.

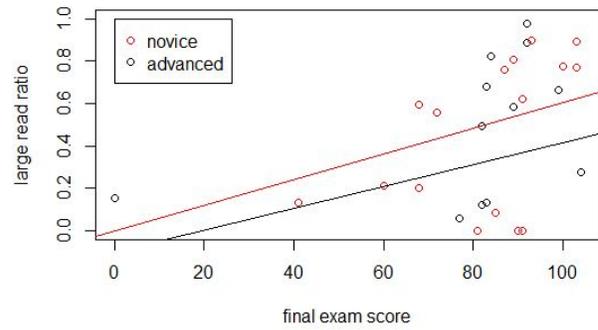


Figure 6. Final score vs. Large read rate

## 5. CONCLUSIONS

### 5.1 Summary

In this study, we designed a programming information seeking framework with a browser plugin to collect students' programming information seeking behavior data from discussion forum StackOverflow. Students' query intention, time spent and all actions were logged. We modeled programming learners' query formulation, refinement, results examination, and reading processes with Hidden Markov Model. We conducted sequence pattern mining. The results showed that programming learners indeed seek for programming related information from discussion forums by actively searching on the site and reading posts progressively according to course schedule topics.

The result of this study showed that programming novices usual spend more time in browsing search result and reading, while the sequential due to their lack of pre knowledge. As long as they can read as well as advanced students, they can learn as much as advanced students according to the learning evaluation result.

All the study results shed lights on programming learners seek for learning resources from large-scale online discussion forums. We anticipate this work serves as guidelines for educational technologists to design better effective tools to facilitate learning via programming information seeking process.

### 5.2 Limitations and Future Work

There are a few limitations in current study. First of all, after students log in from the browser at least once, all their activities on StackOverflow will be recorded. However, when students search from search engines (i.e. Google) and land on StackOverflow site, their initial queries will not be captured. A more completed data collection should include all queries that the students search in information seeking.

Moreover, we mainly take into account of students' query and mouse actions without considering other keystrokes' actions. Another common information seeking behavior is to use Ctrl+F on the keyboard to search keyword with in a web page, which was not captured in the study. This operation can be a convenient and fast method to locate useful information when browsing web pages, including discussion forums.

In the future, we will consider a more completed data collection and more exhaustive evaluation. Most importantly, we aim to design an adaptive programming information seeking tool to help novices effectively navigate search results.

## 6. REFERENCES

- [1] Vasilescu, B., Serebrenik, A., Devanbu, P., & Filkov, V. (2014, February). How social Q&A sites are changing

<sup>1</sup> <http://mallet.cs.umass.edu>

- knowledge sharing in open source software communities. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 342-354). ACM.
- [2] Treude, C., O. Barzilay, and M. Storey. How do programmers ask and answer questions on the web?: NIER track. in *Software Engineering (ICSE), 2011 33rd International Conference on*. 2011.
- [3] Seaman, C.B. The information gathering strategies of software maintainers. in *Software Maintenance, 2002. Proceedings. International Conference on*. 2002. IEEE.
- [4] Sharif, K.Y. and J. Buckley. Developing schema for open source programmers' information-seeking. in *Information Technology, 2008. ITSIM 2008. International Symposium on*. 2008. IEEE.
- [5] Sim, S.E., *Supporting multiple program comprehension strategies during software maintenance*. 1998, University of Toronto.
- [6] Stylos, J. and B.A. Myers. Mica: A Web-Search Tool for Finding API Components and Examples. in *Visual Languages and Human-Centric Computing, 2006. VL/HCC 2006. IEEE Symposium on*. 2006.
- [7] Kuhlthau, C.C., Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 1991. 42(5): p. 361-371.
- [8] Eickhoff, C., Teevan, J., White, R., & Dumais, S. (2014, February). Lessons from the journey: a query log analysis of within-session learning. In *Proceedings of the 7th ACM international conference on Web search and data mining* (pp. 223-232). ACM
- [9] Chi, M.T.H. and R. Wylie, The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 2014. 49(4): p. 219-243.
- [10] McNamara, D. S. (2001). Reading both high-coherence and low-coherence texts: Effects of text sequence and prior knowledge. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 55(1), 51.
- [11] Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2008, June). Why tutored problem solving may be better than example study: Theoretical implications from a simulated-student study. In *Intelligent Tutoring Systems* (pp. 111-121). Springer Berlin Heidelberg.
- [12] Catrambone, R. (1996). Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(4), 1020.
- [13] Puustinen, M. and J.-F. Rouet, Learning with new technologies: Help seeking and information searching revisited. *Computers & Education*, 2009. 53(4): p. 1014-1019.
- [14] Zimmerman, B.J. and M.M. Pons, Development of a Structured Interview for Assessing Student Use of Self-Regulated Learning Strategies. *American Educational Research Journal*, 1986. 23(4): p. 614-628.
- [15] Marchionini, G., Exploratory search: from finding to understanding. *Communications of the ACM*, 2006. 49(4): p. 41-46.
- [16] Dave, K., M. Wattenberg, and M. Muller, Flash forums and forumReader: navigating a new kind of large-scale online discussion, in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 2004, ACM: Chicago, Illinois, USA. p. 232-241.
- [17] Indratno, J. Vassileva, and C. Gutwin, Exploring blog archives with interactive visualization, in *Proceedings of the working conference on Advanced visual interfaces*. 2008, ACM: Napoli, Italy. p. 39-46.
- [18] Guerra, J., Sahebi, S., Lin, Y. R., & Brusilovsky, P. (2014). The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises. in *The 7th International Conference on Educational Data Mining*. 2014: London, UK.
- [19] Wen, M., D. Yang, and C. Rose. Sentiment Analysis in MOOC Discussion Forums: What does it tell us? in *The 7th International Conference on Educational Data Mining*. 2014. London, UK.
- [20] Wang, X., Yang, D., Wen, M., Koedinger, K., & Rosé, C. P. Investigating how student's cognitive behavior in MOOC discussion forums affect learning gains, in *The 8th International Conference on Educational Data Mining*. 2015: Madrid, Spain.
- [21] Enamul Hoque, G.C., Shafiq Joty. Interactive Exploration of Asynchronous Conversations: Applying a User-Centered Approach to Design a Visual Text Analytic System. in *Workshop on Interactive Language Learning, Visualization, and Interfaces*. 2014. Baltimore, Maryland.
- [22] Wise, A. F., Speer, J., Marbouti, F., & Hsiao, Y. T. (2013). Broadening the notion of participation in online discussions: examining patterns in learners' online listening behaviors. *Instructional Science*, 41(2), 323-343.
- [23] Sande, C.v.d., Free, open, online, mathematics help forums: the good, the bad, and the ugly, in *Proceedings of the 9th International Conference of the Learning Sciences - Volume 1*. 2010, International Society of the Learning Sciences: Chicago, Illinois. p. 643-650.
- [24] Hanrahan, B.V., G. Convertino, and L. Nelson, Modeling problem difficulty and expertise in stackoverflow, in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*. 2012, ACM: Seattle, Washington, USA. p. 91-94.
- [25] Posnett, D., Warburg, E., Devanbu, P., & Filkov, V. (2012, December). Mining stack exchange: Expertise is evident from initial contributions. In *Social Informatics (SocialInformatics), 2012 International Conference on* (pp. 199-204). IEEE.
- [26] Han, S., Z. Yue, and D. He. Automatic detection of search tactic in individual information seeking: A hidden Markov model approach. in *iConference 2013*. 2013. arXiv preprint arXiv:1304.1924.
- [27] Hsiao, I. H., Han, S., Malhotra, M., Chae, H. S., & Natriello, G. (2014, June). Survey sidekick: Structuring scientifically sound surveys. In *Intelligent Tutoring Systems* (pp. 516-522). Springer International Publishing.
- [28] Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012, February). Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 153-160). ACM
- [29] Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1), 164-171.

# Classifying behavior to elucidate elegant problem solving in an educational game

Laura Malkiewich  
Teachers College,  
Columbia University  
525 W 120<sup>th</sup> St.  
New York, NY 10027  
Laura.malkiewich@  
tc.columbia.edu

Ryan S. Baker  
Teachers College,  
Columbia University  
525 W 120<sup>th</sup> St.  
New York, NY 10027  
baker2@  
tc.columbia.edu

Valerie Shute  
Florida State  
University  
3205G Stone Building  
1114 West Call St.  
Tallahassee, FL  
32306  
vshute@fsu.edu

Shimin Kai  
Teachers College,  
Columbia University  
525 W 120<sup>th</sup> St.  
New York, NY  
10027  
smk2184@  
tc.columbia.edu

Luc Paquette  
University of Illinois,  
Urbana Champaign  
383 Education  
Building  
1310 S. Sixth St.  
Champaign, IL 61820  
lpaq@illinois.e

## ABSTRACT

Educational games have become hugely popular, and educational data mining has been used to predict student performance in the context of these games. However, models built on student behavior in educational games rarely differentiate between the types of problem solving that students employ and fail to address how efficacious student problem solutions are in game environments. Furthermore, few papers assess how the features selected for classification models inform an understanding of how student behaviors predict student performance. In this paper, we discuss the creation and consideration of two models that predict if a student will develop an elegant problem solution (the Gold model), or a non-optimal but workable solution (the Silver model), in the context of an educational game. A pre-determined set of features were systematically tested and fit into one or both of these models. The two models were then examined to understand how the selected features elucidate our understanding of student problem solving at varying levels of sophistication. Results suggest that while gaming the system and lack of persistence indicate non-optimal completion of a problem, gaining experience with a problem predicts more elegant problem solving. Results also suggest that general student behaviors are better predictors of student performance than level-specific behaviors.

## Keywords

Educational games; Problem solving, Classifiers.

## 1. INTRODUCTION

Educational games can be a great way to enhance learning; in some cases games lead to better learning than standard instructional activities [5, 22]. Yet while understanding how students learn in educational games is important, not much work has been done on modeling student learning in educational games that are open-ended, where students have a lot of freedom to explore. Furthermore, although there has been work on modeling behavior in games and educational learning environments to predict performance in these environments [6, 10, 13, 14, 16, 20] or more generally in school [4], there is not a lot of work that specifically looks at student problem solving strategies in games. Analyzing how students solve complex problems is a key part of understanding student learning in a domain [1, 3, 12], especially in open-ended environments [2]. For this reason, we are investigating student problem solving techniques in order to better understand the nature of student behavior and performance in open-ended educational games.

One key problem solving skill for learning is the ability to produce elegant solutions as well as workable solutions [8, 17],

especially as one of the key markers of expertise in a field is the ability to solve problems more elegantly than a novice [11]. Even though there has been research on how to model different student approaches to problem solving [7] there has not yet been sufficient work on modeling the behaviors associated with elegant problem-solving vs. creating workable but less-optimal solutions to problems, especially in game environments. This paper examines how students solve problems to create elegant versus non-optimal, workable, solutions to problems in open-ended educational games. We study this issue in the context of Physics Playground, an open-ended discovery based learning game where students learn about Newtonian physics while trying to solve problems.

## 2. THE GAME: PHYSICS PLAYGROUND

Physics Playground, formerly called Newton's Playground [19], is an educational game that measures and supports knowledge of conceptual physics for middle and high school students. The game requires students to draw simple machines (consisting of ramps, levers, pendulums, and springboards) that act in accordance with Newton's laws of force and motion. In each level of the game, students are tasked with freehand drawing these machines, which are used to get a green ball to hit a red balloon. In addition to drawing machines, students can draw objects that interact with the ball directly in order to get the ball to reach the balloon. For example, students can draw objects made to fall and hit the ball directly, causing the ball to move. These objects are called "divers" in the context of the game. Students can also draw objects through the ball to move it up slightly. This technique is called "stacking" and is considered a form of "gaming the system" [21]. Similarly, students can click on the ball to "nudge" it forward slightly, if need be, without drawing an object at all. When students finally find a way to hit the red balloon with the green ball, they have completed the level, and are awarded a badge based on their performance.

Students can either receive a gold badge, silver badge, or no badge, depending on their performance in any given level. Badges are awarded according to the efficiency of the student's solution to a problem — determined by the number of objects a student draws in his or her attempt to solve a given problem. For most levels, gold badges are awarded if the student solves the problem by drawing three or fewer objects. Silver badges are awarded if the student solves the problem, but draws more objects. Each level is designed so that one simple machine (a ramp, springboard, pendulum or lever) will optimally solve the given problem. Accordingly, badges for performance are also tied to the type of machine that a student drew in the given level. For example, if a student creates an efficient solution to a level using a ramp, then

the student would be awarded a “gold ramp” badge upon completion of the level. Badges are awarded as a means to give students feedback about the efficiency of their solution, so students can reflect on their solution quality. Badges are not necessarily constructed for motivational purposes. Student badges are referred to as “trophies” in the context of the game, and are displayed in the top right hand side of the screen upon level completion.

The game consists of seven “playgrounds”, or game worlds, that each contains 10-11 problems. In total there are 74 problems in the entire game. Problems are ordered by difficulty, and problem difficulty is determined by a number of factors including the location of the ball to the target, the magnitude and location of obstacles between the ball and the balloon, the number of agents required to get the ball to the balloon, the novelty of the problem. Students do not have to move through the game in a linear fashion. All levels are unlocked and accessible to students when the game starts (i.e., level access does not depend on a student’s performance or progress in the game). Therefore, students can choose to go to any playground and work on any problem that they wish. That being said, there is a logical ordering to the levels, and many students do choose to go through the game in a linear fashion.

### 3. METHOD

#### 3.1 The Study

This project is based on data collected during a prior study using Physics Playground. A more detailed description of the study population and methods can be found in [9, 18].

##### 3.1.1 Participants

This data is from a study on 137 8<sup>th</sup> and 9<sup>th</sup> grade students who attended a diverse K-12 school in the southeastern United States.

##### 3.1.1.1 Procedure

Students played the game in class for about 2.5 hours across four days of the study. Days 1 and 4 of the study consisted of student assessments, including a pretest and isomorphic posttest of students’ knowledge of physics concepts. Learning data will not be discussed in the context of this paper [for learning data see 9, 18]. Days 2 and 3 of the study, as well as the first half of Day 4, consisted entirely of gameplay.

##### 3.1.1.2 Measures

Physics Playground captured student log data during gameplay. The final data set consisted of 2,603,827 lines of action codes across the 137 students. Data collected included over seventy variables including information on student progression through the game, time stamps for actions, metrics on student drawings, gameplay actions, and badge awards. Across the 137 students, 919 levels were completed, 203 gold badges were awarded and 500 silver badges were awarded.

#### 3.2 Model Selection

Two models were built for the purpose of distinguishing which features indicate elegant problem solving, and which indicate non-optimal problem solving. The first model was built to classify the award of a gold badge, where problem solutions are optimal (Gold Model). The second model was built to classify the award of a silver badge, where students solve a level, but in a non-optimal way (Silver Model). Levels that a student attempted but did not

complete (levels where the student was not awarded a badge) were not used in this analysis.

By building two models, we were able to more effectively differentiate between features that predict elegant problem solving and features that predict non-optimal problem solutions more effectively. For example, creating two models allows for the identification of features that positively load onto one model but negatively load onto another. In turn, understanding these distinctions allows for a deeper understanding of how different levels of various features are indicative of the two types of problem solving. Badges were used as labels because they are the game’s proxy for assessing student problem solution quality by marking the efficiency of a student’s solution. Although badges in many modern games are used for motivational purposes, for the purpose of this project, we were only interested in what badges indicated about the elegance of a student’s problem solution.

Features were created, tested, and iteratively improved upon, across a variety of classification algorithms. During this process, the J48 algorithm, which is Weka’s implementation of the C4.5 algorithm [15], consistently provided the strongest predictive power, while protecting against over fitting. For this purpose, when it came to final feature selection and model creation, J48 was the sole algorithm used.

The models were built on less than half of the student data (61 students) so that the remaining test set could later be used to validate and test the final models. In order to validate the models during model creation and feature selection, batch level cross-validation was used. Each student was randomly assigned into 1 of 10 batches, and 10-fold validation was used to assess model goodness. Kappa was used as a measure of model fit.

#### 3.3 Feature Selection

To make the two models, gold and silver labels were made. The gold label had a value of 1 if the student was awarded a gold badge, and a value of 0 if the student was awarded any other kind of badge (or no badge). A label for silver was created in the same way. The original log data tied each badge to the type of machine it awarded a badge for, but for the purpose of this project badge color and machine type were separated into two different features. This was done in part because we wanted to see if machine type affected which type of badge was awarded and in part because making machine type part of the label would result in models predicting what machine the student was building. Instead, we wanted to simply assess how successful students were at solving any given problem, regardless of the nature of the problem given.

Over fifty features were created and assessed for their goodness in predicting badge awards on any given level. The feature engineering process started with a restructuring of the raw student data logs to the problem-level (raw logs came at the action level) because the label of interest categorized student performance at the problem-level grain size. This process was then followed by a descriptive analysis of the variables that came out of this restructured data, followed by structured brainstorming to elicit ideas about the types of features that could be built out of this data. Features were then created to measure certain constructs (e.g., time on task, gaming the system behavior, etc.) and behaviors of interest. Once a core set of features was created, colleagues and system experts were consulted about the quality, interest, and potential effectiveness of those features. Features were then iterated on. New features were created in an attempt to both measure constructs in more ways (e.g., measuring time on

task by looking at time on level, standardized time, or just time spent drawing objects) and to measure different student behaviors and constructs that the first set of features failed to measure. Features were then refined based on colleague and system expert feedback and used in single feature models to assess feature quality. An iterative process of feature creation, peer consulting, and feature refinement then continued for several more cycles until the final set of fifty features had been created.

Once all features had been created, single-feature models were used to choose the seventeen features that were the best predictors of any given construct. For example, Time on Level in minutes was determined to be a better classification of the amount of time that a student spent on a level than standardized time.

The final seventeen features were then ordered in terms of their goodness within a single-feature J48 model, under student-level cross-validation. The best feature was added, and then a recursive process was used where additional features were tested in the same order to determine whether adding that feature improved model goodness, as measured by an increase in kappa. Only features that improved kappa were added. The final gold model contained fourteen features, and the final silver model contained nine features.

### 3.4 Feature Descriptions

The final seventeen features used for model creation are listed and described below in addition to which model they ended up being included in. Features are listed in the order that they were tested and selected.

**Sum Elapsed (silver):** The total amount of time that a student spent actively drawing objects up until that point in the game. For example, if a student spends 90 seconds actively drawing objects in Level 1, and then 30 seconds drawing during Level 2, then Sum Elapsed by the end of Level 2 would have a value of 120 seconds.

**Time on Level (both):** The total amount of time spent playing the level that the student is being awarded the badge for (in seconds).

**Nudge Count (gold):** The total number of times that the student pressed the ball to nudge it forward a little in the level.

**Number of Objects (both):** The total number of distinct objects (machines, random lines, weights, etc.) the student drew in the level.

**Diver Count (none):** The total number of divers that a student created in the level.

**Pause Before End (both):** Binary indicator of whether or not the student hit the pause button as their last action before the level ended. Usually this happens when students want to exit out of a level before completing the level. In this case, students would neither be awarded a gold badge nor a silver badge.

**Ball Count (both):** The number of balls a student uses in a level. If a student knocks a ball off the screen or if the ball provided to the student falls to the bottom of the screen, then it disappears and the student gets a new ball to try again.

**Max Velocity Y (both):** The maximum velocity that any ball a student used in a level traveled in the y direction (up and down). Velocity values in the Physics Playground system are given in

meters-kilogram-second (MKS) units.

**Max Velocity X (gold):** The maximum velocity that any ball a student used in a level ever traveled in the x direction (left and right).

**Erased Object Count (silver):** Number of objects that a student drew, and then erased in the level. Students can erase an object that they have drawn by clicking on it.

**Stack Count (both):** Number of times student drew an object through the ball in order to move the ball up.

**Badge Before (gold):** Binary indicator of whether or not a student has received a badge (of any color) on this level before.

**Played Before (gold):** Binary indicator of whether or not a student has played this level before.

**Average Free-fall Distance (gold):** Free-fall distance is a measure of how far any divers fell before striking a ball. This feature averages across all those distances in the level. Units are percentage of the game screen. So if the diver falls half the distance of the game screen, this would have a value of 0.5.

**Restart Count (gold):** The number of times a student re-started the level.

**Play Count (gold):** The number of times that a student has played the current level before. Restarts are not included in this count. A student has to have either completed the level or made an attempt at the level, left the level, and then returned, in order for it to contribute towards this play count.

**Machine (both):** The type of machine that should be created to optimize movement of the ball to the target. There is one machine per level and they can take the form ramp, lever, pendulum, or springboard.

### 3.5 Final Models

The final J48 gold classification model with ten-fold student batch cross-validation, which was built on half the data, had a Kappa value of 0.69, and the silver classification model had a Kappa of 0.83. The other half of the data was held out for future analysis comparing the models developed here to other, future models. As is evident from the features mentioned above, seven features fit into both the gold and silver classification models. Those features were Time on Level, Number of Objects, Pause Before End, Ball Count, Max Velocity Y, Stack Number, and Machine. Seven features only fit the gold classification model; those were Nudge Count, Max Velocity X, Badge Before, Played Before, Average Free-fall Distance, Restart Count, and Play Count. Finally, two features only fit the silver classification model. Those were Sum Elapsed and Erased Object Count.

### 3.6 Qualitative Analysis of Models

The primary goal of this project was to use classification models to help elucidate how student behavior predicts gold and silver badge acquisition differently. For this reason, we take a more qualitative look at which features were included in each model, which were included in both, and which were included in neither.

Table 1 indicates how each of the features loaded onto each of the models when used in a single-feature model (machine type does not have a numeric value, so it is not included in the table). Since both models were built using J48 decision trees, this is simply a proxy for the general loading of each feature on the model outcomes, and not a comprehensive measure of how each feature fits into each model.

**Table 1. Feature loadings onto each model**

Feature	Gold Model	Silver Model
Sum Elapsed	-	Negative
Time on Level	Negative	<b>Positive</b>
Nudge Count	Negative	-
Number of Objects	Negative	<b>Positive</b>
Diver Count	-	-
Pause Before End	Negative	Negative
Ball Count	<b>Positive</b>	Negative
Max Velocity Y	Negative	<b>Positive</b>
Max Velocity X	<b>Positive</b>	-
Erased Object Count	-	<b>Positive</b>
Stack Count	Negative	<b>Positive</b>
Badge Before	Negative	-
Played Before	Negative	-
Average Free-fall Distance	Negative	-
Restart Count	<b>Positive</b>	-
Play Count	<b>Positive</b>	-

### 3.6.1 Features included in both models

Features that were included in both models mostly helped indicate whether the student was able to achieve optimal performance or simply workable solutions. For the majority of the features that were in both models, the value was higher for non-gold and higher for silver, indicating that these behaviors were typical of students who developed workable yet non-optimal solutions.

For example, Time on Level was a good indicator of which students produced non-optimal, yet workable solutions. Students who spent a very short time on the level could have entered a level and then immediately quit, so they were likely to not receive a badge. However, longer time in level is associated with a badge but not a gold badge. This loading is likely because students who spend a long time on a level are struggling more or drawing more and those students are therefore less likely to develop the most optimal solution in a single level attempt.

Other features that were higher for non-gold and silver were Number of Objects, Max Velocity Y, and Stack Count. It makes sense that students who drew more objects would get silver, because they are doing more work than students who quit the level (no badge) and students who developed optimal solutions (gold badge). Also, badges are awarded in accordance with the number of objects a student draws in his or her attempt to solve a given problem, so it makes sense that this feature would be a significant indicator of performance. Stack Count could have been a good indicator of whether students solved a problem optimally or non-optimally because students who are stacking a lot could be

trying to game the system, likely because they don't know how to solve the problem more effectively using machines. These students are likely to get a silver badge if they complete the problem, because stacking requires drawing many objects.

Only one feature that appeared in both models was higher for both non-gold and non-silver, Pause Before End. This is likely because students who paused before the end of the level were quitting, and therefore did not receive a badge at all. However, that was not always the case.

It is curious that students who had a higher Ball Count per level were more likely to produce optimal solutions; the value for ball count was higher for gold and non-silver indicators. This may be because students who created optimal solutions were experimenting more, and therefore going through more balls, but without spending too much time or drawing too many objects. This behavior could be indicative of students who are quickly iterating on a single idea, or thinking of what to do before drawing objects. (On some levels balls keep dropping down until you draw an object underneath to catch the ball, so the longer you spend without drawing an object, the more balls you use).

### 3.6.2 Features that only fit the gold model

Features that only fit the gold model are interesting because they specifically separate those who were able to solve problems elegantly as opposed to students who could not find an optimal solution to the problem. The features fit three general categories, relative to whether or not they indicate experience, shallow strategies, or efficiency.

Features that indicate experience include Badge Before, Played Before, Play Count, and Restart Count. It is interesting that Badge Before and Played Before, which are both binaries, indicate non-gold performance while Play Count and Restart Count indicate gold performance. This indicates that if a student is working on a problem they have completed or played once before, they are not likely to develop an optimal solution, but the more they play a level, the closer they are to get to an optimal solution. Students who have played the level before have some experience with the problem space, even if they did not complete the level previously and that experience could help them determine an optimal problem solution. Play Count and Restart Count tell the model the precise amount of experience the current student has had with a level. Students who re-start or play a level more often might be optimizers, aiming to iterate several times on their problem solution in an attempt to find the best approach to solving the problem. They might be thinking more critically about the choices they are making and choosing to come back to a level or start it again when they've determined that they have acquired the skill or knowledge necessary to now perform more effectively. Resetting also enables students to clear their screens of all objects, and start over, so they can approach the problem afresh. This can be a good strategy for students who want to try going in a different direction instead of iterating on an earlier idea, and it can lead to more efficient problem attempts later.

Nudge Count is a feature that indicates shallow strategies, or even potentially gaming the system. Students who nudge the ball a lot are trying to make the ball move without using a drawn machine to move the ball. This could lead to effectively moving the ball without drawing more objects, which could lead to a problem solution despite a low object count, which would result in a gold badge. Or, it could indicate a student who is nudging because they are struggling a lot with the problem, perhaps because they have

already drawn many objects, but are unable to get the ball to move effectively, so they try to nudge it along.

The other features associated with gold badges but not silver badges measure how efficiently students are building machines. These include Average Free-fall Distance and Max Velocity X. Max Velocity X is a predictor of gold badges while Max Velocity Y can predict gold and silver badges, because Max Velocity X is a more effective measure of how well a student has constructed his or her machine. If a ball is dropped from the starting point, then regardless of how effective the student's machine is, the ball will, in many cases, hit the same maximum velocity as it falls because all balls in the Physics Playground interface follow the laws of physics, and therefore accelerate at  $g$ . However, how fast a ball moves in the  $x$  direction is a direct result of how well a student's designed machine moved the ball in that direction. Likewise, Average Free-fall distance measures student machine efficiency, because students have to carefully choose where to draw divers so that they have a desired effect on ball movement. Divers that are positioned too far away might not hit the desired target, requiring another driver to be drawn for the desired effect. Therefore, both these features are found in this model because they are able to successfully classify effective and efficient student construction choices.

### 3.6.3 Features that only fit the silver model

Only two features were associated with silver badges but not gold badges. They were Sum Elapsed and Erased Object Count. Both of these features describe the behaviors of students who are tinkering to iterate to a solution. Sum Elapsed negatively loads on the model, suggesting that it indicates ineffective tinkering, while Erased Object count positively loads on the model, suggesting that it indicates effective yet inefficient tinkering. Sum Elapsed is a measure of how much effort a student has put into the game, up until that point in time. A student who has spent a lot of time drawing objects across all prior game levels will have a higher Sum Elapsed value. This is higher for non-silver badges, maybe in part because students who spend a lot of time drawing on levels are less likely to complete the level they are on. This could be because students are making long strokes while doodling, or doing other off task work. On the other hand, students who erase many objects are more likely to get a silver badge. This might be because students who erase a lot are pruning their work if they drew too many objects or made mistakes. These students are more dedicated to completing the current problem, to acquire a badge, but they are not likely to solve the problem in an optimal manner. Therefore Erased Object Count measures an effective problem solving strategy that is not efficient.

### 3.6.4 Features that fit neither model

It is important to consider not only the features that fit into the models, but also the features that failed to improve either of the models when added. These included Diver Count and a host of other features that were discarded during the feature engineering process, due to the features' low predictive power for behaviors of interest. Interestingly, more specific features involving specific machines or operators were less predictive of student performance than more general variables. Concrete behavior-specific features like Diver Count and Pin Count (pins are small dots that students can add to a drawing to tack an object in place or create a point for an object to rotate around) were less associated with outcomes than were general features like Object Count and Sum Elapsed, which describe student behaviors that span across several actions or several levels. (Note that divers are objects, so when talking

about a distinction between these features Object Count is a more general category than Diver Count). It could be that student performance on any particular problem was not as predictive of their problem-solving efficacy as that student's overall behavior. This could suggest that problem solving scaffolding and teaching should focus more on students' overall strategies, rather than level specific strategies. On the other hand, it may simply indicate that none of the more specific features, by themselves, are as predictive as the more general categories that cut across and combine different specific features. It is also important to note that in addition to improving prediction, using more general features also reduces the risk of models over-fitting.

## 4. DISCUSSION AND CONCLUSION

This analysis of two models built to predict optimal student performance and non-optimal student performance gives us some interesting insights about the kinds of behaviors that predict student performance, and also about the kinds of features that best fit these types of models. Models that describe student performance more generally are more predictive when fed into a J48 decision tree, which can make cutoffs at different values of those feature variables in order to differentiate students who are solving levels optimally, sub-optimally, and not solving levels at all. In turn, features that differentiate optimal performers from all others focus on student experience with the problem space, shallow strategies, and gaming behaviors in addition to measures of student problem solving efficiency. Classifiers of successful but sub-optimal performance tend to describe more exploratory, tinkering behavior while classifiers of elegant problem solving seem to highlight the value of student exposure to a problem and measures of problem-solving efficiency.

These findings give insight into future designs of Physics Playground and other games and open-ended learning environments. To encourage more elegant student problem solving, the learning environment can encourage students to revisit problems, especially after they've created a workable solution, but failed to create an elegant one. Additionally, student feedback about how effective their solution is or what kind of metrics are needed for an optimal solution (e.g., a prompt indicating that for the ball to reach the target it must hit a certain  $x$  velocity) could aid students in understanding what more proximal goals they need to fulfill in order to ultimately solve the problem at hand in the most efficient way.

Future work can explore whether similar features are effective for predicting student problem solving in other games. The models discussed here were built on only one game with a unique form of gameplay and specific design constraints, so the study is limited in its generalizability. However, there is the potential for the results of this paper to be used for constructing models for classifying student performance to differentiate between elegant and non-optimal problem solving strategies in other games or open-ended learning environments.

## 5. ACKNOWLEDGMENTS

We would like to thank the Bill & Melinda Gates Foundation (SOL1071343/APP181499) for their generous support provided for this research, and Ed Dieterle for helpful suggestions and advice. We would also like to thank the students who participated in the study.

## 6. REFERENCES

- [1] Anderson, J. R. (1993). Problem solving and learning. *American Psychologist*, 48(1), 35.
- [2] Blikstein, P. (2011, February). Using learning analytics to assess students' behavior in open-ended programming tasks. In *Proceedings of the 1st international conference on learning analytics and knowledge* (pp. 110-116). ACM.
- [3] Chi, M. T., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5(2), 121-152.
- [4] Chi, M., Schwartz, D. L., Chin, D. B., & Blair, K. P. (2014, July). Choice-based Assessment: Can Choices Made in Digital Games Predict 6 th-Grade Students' Math Test Scores?. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*(pp. 36-43).
- [5] Clark, D. B., Tanner-Smith, E. E., & May, S. K. (2013). *Digital games for learning: A systematic review and meta-analysis*.
- [6] Conrad, S., Clarke-Midura, J., & Klopfer, E. (2014). A framework for structuring learning assessment in a massively multiplayer online educational game: experiment centered design. *International Journal of Game Based Learning*, 4(1), 37-59.
- [7] Eagle, M., & Barnes, T. (2014, July). Exploring differences in problem solving with data-driven approach maps. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*(pp. 76-83).
- [8] Ertmer, P. A. (2015). *Essential Readings in Problem-based Learning*. Purdue University Press.
- [9] Kai, S., Paquette, L., Baker, Bosch, N., D'mello, S., Ocumpaugh, J., Shute, V., & Ventura, M. (2015). A Comparison of face-based and interaction-based affect detectors in physics playground. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*(pp. 77-84).
- [10] Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013, July). Using data-driven discovery of better student models to improve student learning. In *Artificial intelligence in education* (pp. 421-430). Springer Berlin Heidelberg.
- [11] Larkin, J., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. *Science*, 208(4450), 1335-1342.
- [12] Larkin, J. H., & Reif, F. (1979). Understanding and teaching problem-solving in physics. *European Journal of Science Education*, 1(2), 191-203.
- [13] Martin, J., & VanLehn, K. (1995). Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, 42(6), 575-591.
- [14] Olsen, J. K., Aleven, V., & Rummel, N. Predicting Student Performance In a Collaborative Learning Environment. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*(pp. 211-217).
- [15] Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufman: New York.
- [16] Rowe, E., Baker, R., Asbell-Clarke, J., Kasman, E., & Hawkins, W. (2014, July). Building automated detectors of gameplay strategies to measure implicit science learning. In *Poster presented at the 7th annual meeting of the international educational data mining society* (pp. 4-8).
- [17] Savransky, S. D. (2000). *Engineering of creativity: Introduction to TRIZ methodology of inventive problem solving*. CRC Press.
- [18] Shute, V.J., D'Mello, S., Baker, R.S.J., Cho, K., Bosch, N., Ocumpaugh, J., Ventura, M., & Almeda, V. (2015). Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Computers & Education*, 86, 224-235.
- [19] Shute, V.J., Ventura, M., & Kim, Y.J. (2013). Assessment and learning of informal physics in Newton's Playground. *The Journal of Educational Research*, 106, 423-430.
- [20] VanLehn, K. (1988). Student modeling. *Foundations of intelligent tutoring systems*, 55, 78.
- [21] Wang, L., Kim, Y. J., & Shute, V. (2013). "Gaming the system" in Newton's Playground. In *AIED 2013 Workshops Proceedings Volume 2 Scaffolding in Open-Ended Learning Environments (OELEs)* (p. 85).
- [22] Wouters, P., Van Nimwegen, C., Van Oostendorp, H., & Van Der Spek, E. D. (2013). A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105(2), 249.

# Predicting Dialogue Acts for Intelligent Virtual Agents with Multimodal Student Interaction Data

Wookhee Min  
North Carolina State University  
Raleigh, NC 27695  
wmin@ncsu.edu

Joseph B. Wiggins  
North Carolina State University  
Raleigh, NC 27695  
jbwigg3@ncsu.edu

Lydia G. Pezzullo  
Tufts University  
Medford, MA 02155  
lydia@learndialogue.org

Alexandria K. Vail  
North Carolina State University  
Raleigh, NC 27695  
akvail@ncsu.edu

Kristy Elizabeth Boyer  
University of Florida  
Gainesville, FL 32611  
keboyer@ufl.edu

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

Megan H. Frankosky  
North Carolina State University  
Raleigh, NC 27695  
rmhardy@ncsu.edu

Eric N. Wiebe  
North Carolina State University  
Raleigh, NC 27695  
wiebe@ncsu.edu

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

## ABSTRACT

Recent years have seen a growing interest in intelligent game-based learning environments featuring virtual agents. A key challenge posed by incorporating virtual agents in game-based learning environments is dynamically determining the dialogue moves they should make in order to best support students' problem solving. This paper presents a data-driven modeling approach that uses a Wizard-of-Oz framework to predict human wizards' dialogue acts based on a sequence of multimodal data streams of student interactions with a game-based learning environment. To effectively deal with multiple, parallel sequential data streams, this paper investigates two sequence-labeling techniques: long short-term memory networks (LSTMs) and conditional random fields. We train predictive models utilizing data corpora collected from two Wizard-of-Oz experiments in which a human wizard played the role of the virtual agent unbeknownst to the student. Empirical results suggest that LSTMs that utilize game trace logs and facial action units achieve the highest predictive accuracy. This work can inform the design of intelligent virtual agents that leverage rich multimodal student interaction data in game-based learning environments.

## Keywords

Game-Based Learning, Virtual Agents, Deep Learning, Multimodal.

## 1. INTRODUCTION

Recent years have witnessed a growing interest in intelligent game-based learning environments because of their potential to

simultaneously promote student learning and create engaging learning experiences [23]. These environments incorporate personalized pedagogical functionalities delivered with adaptive learning techniques and the motivational affordances of digital games featuring believable characters and interactive story scenarios situated in meaningful contexts [13, 23]. A key feature of game-based learning environments is their ability to embed problem-solving challenges within interactive virtual environments, which can enhance students' engagement and facilitate learning through customized narratives, feedback, and problem-solving support [18, 25].

Game-based learning environments offer considerable opportunities for implementing virtual agents by delivering visually contextualized pedagogical strategies [14]. Intelligent virtual agents have been shown to deliver motivational benefits, promote problem-solving, and positively affect students' perception of learning experiences [14]. Virtual agents play a variety of roles in interactive learning environments including intelligent tutors, teachable agents, and learning companions [4].

A key challenge in developing intelligent virtual agents is devising accurate predictive models that dynamically attune pedagogical strategies to individual students using evidence from students' interactions with the learning environment. Previous research has focused on when to intervene [21] and what types of dialogue moves to make during students' problem-solving activities [3] to provide support in a timely, contextually relevant manner. Selecting appropriate pedagogical dialogue moves is critical [24] because failing to provide effective feedback may lead to decreased learning in a student experiencing boredom [1], lead a student who is confused to become disengaged [10], or negatively impact the outcome of dialogues [5].

Much of the previous work in this line of investigation has addressed this challenge through computationally modeling agents' *dialogue acts*, the underlying intention (e.g., greeting, question, suggestion) of the utterances, by utilizing sequences of actions within learning environments as evidence [2]. The current work builds on this by examining multimodal data streams, which

can provide rich evidence of students' cognitive and affective states, in addition to evidence captured from game trace logs. To effectively deal with the granular sequential data in parallel multimodal data streams, we investigate two sequence labeling techniques: a deep-learning technique, long short-term memory networks (LSTMs) [11]; and a competitive baseline approach, conditional random fields (CRFs) [26]. This work is inspired by the recent success of LSTMs in dealing with low-level data (e.g., speech signals), and particularly by their state-of-the-art performance in speech recognition tasks [16]. Additionally, hierarchical representation learning supported by deep learning provides advantages over other machine learning techniques by avoiding the need for labor-intensive feature engineering [16].

Our sequence labeling models are evaluated with 211 dialogue acts made by human wizards who interacted with 11 students playing CRYSTAL ISLAND, a game-based learning environment for middle school microbiology [23]. The interaction data include game trace logs, facial action units [17] processed from facial video recordings, and galvanic skin responses, all of which are utilized as input features for devising predictive models. Wizards used pre-designed utterances, which they selected from menus organized by dialogue act. Each selected utterance was then delivered to the student via speech synthesis. Wizards could observe the student's face, gaze, game screen, and voice while selecting dialogue moves, but facial action units, galvanic skin responses, and game trace logs were not directly accessible. We hypothesize that these unobserved multimodal data streams serve as proxies for the wizards' dialogue decisions and examine these as explanatory variables to predict the next dialogue act that a human wizard might choose.

LSTM and CRF models are devised utilizing subsets of the parallel multimodal data streams. Student-level cross-validation studies indicate that LSTMs utilizing game trace logs and facial action units outperform both CRFs and the majority class-based baseline with respect to predictive accuracy. Further, we find that the LSTM model effectively takes advantage of multimodal data streams, and it most effectively utilizes both game trace logs and facial action unit data. The results suggest that LSTM models can serve as the foundation for dialogue act modeling for intelligent virtual agents that dynamically adapts dialogues to individual students.

## 2. RELATED WORK

Recent work in game-based learning has explored a broad spectrum of subject matters ranging from computer science [18] and language to cultural learning [13]. Narrative-centered learning environments, which provide narrative adaptation for individual students in the context of intelligent game-based learning, have been found to deliver experiences in which learning and engagement are synergistic [13, 23]. Student interaction data from game-based learning activities has provided a rich source of information from which students' development of competencies [18, 25] and progress towards learning goals [19, 20] are diagnosed. Game-based learning environments can also be populated by virtual agents, whose design should consider students' cognitive and affective states [4, 14].

In parallel work on tutorial dialogue, it has been found that tutorial planning can take into account students' cognitive and affective states [7]. Planning dialogue moves and inducing turn-taking policies have been widely examined in supervised learning (e.g., hidden Markov models [2], directed graph representations [5]) and reinforcement learning [3, 21]. The approach described in

this paper is the first to investigate dialogue move classification using LSTMs and CRFs that take as input sequential multimodal data streams, which can serve as the foundation for guiding the dialogue of intelligent virtual agents in game-based learning environments.



Figure 1. The CRYSTAL ISLAND game-based learning environment.

## 3. CRYSTAL ISLAND

Over the past several years, our lab has been developing CRYSTAL ISLAND (Figure 1), a game-based learning environment for middle school microbiology [23]. Designed as a supplement to classroom science instruction, CRYSTAL ISLAND's curricular focus has been expanded to include literacy education based on Common Core State Standards for reading informational texts. The narrative focuses on a mysterious illness afflicting a research team on a remote island. Students play the role of a visitor who is drawn into a mission to save the team from the outbreak. Students explore the research camp from a first-person viewpoint, gather information about patient symptoms and relevant diseases, form hypotheses about the infection and its transmission source, use virtual lab equipment and a diagnosis worksheet to record their findings, and report their conclusions to the camp's nurse.

Extending the previous edition of CRYSTAL ISLAND, we incorporated a prototype virtual agent into the game to investigate both affective and cognitive influences on students' learning processes. This virtual agent, a young female scientist named Layla (Figure 2), was designed as a near-peer mentor who supports the student through dialogue-based interactions.

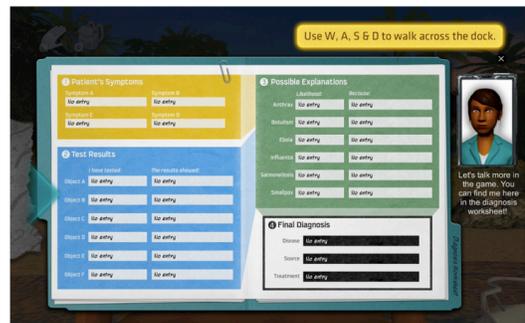


Figure 2. CRYSTAL ISLAND virtual agent.

In CRYSTAL ISLAND's virtual world, students interact with learning resources such as books and posters, as well as with non-player characters through informative menu-based dialogue. As students progress through the game, they collect evidence and record their hypotheses in a "diagnosis worksheet." The student meets Layla when the diagnosis worksheet is opened (Figure 2).

With Layla’s visual and speech synthesis prototypes in place, but no adaptive dialogue model implemented yet, a Wizard of Oz system was implemented to enable a human operator to provide the intelligence behind Layla’s dialogue. When the human “wizard” decides to initiate a dialogue move, she chooses one of six dialogue acts (Table 1) from a menu interface, then selects a dialogue utterance from the act’s set of pre-determined utterances. Layla then speaks the utterance through speech synthesis. The selection of dialogue moves was informed by the literature on dialogue systems for learning [8], as well as experience with a recent study conducted in the same middle school, in which pairs of middle school students interacted with CRYSTAL ISLAND together.

Three wizards controlled Layla’s dialogue in the game from a room separated from the students, while observing the students through a live feed that included the student’s facial video, the student’s gaze superimposed in real time over a video capture of the game screen, and the student’s voice as recorded through a headset microphone.

Data was collected in two studies implemented in the spring and summer of 2015 at a public middle school in Raleigh, North Carolina. In the spring study, participants were drawn from an after-school activity, and the summer study’s participants were from classroom pull-outs. Of the 11 students who participated, 7 were female and 4 were male, with an average age of 12 (SD = 1.1). The data corpus contains 211 virtual agent dialogue acts across the students (average number of acts: 19.2, maximum number of acts: 41, and minimum number of acts: 3).

**Table 1. Agent’s dialogue acts and distributions of their use.**

Dialogue Act	Distributions	Dialogue Act	Distributions
<i>Greeting</i>	58 (27.5%)	<i>Suggestion</i>	51 (24.2%)
<i>Question</i>	35 (16.6%)	<i>Feedback</i>	8 (3.8%)
<i>Acknowledgement</i>	43 (20.4%)	<i>Affective Statement</i>	16 (7.6%)

## 4. MULTIMODAL DATA

During the students’ interactions with CRYSTAL ISLAND, both game actions and parallel sensor data were captured to collect both cognitive and affective features of students’ experience. In the following subsections, we describe the three types of input data investigated in the present work.

### 4.1 Game Trace Logs

Students play CRYSTAL ISLAND using a keyboard and mouse. Student actions are logged for gameplay analysis and game telemetry [20]. In the present modeling work, seven key categories of actions are examined: moving around the camp, using the laboratory’s equipment to test a hypothesis about the disease and its source, conversing with non-player characters, reading complex informational texts about microbiology concepts, taking embedded assessments associated with the informational texts, interacting with the diagnosis worksheet, and experiencing dialogue moves with the virtual agent. The total number of distinct actions is 143.

A total of 4,117 student actions were logged along with 211 dialogue acts by the virtual agent in the training data. Students took an average of 19.5 actions between two adjacent dialogue acts, where the minimum and maximum number of actions between any two adjacent dialogue acts are 1 and 217, respectively.

## 4.2 Galvanic Skin Response

Galvanic skin response (GSR) is a measurement of the level of conductance across the surface of the skin, which is driven by the activity of the sympathetic nervous system. GSR reflects a variety of cognitive and affective processes, including attention and engagement [6, 22]. In addition, the presence of significant spikes in students’ GSR in response to certain events during a technology-supported learning activity has been found to be associated with learning-linked emotions and learning outcomes [12]. In this study, Empatica E4 bracelets on both wrists were used for GSR recording. These bracelets were chosen because, unlike palmar and fingertip GSR recording devices, they do not restrict the range of hand movement needed to play the game.

## 4.3 Facial Action Units

Facial expressions have been shown to have a relationship to self-reported and judged learning-centered affective states [1, 17]. Previous work has also found that facial expressions during learning can help predict a student’s learning gains, frustration, and engagement [27]. Facial expressions can be examined non-invasively through video recordings taken during a student’s interaction with a learning environment.

In this work, we observe facial expressions by analyzing a student’s facial action units, which capture movement of the muscles in the face. Facial action units are grounded in the Facial Action Coding System, which was devised to make observations about facial movements [9]. In this study, facial videos were recorded via a webcam and analyzed using FACET, an automated system devised for tracking facial action units, because it allows for frame-by-frame tracking in the facial videos without the time intensive effort of human-tagging facial action units. FACET is the next generation of the Computer Expression Recognition Toolbox [17], which has been validated for both adults and children. In this study, we considered the subset of facial action units provided by FACET (Table 2). In the following section, we describe the deep learning-based dialogue act classifier that utilizes these three data sources.

**Table 2. Facial action units examined.**

Inner Brow Raiser (AU1)	Upper Lip Raiser (AU10)	Tightener (AU23)
Outer Brow Raiser (AU2)	Lip Corner Puller (AU12)	Lip Pressor (AU24)
Brow Lowerer (AU4)	Dimpler (AU14)	Lips Part (AU25)
Upper Lid Raiser (AU5)	Lip Corner Depressor (AU15)	Jaw Droop (AU26)
Cheek Raiser (AU6)	Chin Raiser (AU17)	Lip Suck (AU28)
Lid Tightener (AU7)	Puckerer (AU18)	
Nose Wrinkler (AU9)	Lip Stretcher (AU20)	

## 5. LSTM-BASED DIALOGUE MOVE DECISION MODEL

Long short-term memory networks (LSTMs) have demonstrated significant success in dealing with a series of raw signals, such as speech, yielding state-of-the-art performance in speech recognition tasks [16]. This inspires our work, which deals with low-level sensor data such as GSRs and facial AUs. In the following subsections, we present a high-level description of LSTMs [11], introduce how multimodal input data are synchronized and encoded into a trainable format, and describe how the LSTM-based dialogue move prediction models are configured.

## 5.1 LSTM Background

LSTMs are a type of gated recurrent neural network specifically designed for sequence labeling on temporal data. LSTMs, like standard recurrent neural networks, take the approach of sharing weights across layers at different time steps. LSTMs feature a sequence of memory blocks that include one or more self-connected memory cells along with three gating units [11]. In LSTMs, the input and output gates modulate the incoming and outgoing signals to the memory cell, and the forget gate controls whether the previous state of the memory cell is remembered or forgotten. This structure allows the model to preserve gradient information over longer periods of time [11].

In the implementation of LSTMs investigated here, the input gate ( $i_t$ ), forget gate ( $f_t$ ), and candidate memory cell state ( $\tilde{c}_t$ ) at time  $t$  are computed by Equations (1)–(3), respectively, in which  $W$  and  $U$  are weight matrices for the input ( $x_t$ ) at time  $t$  and the cell output ( $h_{t-1}$ ) at time  $t-1$ ,  $b$  is the bias vector of each unit, and  $\sigma$  and  $\tanh$  are the logistic sigmoid and hyperbolic tangent function, respectively.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

Once these three vectors are computed, the current memory cell’s state is updated to a new state ( $c_t$ ) by modulating the current memory cell state candidate value ( $\tilde{c}_t$ ) via the input gate ( $i_t$ ) and the previous memory cell state ( $c_{t-1}$ ) via the forget gate ( $f_t$ ). Through this process, a memory block decides whether to keep or forget the previous memory state and regulates the candidate of the current memory state via the input gate. This step is described in Equation (4), in which  $\odot$  denotes element-wise multiplication:

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \quad (4)$$

The output gate ( $o_t$ ) calculated in Equation (5) is utilized to compute the memory cell output ( $h_t$ ) of the LSTM memory block at time  $t$ , modulating the updated cell state ( $c_t$ ) (Equation 6):

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Once the cell output ( $h_t$ ) is calculated at time  $t$ , the next step is to use the computed cell output vectors to predict the label of the current training example. For the dialogue move decision model, we use the final cell output vector ( $h_t$ ), assuming that  $h_t$  captures long-term dependencies from the previous time steps.

## 5.2 Data Encoding for Dialogue Move Decision Model

Each data stream from a suite of multimodal interaction data is of a sequential form. Because these data include fixed-rate recordings (e.g., facial action units and galvanic skin responses) with rates that differ between streams, as well as in-game action-driven recordings (e.g., game trace logs) with no set rate, the first step of data encoding is synchronizing input data across modalities.

We obtained from each student two series of galvanic skin responses (GSRs), one each for the left and right hand, as well as 19 facial action units (AUs). In the modeling work reported here, only the GSR information from the subject’s dominant hand is utilized, so GSR is represented by a one-dimensional vector. AUs are represented by a 19-dimensional vector space per time stamp. GSR and AUs were logged with the frequencies of approximately 4Hz and 30 Hz, respectively. Game traces were recorded as events

were triggered in the game, whenever the actions described in Section 4.1 were performed.

In contrast to GSR or AUs, which have continuous values, the game trace logs (GAME) consist of discrete indices for specific actions, indexed 1 to 143. To represent actions in a vector format, we employ the *one-hot-encoding* technique, in which a bit vector whose length is the total number of actions (143 in this work) is created while only the associated action bit is on (i.e., 1) while all other bits are off (i.e., 0). Once the vector representations for GAMEs are created, the next step is to synchronize the three data representations into an integrated representation.

To keep the length of data sequences manageable while preserving key game actions, we synchronize the multimodal data based on the game trace logs. All GSR and AU data collected between any two adjacent game actions are transformed into two vectors, using the following method:

- Vector 1: (75th percentile minus 50th percentile) per feature across all the data points between the two adjacent actions
- Vector 2: (50th percentile minus 25th percentile) per feature across all the data points between the two adjacent actions

We hypothesize that these two quartile-based vectors can capture variance of signals within an interval, while effectively avoiding outliers, smoothing out individual differences, and keeping the number of input features (183, or the sum of 143 for GAME, 38 for AU, and 2 for GSR) small enough to efficiently train LSTMs. Once these two vectors are created for the GSR stream and for each AU, the vectors are concatenated to the game trace log vector.

## 5.3 LSTM Model Configurations for Dialogue Move Decision

Prior to training LSTMs, the hyperparameters of the models must be determined. LSTM hyperparameters have often been explored using grid search or random search settings in the process of minimizing validation errors [20]. We adopt the grid search approach to empirically find an optimal configuration for a set of hyperparameters. In this work, we consider two hyperparameters: the number of hidden units for LSTMs among {32, 64} and the dropout rate [16], a model regularization technique, among {0.4, 0.7}. Both hyperparameters have significant influence on the performance of deep neural networks [11, 20].

In addition to LSTM-wide hyperparameters, this work also analyzes the isolated impacts of multimodal data sources. In order to perform this analysis, we examine all possible combinations of features, generating the following seven input feature sets: galvanic skin responses (GSRs), facial action units (AUs), game trace logs (GAMEs), GSRs and AUs, AUs and GAMEs, GSRs and GAMEs, and all three data sources. The dimension of a feature set is decided by summing up the dimensions of the features (see Section 5.2) that comprise the feature set.

In addition to the hyperparameters examined in the grid search, we apply a fixed value to the following hyperparameters for LSTMs: employing a softmax layer for classifying given sequences of interactions, adopting mini-batch gradient descent with a mini-batch size of 32, utilizing categorical cross entropy for the loss function, and employing a stochastic optimization method. The training process stops early if the validation score has not improved within the last 15 epochs. In this work, we evaluate our models using student-level leave-one-out cross validation, and so in each fold, 1 student’s data is used for testing

(completely hidden) out of 11 students, while 8 students' and 2 students' data are utilized as the training and validation set, respectively. Finally, the maximum number of epochs is set to 100.

## 6. EVALUATION

To evaluate the proposed LSTM-based dialogue act classification (cast as six-class classification), we search for an optimal set of hyperparameters through cross-validation in the previously discussed grid search setting, and then perform feature-set level predictive performance analyses based on the chosen hyperparameters. Additionally, we compare each LSTM-based computational model to a competitive approach based on linear-chain conditional random fields (CRFs) [26] as well as a majority class baseline using the same cross-validation split for a pairwise comparison. CRFs are trained using the Block-Coordinate Frank-Wolfe optimization technique [15], and we adjust the regularization parameter for the optimization technique among  $\{0.1, 0.5, 1.0\}$  to find optimal CRFs as we do in LSTMs.

Table 3 presents feature-set-level cross-validation results. LSTMs with the hyperparameter configuration of 64 hidden units and 0.7 dropout rate achieve the highest predictive accuracy (34.1%), and CRFs trained with the regularization parameters of 0.5 achieved the second highest accuracy (32.2%). We use raw correct and incorrect prediction counts to calculate accuracy rates rather than reporting fold-based averaged accuracy rates, in an effort to avoid the potential for skew brought on by the wide variation in the number of data points per student (min: 3; max: 41).

**Table 3. Student-level leave-one-out cross validation results across feature sets (64 hidden units and 0.7 dropout rate for LSTMs and 0.5 regularization parameter for CRFs).**

	LSTMs	CRFs
GSRs	28.0%	19.9%
AUs	21.8%	25.6%
GAMEs	29.4%	<b>32.2%</b>
GSRs / AUs	26.1%	22.3%
AUs / GAMEs	<b>34.1%</b>	30.8%
GSRs / GAMEs	29.9%	29.4%
GSRs / AUs / GAMEs	31.3%	27.0%

In the evaluation, LSTMs that achieve the highest predictive accuracy utilize AUs and GAMEs (LSTM<sub>AU/GAME</sub>), the accuracy of which constitutes a 43.9% marginal improvement over the baseline accuracy (23.7%). Note that the baseline accuracy is different from Table 1, because it is influenced by the random split made in cross validation. We conducted a Wilcoxon signed rank, a non-parametric statistical test for two related samples, to compare cross-validation results between the LSTM<sub>AU/GAME</sub> and the majority class baseline per fold. The test finds a statistically significant difference between LSTM<sub>AU/GAME</sub> and the baseline ( $Z=-2.25, p=0.024$ ). The differences between LSTM<sub>AU/GAME</sub> and the best performing CRFs ( $p=0.67$ ) and between the CRFs and the baseline ( $p=0.095$ ) are not statistically significant.

It is noteworthy that AUs by themselves do not achieve a high predictive accuracy. This can be partially explained by noting that the facial action unit data stream was often temporarily lost (a vector filled with zeros is used in this case for the missing data), usually when the subject's face was not properly situated within the camera screen. It is surprising, however, to see that partially-missing AUs synchronized with GAMEs data helped improve the prediction of the next virtual agent dialogue act by outperforming GAMEs models ( $Z=-1.71, p=0.088$ ) as well as AUs models ( $Z=-2.24, p=0.025$ ).

The LSTM<sub>AU/GAME</sub>'s outperformance might be explained by the information available to the human wizards as they chose dialogue acts: they were able to watch the subject's game play as well as facial expressions during the interaction with the game, which together potentially influenced the dialogue decisions. On the other hand, the AUs likely characterize aspects of the subject's affective states, and they can contribute to the improved predictive performance synergistically with GAMEs in LSTMs.

Overall, GAMEs serve as a strong predictor relative to other independent data sources: GAMEs models (29.4%) outperform the other two independent models induced utilizing GSRs (28.0%) or AUs (21.8%); in the meantime, each feature set that leverages GAMEs in addition to other data sources outperforms the corresponding feature set without the GAMEs (e.g., GSRs, AUs, and GAMEs (31.3%) vs. GSRs and AUs (26.1%)). Sequences of actions in the GAMEs may reflect students' underlying cognitive states such as plans, goals, and knowledge during problem-solving activities [19, 20], which wizards attempted to address through their dialogue act choices. It is expected that LSTMs' capacity for hierarchical feature abstraction enables them to recognize these high-level patterns from low-level action sequences.

It is interesting to observe that GSRs by themselves outperform the baseline but incorporating GSRs with AUs and GAMEs (31.3%) does not outperform LSTM<sub>AU/GAME</sub> (34.1%). Although much of the previous research has used GSR data streams as evidence for modeling humans' affective and cognitive states [22], the findings of the study presented here suggest that GSR collected using wrist sensors may not be the most informative data source for predicting a human-operated virtual agent's next dialogue act, particularly when other data sources are available.

## 7. CONCLUSION AND FUTURE WORK

Dialogue modeling is a critical functionality for pedagogically adaptive virtual agents. This paper has presented two sequence-modeling approaches to classifying human wizards' dialogue moves when utilizing multimodal observation sequences. Both conditional random fields (CRFs) and long short-term memory networks (LSTMs) have demonstrated significant promise as effective modeling techniques on the sequential, parallel, multimodal data from game trace logs, galvanic skin response, and facial action units. Both CRFs and LSTMs outperform the majority class-based baseline with respect to predictive accuracy, while LSTMs achieve the highest predictive accuracy. Feature-level analyses of LSTMs suggest that even incomplete facial action unit data can augment LSTMs' predictive performance along with game trace logs, while game trace logs serve as strong predictor in both computational approaches. Along with achieving a substantial improvement in the use of sequence labeling techniques, this work suggests a number of directions for future work.

First, it will be important to extend the current models to determine the timing of dialogue acts. Together with the current work, this will further enhance the potential capacity for intelligent virtual agents to provide adaptive pedagogical support. Second, it will be important to examine the relationships between students' cognition and affect as perceived by human wizards, and to investigate how they influence wizards' dialogue decision-making. Because multimodal interaction data may reflect students' affective and cognitive states, identifying the relationship between student models and dialogue acts can guide the design of advanced tutorial dialogue management capabilities for pedagogical agents.

## 8. ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation through Grant CHS-1409639. Any opinions, findings, conclusions, or recommendations expressed are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

## 9. REFERENCES

- [1] Baker, R., D’Mello, S., Rodrigo, M.M. and Graesser, A. 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.
- [2] Boyer, K., Phillips, R., Ha, E., Wallis, M., Vouk, M. and Lester, J. 2010. Leveraging Hidden Dialogue State to Select Tutorial Moves. *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*. 66–73.
- [3] Chi, M., Vanlehn, K., Litman, D. and Jordan, P. 2011. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*. 21, 1-2, 83–113.
- [4] Chou, C.Y., Chan, T.W. and Lin, C.J. 2003. Redefining the learning companion: The past, present, and future of educational agents. *Computers and Education*. 40, 3, 255–269.
- [5] D’Mello, S.K., Olney, A. and Person, N.K. 2010. Mining Collaborative Patterns in Tutorial Dialogues. *Journal of Educational Data Mining*. 2, 1, 1–37.
- [6] Dawson, M.E., Schell, A.M. and Filion, D.L. 2007. The Electrodermal System. *The Handbook of Psychophysiology*. 200–223.
- [7] DeVault, D. et al. 2014. SimSensei Kiosk : A Virtual Human Interviewer for Healthcare Decision Support. *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*. 1061–1068.
- [8] Dweck, C.S. 2002. The development of ability conceptions.
- [9] Ekman, P. and Friesen, W. V 1977. Facial action coding system.
- [10] Forbes-Riley, K. and Litman, D. 2012. Adapting to Multiple Affective States in Spoken Dialogue. *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 217–226.
- [11] Graves, A. 2012. *Supervised sequence labelling with recurrent neural networks*. Springer.
- [12] Hardy, M., Wiebe, E., Grafsgaard, J., Boyer, K. and Lester, J. 2013. Physiological Responses to Events During Training: Use of Skin Conductance to Inform Future Adaptive Learning Systems. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2101–2105.
- [13] Johnson, W.L. 2010. Serious use of a serious game for language learning. *International Journal of Artificial Intelligence in Education*. 20, 175–195.
- [14] Johnson, W.L. and Lester, J.C. 2015. Face-to-Face Interaction with Pedagogical Agents, Twenty Years Later. *International Journal of Artificial Intelligence in Education*. 25, 25–36.
- [15] Lacoste-Julien, S., Jaggi, M., Schmidt, M. and Pletscher, P. 2013. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. *Proceedings of the 30th International Conference on Machine Learning*. 28, 9.
- [16] LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep Learning. *Nature*. 521, 7553, 436–444.
- [17] Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J. and Bartlett, M. 2011. The Computer Expression Recognition Toolbox (CERT). *Automatic Face Gesture Recognition and Workshops (FG 2011)*. 298–305.
- [18] Min, W., Frankosky, M., Mott, B., Rowe, J., Wiebe, E., Boyer, K. and Lester, J. 2015. DeepStealth: Leveraging Deep Learning Models for Stealth Assessment in Game-Based Learning Environments. *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, 277–286.
- [19] Min, W., Ha, E.Y., Rowe, J., Mott, B. and Lester, J. 2014. Deep Learning-Based Goal Recognition in Open-Ended Digital Games. *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. 37–43.
- [20] Min, W., Mott, B., Rowe, J., Liu, B. and Lester, J. 2016. Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks. *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. In Press.
- [21] Mitchell, C., Boyer, K. and Lester, J. 2013. Evaluating State Representations for Reinforcement Learning of Turn-Taking Policies in Tutorial Dialogue. *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 339–343.
- [22] Poh, M.Z., Swenson, N.C. and Picard, R.W. 2010. A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *IEEE Transactions on Biomedical Engineering*. 57, 5, 1243–1252.
- [23] Rowe, J., Shores, L., Mott, B. and Lester, J. 2011. Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*. 21, 1-2, 115–133.
- [24] Shute, V.J., D’Mello, S., Baker, R., Cho, K., Bosch, N., Ocumpaugh, J., Ventura, M. and Almeda, V. 2015. Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Computers & Education*. 86, 224–235.
- [25] Shute, V.J. and Ventura, M. 2013. *Measuring and supporting learning in games: Stealth assessment*. Cambridge, MA: The MIT Press.
- [26] Sutton, C. and McCallum, A. 2012. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*. 4, 4, 267–373.
- [27] Vail, A., Grafsgaard, J., Wiggins, J., Lester, J. and Boyer, K. 2014. Predicting Learning and Engagement in Tutorial Dialogue: A Personality-Based Model. *Proceedings of the 16th ACM International Conference on Multimodal Interaction*. 255–262.

# Exploring the Impact of Data-driven Tutoring Methods on Students' Demonstrative Knowledge in Logic Problem Solving

Behrooz Mostafavi  
North Carolina State University  
Raleigh, NC 27695  
bzmostaf@ncsu.edu

Tiffany Barnes  
North Carolina State University  
Raleigh, NC 27695  
tmbarnes@ncsu.edu

## ABSTRACT

We have been incrementally adding data-driven methods into the Deep Thought logic tutor for the purpose of creating a fully data-driven intelligent tutoring system. Our previous research has shown that the addition of data-driven hints, worked examples, and problem assignment can improve student performance and retention in the tutor. In this study, we investigate how the addition of these methods affects students' demonstrative knowledge of logic proof solving using their post-tutor examination scores. We have used data collected from three test conditions with different combinations of our data-driven additions to determine which methods are most beneficial to students who demonstrate higher or lower knowledge of the subject matter. Our results show that students who are assigned problems based on profiling proficiency compared to prior exemplary students with similar problem-solving behavior show higher examination scores overall, and the use of proficiency profiling increases retention and reduces the amount of time taken in-tutor for lower performing students in particular. The results from this study also helps differentiate the behavior of higher and lower performing students in tutor, which can allow quicker interventions for lower proficiency students.

## Keywords

Data-driven Methods, Proficiency Profiling, Tutoring Systems

## 1. INTRODUCTION

We have been incrementally adding data-driven methods for problem assignment[9, 10], hint generation[3], and worked examples[11] to the Deep Thought logic tutor to create a fully data-driven tutoring system. While we have observed improvements in student retention and tutor scores with each of these additions, we have not studied the difference in post-tutor examinations when these methods are combined in different test conditions. We seek to understand how the

specific methods of problem assignment and combination of hints and worked examples may have impacted student performance on related questions on the course midterm exam.

In this paper we compare two classrooms of students using different test conditions of Deep Thought, with different combinations of problem assignment, hints and worked examples. Students' knowledge of logic were evaluated in two problems on a mid-term exam, and these scores were used to differentiate high and low proficiency students for our analysis. The results from our analysis show that high performing students benefit most from problem-solving opportunities, while low performing students benefit most from problem assignment based on proficiency profiling, comparing current students to prior exemplary students with similar behavior. We conclude that the use of proficiency profiling is the most effective method for increasing retention and reducing time spent in the Deep Thought tutor, and result in higher overall examination scores. The results from this study also help differentiate the behavior of higher and lower performing students in tutor, allowing for quicker interventions for lower proficiency students who need additional instructional support.

## 2. RELATED WORK

Koedinger et al.[6] summarized the general process of intelligent tutoring systems: the system selects an activity for the student, evaluates each student action, suggest a course of action (either via hints, worked examples, or another form of feedback), and finally updates the system's evaluation of the student's skills. An effective tutor should adapt instruction according to the student's current knowledge level [1]. However, in order to make instructional decisions, most ITSs either use fixed pedagogical policies providing little adaptability, or expert-authored pedagogical rules based on existing instructional practices [1, 14]. Intelligent tutoring systems with data-driven methods can be more adaptive by leveraging previous student data in order to complete one or more of these steps. Data-driven approaches to making effective pedagogical decisions – in particular selecting problems, when to apply worked examples, and the type of hint or feedback to provide – would mostly bypass the need for expert involvement in creating and improving the effectiveness of ITSs. In practice, incorporating student data has been shown to increase learning efficiency and predict student behavior. This, in particular is why we use data-driven knowledge tracing (DKT) of rule applications within

the Deep Thought logic tutor to facilitate profiling of students' proficiency.

In the remainder of this section, we describe the Deep Thought logic tutor and the data-driven additions implemented. We then describe the system and data used to evaluate the effectiveness of these data-driven methods in Deep Thought. After reporting the results of this evaluation, we discuss the implications for future design decisions in the tutor, and present our conclusions.

## 2.1 The Deep Thought Tutor

We have been examining the potential for data-driven methods to improve learning gains in a complex problem solving domain by incrementally augmenting the Deep Thought logic tutor. Deep Thought is a tutor for graphically constructing propositional logic proofs. Deep Thought presents proof problems consisting of logical premises and a conclusion to be derived using logical axioms. Deep Thought is divided into 6 levels of logic proof problems. In previous work with the Deep Thought logic tutor, we have been implementing data-driven methods for several of the intelligent tutor steps. We implemented a data-driven mastery learning system (DDML) to track student actions and assign appropriate problems based on the student's current level of proficiency [9]. The problem set was split into two tracks: a high proficiency track and a low proficiency track for Levels 2–6, with Level 1 containing a common set of problems for initial track assignment. We tracked student actions throughout their time in the tutor, and in particular their application of logical rules to construct logic proofs. Based on their correct or incorrect application of logical rules, the DDML updated a set of rule scores, one score for each logical rule. At the end of each level, the students' rule scores were weighted based on expert-determined priorities; rules deemed by experts to be of high importance to solving the problems in that level were weighted higher than rules that were not. These weighted scores were summed together, and compared to the average rule scores in the previous semester's data; based on this comparison, students were assigned to the higher or lower proficiency path. We tested Deep Thought with the DDML incorporated and found students completed, on average, 79% of all six levels in the tutor assignment. Student retention rate was 55%. This was an improvement over the non-DDML version of Deep Thought (61% tutor completion on average, and 31% retention rate).

We later incorporated a data-driven proficiency profiler (the DDPP) to replace the expert-determined priorities [10][8]. The DDPP is a system that 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, with rule scores weighted as determined through principal component analysis (PCA). Based on how similar exemplary students were assigned in subsequent levels, 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 determined similar problem solving strategies among the exemplars by clustering the exemplars' rule scores based on

hierarchical clustering. Expert weighting was replaced by PCA of the frequency of the rules used for each exemplar for each level, accounting for 95% variance of the results. For each rule, its PCA coefficient is the new weight for that rule score. 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 examines 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. 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, the student's proficiency is calculated using the average scores, as in the original DDML system.

Providing hints to students in the course of an intelligent tutor as a possible form of step-based feedback has the potential to increase learning gains. Razzaq, Leena, and Hefernan [12] found that learning gains increased for students given on-demand hints in comparison to students who were provided hints proactively. In Deep Thought, the hint system used is called Hint Factory. Hint Factory is an automatic data-driven hint generator that converts an interaction network graph of student trace behavior into a Markov decision process (MDP) to automatically select on-demand hints for students upon request, based on their individual performance on specific problems. The MDP is data-driven, using actions logs from previous Deep Thought use in the classroom to assign weight to proof-state actions based on whether or not that action ultimately led to successful completion of the proof. These hints help students solve problems by suggesting what step should be taken next on a multi-step problem. Hint Factory has been implemented in the Deep Thought logic tutor to automatically deliver context-specific hints to students during problem-solving [4]. In a previous study Hint Factory was shown to provide context-specific hints over 80% of the time [3]. In a pilot study, Barnes & Stamper found that Hint Factory can provide sufficient, correct, and appropriate hints for the Deep Thought Logic tutor and help students to solve more logic proof problems in the same span of time [4]. However, we currently cannot determine the effect hints would have in addition to the DDML or DDPP; so far, students using either of those versions of Deep Thought did not use hints often enough for any meaningful analysis.

Adding worked examples as a supplement to traditional problem solving can also be beneficial [2, 13]. Hilbert and Renkl [5] found that improved learning outcomes occurred when providing worked examples with a prompt, and proposed that this was due to allowing the students to have a greater cognitive load at once. McLaren and Isotani [7] compared three tutors using all worked examples, all traditional unguided problem solving, and a mix of worked examples and problem solving. Each group achieved similar learning gains, but the students who were given all worked examples required less time to achieve those gains. We added worked

examples to the version of Deep Thought with the DDML incorporated[11]. Worked examples were generated based on previous best student solutions, and procedurally annotated. They were presented to students randomly on a per-problem basis, based on the number of problems they had solved in that level already. We found that student retention overall was 90%, and students completed 94% of the tutor on average. This percentage was significantly higher than that of the DDML alone.

### 3. METHODS

Deep Thought was used as a mandatory homework assignment by students in an undergraduate “discrete mathematics for computer scientists” course in Fall 2015 and Spring 2016. Students in the two semesters were taught by different instructors. Students were assigned Levels 1–6 of Deep Thought for full credit, with partial credit awarded proportional to the number of levels completed. For this study, we compare the data from three Deep Thought test conditions used across the two semesters to differentiate the effect of our data-driven methods on student performance.

The first group evaluated for this study were assigned only problem-solving opportunities (PS group,  $n = 26$ ). The problem assignment system used was the DDML system described in the previous section, where students were assessed between levels and placed on either a high or low proficiency track in the next level. This group of students were taken only from the Fall 2015 semester, as there existed no equivalent test condition in Spring 2016.

The second group of students were randomly assigned either problem-solving opportunities or worked examples of the same problems within each level (PS/WE group,  $n = 179$ ), with the number of problem-solving opportunities controlled to match the number of problems solved by the PS group. Like the PS group, the PS/WE group were assigned proficiency tracks using the DDML. However, because individual rule application scores were updated at each step in worked examples as if a student had applied that rule in while problem solving, most students were consistently assigned to the high track in most levels, and were only assigned the low track when their individual performance was below satisfactory. This group of students were taken from both the Fall 2015 and Spring 2016 semesters.

The third group of students were randomly assigned problem-solving opportunities or worked examples in the same manner as the PS/WE group, but with the DDPP method assigning proficiency tracks instead of the DDML, where students were assigned the same proficiency track as prior students who most closely matched their rule application behavior (DDPP group,  $n = 61$ ). This group of students were also taken from both the Fall 2015 and Spring 2016 semesters. Students in all three groups had access to on-demand hints.

All students were evaluated using two proof problem questions as part of a mid-term examination, which was used as a post-test for this study. Students performance in the post-test for both Fall 2015 and Spring 2016 were graded by the same teaching assistant, ensuring consistent evaluation across all results. Students were separated for evaluation

by performance on the post-test and by the predominant track level in Deep Thought. The post-test was a set of two proofs students had to solve on paper for a midterm exam. These questions were hand-graded with partial credit given based on the percentage of the proofs completed and points taken off for misapplication of rules and skipping non-trivial rules. We considered two performance levels: post-test scores greater than or equal to 80% (AB), or less than 80% (CDF). The post-test scores mark the final evaluation of students’ ability to solve proof problems, and occurs immediately following the Deep Thought tutor homework assignment.

The second dimension we studied was the proportion of high to low proficiency track levels the students completed. Students who were assigned to the high proficiency track in a level had the ability to finish on either the high or low proficiency track depending on the number of problems skipped within that level. Students who completed more levels on the high track than the low track were marked as high track students, and students who completed more levels on the low track than the high track were marked as low track students. The track assignments indicate the number and complexity of problems students received, with the low track having more problems of lower complexity, and the high track having fewer problems of higher complexity. The tracks were designed so that students would have a similar number of rule applications across the tracks, even though the number of problems differs. Typically, the low track has three problems with expert solutions using 5 rule applications, and the high track has 2 problems with expert solutions using 7 – 8 rule applications - meaning that both tracks minimally required about 15 total rule applications (though students typically used more).

In addition to post-test and predominant track level, we examined total time in tutor, average time spent per problem, percentage of correct rule applications out of all rule applications, and the total number of rule applications. We also looked at ancillary behaviors (hint usage, skipped problems, and reference requests) that could differentiate high and low performing students. We compared these metrics to better understand the impact of worked examples, hints, and data-driven track selection on student performance. The results of this descriptive analysis are presented in the next section.

### 4. RESULTS

Table 1 displays the percentage of AB students in each of the PS, PS/WE, and DDPP groups for all students, as well as students who completed the majority of the tutor in either the high or low tracks. Table 1 also displays the percentage of students in each group and each track who dropped out of the tutor before full completion, as this is one of the metrics we have used to judge the effectiveness of our data-driven methods. In our previous work using the same version of Deep Thought, we found that students completed 94% of the tutor on average, with a retention rate of 90%. The average percent tutor completion for the groups in this study were consistent with these numbers (PS: 95%, PS/WE: 93%, DDPP: 94%).

The first interesting result of note is that the percentage of students who performed better on the post-test was higher

**Table 1: Percentage of AB Students and Percentage of dropped students in the PS, PS/WE, and DDPP groups.**

Condition	ALL	High Track	Low Track
	<i>n</i>	% AB Students	
PS	26	65.38	63.16
PS/WE	179	49.72	36.67
DDPP	61	63.93	61.76
	<i>n</i>	% Dropped Students	
PS	26	3.85	5.26
PS/WE	179	11.73	36.67
DDPP	61	9.84	8.82

for for the PS (65%) and DDPP (64%) groups than for the PS/WE group (50%), across all the students, as well as within the high and low track groups. In the PS group, students who completed more levels on the high track displayed a higher overall proficiency of the subject matter than those who finished more often on the low track (71% vs 63%, respectively), as did students in the PS/WE group (52% vs 37%).

However, students in the DDPP group showed a consistent level of proficiency regardless of the tracks completed (66% vs 61%), which makes sense considering that these students were matched to previous successful students who displayed similar rule-application behavior, and had a more even placement within the high and low tracks compared to the PS group, who had even placement among tracks, but within the context of their own performance compared to expert-decided thresholds. The DDPP group also had higher placement compared to the PS/WE group, who were placed on the high track much more often than not due to the inclusion of worked examples. A Kruskal-Wallis test for one-way analysis of variance showed no significant difference between groups ( $p = 0.22$ ).

Students also had a higher retention rate in both the PS (4%) and DDPP (10%) groups compared to the PS/WE group (12%). It is especially interesting to see the drop rate among low track students in the PS/WE group, who had a much lower retention rate among all the students in the study. Because students in the PS/WE group were more often that not placed in the high track in each level, for students to end up on the low track indicates a high level of problem-skipping among these students. We can conclude that low performing students who are not intelligently assigned problems based on their problem-solving performance appear to gain little from worked examples.

While it may be tempting to declare problem-solving opportunities with no worked examples as the best performing pedagogical choice among the three groups based on these numbers alone, a look into additional performance metrics gives some more insight. Table 2 presents the amount of time spent in tutor and on each problem, as well as the percentage of correct and total rule applications for each group, separated by track. The numbers presented are the median values for each metric, since the distributions of scores were highly skewed and non-normal, and none of the differences were significant due to low sample size within each subgroup.

As shown in Table 2, among AB students in all three groups, the total time spent in tutor appears similar, although the mean time for high-track students was lower for DDPP ( $M = 3.95hr$ ,  $SD = 6.21hr$ ) compared to PS/WE ( $M = 4.46hr$ ,  $SD = 9.13hr$ ) and PS ( $M = 6.66hr$ ,  $SD = 9.91hr$ ). The mean time for low-track students was lower for PS ( $M = 4.63hr$ ,  $SD = 9.55hr$ ) and DDPP ( $M = 5.48hr$ ,  $SD = 5.42hr$ ) than the PS/WE ( $M = 7.74$ ,  $SD = 9.76$ ). The means of average problem time, percentage of correct rule applications, and number of rule applications were consistent with the median values presented in Table 2 across all three groups. Note that low-track students in the PS/WE groups had the lowest percentage of correct rule applications, and the highest number of total rule applications among all the groups. This means they are doing more work, but a lower percentage of it is correct.

As shown in Table 2, among CDF students in all three groups, the total time spent in tutor is dramatically different, with PS spending 3 to 4 times as long in the tutor than PS/WE and DDPP groups. This ratio is also similar in the average problem time for high and low track students, and the number of total rule applications for high track students. Therefore, while problem-solving only (PS) may have a slightly higher overall success rate in helping students learn proof problem solving and remain in the tutor than the DDPP students, for students who are less prepared, PS results in a much higher time spent in the tutor, with little return on the time investment. Therefore, for students who have a better grasp of the subject matter, pure problem-solving may offer a slightly better option for getting through the assigned tutor, although the differences between problem solving, problem solving and worked examples, and proficiency profiled assigned problem solving and worked examples are minimal. However, for less prepared students, pure problem-solving opportunities offer little to guide students to higher understanding of the material, and in general, the DDPP offers a much better path to completing the tutor in far less time for both AB and CDF students, giving students the opportunity to encounter all the subject matter and have a greater chance of learning the material, resulting in higher overall post-test scores.

Completing the tutor assignment is important for students; however, since we want to make sure that students are learning the material well, mid-term examination scores are ultimately a higher gauge for learning success. Among all the experimental groups in this study, at most 65% of students were performing at A or B grade level on the mid-term examination. We would like to increase this percentage of AB students, so the question at this point is: Is it possible for us to predict low exam scores based on in-tutor data for early intervention?

We first look at the differences between AB and CDF students in Table 2, with the assumption that the DDPP method offers the best overall chance of success for students. For high track students, total tutor time, average problem time, percentage of correct rule applications, and total rule applications are consistent between AB and CDF students. However, for low track students, average problem time, percentage of correct rule applications, and total rule applications show a higher difference. CDF students spent twice as

Table 2: Total Time, Average Problem Time, Percentage of Correct Rule Applications, and Total Rule Applications for AB and CDF students in the PS, PS/WE, and DDPP groups, separated by High and Low Track. The numbers listed are all median values.

		AB STUDENTS			CDF STUDENTS			
		PS	PS/WE	DDPP	PS	PS/WE	DDPP	
<i>HIGH TRACK</i>	<i>n</i>	5	78	18	<i>n</i>	2	71	9
<i>Total Tutor Time (hr)</i>		2.47	2.37	2.80		12.8	3.75	3.17
<i>Average Problem Time (min)</i>		9.89	11.1	12.1		52.3	18.4	16.0
<i>% Correct Rule Applications</i>		60.8	63.5	58.5		64.1	56.9	62.3
<i>Total Rule Applications</i>		258	214	203		471	255	204
<i>LOW TRACK</i>	<i>n</i>	12	11	21	<i>n</i>	7	19	13
<i>Total Tutor Time (hr)</i>		1.80	3.33	3.67		17.2	5.96	4.98
<i>Average Problem Time (min)</i>		6.76	15.2	15.0		60.1	25.0	30.4
<i>% Correct Rule Applications</i>		68.8	45.5	57.0		48.7	45.7	47.0
<i>Total Rule Applications</i>		201	404	291		382	394	389

long on average per problem than AB students, and applied rules correctly less than half of the time, while AB students applied rules more than half of the time. CDF students also attempted applying rules 25% more overall than AB students.

Since the performance differences between AB and CDF students are not as apparent for high track students, we look at ancillary tutor behavior to make a better distinction. Table 3 shows the number of requested hints, the number of skipped problems, and the number of rule reference requests (descriptions of logic rule operations) made by students in all groups. For the DDPP group, the most apparent difference among AB and CDF students are the number of hints requested, with the CDF group requesting 32 hints ( $M = 50, SD = 57$ ) compared to 17 ( $M = 32, SD = 42$ ) for the AB group. This difference in hints requested between AB and CDF students is also consistent across all groups and both high and low track students. We conclude that for high track students, we can differentiate between higher and lower proficiency students using hint request behavior, and for low track students, we can differentiate higher and lower proficiency students using the amount of time spent on average per problem and the percentage of correct rule applications. This allows the possibility of making an intervention during a student's progress through Deep Thought in the case that a student requires additional feedback or aid from an instructor due to a lesser understanding of the subject matter.

Table 3: Number of Hints, number of Skips, and number of Rule Reference requests for AB and CDF students in the PS, PS/WE, and DDPP groups, separated by High and Low Track. The numbers listed are all median values.

	PS		PS/WE		DDPP	
	AB	CDF	AB	CDF	AB	CDF
<i>HIGH</i>						
<i># Hint</i>	95	166	12	26	17	32
<i># Skip</i>	5	16	1	1	0	2
<i># Ref</i>	151	168	76	145	111	92
<i>LOW</i>						
<i># Hint</i>	30	104	31	44	19	26
<i># Skip</i>	1	0	30	24	3	15
<i># Ref</i>	77	224	60	271	55	109

## 5. CONCLUSION

In this paper we compared two classrooms of students using different test conditions of Deep Thought, with different combinations of problem assignment (DDML or DDPP) and the addition of worked examples, for the purpose of understanding how the specific methods of problem assignment and combination of hints and worked examples affect high and low performing students, as evaluated using mid-term examination scores. We found that for higher proficiency students who have a firmer grasp of the subject matter, problem-solving opportunities offer the best chance of completing the tutor in a timely manner; however, the addition of worked examples does not significantly detract from these students' learning experience. The method of problem assignment (DDML or DDPP) does not have a noteworthy effect on high student performance.

For lower proficiency students, we found that problem-solving opportunities alone with DDML problem assignment offered little to guide students to higher understanding of the material, and greatly extended the amount of time students spent in the tutor with little learning benefit. The addition of worked examples helped these students get through the tutor faster, however these students had a lower retention rate than any other students and lower examination scores. We conclude from these results that updating our data-driven skill estimates equally for viewing or applying rules resulted in students being assigned to the high-track when they were not prepared to solve harder problems. With proficiency profiling – matching students to previously successful students and the paths they take through the tutor – we can reduce the amount of time spent in tutor, increase retention, and make better use of worked examples by giving them alongside problems that better match an individual student's proficiency level. This results in similar performance to problem solving alone in terms of retention and knowledge gained, but with a lot less time spent in the tutor for lower-proficiency students. We conclude that our DDPP method offers the best overall possibility of success for students completing the Deep Thought tutor in a timely manner, learning the subject matter, and performing well on post-tutor examinations.

## 6. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grants 1432156 and 0845997.

## 7. REFERENCES

- [1] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive Tutors: Lessons Learned. *The journal of the learning sciences*, 4(2):167–207, 1995.
- [2] R. K. Atkinson, S. J. Derry, A. Renkl, and D. Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, 70(2):181–214, 2000.
- [3] T. Barnes, J. Stamper, L. Lehmann, and M. J. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197 – 201, 2008.
- [4] T. Barnes, J. Stamper, L. Lehmann, and M. J. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197 – 201, 2008.
- [5] T. S. Hilbert and A. Renkl. Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior*, 25(2):267–274, 2009.
- [6] K. R. Koedinger. New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization. *AI Magazine*, 34(3):27–41, 2013.
- [7] B. M. McLaren and S. Isotani. When is it best to learn with all worked examples? In *Artificial Intelligence in Education*, pages 222–229, 2011.
- [8] B. Mostafavi and T. Barnes. Data-driven Proficiency Profiling - Proof of Concept. In *Proceedings of the 6th International Conference on Learning Analytics and Knowledge (LAK 2016)*. In Press., 2016.
- [9] B. Mostafavi, M. Eagle, and T. Barnes. Towards data-driven mastery learning. In *Proceedings of the 5th International Conference on Learning Analytics and Knowledge (LAK 2015)*, pages 270–274, 2015.
- [10] B. Mostafavi, Z. Liu, and T. Barnes. Data-driven Proficiency Profiling. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, pages 335–341, 2015.
- [11] B. Mostafavi, G. Zhou, C. Lynch, M. Chi, and T. Barnes. Data-driven Worked Examples Improve Retention and Completion in a Logic Tutor. In *Proceedings of the 17th International Conference on Artificial Intelligence in Education (AIED 2015)*, pages 726–729, 2015.
- [12] L. Razzaq and N. T. Heffernan. Hints: is it better to give or wait to be asked? In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*, pages 349–358. Springer, 2010.
- [13] J. Sweller and G. A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):59–89, 1985.
- [14] K. VanLehn. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16(2):227–265, 2006.

# Properties and Applications of Wrong Answers in Online Educational Systems

Radek Pelánek  
Masaryk University Brno  
xpelane@mail.muni.cz

Jiří Řihák  
Masaryk University Brno  
thran@mail.muni.cz

## ABSTRACT

In online educational systems we can easily collect and analyze extensive data about student learning. Current practice, however, focuses only on some aspects of these data, particularly on correctness of students answers. When a student answers incorrectly, the submitted wrong answer can give us valuable information. We provide an overview of possible applications of wrong answers and analyze wrong answers from three different educational systems (geography, anatomy, basic arithmetic). Using this cross-system comparison we illustrate some common properties of wrong answers. We also propose techniques for processing of wrong answers and their visualization, particularly an approach to item clustering based on community detection in a confusion graph.

## 1. INTRODUCTION

A key advantage of computerized educational systems is their potential for personalization. By analyzing students' answers we can estimate their knowledge using student modeling techniques and adapt the behaviour of a system to the needs of individual students. Student models [6] typically utilize only information about correctness of answers. Online systems, however, collect (or can easily collect) much richer information, e.g., timing information [18] and specific details about answers and individual steps. In this work we focus on analysis of wrong answers.

Wrong or incomplete answers from online educational systems have been studied previously, but mostly just as a supplementary analysis to other research interests. For example, analysis of programming assignments in MOOCs [9, 14] shows that the distribution of wrong answers is highly skewed, containing few very common wrong answers. This research does not, however, focus on analysis of wrong answers, but rather on finding similar or equivalent solutions and their visualizations (as there are many ways how to write the same program) [7].

The observation that distribution of wrong answers is highly skewed holds not only for programming assignments, but also for other domains. For example, common wrong answers have been used for student modeling in mathematics [29], but this work uses only information about whether the wrong answer is common or not, it does not utilize actual values of wrong answers. Specific student answers were also modeled [8], but authors present only overall accuracy of the proposed model without discussion of specific mistakes.

Data analysis techniques has been used for analysis of mathematical errors with the goal of classification (explanation) of answers [13, 24]. The results show that it is possible to classify most wrong answers into one of few categories. Other data-driven techniques in educational data mining have focused mainly on programming assignments [10, 21]. Rather than “wrong answers” they utilize “incomplete solutions” and use them for automatic generation of hints (changes towards a correct solution).

In the wider context, wrong answers are related to misconceptions, which are intensively studied in pedagogical literature, e.g., misconceptions in mathematics [26] or chemistry [22]. This line of research focuses on understanding “buggy rules” used by students [4]. These rules are useful not just for educating teachers about student thinking, but also in development of intelligent tutoring systems. They can be also used as a basis of erroneous examples [1, 11]. Research in this direction is typically based on expert insight using only relatively small (and often qualitative) data and the focus is typically on complex skills.

In this work we focus on automatic techniques for analysis of large quantitative data, dealing with simple skills (learning of declarative knowledge and simple procedures). We describe analysis of wrong answers from three educational systems. Although the used systems share similar basic principles they cover widely different domains (geography, anatomy, basic arithmetic) and different learner populations (from kindergarten to university students). Thanks to the size of the used data set (millions of answers), results provide interesting insights into properties and potential of wrong answers. We describe specific examples of analysis and propose novel techniques for analysis and visualization of wrong answers. A key observation is that wrong answers in our three domain (geography, anatomy, basic arithmetic) share many properties and thus it should be feasible to carry insights and analysis techniques across domains.

## 2. POTENTIAL APPLICATIONS OF WRONG ANSWERS

In this section we outline potential applications of wrong answers. The presented applications are rather general and for a specific application they need to be more precisely quantified. In the next section we provide such specific analysis for three particular domains.

### 2.1 Student and Domain Modeling

Student and skill models [6] typically utilize only binary information about correctness of an answer (correct/incorrect). A more thorough analysis of wrong answers may improve student and skill modeling in several directions.

In modeling of cognitive skills, wrong answers may help to distinguish between absence of understanding and slips (careless errors, typos). Highly uncommon wrong answer is more likely to be a careless error, whereas common wrong answer is more likely to be a genuine mistake (unless caused by poorly designed user interface). Wrong answers may also be indicative of the level of knowledge and strategies that students are using. Consider for example a multiplication  $5 \times 5$ : a student A answers quickly 30, whereas a student B answers 24 after a long time. This may indicate that the student A retrieved the answer (incorrectly) from declarative memory, whereas the student B made an error in a procedural strategy. Wrong answers can thus be useful for modeling cognitive processes of learners [27]. Moreover, they may be useful also for modeling affect and motivation [29]. Irrelevant, highly uncommon wrong answers (particularly when repeated and quickly delivered) are probably indication of disengagement rather than lack of knowledge.

Wrong answer may be useful also for domain modeling. Common wrong answers may indicate relations between topics and thus may be used for automatic detection of knowledge components. Even through these may be misconceived relations, when they are common, they may be useful for student modeling. Relations between items based on wrong answers may also be taken into account in the design of the user interface or in the item selection algorithm. Wrong answers can also be used for student clustering – different groups of students make different types of mistakes and need different treatment from the educational system (e.g., students with dyslexia or dyscalculia).

### 2.2 Construction of Items and Hints

A basic observation about wrong answers, which seems to be valid in many different domains, is that the distribution of wrong answers is often highly skewed, i.e., some mistakes are much more common than others. This feature of wrong answers is potentially very useful for construction of questions and hints (both manual and automatic).

Common wrong answers may highlight student misconceptions and thus provide inspiration for new items (problems). In the case of items with simple structure, wrong answers may even be used automatically, e.g., as competitive distractors in multiple choice questions [16]. Previous work [1, 11] explored the possibility of using erroneous examples in education. Common wrong answers provide useful material for creation of such examples.

Wrong answers may also be useful for development of hints, feedback to students, and other scaffolding aids. If the hints are developed manually by experts, wrong answers provide good way to prioritize the expensive work of an expert. Due to the skewed distribution of wrong answers it may be possible to quickly provide answer-specific feedback to most answers even in open environments [9]. It is also possible to generate hints automatically based on actions of other students with the same wrong answer [23].

### 2.3 Feedback for Learners, Teachers, and Tool Developers

Analysis of wrong answers can also bring more pragmatic advantages. A useful feature of personalized educational systems is an overview of mistakes made by a learner or a class. Such an overview can serve for example as a base for a review session. Teachers may use such overview to quickly detect common problems of their students and thus focus on problematic parts in classroom time or in personal consultations.

For tool developers common wrong answers may be useful as an indicator of problems with a user interface. For example, in a prototype of one of the systems used in this study there was a common wrong answer “1” in cases where the answer should have been “10”. This turned out to be a user interface issue – the system was expecting a single click on a “10” button, whereas users were trying to click buttons “1” and “0”.

For these types of applications, basic analysis of wrong answers should be easily accessible in educational systems for both teachers and system developers. Since there can be a large number of mistakes, it is important to make the listing of mistakes easy to navigate. To achieve this goal, we need to understand common features of wrong answers.

## 3. ANALYSIS OF WRONG ANSWERS

After the general discussion of properties and possible applications of wrong answers, we turn to analysis of specific data.

### 3.1 Used Systems and Data

The used systems cover three different domains (geography, anatomy, basic arithmetic) and are used by very different learners, but they have been developed by the same research group and share the basic principles. All of them focus on adaptive practice of declarative knowledge or simple procedures. Systems estimate learners’ knowledge and based on these estimates they adaptively select questions of suitable difficulty. They use a target success rate (e.g., 75%) and adaptively selects questions in such a way that the learners’ achieved performance is close to this target.

The used questions are either multiple-choice questions or “open questions” – either a free text answer or selection of any item from a provided context (e.g., “select Rwanda on the map of all African states”). For the analysis we use only answers to open questions, since the used multiple-choice questions have adaptively chosen distractors and this feature makes analysis difficult (due to the presence of feedback loops [19]).

The first system is Outline Maps ([outlinemaps.org](http://outlinemaps.org)) for practice of geography facts (e.g., names and locations of countries, cities, mountains). Details of the behaviour of the system are described in [15, 16]. The used data set contains more than 10 million answers (with more than 1 million wrong answers) and is publicly available [17]. This data set is the largest of the three used data sets and it is at the core of the presented analysis. The application is currently used by hundreds of learners per day, majority of learners is from the Czech Republic since the interface was originally only in Czech. The geographical origin and language of students clearly influence interpretation of below presented results. However, our main point is not interpretation of particular results, but rather illustration of different insight that can be gained by the analysis of the data.

The second system is Practice Anatomy for practicing human anatomy ([practiceanatomy.com](http://practiceanatomy.com)). The main target audience of the system consists of junior medical students preparing for their anatomy exams. Currently, the system offers practice of more than 1800 items organized into 14 organ systems and 9 body parts. Learners can practice a selected organ system or a body part, or specify a more advanced practice filter as an intersection of a set of organ systems and a set of body parts. The system is available in Czech (with Latin terminology) and English. Most users are from the Czech republic. The analyzed data set contains over 380 000 answers.

The third system is MatMat ([matmat.cz](http://matmat.cz)) for practice of basic arithmetic; its functionality is similar to for example Math Garden [24]. The system contains examples divided into 5 high level concepts (counting, addition, subtraction, multiplication, division), each of these concepts contains around 50-700 items, over 2 000 items in total. The system behaviour and the used student modeling approach are described in [28]. The analyzed data set contains over 180 000 answers.

Student knowledge and mistakes in the used domains have been analyzed before, e.g., recall and mistakes in knowledge of US states [20] or knowledge of Europe by Turkish students [25]. These works focused on difficulty of recall of individual countries and on factors which influence this difficulty (e.g., borders), they did not analyze wrong answers. Moreover, we use a data set that is orders of magnitudes larger than those used in previous research on geography knowledge. The domain of basic arithmetic has been studied intensively before, even with the focus on mistakes. A well-known example is the repair theory [4] with case study for subtraction problems. Particularly multiplication has been studied in detail, e.g., description of effects influencing difficulty (size effect, five effect, tie effect), connectionist model of retrieval [27], classification of errors [5, 24]. Our contribution in this domain is mainly in aligning the results with analysis from different domains (learning declarative knowledge in geography and anatomy).

### 3.2 Common Wrong Answers

Generally the distribution of wrong answers is highly skewed, most wrong answers are comprised from just few items. Analysis of commonly confused countries shows that the most important factors are whether the countries have com-

mon border, if they have similar size (important factor particularly if they have a common border) and whether their name starts with the same first letter (important factor particularly if they do not have a common border). There are differences between the skewness of the distribution of wrong answers for individual items. For some countries there are few very typical mistakes – for Bulgaria more than 40% of wrong answers are Romania, for Finland the two most common wrong answers (Sweden and Norway) comprise nearly three quarters of wrong answers. Some countries, however, have much more even distribution of wrong answers, e.g., for Switzerland or Croatia the most common mistake comprises only 10% of wrong answers.

The context of questions is also important. In the used system countries can be practiced either in the context of a single continent or of the whole world. In most cases the mistakes on the world map are within the same continent (i.e., the wrong answers on the world map are very similar to wrong answers within the continent map). There is, however, nontrivial number of exceptions, for example: countries with similar names, e.g., Guinea, Guyana, and Papua New Guinea, which have confusingly similar names and are on three different continents; countries close to continent borders, e.g., Turkey is confused with European countries and Arab countries in Africa and Asia confused; islands are confused together, e.g., Madagascar is not confused with other African countries, but with other islands. These examples illustrate the importance of proper practice context for some items, e.g., it is not very useful to practice Madagascar on the map of Africa, Madagascar should be practiced mainly on the map of the whole world. Such results can have direct consequences for the design of the behaviour of educational systems.

The data from the MatMat application contain similar patterns – the distribution of wrong answers is skewed, but the skewness of the distribution differs among items. Some items have very typical wrong answer (e.g.,  $1 \times 1 = 2$ ,  $4 \times 9 = 32$ ), for other items wrong answers are more uniformly distributed (e.g.,  $6 \times 8$  with answers 42, 54, 56, 64, 78). Previous work [24] has analyzed classification of errors in basic arithmetic (particularly in multiplication), using categories like near miss ( $\pm 1$ ), typo, operation error, or operand related error. In agreement with previous research [13, 24], large part of wrong answers fit into one of these categories, and the dominant categories are as expected – for counting and addition the dominant error type is “near miss”, whereas for multiplication a common error is operand related, e.g.,  $4 \times 9 = 32$  (which is  $4 \times 8$ ). There are, however, interesting differences between items of the same type. For division the typical mistake is “near miss” ( $\pm 1$ ). For division by 1 and 10, however, the typical mistakes are rather answers 1 and 10; for items of the type  $N/N$  common wrong answers are  $N$  or 0. For small operands (e.g.,  $4/2$ ) operation errors (multiplication instead of division) sometimes occur, whereas this does not happen for larger operands (e.g.,  $54/6$ ).

### 3.3 Categories of Wrong Answers

To provide a more quantitative analysis and comparison across educational systems, we define a coarse classification of wrong answers and analyze properties of individual categories. We propose the following classification of wrong an-

swers into four categories (note that the defined categories can be seen as “degrees of wrongness” of an answer with a natural ordering). *TopWA* is the most common wrong answer for a given item. *CWA* is a common wrong answer other than the most common one (as a definition of “common” we require that the number of occurrences is more than 5% of all wrong answers for the given item, it must also be larger than 1). *Other* is any nonempty answer that is not common. *Missing* is an empty answer. Previous research [29] used 10% bound for definition of common wrong answers, but they did not treat the top wrong answer separately.

Figure 1 (top) shows distribution of answers among these classes. Although there are some differences between the used systems (respectively specific maps in the geography system), overall the distribution is quite balanced, i.e., the used definitions of classes provide reasonable partition of wrong answers. The rest of Figure 1 shows characteristics of student behaviour related to answers from individual categories. Since in this work we are interested mainly in relative comparison among types of answers (and not among systems), the results are normalized with respect to correct answers (for each system). The reported characteristics are computed globally. We have also analyzed more detailed results (e.g., for specific practice contexts like European countries or one digit multiplication), the results show similar trends.

The results show clear trends that are very similar across the three used systems. The median response time is larger for more wrong answers, with the exception of missing answers. The probability of leaving the system directly after an answer is much higher for wrong answers than for correct answers. Also within the wrong answers there is a clear trend (the probability of leaving increasing with wrongness). Finally, the last two graphs analyze future success of a student; the probability of success on the next question about the same item, the probability of success on the next question within the system (global). In both cases there the probability of future success decreases with wrongness of the current answer.

We see that there are systematic differences between different types of wrong answers. The general nature of these differences is rather intuitive, the main interesting aspects of these results are the similarity of results across three different domains and the consistently linear nature of these relationships, i.e., we can say that the distance between *TopWA* and *CWA* is the same as the distance between *CWA* and *Other*. The bottom line is that the wrongness of answers can be treated as an interval variable and it may be useful to utilize it as such for student modeling (for modeling both knowledge and affect).

### 3.4 Confusion Graph and Item Clustering

So far we have analyzed wrong answers for each item separately. But mistakes for individual items are clearly interconnected. We can analyze these interconnections with a “confusion graph” (a similar analysis has been done previously for the domain of statistics [12], but for much smaller data). In a confusion graph nodes are individual items, and edges correspond to wrong answers – we consider a weighted graph where a weight of an edge  $(u, v)$  is given by a frequency

of a particular wrong answer  $v$  among all wrong answers on an item  $u$ . This definition leads to a directed graph, to obtain an undirected graph we compute the weight of an undirected edge by averaging the weights of the corresponding directed edges.

Figure 2 shows the confusion graph for European countries. The confusion graph contains distinct clusters of items, this observation holds also for confusion graphs of other practice contexts in the used systems. To automatically detect these clusters we use a community detection algorithm [3]. The resulting clusters are meaningful and can provide useful insight for teachers and developers of educational system (Figure 2 for an illustration). The presented clustering was obtained by off-the-shelf implementation of the community detection algorithm [2] without any tuning. For a specific application of such clustering it may be useful to experiment with different community detection algorithms and specific definitions of the confusion graph.

### 3.5 Other Properties of Wrong Answers

Wrong answer may help us to (quickly) differentiate between different groups of users. For example in the geography domain we can see some important differences in wrong answers of students of different geographical origin, e.g., confusions between Slovakia and Slovenia, which is much more common mistake for US students than for Czech students, or wrong answers for Belarus (Bulgaria for US students, Ukraine for Czech students).

Wrong answers differ in their “persistence”, i.e., probability that the mistake will be repeated (by the same student) in future. For example, consider wrong answers for Ireland. United Kingdom is more probable mistake than Italy, but the second one is more likely to persists. Other similar examples are Moldova (answers Macedonia versus Kosovo) or Benin (answers Burundi versus Ghana). Some mistakes are very likely to be repeated, e.g., confusion between Zambia and Zimbabwe, Gambia and Senegal, or Guinea-Bissau and Burkina Faso.

## 4. CONCLUSIONS

Our analysis suggests that wrong answers are underused resource in online educational systems. They are easy to collect and can provide interesting insight applicable in many different ways (student modeling, automatic question and hint construction, feedback and inspiration for teachers and system developers). We provide a systematic overview of potential applications of wrong answers and many illustrative examples of interesting insights from educational applications.

We also propose specific novel approaches to analysis and utilization of wrong answers, particularly a classification of wrong answers into four categories (which can be treated as “degrees of wrongness”) and clustering of items using a confusion graph (based on wrong answers) and a community detection algorithm. The results of analysis from three different domains (geography, anatomy, basic arithmetic) show that properties of wrong answers are rather consistent and thus the developed approaches should be applicable also for other domains.

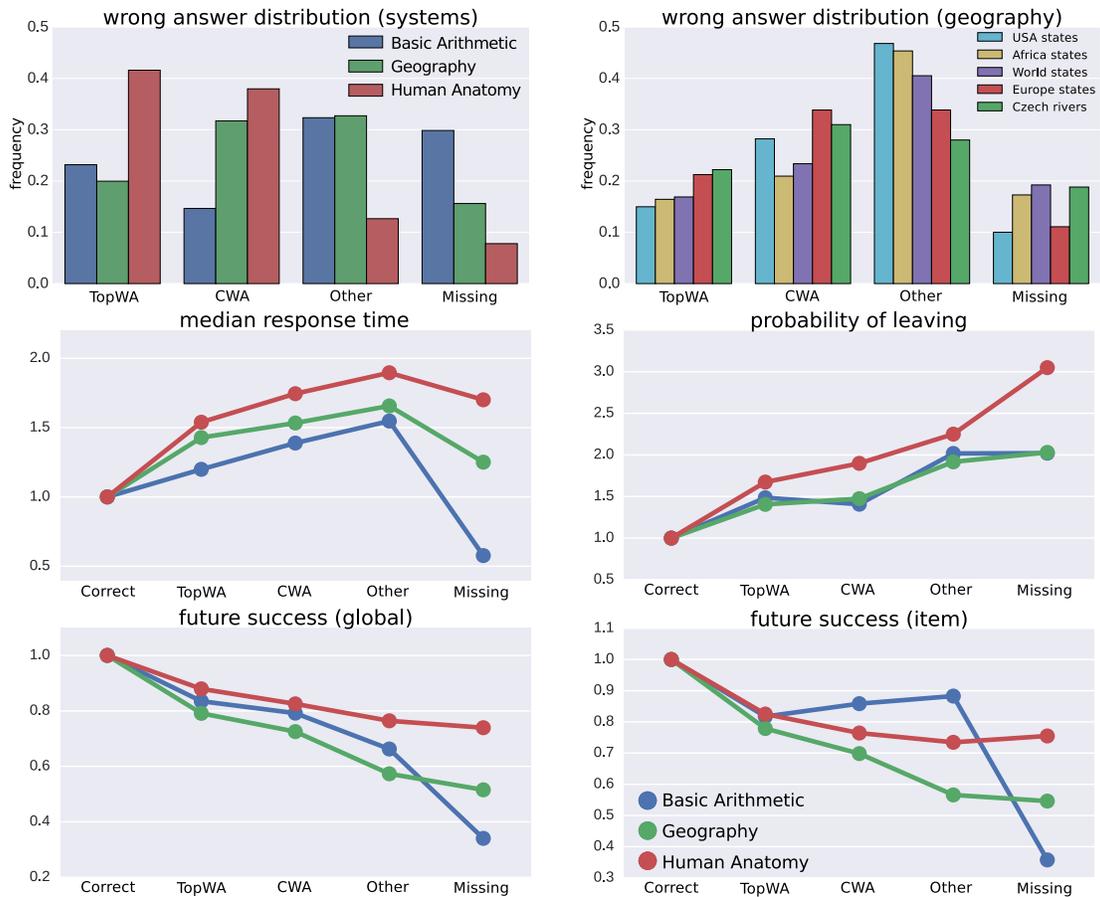


Figure 1: The first line shows frequency of different categories of wrong answers for different systems and for selected maps in geography system. The rest of the figure shows properties of different categories of answers normalized with respect to correct answers.

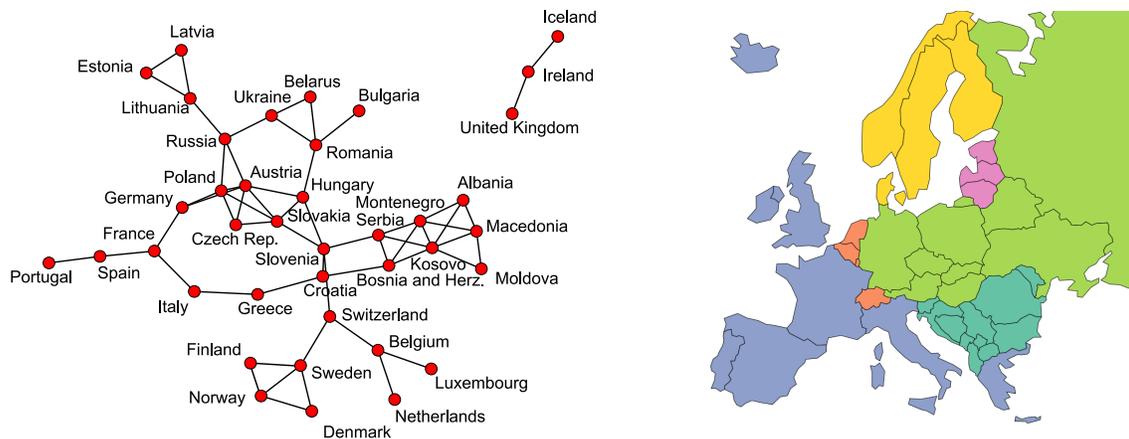


Figure 2: Left: A confusion graph for European countries (showing only the most significant edges). Right: Clustering of European countries based on community detection in the confusion graph.

## 5. REFERENCES

- [1] Deanne M Adams, Bruce M McLaren, Kelley Durkin, Richard E Mayer, Bethany Rittle-Johnson, Seiji Isotani, and Martin van Velsen. Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior*, 36:401–411, 2014.
- [2] Thomas Aynaud. Community detection for networkx, 2009.
- [3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [4] John Seely Brown and Kurt VanLehn. Repair theory: A generative theory of bugs in procedural skills. *Cognitive science*, 4(4):379–426, 1980.
- [5] Brian Butterworth, Noemi Marchesini, Luisa Girelli, and AJ Baroody. Basic multiplication combinations: Passive storage or dynamic reorganization? *The Development of Arithmetic Concepts and Skills: Constructive Adaptive Expertise*, pages 187–201, 2003.
- [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] Elena L Glassman, Jeremy Scott, Rishabh Singh, Philip J Guo, and Robert C Miller. Overcode: Visualizing variation in student solutions to programming problems at scale. *ACM Transactions on Computer-Human Interaction*, 22(2):7, 2015.
- [8] George Gogvadze, Sergey Sosnovsky, Seiji Isotani, and Bruce McLaren. Evaluating a bayesian student model of decimal misconceptions. In *Educational Data Mining 2011*, 2010.
- [9] Jonathan Huang, Chris Piech, Andy Nguyen, and Leonidas Guibas. Syntactic and functional variability of a million code submissions in a machine learning mooc. In *AIED 2013 Workshops Proceedings Volume*, page 25, 2013.
- [10] Barry Peddycord Iii, Andrew Hicks, and Tiffany Barnes. Generating hints for programming problems using intermediate output. In *Educational Data Mining*, 2014.
- [11] Seiji Isotani, Deanne Adams, Richard E Mayer, Kelley Durkin, Bethany Rittle-Johnson, and Bruce M McLaren. Can erroneous examples help middle-school students learn decimals? In *Towards Ubiquitous Learning*, pages 181–195. Springer, 2011.
- [12] Jaclyn K Maass and Philip I Pavlik Jr. How spacing and variable retrieval practice affect the learning of statistics concepts. In *Artificial Intelligence in Education*, volume 9112 of *LNCS*, pages 247–256. Springer, 2015.
- [13] Thomas S McTavish and Johann Ari Larusson. Labeling mathematical errors to reveal cognitive states. In *Open Learning and Teaching in Educational Communities*, pages 446–451. Springer, 2014.
- [14] Andy Nguyen, Christopher Piech, Jonathan Huang, and Leonidas Guibas. Codewebs: scalable homework search for massive open online programming courses. In *Inter. conf. on World Wide Web*, pages 491–502. ACM, 2014.
- [15] Jan Papoušek and Radek Pelánek. Impact of adaptive educational system behaviour on student motivation. In *Artificial Intelligence in Education*, volume 9112 of *LNCS*, pages 348–357, 2015.
- [16] 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.
- [17] Jan Papoušek, Radek Pelánek, and Vít Stanislav. Adaptive geography practice data set, 2015. <http://www.fi.muni.cz/adaptivelearning/>.
- [18] Radek Pelánek and Petr Jarušek. Student modeling based on problem solving times. *International Journal of Artificial Intelligence in Education*, 25(4):493–519, 2015.
- [19] Radek Pelánek, Jiří Řihák, and Jan Papoušek. Impact of data collection on interpretation and evaluation of student model. In *Learning Analytics & Knowledge*, pages 40–47. ACM, 2016.
- [20] James A Reffel. Cued vs. free recall in long-term memory of the fifty united states. *Current Psychology*, 16(3-4):308–315, 1997.
- [21] Kelly Rivers and Kenneth R Koedinger. Automatic generation of programming feedback: A data-driven approach. In *Workshop on AI-supported Education for Computer Science*, page 50, 2013.
- [22] Hans-Jürgen Schmidt. Students’ misconceptions—looking for a pattern. *Science education*, 81(2):123–135, 1997.
- [23] John Stamper, Tiffany Barnes, Lorrie Lehmann, and Marvin Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Intelligent Tutoring Systems Young Researchers Track*, pages 71–78, 2008.
- [24] Marthe Straatemeier. *Math Garden: A new educational and scientific instrument*. PhD thesis, Universiteit van Amsterdam, Faculty of Social and Behavioural Sciences, 2014.
- [25] Ilkay Sudas and Cemil Gokten. Cognitive maps of europe: geographical knowledge of turkish geography students. *European Journal of Geography*, 3(1):41–56, 2012.
- [26] Dina Tirosh. Enhancing prospective teachers’ knowledge of children’s conceptions: The case of division of fractions. *Journal for Research in Mathematics Education*, pages 5–25, 2000.
- [27] Tom Verguts and Wim Fias. Interacting neighbors: A connectionist model of retrieval in single-digit multiplication. *Memory & cognition*, 33(1):1–16, 2005.
- [28] Jiří Řihák. Use of time information in models behind adaptive system for building fluency in mathematics. In *Educational Data Mining, Doctoral Consortium*, 2015.
- [29] Yutao Wang, Neil T Heffernan, and Cristina Heffernan. Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Learning Analytics And Knowledge*, pages 31–35. ACM, 2015.

# Using Inverse Planning for Personalized Feedback

Anna N. Rafferty  
Department of Computer  
Science  
Carleton College,  
Northfield, MN 55057 USA  
arafferty@carleton.edu

Rachel A. Jansen  
Department of Psychology  
University of California,  
Berkeley, CA 94720 USA  
racheljansen@berkeley.edu

Thomas L. Griffiths  
Department of Psychology  
University of California,  
Berkeley, CA 94720 USA  
tom\_griffiths@berkeley.edu

## ABSTRACT

An increasing number of automated models can make inferences about learners' understanding based on their problem solving choices in interactive educational technologies. One potential use of these models is to personalize feedback interventions. We investigate using the output of an inverse planning model to choose feedback activities for learners. The inverse planning model uses the patterns of how a learner solves algebraic equations to estimate her proficiency on several discrete skills. The personalized feedback then focuses on the skill which is least proficient and includes a combination of existing educational content and scaffolded practice. We experimentally tested the effectiveness of personalizing the feedback based on the algorithm's estimate compared to simply providing a random feedback activity. The results show that completing the feedback was associated with performance improvements from pre- to post-test, but that personalized feedback was not associated with reliably more improvement. However, participants who received feedback about a skill that was far from mastery did show reliably more improvement than those who received feedback about an already-mastered skill. This suggests that there is potential in using the inverse planning algorithm to provide more effective learning experiences.

## 1. INTRODUCTION

Cognitive models of people's learning are often useful for better understanding behavior and can highlight what a particular learner knows and where she may be struggling. There are also an increasing number of educational resources available for learning specific topics, such as online videos, which might be effective for remediating a learner's struggles. However, there can be challenges when trying to close the loop between estimating a model of what someone knows and creating interventions based on that model to address misunderstandings or gaps in knowledge. The model is not a perfect assessment, and many interventions may be effective for a particular learner, making it difficult to determine if personalizing the intervention is valuable. While there are a

number of models that have been used to change the behavior of an educational technology, such as providing problems until mastery [2], there has been less of a focus on using models based on behaviors in more open-ended learning environments to guide feedback and remedial interventions in these settings.

We address the problem of closing the loop between a model-based assessment of a learner's algebra skills and the experience the learner has in a web-based algebra activity. The model was an inverse planning model for algebra, which provides an assessment of specific algebra skills based on the pattern of how someone solves equations. While the model provides a profile of what a person may misunderstand, suggesting that it could be used to guide feedback interventions, its estimates also have some error, meaning that it will not perfectly identify misunderstandings for every person. Additionally, the model's assessment is based on a collection of problem solutions, meaning the feedback must be targeted at an overall skill or misunderstanding rather than performance on a specific problem. This differs from many contexts where feedback is provided in interactive educational technologies, but has the potential to facilitate longer interventions about specific concepts or skills. This type of feedback could connect a learner with existing resources about particular concepts, since rather than assisting with a single question, the feedback is remediating a more abstract area of struggle. Thus, we explore how the model's assessment of understanding can be used to provide feedback to learners that targets their misunderstandings.

We investigate this question by designing feedback interventions for specific skills and experimentally testing how people's performance changes from pre- to post-test based on the intervention that they are given. The feedback interventions combined relevant content from existing sources and scaffolded opportunities for practicing a particular algebra skill. In an experiment, we compared performance for people who completed a feedback intervention based on the algorithm's estimate of their skills versus those who completed an intervention that was chosen randomly. We found that both groups showed significant performance improvements from pre- to post-test, but the two groups did not differ in their amount of improvement. However, completing feedback about a skill that one was less proficient in was reliably associated with more improvement than completing feedback about a skill that was near mastery. These results suggest that the algorithm's assessment may be used to di-

rectly improve the educational technology, although there are a number of subtleties in how to do this effectively.

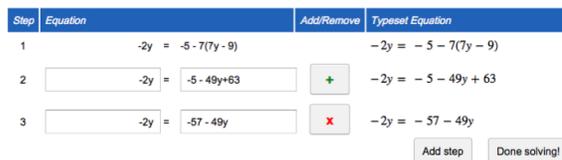
## 2. BACKGROUND

There has been a great deal of previous work related to assessing student understanding and providing interactive feedback to improve understanding. In our work, we are most interested in techniques where a student’s actions or choices are used as part of the assessment of understanding, such as in open-ended learning environments (OELEs). OELEs are often used in science education, as they can provide opportunities for students to generate and test their own hypotheses [6]. Educational data mining has been used to better understand what behaviors are associated with learning in some of these environments, such as Betty’s Brain [4], and these environments may provide feedback to students about their progress (e.g., [12]). Data mining is also used in these environments for assessments of skills, especially those like experimentation that are more difficult to measure in other environments [3]. However, it is rarer for the data mining to be used directly to inform feedback to students, and the feedback that is provided is frequently in the form of a short hint or suggestion about what to do. In mathematics education, there exist several systems, such as the Cognitive Tutor [2], that maintain a model of student learning and use this to adapt instruction, such as providing more problems on an unmastered skill; typically these systems assess student knowledge based on final answers rather than on what actions are taken to generate a solution. In both the science and mathematics systems the type of adaptive feedback differs from our focus on providing a somewhat longer session of feedback focused on re-teaching a particular skill.

While formative feedback to learners is an effective way to improve understanding and help create a more integrated base of knowledge [13], the problem of determining what type of feedback will be most effective is an area of active research. Much of the previous work on feedback in mathematics tutors has focused on progressively more informative hints (e.g., [5]). More holistic information based on assessments of skills may be provided to students, such as when making a learner model “open” to the learner [1], but this is not necessarily paired with feedback or interventions to remediate understanding. Research about teachers’ responses to student work in educational technologies has found that teachers may customize their instruction in a variety of ways to adjust to student misunderstandings [8]. Based on this work, we were interested in how more holistic feedback that focuses on a particular skill that a student is struggling with, rather than a specific problem, might affect learning.

## 3. INVERSE PLANNING

In order to get a holistic assessment of a learner’s algebra skills based on observing their behavior, we used a Bayesian inverse planning approach [11]. Bayesian inverse planning takes as input a set of step-by-step actions from a learner, and outputs a posterior distribution over possible levels of proficiency for various skills. This approach allows us to interpret people’s patterns of behaviors while they solve algebraic equations in a relatively freeform interface. In this interface, shown in Figure 1, learners have the ability to enter step-by-step solutions to equations, with no constraints on whether individual steps are correct before entering a new



**Figure 1: A screenshot of the step-by-step interface for solving algebraic equations. The user may solve the problem using any steps she chooses and record them in the interface.**

step. The Bayesian inverse planning algorithm uses both the mathematical correctness of each step and the way it moves the learner towards the solution to diagnose proficiency; the model is substantially similar to that described in [10]. We provide a brief overview of the algorithm and its underlying model of problem solving.

Bayesian inverse planning is based on a generative model: it models how likely a person would be to choose each possible solution step if she had a particular understanding of algebra, and then uses this model to infer what understanding is most likely to have resulted in the observed solutions. To create this generative model, we need to specify how choices about solution steps are made as well as specifying the representation of possible understandings. Inverse planning treats algebraic equation solving as a Markov decision process (MDP), in which people choose actions to bring them closer to the goal of solving an equation with as few steps as possible. With each action, the person moves from one (partially solved) equation to another. In an MDP, the value  $Q_h(s, a)$  of taking an action  $a$  given that the current equation is  $s$  can be approximated using dynamic programming. This long-term value is dependent on the person’s understanding of algebra, denoted as  $h$ , since that understanding may change what actions she believes are possible or what next equation she generates from the current equation. We model people as following a noisy optimal policy when choosing actions:  $p(a|s) \propto \exp(\beta \cdot Q_h(s, a))$ , where  $\beta$  controls the level of noise. Intuitively, this policy assumes people tend to choose actions that they think will help them solve the problem efficiently but they do not do so deterministically. The parameter  $\beta$  is estimated for each individual, as described below.

In this model, understanding is represented by the level of proficiency for several skills. For each skill, the proficiency indicates whether the person generally applies the skill correctly or if she makes a particular type of error. The different levels of proficiencies form a *hypothesis space* of possible algebra understandings. The hypothesis space was based on past education and psychology research and consists of parameters for six skills (see [10] for details): moving terms, dividing by the coefficient of a term, applying the distributive property, combining terms, arithmetic, and planning. The first four parameters relate to specific rules of manipulating algebraic equations, while the latter two apply more broadly.

Each of the four algebra-specific parameters indicates whether the person is prone to a particular type of error or “mal-

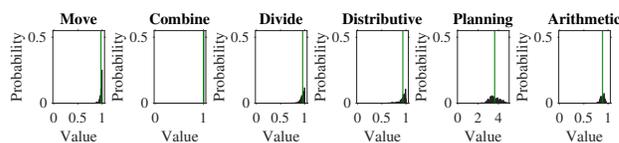
rule” [9]. For moving terms, the mal-rule is failing to flip the sign of a term when moving it from one side of the equation to another; the inferred parameter is the probability of not following this mal-rule when moving a term. For dividing by the coefficient of a term, the mal-rule is multiplying rather than dividing (i.e., not using the reciprocal), and for applying the distributive property, the mal-rule is only distributing the coefficient to the first term rather than all terms. Both of these parameters, like moving terms, are probabilities. For combining terms, the mal-rule is combining unlike terms, such as a variable and a constant. This parameter is binary: the person either considers combining unlike terms when choosing actions or she does not.

The final two parameters for the hypothesis space are the arithmetic parameter and the planning parameter. The arithmetic parameter is the probability that a person accurately computes a calculation. The planning parameter is the parameter  $\beta$  in the noisily optimal policy: higher values for this parameter indicate very high probability of choosing the most efficient action for moving towards a solution, while values close to zero indicate very different choices from those expected by the model, such as choosing an action that does not make progress towards the solution or giving up prior to reaching a solution. This parameter is the only parameter not targeted for feedback, as a mixture of cognitive and motivational feedback might be most effective for improving planning and lessening the rate of non-answers.

The parameters above form a six-dimensional, continuous hypothesis space  $\mathcal{H}$ , where each point in the space represents one possible set of skill proficiencies  $h$ . Given this hypothesis space, the posterior distribution after observing the person’s problem solutions  $D$  is calculated using Bayes’ rule: for each  $h \in \mathcal{H}$ ,  $p(h|d) \propto p(h) \prod_{d \in D} p(d|h)$ , where  $p(h)$  is the prior distribution over the hypothesis space and  $p(d|h)$  is the likelihood that the person would produce the observed step-by-step solution if she had the skill levels indicated in  $h$ . The prior favors higher levels of proficiency; intuitively, this means that the algorithm favors the part of the hypothesis space indicating normative algebra understanding unless it observes evidence in the solutions that non-normative steps are being taken. Because the hypothesis space is continuous, the posterior distribution cannot be calculated exactly. Instead, Markov chain Monte Carlo (MCMC) methods are used to compute an approximate posterior distribution. As shown in Figure 2, the resulting posterior distribution indicates both the most likely proficiency for each skill as well as the algorithm’s confidence. In the figure, both the parameter for moving terms and the distributive property are close to one, but the estimate for moving terms is more certain; there is also a lower estimated proficiency for arithmetic than for the other skills. In order to use the posterior distribution for feedback, we calculate the mean value of the posterior on each skill dimension (shown as green lines in Figure 2).

#### 4. FEEDBACK DESIGN

Given the output of the inverse planning algorithm, our goal was to “close the loop” by providing learners with a feedback activity that could help to remediate their understanding of a particular skill. In an attempt to minimize differences in feedback effectiveness due to quality rather than topic, all of the feedback interventions followed the same pattern. First,



**Figure 2: The inverse planning algorithm’s assessment for a learner from the experiment. Each plot shows the posterior distribution for one skill. Larger values are closer to mastery.**

an overview screen showed the learner two skills: the skill closest to mastery and the skill she would receive feedback about. In both cases, she was shown her proficiency level as a colored bar and a short description of the skill was provided. The bottom of the page told her that she would be learning more about the second skill that was shown; we refer to this skill as the *feedback skill*. On the next page, learners were shown a 2–5 minute embedded video about the feedback skill. Since there already exist a large number of freely available educational videos, we aimed to connect learners to a relevant resource rather than create new tutorial content. All videos were sourced from Khan Academy<sup>1</sup>, and were chosen because they targeted one of the five skills.

After the video, several stages of scaffolded practice were provided. For the four skills related to algebraic rules, the scaffolded practice began with four problems to highlight the core skill being practiced. For example, only the feedback focused on correctly applying the distributive property included practice on the distributive property. For these problems, the learner’s steps were checked for correctness with each new step. If a mathematical error was detected, the step was highlighted and she was asked to fix it before continuing. After each problem, the learner was told the correct answer. Following these problems, eight problems were provided that still focused on the feedback skill, but checking of correctness was only provided after the learner submitted her answer. At that point, steps with errors were highlighted and the learner was given the opportunity to review them before continuing. These problems thus targeted the feedback skill, but included slightly less immediate assistance than the first set of problems. For the feedback targeting arithmetic, all twelve practice problems were arithmetic computations to complete rather than algebraic equations. Finally, all feedback finished with twelve algebra problems that were not specialized based on the feedback skill, with the intention for people to practice in context what they had learned from the skill-specific problems. The interface for these problems was the same as when doing general problem solving on the website: people had the opportunity to enter individual problem steps, and they were told whether they were correct before moving to the next problem.

#### 5. EXPERIMENT

When we designed the feedback, our goal was to personalize what feedback someone was given based on the algorithm’s assessment of their skills by assigning the person to complete feedback on their least proficient skill. While it is intuitively plausible that personalized feedback based on

<sup>1</sup><http://www.khanacademy.org/>

this assessment might be more helpful than non-personalized feedback, there are several reasons to be skeptical. First, the algorithm's diagnosis is an approximation: there is error both in the MCMC estimate, and in the model itself. In general, the algorithm can interpret most problem solutions [10], but some people's behavior may be poorly fit by the model, resulting in poor accuracy for an individual. Additionally, the algorithm does not account for learning within the period that the skills are being assessed and depending on the person's behavior, there may be some skills about which we have very limited information. For example, a person might solve only a few problems using the distributive property, giving a relatively large confidence interval for possible skill proficiencies. A second concern about personalizing feedback is that learners who are struggling may be struggling in many skills. In that case, it may be that the personalization is unnecessary: most students who benefit from one feedback activity would also benefit from any of the other feedback activities. Thus, we ran an experiment to test whether the feedback activities were associated with learning and to examine whether personalized feedback produced larger learning gains than feedback that was not personalized based on the algorithm's assessment.

## 5.1 Methods

**Participants.** 200 participants in the USA were recruited from Amazon's Mechanical Turk (AMT) and compensated \$4 for session 1, \$6 for session 2, and \$8 for session 3. Participants had taken an algebra course and had not completed college math classes beyond algebra.

**Stimuli.** Participants completed a multiple-choice assessment, solved algebra problems on a website, and responded to several surveys. The twelve question multiple-choice assessment was based on College Board ACCUPLACER<sup>®</sup> tests used for math placement in many postsecondary institutions[7]. The questions were substantially similar to the Elementary Algebra questions used in [11], but the numbers were changed to create two versions of the assessment.

All problem solving on the website used a similar interface to that shown in Figure 1. In sessions 1 and 3, learners were told whether or not they were correct immediately after submitting a problem. During the feedback in session 2, the interface behaved as described in the previous section.

In the surveys, participants indicated their demographics as well as prior math class experience. They also completed 18 questions focused on the usability of the website and the perceived helpfulness of the feedback.

**Procedure.** Participants completed three sessions, separated by at least one day. In the first session, all participants solved 24 equations on the algebra website, followed by the multiple-choice questions about elementary algebra topics. The website included a short tutorial about how to use the interface, and the 24 problems were generated based on templates. For example, one template was a constant plus a variable equal to a constant. The constants and coefficients for variables were generated randomly, but all participants shared the same templates. After a participant completed all problems on the website, the diagnosis for that participant was computed automatically by the inverse plan-

ning algorithm, and results were stored in the database for the participant's next session. Participants were randomly assigned to receive version one of the multiple-choice questions or version two; these versions were identical except for changes to the exact numbers used in the problems.

In the second session, participants completed one of the feedback activities. They were randomly assigned to either *targeted* or *random* feedback. Those receiving targeted feedback completed the feedback activity for the skill which the algorithm estimated they had least proficiency; those receiving random feedback completed one of the five feedback activities selected uniformly at random.

In the third session, participants again solved 24 equations on the algebra website, followed by the multiple-choice questions about elementary algebra topics. Just as in the first session, participants all completed problems on the website that used the same templates. For the multiple-choice questions, each participant completed the version of the questions that they did not complete in the first session. Finally, participants ended the third session by completing the demographics and usability surveys.

## 5.2 Results

82% of participants completed all three parts of the experiment in a single session. Several participants were removed due to technical problems, such as needing to restart the computer during a session and thus losing their place in the activity. The results that follow include only the 164 participants who completed all parts of the experiment.

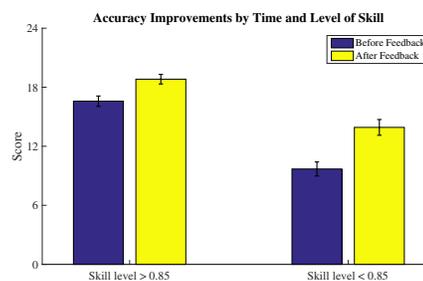
Responses to our demographics questions suggest that participants came in with varying levels of mathematics background and that for most, significant time had passed since they had last studied algebra in school. 98% of participants reported what previous math classes they had taken, in college or in high school. 62% of those who responded had taken no math classes beyond geometry (typically at a high school level); the remaining participants had taken trigonometry, pre-calculus, or calculus at a high school level. A number of participants who reported taking one of these higher-level courses in high school also reported taking a college algebra class. Thus, we would expect all participants to have prior experience with solving equations, but to be likely to have some gaps in their knowledge.

We first examined changes in participants' performance between the first session, before getting feedback, and the final session, after getting feedback. Results from the first session confirmed that participants were on average far from ceiling on the task: they correctly answered an average of 7.2 multiple-choice questions out of a total of 12, and correctly answered an average of 12.4 out of the 24 algebra problems on the website. There was a small increase in the number of multiple-choice questions answered correctly in the final session. Using a repeated-measures ANOVA with factors for time of test, condition, and a random factor for participant, we found that this main effect was reliable ( $F(162, 1) = 15.7, p < .001$ ), but there was no interaction between condition (targeted versus random feedback) and time of test. Given that many of the questions focused on skills that were not directly targeted by our intervention, in-

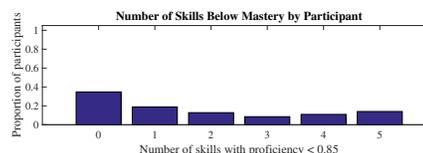
cluding some quadratic equations and linear inequalities, it is not surprising that we see only a small improvement from the first to the final session. The increase in performance was somewhat larger for the algebra equations solved on the website: participants correctly answered 23% more problems correctly, for a mean of 16.6 problems correct in the final session. We again used a repeated-measures ANOVA with factors for time of test, condition, and a random factor for participant to analyze the reliability of this finding, and found that there was a main effect for time of test ( $F(162, 1) = 89.9, p < .001$ ), but no interaction between time of test and condition.

To better understand why there was no interaction between condition and the amount of improvement, we examined the estimated proficiency level of the skills for which feedback was given. On average, the targeted condition selected skills that had lower levels of proficiency (average proficiency level of 0.56 versus 0.88;  $t(162) = 7.03, p < .001$ ), indicating that in many cases, there were large differences between the least mastered skill and a random skill. However, there were a number of participants in the random condition who received feedback about a skill with which they were struggling as well as participants in the targeted condition who did not have any skills that were far from mastered. To test whether participants who received feedback that was more appropriate for them improved more than participants who received feedback that was less appropriate for them, we divided all participants into two categories: those who received feedback about a skill that was estimated to be less than a proficiency level of 0.85 (an *unmastered* skill) and those who received feedback about a skill that was at a proficiency level greater than or equal to 0.85 (a *mastered* skill). This criterion categorizes 46% of participants as receiving feedback about an unmastered skill. As shown in Figure 3, participants who received feedback about an unmastered skill improved more than those who received feedback about a mastered skill. A repeated-measures ANOVA with factors for whether the feedback skill was already mastered, time of test, and a random factor for the participant showed that there was a main effect of time of test as well as an interaction between time of test and whether the feedback skill was already mastered ( $F(162, 1) = 9.42, p < .01$ ). To ensure that this result was not simply due to the cutoff level we chose for mastery, we also examined a categorization based on mastery level 0.9, and found the same trends ( $F(162, 1) = 46, p < .05$ ). While these results must be interpreted with some caution, as participants were not randomly assigned to the two categories, they suggest that receiving feedback that the algorithm indicates is more appropriate can result in greater improvements in performance.

Based on the fact that proficiency level influenced the effectiveness of the feedback, we examined the distribution of proficiencies for individual participants. We were interested in whether participants tended to have all skills at a similar level or whether they usually had some skills that were mastered and some that were unmastered. As shown in Figure 4, 35% of participants were at mastery for all skills, where mastery is defined as proficiency of at least 0.85, and 14% of participants were not at mastery for any skills. The remaining 51% of participants who had some mastered skills and some unmastered skills are arguably those that might most bene-



**Figure 3: Improvement from first to last session in accuracy on website problems, categorizing participants based on prior level of proficiency in feedback skill. Participants who received feedback about an unmastered skill improved more from the first to the final session than those who received feedback about a mastered skill.**



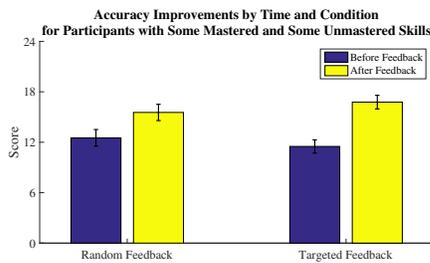
**Figure 4: Count of the number of unmastered skills by participant.**

fit from targeted rather than random feedback. A repeated-measures ANOVA with factors for time of test, condition, and a random factor for participant shows that there is a significant interaction between time of test and condition when restricting the data to these participants: as shown in Figure 5, those who completed targeted feedback improved almost twice as much those who completed random feedback (average improvement 5.3 versus 3.0;  $F(82, 1) = 5.64, p < .05$ ).<sup>2</sup> This suggests that inverse planning can provide a benefit for these participants: it allows us to determine what skill(s) will be appropriate targets for feedback.

## 6. DISCUSSION

Our goal in the feedback design and the experiment was to evaluate the benefit of connecting the holistic assessment and the feedback activities. While many of the feedback problems provided practice on multiple skills, since multiple skills are required to solve the algebraic equations, there was specialization in our feedback based on the algorithm's assessment. Our results show that overall, participants' performance improved after completing the feedback activities. The effects of personalization on the size of this effect were mixed: across all participants, feedback targeted at someone's weakest skill was not associated with reliably more improvement than feedback about a random skill, but restricted to those who had some mastered and some unmastered skills, we observed more improvement for those receiving the targeted feedback compared to those receiving the random feedback. This suggests that there is promise in using the inverse planning algorithm's assessment to connect

<sup>2</sup>With mastery level set at 0.9, this effect is marginally significant (average improvement 4.2 versus 2.7;  $F(103, 1) = 3.33, p = .07$ ).



**Figure 5: Improvement from first to last session in accuracy on website problems, restricted to participants with some unmastered and some mastered skills. Participants show reliable improvement, and participants who received targeted feedback tended to improve more than those who received random feedback.**

learners to relevant resources and personalize feedback activities, although further investigation is needed to determine ways to make this personalization even more effective.

There are several limitations of this work. First, our population of AMT workers may not be typical of algebra learners. These people were paid to participate in the study, and may differ in motivation and background from those who would use the website by choice. However, their varied backgrounds may be typical of adult learners who are trying to surmount barriers such as algebra at the community college level, a group we are particularly interested in reaching. Second, this experiment does not separate whether the content of a feedback intervention is helpful from whether the targeting of that feedback is accurate. We intend to further evaluate these two components to better understand what the maximum benefit of this type of feedback would be if targeting was perfectly accurate, but any evaluation of the overall effectiveness of the knowledge diagnosis-feedback loop must acknowledge that inaccuracies in the diagnosis may lead to the personalization being less effective.

In future work, there are a number of ways we will explore how to design more effective personalized feedback and investigate variations in how to use the algorithm for personalization. Our intervention was relatively short, with most participants taking about an hour for the session in which feedback was provided. One might expect the effects of personalization to be cumulative, with targeted feedback being most helpful when learning over a longer period; in that case, the targeting could be used to remediate the same skill multiple times if struggles were still evident or to recognize that say, one session of feedback had resulted in several skills reaching mastery and skipping the already mastered skills. Such longer interventions are likely to have larger effects, and may highlight whether targeted feedback is overall more effective or whether there is a subset of participants for which targeting makes a difference. Another area to explore is providing the profile generated by the inverse planning algorithm to the learner and using this in conjunction with targeted feedback, random feedback, or feedback chosen by the learner. The current system provides learners with the algorithm’s assessment of several of their skills, but it does not allow them to make choices about what feedback they re-

ceive. Choice might be useful for those not well-modeled by the algorithm or in cases where several non-mastered skills have been identified; however, it is also possible that struggling learners are unable to understand the possible types of feedback in order to make a good choice. Finally, there are several ways we might adjust how the algorithm’s output is linked to feedback. The diagnosis includes information about the algorithm’s certainty. This might be used to focus on skills that we are confident are unmastered. Additionally, the algorithm outputs a diagnosis of planning efficiency, but this was not used for feedback. Low levels of this parameter can be indicative of someone who frequently gives up or who is not well fit by the model. In either situation, it may not be appropriate to simply give feedback about the least proficient skill. Overall, the results in this paper serve as first steps for a larger investigation into how to effectively close the loop between holistic assessments of misunderstandings and guiding personalized feedback interventions for learners.

**Acknowledgements.** This work was funded by NSF grant number DRL-1420732 to Thomas L. Griffiths. Thanks go to Jonathan Brodie and Sam Vinitzky for programming parts of the feedback.

## 7. REFERENCES

- [1] S. Bull and J. Kay. Student models that invite the learner in: The SMILI open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120, 2007.
- [2] A. T. Corbett, K. R. Koedinger, and W. Hadley. *Cognitive tutors: From the research classroom to all classrooms*, pages 235–263. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2001.
- [3] J. D. Gobert, M. Sao Pedro, J. Raziuddin, and R. S. Baker. From log files to assessment metrics: Measuring students’ science inquiry skills using educational data mining. *Journal of the Learning Sciences*, 22(4):521–563, 2013.
- [4] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students’ learning behavior patterns. *JEDM-Journal of Educational Data Mining*, 5(1):190–219, 2013.
- [5] K. R. Koedinger and V. Alevan. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.
- [6] S. M. Land. Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3):61–78, 2000.
- [7] K. D. Mattern and S. Packman. Predictive validity of accuplacer scores for course placement: A meta-analysis. Technical report, College Board, December 2000.
- [8] C. F. Matuk, M. C. Linn, and B.-S. Eylon. Technology to support teachers using evidence from student work to customize technology-enhanced inquiry units. *Instructional Science*, 43(2):229–257, 2015.
- [9] S. Payne and H. Squibb. Algebra mal-rules and cognitive accounts of error. *Cognitive Science*, 14(3):445–481, 1990.
- [10] A. N. Rafferty and T. L. Griffiths. Interpreting freeform equation solving. In *Artificial Intelligence in Education*, pages 387–397. Springer International Publishing, 2015.
- [11] A. N. Rafferty, M. M. LaMar, and T. L. Griffiths. Inferring learners’ knowledge from their actions. *Cognitive Science*, 39(3):584–618, 2015.
- [12] J. R. Segedy, J. S. Kinnebrew, and G. Biswas. The effect of contextualized conversational feedback in a complex open-ended learning environment. *Educational Technology Research and Development*, 61(1):71–89, 2013.
- [13] V. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008.

# Pattern mining uncovers social prompts of conceptual learning with physical and virtual 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](mailto:marau@wisc.edu)

## ABSTRACT

To succeed in STEM, students need to connect visual representations to domain-relevant concepts, which is a difficult task for them. Prior research shows that physical representations (that students manipulate with their hands) and virtual representations (that they manipulate on a computer) have complementary advantages for conceptual learning. Further, physical and virtual representations are often embedded into different social classroom practices. Thus, to optimally combine these representation modes, we need to understand what social events prompt students to connect representations to concepts, and if different representation modes afford different social prompts. A multiple-case study with 12 high-school students addresses this question. Student pairs worked with physical and virtual representations of chemistry. Frequent patterns obtained from discourse data show that students incrementally co-construct concept-representation connections, and that instructor prompts are key triggers of these connections for both representation modes. Meta-cognitive statements serve as important prompts in the absence of an instructor when students work with virtual representations. I discuss implications for interventions that combine physical and virtual representations.

## Keywords

Physical and virtual representations, educational technology, collaboration, conceptual and social learning processes, STEM.

## 1. INTRODUCTION

Novice students in science, technology, engineering, and math (STEM) domains grapple with a *representation dilemma* [1]: they have to use visual representations they have never seen before to make sense of concepts they have not yet learned. Educators often take for granted that students can see meaningful concepts in representations [2]. However, much evidence shows that students struggle in connecting concepts to visual representations [3]. Their failure to make such concept-representation connections can impede their learning [4]. For example, in chemistry, difficulties in making concept-representation connections affect students' understanding of key concepts related to atomic structure and

bonding [5]. This issue applies to most STEM domains: because many key concepts cannot be directly observed, STEM domains heavily rely on visual representations [3]. Thus, STEM instruction typically provides conceptual prompts to help students make concept-representation connections [6].

Research in many STEM domains—including chemistry—shows that different *representation modes* provide different types of prompts for concept-representation connections [7]. *Physical representations* are tangible objects that students manipulate with their hands (Figure 1, top). In physical representations, haptic sensory input, experiences of movement, and continuous changes serve as prompts by making concepts intuitively accessible [7, 8]. By contrast, *virtual representations* are digital visualizations that students manipulate via mouse or text input (Figure 1, bottom). In virtual representations, visualizations and manipulations of invisible processes and immediate feedback can serve as prompts for concept-representation connections [7]. Thus, physical and virtual representations serve complementary roles in prompting for students to make concept-representation-connections [7, 9].

Besides providing different types of conceptual prompts for concept-representation connections, physical and virtual representations may provide different types of *social prompts*. Social prompts are discourse events that elicit collaborative co-construction of such connections [10]. Such events can emerge from student-student or student-instructor interactions. Because *physical representations* are typically used in collaborative contexts, interactions among students and instructors may prompt concept-representation connections [11]. By contrast, *virtual representations* are embedded in educational technologies that provide help in making concept-representation connections. In this context, students may work individually or collaboratively, typically with less help from an instructor [12]. Hence, interactions with instructors may be less important in prompting concept-representation connections. Thus, because physical and virtual representations are embedded in different social classroom practices, they may yield different social prompts for concept-representation connections.

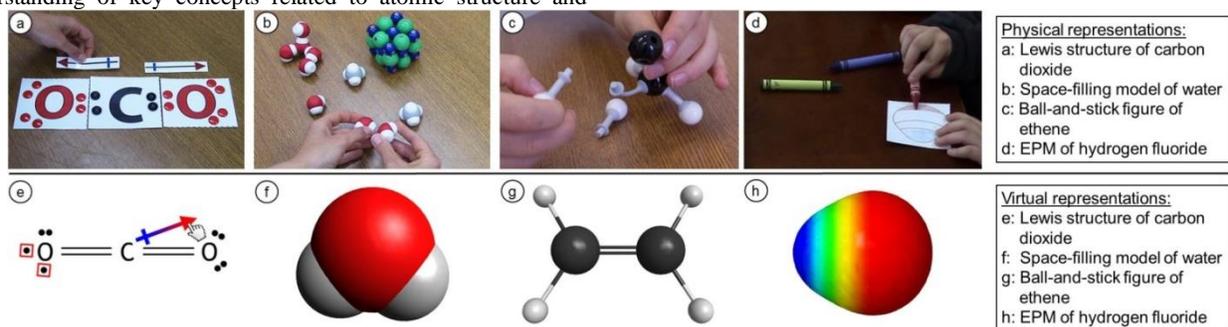


Figure 1. Physical representations (top) and virtual representations (bottom) of chemical molecules

Considering what social events serve as prompts for concept-representation connections is important for the design of instructional interventions that combine physical and virtual representations. Prior research has not investigated whether different representation modes afford different types of social prompts for concept-representation connections. At a theoretical level, addressing this question will help us understand the mechanisms by which representation modes affect students' ability to make concept-representation connections. It will also help us understand why one representation mode may be more effective than another for a given concept. At a practical level, it will allow us to design instructional activities that take advantage of the social prompts that the different representation modes afford.

The goal of this paper is to take a first step towards identifying social prompts of concept-representation connections for physical and virtual representation modes. To this end, I used a multiple-case study approach; specifically, I observed and recorded collaborative discourse among six student pairs over an extended learning period. Case-study approaches are particularly appropriate for investigating how social processes unfold over a longer learning intervention within the given social classroom practices [13]. The study compared two instructional contexts: (1) student pairs working with physical representations while receiving support from an instructor and (2) student pairs working with virtual representations embedded in an educational technology.

To identify social prompts of concept-representation connections, I applied frequent pattern mining to discourse data. This analysis identified social prompts that are successful for both representation modes and social prompts that were specific to a particular representation mode. I discuss implications for blending interventions that combine physical and virtual representations.

## 2. METHODS

### 2.1 Multiple-Case Study

Participants were 12 students from a small charter high school in the Midwestern U.S. The study was conducted as part of a chemistry workshop. Students had very limited prior knowledge about the concepts and the visual representations. The study took place as part of an in-school workshop on 3 days spread across 4 weeks. Each study day was 3h long. Prior to day 1, the teacher gave an introduction on chemical bonding. On day 1, students received an introduction into collaborative strategies and then worked on the chemistry workshop materials for the remaining study days.

All students were randomly assigned to pairs for the duration of the study. For each study day, the pairs were randomly assigned to a sequence of representation mode (i.e., physical-then-virtual, virtual-then-physical). For example, a pair might be assigned to the physical-then-virtual order for day 1. This pair would work with physical representations for the first half of day 1 and then switch to virtual representations for the second half of day 1. On day 2, the pair was randomly assigned to a new sequence.

The workshop covered basic concepts related to the polarity of chemical bonds. Students were presented with the visual representations shown in Figure 1: Lewis structures, ball-and-stick models, space-filling models, and electrostatic potential maps. Each was presented in the physical and virtual mode. When working with *physical representations*, students received a worksheet that asked them to construct a physical representation of a molecule, answer questions about the target concepts (e.g., about electronegativity) and about how the representation depicts these concepts. Each student pair was teamed up with an instructor—a research assistant who was trained on facilitating student collaboration and on

the chemistry concepts covered. Instructors provided feedback and assistance as students solved the problems.

*Virtual representations* were integrated in an educational technology for chemistry: Chem Tutor [14]; a type of intelligent tutoring system designed specifically to help students make concept-representation connections. To this end, Chem Tutor provides interactive virtual representations that students manipulate to solve problems about bonding. Chem Tutor prompts students to reflect on how each visual representation depicts particular concepts. Chem Tutor provides error-specific feedback and hints on demand. Chem Tutor was shown to significantly enhance learning of chemistry knowledge and conceptual understanding of representations [14]. While working with Chem Tutor, students could request help from an instructor who circulated the classroom.

### 2.2 Analysis

The goal of the analysis was to identify social events that prompt students' concept-representation connections and to investigate whether these prompts differ between representation modes.

The first step in the analysis was to code discourse data. All interactions among students and instructors were video-taped and transcribed. To develop a coding scheme, we used a grounded, bottom-up approach: we summarized discourse utterance-by-utterance to discover emerging themes. Next, we formalized these themes as codes, and then applied the codes to the discourse data. The coding scheme comprises 45 codes (see Table 1 for examples). Inter-rater reliability was substantial with kappa = .77.

The second step in the analysis was to identify discourse segments in which students succeed in making a concept-representation connection, defined as establishing the relation between a visual feature in a representation and the domain-relevant concept it illustrates [6]. Hence, a concept-representation connection was operationalized as an utterance made by a student that correctly refers to a concept and a representation (e.g., Table 2, #5).

The third step in the analysis was to operationalize social events that may prompt students to make concept-representation connections. In principle, any aspect of student-student or instructor-student discourse could serve as a social prompt: mentioning a concept, encouragement, evaluating, a meta-cognitive statement, a mistake, etc. Hence, I considered any code as a potential prompt.

The fourth step in the analysis was to specify the unit of analysis. Because I was interested in *social* events as prompts, I defined two consecutive discourse turns as the unit of analysis (i.e., utterances by two different speakers). I segmented the discourse data in the following way. First, I identified turns with concept-representation connections (e.g., Table 2, row 5). Second, I identified the two prior turns and considered them as a case (e.g., rows 3-4 in Table 2). This case was labeled as 'connection present' (i.e., a concept-representation connection occurs in the next turn). Third, I segmented the remaining discourse data such that two consecutive turns serve as a case (e.g., rows 1-2 in Table 2), labeled as 'connection absent' (i.e., no concept-representation connection in the next turn). Thus, each case was composed of two consecutive turns, labeled as connection-present/absent, annotated with codes, speaker (student or instructor) and mode (physical or virtual). Table 3 shows an overview of the dataset.

The final step in the analysis was to search for social events that trigger concept-representation connections. Given the focus on social mechanisms, I was interested in discovering which codes co-occur in collaborative discourse. To this end, I used frequent pattern mining to identify undirected patterns that describe which

**Table 1. Subset of codes in the coding scheme with examples from the transcripts.**

Code	Definition	Example
Concept	Utterances that relate something to a scientific concept	“They want to be able to make a complete number, a complete number of the eight on the outside”
Concept-request	Suggesting / prompting utterances that relate something to a concept	“What’s the rule for the bonding?”
Representation	Utterances that relate something to the representation; utterances that explain information shown by a representation	[pointing at a representation] “So, one, two, three, four, five. He have five.”; [pointing at a representation] “So, wait, that’s carbon?”
Representation-request	Suggesting / prompting utterances that relate something to the representation; utterances that explain information shown by a representation	“By looking at the Lewis structure, can you answer the question about electronegativity?”; “What are these things [points at dots in Lewis structure]?”
Assent	Expression of approval or agreement	“yeah”; “ok”; “I know.”; “Mmhhh.”
Meta-confusion	Utterances about oneself that describe confusion about how to proceed or about a concept, or about not knowing a concept	I don’t know.”; “this is very confusing.” “Maybe.”; “This is hard.”; “So, now we’re stuck.”; “I don’t get it why it’s lines.”
Meta-understanding	Utterances about oneself that describe a novel insights or understanding of how to proceed or of a concept	“Got it “; “Well, I know that part”; “I like this explanation.”; “then I was like, well, duh”; “We’ve been making this so much harder than it is!”
Reading	Reading the problems statement or instructions or hints / feedback from Chem Tutor	“well it says right here that, “Choose the letters that show each atom,”
Explanation	Utterances that explain / elaborate a concept	“But when they say dinitrogen, means they bonded.”; “I’ll give a little bit more help.”; “So, carbon has more electrons than hydrogen.”
Explanation-request	Suggesting / prompting utterances that explain / elaborate a concept	“So what do you think that that is?”; “Could you try, try to put as a complete sentence”; “But why?”; “How did you know?”
Metaphor	Utterances that use a metaphor, intuitive example, embellished language to describe an abstract concept	“To make it lock on kind of.”; “can I borrow your electrons”; “It’s the same pulling forces.”; “So, like magnetic, plus and minus.”;

**Table 2. Excerpt transcript showing 4 turns before a concept-representation connection (turn #5), with codes assigned to each turn. All student names are fake.**

#	Speaker	Utterance	Codes
1	Brigid	Electronegativity are the same so makes it covalent which is no difference.	Concept
2	Adriana	[reads] Does the Lewis structure show the polarity? Why or why not? Um. I’d say- I feel like no, be- Well, yeah. I don’t know.	Reading; meta-confusion
3	Brigid	What does polarity mean?	Explanation-request; concept-request
4	Instructor	Polarity means plus and minus. Polarity means- This [points at representation] By looking at this one, can you see it has like electronegativity or stuff. Polarity means that-	Explanation; metaphor; representation-request; concept-request
5	Adriana	I mean, like yeah, it doesn’t like show really like the pulling or the not pulling or the same.	Explanation; <b>representation;</b> <b>concept;</b> metaphor

codes often occur together [15, 16]. I ran this algorithm separately for cases with connections present or absent and for physical and virtual representations. Essentially, this analysis discovered:

1. Frequent patterns for cases with concept-representation connections *present* for *physical* representations
2. Frequent patterns for cases with concept-representation connections *absent* for *physical* representations
3. Frequent patterns for cases with concept-representation connections *present* for *virtual* representations
4. Frequent patterns for cases with concept-representation connections *absent* for *virtual* representations

Comparing findings 1 and 2 identified prompts of concept-representation connections for physical representations. Comparing findings 3 and 4 identified prompts of concept-representation connections for virtual representations. Comparing findings 1 and 3 identified differences between representation modes.

### 3. RESULTS

In the following, I first discuss which discourse patterns were found to prompt concept-representation connections with physical representations or with virtual representations. Then, I compare the physical and virtual representation modes.

#### 3.1 Physical models

To identify prompts of concept-representation connections with physical representations, I considered patterns found only for cases with a *present* concept-representation connection (i.e., cases that correspond to two turns followed by a concept-representation connection). Table 4 shows statistics for the patterns.

Several results are worth noting. First, it stands out that all patterns involve either a reference to a concept or to a representation.

**Table 3. Number of cases by representation mode and speaker.**

Representation mode	Label		Speaker	
	Connection present	Connection absent	Student	Instructor
Physical	229 (7.33%)	2,895 (92.67%)	2,115 (67.70%)	1,009 (32.30%)
Virtual	67 (3.28%)	1,976 (96.72%)	1,780 (86.13%)	263 (12.87%)

**Table 4. Frequent patterns for physical representations (underlined: instructor utterances, italics: patterns that overlap with virtual representations).**

Frequent pattern	Support	Confidence
1. <u>instructor-assent</u> ; <i>student-concept</i>	0.100	0.410
2. <u>instructor-assent</u> ; <i>student-representation</i>	0.087	0.377
3. <u>instructor-representation-request</u> ; <i>instructor-concept-request</i>	0.074	0.684
4. <i>student-representation</i> ; <i>student-concept</i>	0.201	0.803
5. <u>instructor-assent</u> ; <i>student-representation</i> ; <i>student-concept</i>	0.083	0.536

This finding suggests that it may be easiest for students to make a concept-representation connection if discourse is already focused on the concept or representation. A related finding is that 3 of 5 patterns include references to *both* concepts and representations—either as a request to relate to concepts and representations by the instructor (#3 in Table 4) or by the students themselves (#4 and #5). These patterns have the highest support and confidence. Hence, students may be particularly likely to make a concept-representation connection if it already occurs in previous discourse.

Second, 4 of 5 patterns involve instructor utterances. This finding suggests that instructors may be better than students at prompting concept-representation connections.

Finally, 3 of 5 patterns include assent by the instructor. Assent is defined as agreement with a previous statement (see Table 1), often in the form of encouragement (e.g., “mhm”). In the identified patterns, such encouragement co-occurs with references to a concept or to a representation (or both) provided by one of the students or by the instructor. This finding suggests that encouragement by the instructor—when discourse is already focused on a concept or representation—prompts students to elaborate by making a concept-representation connection.

### 3.2 Virtual models

To identify triggers of concept-representation connections with virtual representations, I considered patterns found only for cases with a *present* concept-representation connection. Table 5 shows statistics for these patterns.

The following findings stand out. First, all patterns include a reference to a concept or to a representation. Hence, students may be likely to make a concept-representation connection if discourse is already focused on a concept or on a representation. A related result is that 7 of 16 patterns include a reference to both concept and representation (either as request by the instructor, or a direct reference to both by the instructor or the student). These patterns

**Table 5. Frequent patterns for virtual representations (underlined: instructor utterances, italics: overlap with physical representations).**

Frequent pattern	Support	Confidence
1. <u>instructor-assent</u> ; <u>instructor-concept</u>	0.075	0.420
2. <i>student-metaConfusion</i> ; <i>student-representation</i>	0.104	0.393
3. <i>student-metaUnderstanding</i> ; <i>student-representation</i>	0.075	0.471
4. <i>student-metaUnderstanding</i> ; <i>student-concept</i>	0.075	0.476
5. <i>student-metaConfusion</i> ; <i>student-concept</i>	0.075	0.386
6. <i>student-concept</i> ; <i>student-assent</i>	0.134	0.388
7. <i>student-representation</i> ; <i>student-assent</i>	0.134	0.378
8. <u>instructor-concept-request</u> ; <u>instructor-concept</u>	0.060	0.468
9. <u>instructor-representation-request</u> ; <u>instructor-representation</u>	0.060	0.468
10. <u>instructor-representation-request</u> ; <u>instructor-concept</u>	0.060	0.508
11. <i>student-assent</i> ; <u>instructor-representation</u> ; <u>instructor-concept</u>	0.060	0.568
12. <i>student-metaConfusion</i> ; <i>student-representation</i> ; <i>student-concept</i>	0.075	0.468
13. <u>instructor-representation-request</u> ; <u>instructor-representation</u> ; <u>instructor-concept</u>	0.060	0.637
14. <i>student-metaUnderstanding</i> ; <i>student-concept</i> ; <i>student-representation</i>	0.060	0.463
15. <i>student-assent</i> ; <i>student-concept</i> ; <i>student-representation</i>	0.119	0.550
16. <u>instructor-representation</u> ; <i>student-assent</i>	0.060	0.299
17. <u>instructor-assent</u> ; <i>student-concept</i>	0.090	0.374
18. <u>instructor-assent</u> ; <i>student-representation</i>	0.104	0.428
19. <u>instructor-representation-request</u> ; <u>instructor-concept-request</u>	0.075	0.714
20. <i>student-concept</i> ; <i>student-representation</i>	0.254	0.792
21. <u>instructor-assent</u> ; <i>student-representation</i> ; <i>student-concept</i>	0.090	0.539

had the highest support and confidence. Hence, students may be particularly likely to deepen their discussion about a connection if prior discourse already focuses on the connection.

Second, 7 of 16 patterns involve instructor utterances. This ratio seems surprisingly high, given that students worked without the instructor for most of the time. Recall that when working with virtual representations, instructor support was available only upon request, and that when students worked with virtual representations, they generated 86.13% of the utterances—instructors only 12.87% (see Table 2). Thus, this finding may indicate that students need help from an instructor to make concept-representation connections, even if they receive technology support.

Third, 6 of 16 patterns include assent by the instructor (4 of 6) or a student (2 of 6). Recall that assent is defined as agreement with a previous statement (see Table 1), often in the form of encouragement. Again, assent always co-occurs with a reference to a concept or representation. Hence, this finding suggests that encouragement can prompt a concept-representation connection—regardless of whether it is provided by a student or a tutor.

Fourth, 4 of the 7 patterns that involve instructor utterances involve explicit requests for the student to relate to a concept or a representation. This request is always combined with an instructor reference to a concept or to a representation. This finding suggests that prompts to elaborate on a previously mentioned concept or representation yields concept-representation connections.

Finally, 6 of 16 patterns include a meta-cognitive utterance by the student about understanding (3 of 6) or confusion (3 of 6). All of these meta-cognitive utterances co-occur with a reference to a concept and/or a representation. None of these meta-cognitive utterances co-occur with instructor utterances. This finding suggests that meta-cognitive statements about one's own understanding can prompt concept-representation connections; for example, after a student voices confusion about a concept, the partner may use a representation to explain the concept.

### 3.3 Comparing physical and virtual modes

Finally, I investigated whether prompts of concept-representation connections differ by representation mode. The following commonalities stand out. First, all patterns found for physical representations were also found for virtual representations. Hence, prompts that help students connect concepts to physical representations are also successful prompts for virtual representations.

Second, patterns with highest support and confidence for both representation modes involved relations to concepts and/or representations, indicating that students co-construct concept-representational competencies incrementally, over the course of consecutive social exchanges.

Third, the instructor plays a prominent role in prompting concept-representation connections both for physical and virtual representations: instructor utterances were involved in 4 of 5 patterns for virtual representations and in 7 of 16 patterns for physical representations. This result suggests that the role of an instructor is critical to students' success in making concept-representation connections, regardless of representation mode.

Fourth, assent that co-occurs with a reference to concepts or representations plays an important role for both representation modes. Hence, encouraging students to elaborate by agreeing with prior utterances may prompt concept-representation connections.

Several differences between representation modes stand out. First, students made fewer concept-representation connections with virtual representations (3.28%; see Table 2) than with physical

representations (7.33%). Given the finding that instructors play a critical role for concept-representation connections, it may be that the lower involvement of an instructor when students work with virtual representations accounts for this difference.

Second, when students work with physical representations, assent seems to prompt concept-representation connections only when it is provided by the instructor. By contrast, when students work with virtual representations, assent provided by the student partner also prompts concept-representation connections. Hence, this type of prompt may be one that students can take responsibility for when working collaboratively without instructor support.

Finally, meta-cognitive utterances of confusion or understanding of concepts or representations were important prompts only for virtual representations. Given that none of the patterns that included meta-cognitive utterances included instructor utterances, it seems that meta-cognitive utterances are a major mechanism by which students can prompt concept-representation connections in the absence of instructor support.

## 4. DISCUSSION

My goal was to investigate the representation dilemma: how novice students make connections between new concepts and new representations. I investigated which social events in collaborative classroom practices prompt students' concept-representation connections. Using frequent pattern mining, I identified such prompts for physical and virtual representations.

A key finding was that prompts with the highest confidence and support contained relations to a previously mentioned concept or representation, regardless of representation mode. This finding suggests that the conceptual process by which students make concept-representation connections is mediated by a gradual, incremental social mechanism. Students may first discuss a concept or a representation separately from one another before they negotiate the connection between the two.

A further finding was that instructors played a crucial role in prompting concept-representation connections, regardless of the representation mode. With respect to physical representations, this finding is not surprising because students have no other way of receiving feedback and assistance. However, with respect to virtual representations, this finding is surprising because the representations were embedded in an educational technology that supported concept-representation connections (and was shown to be successful in doing so [14]). Hence technology support for concept-representation connections may not be able to "replace" instructor support—at least when students have little prior knowledge about the concepts or representations.

Finally, the results showed that meta-cognitive statements can prompt concept-representation connections when students work on virtual representations. Meta-cognitive statements were the only successful prompts when an instructor was not involved. The social mechanism underlying this effect may be that a meta-cognitive statement by one student prompts the other to explain the given concept-representation connection.

## 5. LIMITATIONS & FUTURE RESEARCH

Several limitations of the present analysis should be considered when interpreting these results. First, the study used a multiple-case design, which focuses on gaining in-depth insights into social processes that unfold over time rather than on generating generalizable evidence for causal effects. Therefore, this paper does not attempt to make causal claims about which prompts are effective, but to generate new hypotheses about social prompts. Based on

the theoretical consideration that instructional support for concept-representation connections may be most effective if it takes advantage of social prompts that different representation modes afford, one may hypothesize that instructional interventions should be designed to maximize instructors' capacity to assist students, regardless of the representation mode. One might also hypothesize that interventions with virtual representations are particularly effective if students are prompted (or trained) in monitoring their own understanding and communicate their meta-cognitive assessments to their partner. These hypotheses should be tested with study designs that allow for causal claims.

Another limitation regarding the generalizability stems from the focus on the representation dilemma; that is, how novice students see novel concepts in novel representations. Because students in this study had limited prior knowledge about concepts and representations, we do not know if the results generalize to advanced students. One may speculate that the importance of instructor support decreases as students learn, especially if students receive technology support. One might also speculate that the incremental way in which students focus on a concept or a representation alone before connecting them plays a lesser role if students have prior experience with representations or concepts. Hence, future research should examine social prompts among advanced students. A related limitation is that many utterances did not involve concept-representation connections. Consequently, the overall support and confidence for the discovered patterns is rather low. Concept-representation connections are one of many mechanisms of students' learning, so future research may apply the present analysis to other social (or conceptual) mechanisms of learning.

A further limitation results from this study's focus on social mechanisms that may underlie the complementary effects of representation modes on conceptual learning. Consequently, this study did not consider prompts beyond collaborative discourse, such as availability of resources in the classroom, an individual's bodily experiences with physical representations, etc. Future research could examine the role of such distributed and embodied types of prompts for concept-representation connections.

Finally, an assumption of this study was that concept-representation connections are a "desirable" educational outcome. While much research documents the importance of connecting concepts to representations for students' learning [1-12], this study did not test whether concept-representation connections correlate with learning outcomes. Future research could assess learning outcomes and test whether concept-representation connections mediate the effectiveness of physical and virtual representations and of interventions that combine both modes.

## 6. CONCLUSIONS

This study yields new theoretical insights into the representation dilemma by revealing how novice students connect new concepts to new representations. This study identified social events that prompt students to connect concepts to physical and virtual representations. These connections emerge in a co-constructive process that is incremental and requires instructor support. Meta-cognitive statements prompt students to help one another to make connections when an instructor is not always available.

At a practical level, this study yields new hypotheses suggesting that physical and virtual representations are most effective if instructor support is available. If instructor support is not available, interventions with virtual representations may benefit from meta-cognitive support. These hypotheses are empirically testable in studies on combinations of physical and virtual representations.

## 7. ACKNOWLEDGMENTS

We thank participating teachers and students, Sally Wu, Jamie Schuberth, Ashley Hong, Amber Kim, and Tae Ho Lee.

## 8. REFERENCES

- [1] Dreher, A., and Kuntze, S.: 'Teachers facing the dilemma of multiple representations being aid and obstacle for learning: Evaluations of tasks and theme-specific noticing', *Journal für Mathematik-Didaktik*, 1-22 (2014)
- [2] Uttal, D.H., and O'Doherty, K.: 'Comprehending and learning from 'visualizations': A developmental perspective', in Gilbert, J. (Ed.): 'Visualization: Theory and practice in science education' (Springer), 53-72 (2008)
- [3] Gilbert, J.K.: 'Visualization: A metacognitive skill in science and science education', in Gilbert, J.K. (Ed.): 'Visualization: Theory and practice in science education' (Springer), 9-27 (2005)
- [4] NRC: 'Learning to Think Spatially' (National Academies Press). (2006)
- [5] Justi, R., and Gilbert, J.K.: 'Models and modelling in chemical education', in de Jong, O., Justi, R., Treagust, D.F., and van Driel, J.H. (Eds.): 'Chemical education: Towards research-based practice' (Kluwer Academic Publishers), 47-68 (2002)
- [6] Ainsworth, S.: 'DeFT: A conceptual framework for considering learning with multiple representations.', *Learning and Instruction*, 16, 183-198 (2006)
- [7] de Jong, T., Linn, M.C., and Zacharia, Z.C.: 'Physical and virtual laboratories in science and engineering education', *Science*, 340, 305-308 (2013)
- [8] Zacharia, Z.C., Loizou, E., and Papaevripidou, M.: 'Is physicality an important aspect of learning through science experimentation among kindergarten students?', *Early Childhood Research Quarterly*, 27, 447-457 (2012)
- [9] Olympiou, G., and Zacharia, Z.C.: 'Blending physical and virtual manipulatives: An effort to improve students' conceptual understanding through science laboratory experimentation', *Science Education*, 96, 21-47 (2012)
- [10] Roschelle, J.: 'Learning by Collaborating: Convergent Conceptual Change', *Journal of the Learning Sciences*, 2, 235-276 (1992)
- [11] Boulter, C.J., and Gilbert, J.K.: 'Challenges and opportunities of developing models in science education', in Gilbert, J.K., and Boulter, C.J. (Eds.): 'Developing Models in Science Education' (Kluwer Academic Publishers), 343-362 (2000)
- [12] Wu, H.K., Krajcik, J.S., and Soloway, E.: 'Promoting understanding of chemical representations: Students' use of a visualization tool in the classroom', *Journal of research in science teaching*, 38, 821-842 (2001)
- [13] Donmoyer, R.: 'Generalizability and the single-case study', in Eisner, E., and Peshkin, A. (Eds.): 'Qualitative inquiry in education: The continuing debate' (Teachers College Press) 175-200 (1990)
- [14] Rau, M.A.: 'Enhancing undergraduate chemistry learning by helping students make connections among multiple graphical representations', *Chemistry Education Research and Practice*, 16, 654-669 (2015)
- [15] Romero, C., J.M. Luna, J.M., J.R. Romero, J.R., and S. Ventura, S.: 'RM-Tool: A framework for discovering and evaluating association rules', *Advances in Engineering Software*, 42, 566-576 (2011)
- [16] Luna, J.M.: 'Pattern mining: Current status and emerging topics', *Progress in Artificial Intelligence*, 1-6 (2016)

# Predicting Performance on MOOC Assessments using Multi-Regression Models

Zhiyun Ren  
George Mason University  
4400 University Dr,  
Fairfax, VA 22030  
zen4@masonlive.gmu.edu

Huzefa Rangwala  
George Mason University  
4400 University Dr,  
Fairfax, VA 22030  
rangwala@cs.gmu.edu

Aditya Johri  
George Mason University  
4400 University Dr,  
Fairfax, VA 22030  
johri@gmu.edu

## ABSTRACT

The past few years has seen the rapid growth of data mining approaches for the analysis of data obtained from Massive Open Online Courses (MOOCs). The objectives of this study are to develop approaches to predict the scores a student may achieve on a given grade-related assessment based on information, considered as prior performance or prior activity in the course. We develop a personalized linear multiple regression (PLMR) model to predict the grade for a student, prior to attempting the assessment activity. The developed model is real-time and tracks the participation of a student within a MOOC (via click-stream server logs) and predicts the performance of a student on the next assessment within the course offering. We perform a comprehensive set of experiments on data obtained from two openEdX MOOCs via a Stanford University initiative. Our experimental results show the promise of the proposed approach in comparison to baseline approaches and also helps in identification of key features that are associated with the study habits and learning behaviors of students.

## Keywords

Personalized Linear Multi-Regression Models, MOOC, Performance prediction

## 1. INTRODUCTION

Since their inception, Massive Open Online Courses (MOOCs) have aimed at delivering online learning on a wide variety of topics to a large number of participants across the world. Due to the low cost (most times zero) and lack of entry barriers (e.g., prerequisites or skill requirements) for the participants, large number of students enroll in MOOCs but only a small fraction of them keep themselves engaged in the learning materials and participate in the various activities associated with the course offering such as viewing the video lectures, studying the material, completing the various quizzes and homework-based assessments.

Given, this high attrition rate and potential of MOOCs to deliver low-cost but high quality education, several researchers have analyzed the server logs associated with these MOOCs to determine the factors associated with students dropping out. Several predictive methods have been developed to predict when a participant will drop out from a MOOC [4, 5, 6, 14]. Using self reported surveys, studies have determined the different motivations for students enrolling and participating in a MOOC. Participants enroll in a MOOC sometimes to learn a subset of topics within the curriculum, sometimes to earn degree certificates for future career promotion or college credit, social experience or/and exploration of free online education [8]. Students with similar motivation have different learning outcomes from a MOOC based on the number of invested hours, prior education background, knowledge and skills [4].

In this paper, we present models to predict a student's future performance for a certain assessment activity within a MOOC. Specifically, we develop an approach based on personalized linear multi-regression (PLMR) to predict the performance of a student as they attempt various graded activities (assessments) within the MOOC. This approach was previously studied within the context of predicting a student's performance based on graded activities within a traditional university course with data extracted from a learning management system (Moodle) [3]. The developed model is real-time and tracks the participation of a student within a MOOC (via click-stream server logs) and predicts the performance of a student on the next assessment within the course offering. Our approach also allows us to capture the varying studying patterns associated with different students, and responsible for their performance. We evaluate our predictive model on two MOOCs offered using the OpenEdX platform and made available for learning analytics research via the Center for Advanced Research through Online Learning at Stanford University<sup>1</sup>.

We extract features that seek to identify the learning behavior and study habits for different students. These features capture the various interactions that show engagement, effort, learning and behavior for a given student participating in studying; by viewing the various video and text-based materials available within the MOOC offering coupled with student attempts on graded and non-graded activities like quizzes and homeworks. Our experimental evaluation shows accurate grade prediction for different types of homework as-

---

<sup>1</sup>datastage.stanford.edu

assessments in comparison to baseline models. Our approach also identifies the features found to be useful for predicting an accurate homework grade.

## 2. RELATED WORK

Several researchers have focused on the analysis of education data (including MOOCs), in an effort to understand the characteristics of student learning behaviors and motivation within this education model [11]. Brinton et. al. [1] developed an approach to predict if a student answers a question correct on the first attempt via click-stream information and social learning networks. Kennedy et. al. [7] analyzed the relationship between a student’s prior knowledge on end-of-MOOC performance. Sunar et. al. [12] developed an approach to predict the possible interactions between peers participating in a MOOC. Elbadrawy et. al. [3] proposed the use of personalized linear multi-regression models to predict student performance in a traditional university by extracting data from course management systems (Moodle). Our study focuses on MOOCs, which presents different assumptions, challenges and features in comparison to a traditional university environment.

Most similar to our proposed work, Pardos et. al. proposed a model “Item Difficulty Effect Model” (IDEM) that incorporates the difficulty levels of different questions and modifies Bayesian Knowledge Tracing (BKT) model [2] by adding an “Item” node to every question node. By identifying the challenges associated with modeling MOOC data, the IDEM approach and extensions that involve splitting questions into several sub-parts and incorporating resource (knowledge) information [9] are considered state-of-the-art MOOC assessment prediction approaches and referred as KT-IDEM. However, this approach can only predict a binary value grade. In contrast, the model proposed in this paper is able to predict both, a continuous and a binary grade.

## 3. METHODS

### 3.1 Personal Linear Multi-Regression Models

We train a personalized linear multi-regression (PLMR) model [3] to predict student performance within a MOOC. Specifically, the grade  $\hat{g}_{s,a}$  for a student  $s$  in an assessment activity  $a$  is predicted as follows:

$$\begin{aligned} \hat{g}_{s,a} &= b_s + p_s^t W f_{sa} \\ &= b_s + \sum_{d=1}^l (p_{s,d} \sum_{k=1}^{n_F} f_{sa,k} w_{d,k}), \end{aligned} \quad (1)$$

where  $b_s$  is bias term for student  $s$ ,  $f_{sa}$  is the feature vector of an interaction between student  $s$  and activity  $a$ . The features extracted from the MOOC server logs are described in the next Section.  $n_F$  is the length of  $f_{sa}$ , indicating the dimension of our feature space.  $l$  is the number of linear regression models,  $W$  is the coefficient matrix of dimensions  $l \times n_F$  that holds the coefficients of the  $l$  linear regression models, and  $p_s$  is a vector of length  $l$  that holds the memberships of student  $s$  within the  $l$  different regression models [3]. Using lasso [13], we solve the following optimization problem:

$$\underset{(W,P,B)}{\text{minimize}} L(W, P, B) + \gamma(\|P\|_F + \|W\|_F), \quad (2)$$

where  $W$ ,  $P$  and  $B$  denote the feature weights, student memberships and bias terms, respectively. The loss function  $L(\cdot)$  is the least square loss for regression problems.  $\gamma(\|P\|_F + \|W\|_F)$  is a regularizer that controls the model complexity by controlling the values of feature weights and student memberships. Tuning the scalar  $\gamma$  prevents model from over-fitting.

### 3.2 Feature Description

We extract features from MOOC server logs and formulate the PLMR model to predict real-time assessment grade for a given student. Figure 1 shows the various activities, generally available within a MOOC. Fig 1 (a) shows that each homework has corresponding quizzes, each of which has its corresponding video as resources for learning. Fig 1 (b) shows that while watching a video, a student can have a series of actions. Fig 1 (c) shows that while studying using a MOOC, a student can have several login sessions. In order to capture the latent information behind the click-stream for each student, we extract six types of features: (i) session features, (ii) quiz related features, (iii) video related features, (iv) homework related features, (v) time related features and (vi) interval-based features. These features constitute the feature vector  $f_{sa}$  for a student and a homework assessment. The description of these features are as follows:

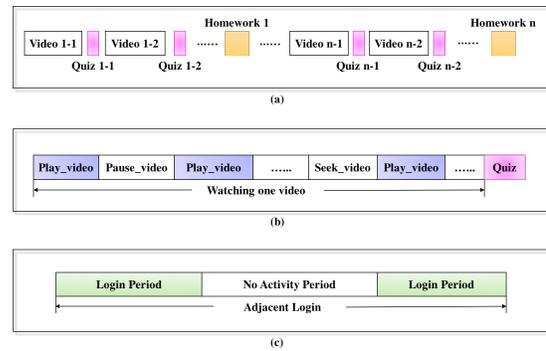


Figure 1: Different activities within a MOOC.

#### (i) Session features.:

A single study session is defined by a student login combined with the various available study interactions that a student may partake in. Since, students do not always log out of a session, we assume that a “no activity” period of more than one hour constitutes a student logging out of a session. We show a “no activity” period for a student between two consecutive sessions in Fig 1 (c).

- **NumSession** is the the average number of daily study sessions a student engages in, before a homework attempt.
- **AvgSessionLen** is the average length of each session in minutes. We calculate the average study time of a study session by

$$\text{AvgSessionLen} = \frac{\text{Total study time}}{\text{NumSession}}. \quad (3)$$

- **AvgNumLogin**. Students are free to choose when to login and study in a MOOC environment. We consider

a day as a “work day” if a student logs into the study system; and a day as “rest day” if a student does not. The rate of “work” and “rest” can capture a student’s learning habits and engagement characteristics.

$$AvgNumLogin = \frac{\# \text{ of “work day”}}{\# \text{ of “work day”} + \# \text{ of “rest day”}} \quad (4)$$

(ii) *Quiz Related features:*

- **NumQuiz** is the number of quizzes a student takes before a homework attempt. This feature reflects the student’s dedication towards the course material and a factor towards performance in a homework.
- **AvgQuiz** is the average number of attempts for each quiz. The MOOCs studied in this paper allow unlimited attempts on a quiz.

(iii) *Video Related features:*

- **VideoNum** denotes the number of distinct video sessions for a student before a homework attempt.
- **VideoNumPause** is the average number of pause actions per video. There are several actions associated with viewing videos, including “pause video”, “play video”, “seek video” and “load video”. Tracking these actions allows for capturing a student’s focus level and learning habits.
- **VideoViewTime** is the total video viewing time.
- **VideoPctWatch**. In a large amount of cases, students do not finish watching a full video. As such, we calculate the average percentage of the watched part of a video.

(iv) *Homework Related features:*

- **HWPProblemSave** is the average number of “save answer” actions for each homework assessment. Before submitting answers for a homework, students are allowed to save their answer sheet and check as many times as they need. This feature is more valuable when the MOOC provides only one chance for a homework answer submission.

(v) *Time Related features:*

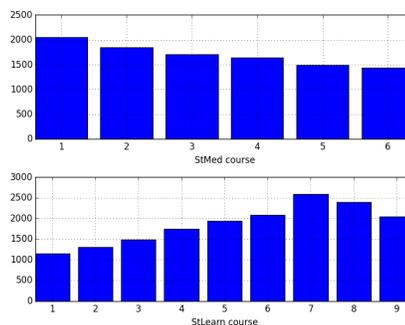
- **TimeHwQuiz** is the time between a homework answer submission and the last quiz attempt.
- **TimeHwVideo** is the time between a homework answer submission and the last video watching activity.
- **TimePlayVideo** is the percentage of study sessions with video watching activity over all the study sessions.
- **HwSessions** is the number of sessions that have homework related activities (save and submit).

(vi) *Interval-Based features:*

It is expected that there will be some changes in study activities once the students know the former homework’s grade. They may study harder if they don’t get a satisfactory score. The interval-based features are aiming to represent different activities between two consecutive homeworks.

- **IntervalNumQuiz**: denotes the number of quizzes the student takes between two homeworks.
- **IntervalQuizAttempt**: is the average number of quiz attempts between two homeworks.
- **IntervalVideo**: is the number of videos a student watches between two homeworks.
- **IntervalDailySession**: is the average number of sessions per day between two homeworks.
- **IntervalLogin**: is the percentage of login days between two homeworks.

We also use the cumulative grade (so-far) on quizzes and homeworks for a student as a feature and denote it by **Meanscore**. For our baseline approach we only consider the averages computed on the previous homeworks.



**Figure 2: Distribution of students attempting each Assessment.** StMed and StLearn had 6 and 9 assessments, respectively.

## 4. EXPERIMENTS

### 4.1 Datasets

We evaluated our methods on two MOOCs: “Statistics in Medicine” (represented as StMed in this paper) taught in Summer 2014 and “Statistical Learning” (represented as StLearn in this paper) taught in Winter 2015.

**StMed:** This dataset includes server logs tracking information about a student viewing video lectures, checking text/web articles, attempting quizzes and homeworks (which are graded). Specifically, this MOOC contains 9 learning units with 111 assessments, including 79 quizzes, 6 homeworks and 26 single questions. The course had 13,130 students enrolled, among which 4337 students submitted at least one assignment (quiz or homework) and had corresponding scores, 1262 students have completed part of the six homeworks and 1099 students have attempted all the homeworks. 193 students attempted all the 79 quizzes and six homeworks. This course had 131 videos and 6481 students had video related activity.

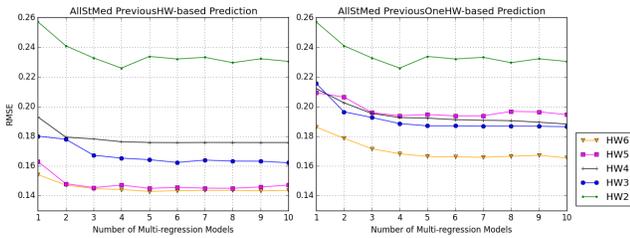


Figure 3: AllStMed Prediction Results. RMSE ( $\downarrow$  is better).

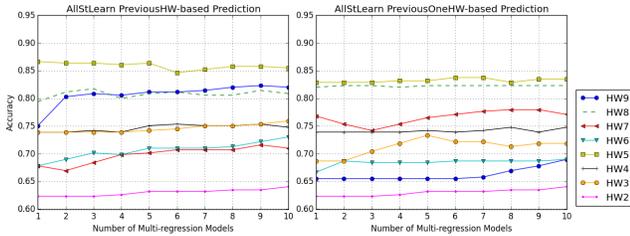


Figure 4: AllStLearn Prediction Results. Accuracy ( $\uparrow$  is better).

**StLearn:** This course had ten units. Except the first one, all units have quizzes and end of unit homeworks, which add up to 103 assessments in total. 52,821 students enrolled in this course, and 4987 students had assessment activities, 3509 students attempted a subsets of the available homeworks while 346 students attempted all the 9 homeworks, and 118 students attempted all the 103 assessments. The key difference between the homeworks in the StLearn in comparison to the StMed is that homeworks have only one question which a student can either get correct or incorrect. As such, scoring in this MOOC is binary instead of continuous. To predict whether a student answers a question correctly, we reformulate the regression problem as a classification problem using a logistic loss function. Figure 2 shows the distribution of students attempting the different assessments available across the two MOOCs studied here.

## 4.2 Experimental Protocol

In order to gain a deep insight of students’ performance in a MOOC, we perform two types of experiments. Given  $n$ , homework assessments represented as  $\{H_1, \dots, H_n\}$  our objective is to predict the score a student achieves in each of the  $n$  homeworks. Depicting the most realistic setting, for the  $i$ -th homework,  $H_i$  we define the training set as all homework and student pairs who attempt and have a score for all homeworks up to the  $H_{i-1}$ . For predicting the score for  $H_i$  for a given student, we use all the features extracted just before attempting the target homework  $H_i$ . We refer to this as **PreviousHW-based Prediction**. Secondly, for the predicting  $i$ -th homework  $H_i$ ’s score, we use training data of student-homework pairs restricted from only the previous one homework i.e.,  $H_{i-1}$ . This experiment is referred by **PreviousOneHW-based Prediction**. Note, in these cases we cannot make any prediction for the first homework ( $H_1$ ) since, we do not have any training information for a

given student.

## 4.3 Data Partition

We partition the students for StLearn and StMed into two groups: the group of students who attempt *all* the requested homeworks, and the group of students who finish *few* of the homeworks. This allows us to consider the different motivations and expectations of students enrolling in a MOOC. For example, the students who aim to learn in a MOOC may choose watching videos over taking all homeworks. While, the students who want to achieve a degree certificate may focus on the homework completeness. We refer to the first group by “Partial homeworks accomplished group”, and the second group by “All homeworks accomplished group”. We evaluate our models on the two groups for the **AllStMed** and **AllStLearn** datasets. Specifically, we name the four group of students as **AllStMed**, **AllStLearn**, **PartialStMed** and **PartialStLearn** based on their group and MOOC class.

HW#	PLMR	Meanscore
2	<b>0.230</b>	0.248
3	<b>0.162</b>	0.176
4	<b>0.176</b>	0.196
5	<b>0.144</b>	0.156
6	<b>0.143</b>	0.150
Avg	<b>0.171</b>	0.185

Table 1: PreviousHW-based RMSE Performance (RMSE) comparison for AllStMed.

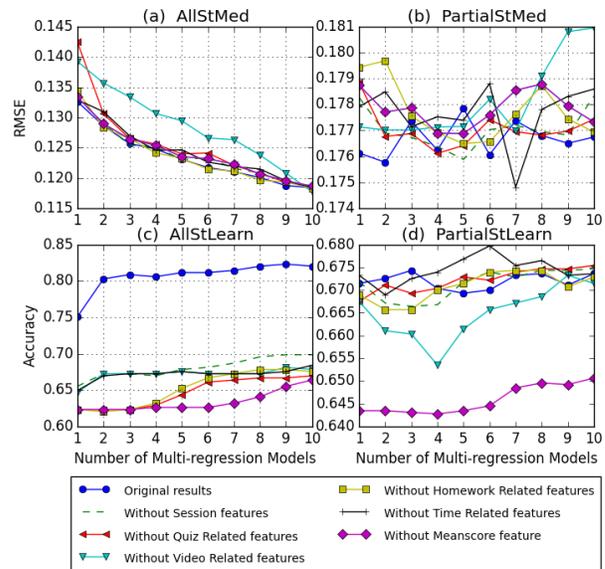


Figure 5: Predictive Performance with Removal of Feature Types.

## 4.4 Evaluation Metrics

StMed course has continuous scores for a homework, which are scaled between 0 and 1. However, the homework score is binary in the StLearn course, indicating whether the student answers a question correctly or incorrectly. For StLearn, we use a logistic loss and formulate a classification problem

HW#	Accuracy ( $\uparrow$ )			$F_1$ ( $\uparrow$ )		
	PLMR	Baseline		PLMR	Baseline	
		Meanscore	KT-IDEM		Meanscore	KT-IDEM
2	0.641	<b>0.646</b>	0.623	0.775	<b>0.777</b>	0.768
3	<b>0.760</b>	0.580	0.681	<b>0.821</b>	0.805	0.810
4	<b>0.754</b>	0.710	0.739	0.838	0.706	<b>0.850</b>
5	<b>0.867</b>	0.809	0.829	<b>0.920</b>	0.880	0.906
6	<b>0.730</b>	0.678	0.667	<b>0.808</b>	0.776	0.800
7	0.716	0.675	<b>0.730</b>	<b>0.887</b>	0.878	0.844
8	<b>0.817</b>	0.762	0.817	<b>0.903</b>	0.849	0.886
9	<b>0.823</b>	0.794	0.777	<b>0.864</b>	0.856	0.853
Avg	<b>0.764</b>	0.707	0.759	<b>0.852</b>	0.816	0.848

Table 2: PreviousHW-based prediction performance comparison for AllStLearn group.

instead of the regression problem as done for the StMed course. To evaluate the performance of our approach, we use the root mean squared error (RMSE) as the metric of choice for regression problem. For classification problem, we use accuracy and the F1-score (harmonic mean of precision and recall), known to be a suitable metric for imbalanced datasets.

## 4.5 Comparative Approaches.

In this work, we compare the performance of our proposed methods with two different competitive baseline approaches.

(i) **Average grade of the previous homeworks.** We calculate the mean score of a given student’s previous homeworks to predict their future performance and is denoted as Meanscore. We use this method to compare our prediction results on StMed.

(ii) **KT-IDEM [10].** KT-IDEM is a modified version of original BKT model. By adding an “item” node to every question node, the model is able to identify different difficulty levels of each question. Since this model can only predict a binary value grade, we use this model to compare our prediction results on StLearn.

## 5. RESULTS AND DISCUSSION

### 5.1 Assessment Prediction Results

Figures 3 and 4 show the prediction results with varying number of regression models for the AllStMed and AllStLearn MOOCs, respectively. Analyzing Figure 3 we observe that as the number of regression models increases the RMSE metric goes lower and use of five models seems to be good choice for all the different homeworks. Comparing the PreviousHW- and PreviousOneHW-based results, we notice that predictions for all the homeworks (HW3, HW4, HW5, and HW6) benefits from using all the available training data prior to those homeworks i.e., to predict grade for  $H_i$  it is better to use training information extracted from  $H_1 \dots H_{i-1}$  rather than just  $H_{i-1}$ . Similar observations can be made while analyzing the prediction results for the AllStLearn cohort which includes nine homework correct/incorrect binary assessments. Figure 4 shows the accuracy scores (higher is better) for the three experiments. For the PreviousOneHW- and PreviousHW-based experiments HW5 shows the best

prediction results. This suggests that in the middle of a MOOC, students tend to have stable study activities and the performance is more predictable than other phases. Also, some homeworks thrive well with just using training data from the previous homework (PreviousOneHW-based, e.g. HW3).

#### 5.1.1 Comparative Performance

Table 1 shows the comparison between baseline approach (Meanscore) and the predictive model for the PreviousHW-based experiments for the AllStMed group. We cannot report results for the KT-IDEM model since, it solves the binary classification problem only. Table 2 shows the comparison of the accuracy and F1 scores of the AllStLearn groups with baseline approaches. We notice that for predicting the second homework, which only uses the information from HW1, the predictive model is not as good as the mean baseline, which reflects that under the situation of lack of necessary amount of information, linear regression models cannot always outperform the baseline. But as the dataset gets larger, our approach outperforms the baseline due to the availability of more training data. From Table 2, we also notice for some homework, KT-IDEM has better performance than PLMR (HW7 and HW4). This could be due to unstable academic activities during these two study periods, which can effect the performance of PLMR.

#### 5.1.2 Feature Importance

We test the effect of each feature set in predicting the assessment scores by training the models under the absence of each feature group. For the StLearn course, since there is no limit on homework attempts, we do not add Interval-Based feature groups to the predictive model. Figure 5 shows the comparison of each prediction result for AllStMed, PartialStMed, AllStLearn and PartialStLearn cohorts. Analyzing these results we observe that for the StLearn MOOC, meanscore is a significant feature and removing it leads to a substantial decrease in accuracy for both All and Partial-cohorts. For the AllStMed, the removal of video related features leads to the most decrease in performance (i.e., increased RMSE). This suggests that features related to the video watching are crucial for predicting the final homework scores. For the PartialStMed, the use of all feature types or a subset does not show a clear winner. This could be due to the varying characteristics of students within these group.

Another way to analyze feature importance is to exclude the influence of the dominant feature, which is meanscore in our study. The evaluation formula of the importance of the  $i_{th}$  feature (excluding meanscore feature) is as follows:

$$I_i = \frac{1}{N} \sum_{n=1}^N \frac{\sum_{d=1}^l |p_{n_S,d} f_{n_S,i} w_{d,i}|}{\sum_{d=1}^l |p_{n_S,d} \sum_{k=1}^{n_F} f_{n_S,k} w_{d,k}|}, \quad (5)$$

where  $N$  is number of test samples,  $n_S$  is the student number corresponding to the  $n_{th}$  test sample.  $f_{n_S,i}$  is the feature value of an interaction between student  $n_S$  and activity  $i$ .  $n_F$  is the number of features.  $l$  is the number of linear regression models.  $w_{d,i}$  is the coefficient of  $d_{th}$  linear regression model with  $i_{th}$  feature, and  $p_{n_S,d}$  is the membership of student  $n_S$  with the  $d_{th}$  regression model. We calculate each feature's importance by calculating the percentage contribution of each feature to the overall grade prediction. Figure 6 shows the feature importance on the AllStMed group, excluding Meanscore feature. We can see **NumQuiz** and **VideoPctWatch** are the most important for AllStMed group besides **Meanscore** feature.

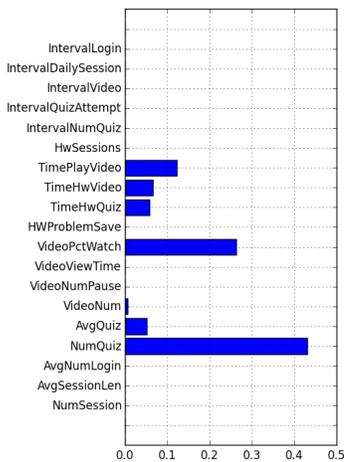


Figure 6: Feature importance for AllStMed.

## 6. CONCLUSION AND FUTURE WORK

In this work we formulated a personalized multiple linear regression model to predict the homework grades for a student enrolled and participating within a MOOC. Our contributions include engineering features that capture a student's studying behavior and learning habits, derived solely from the server logs of MOOCs. We evaluated our framework on two OpenEdX MOOC courses provided by an initiative at Stanford University. Our experimental evaluation shows improved performance in terms of prediction of real time homework scores compared to baseline methods. We also studied on different groups of student participants due to their motivation. Features associated with engagement (logging multiple times), studying materials (viewing videos and attempting quizzes) were found to be important along with prior homework scores for this prediction problem.

## 7. ACKNOWLEDGEMENTS

Funding was provided by NSF Grant, 1447489.

## 8. REFERENCES

- [1] Christopher G Brinton and Mung Chiang. Mooc performance prediction via clickstream data and social learning networks. *To appear, 34th IEEE INFOCOM. IEEE*, 2015.
- [2] 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.
- [3] Asmaa Elbadrawy, Scott Studham, and George Karypis. Personalized multi-regression models for predicting students performance in course activities. *UMN CS 14-011*, 2014.
- [4] Jeffrey A Greene, Christopher A Oswald, and Jeffrey Pomerantz. Predictors of retention and achievement in a massive open online course. *American Educational Research Journal*, page 0002831215584621, 2015.
- [5] Glyn Hughes and Chelsea Dobbins. The utilization of data analysis techniques in predicting student performance in massive open online courses (moocs). *Research and Practice in Technology Enhanced Learning*, 10(1):1–18, 2015.
- [6] Suhang Jiang, Adrienne Williams, Katerina Schenke, Mark Warschauer, and Diane O'dowd. Predicting mooc performance with week 1 behavior. In *Educational Data Mining 2014*, 2014.
- [7] Gregor Kennedy, Carleton Coffrin, Paula de Barba, and Linda Corrin. Predicting success: how learners' prior knowledge, skills and activities predict mooc performance. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 136–140. ACM, 2015.
- [8] Daniel FO Onah and Jane Sinclair. Learners expectations and motivations using content analysis in a mooc. In *EdMedia 2015-World Conference on Educational Media and Technology*, volume 2015, pages 185–194. Association for the Advancement of Computing in Education (AACE), 2015.
- [9] Zachary Pardos, Yoav Bergner, Daniel Seaton, and David Pritchard. Adapting bayesian knowledge tracing to a massive open online course in edx. In *Educational Data Mining 2013*, 2013.
- [10] Zachary A Pardos and Neil T Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, pages 243–254. Springer, 2011.
- [11] Alejandro Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert systems with applications*, 41(4):1432–1462, 2014.
- [12] Ayse Saliha Sunar, Nor Aniza Abdullah, Susan White, and Hugh C Davis. Analysing and predicting recurrent interactions among learners during online discussions in a mooc. *Proceedings of the 11th International Conference on Knowledge Management*, 2015.
- [13] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- [14] Jacob Whitehill, Joseph Jay Williams, Glenn Lopez, Cody Austun Coleman, and Justin Reich. Beyond prediction: First steps toward automatic intervention in mooc student stopout. *Available at SSRN 2611750*, 2015.

# Validating Game-based Measures of Implicit Science Learning

Elizabeth Rowe<sup>1</sup>  
Jodi Asbell-Clarke<sup>2</sup>

Teon Edwards<sup>7</sup>  
EdGE @ TERC

2067 Massachusetts Ave  
Cambridge, MA 02140

[elizabeth\\_rowe@terc.edu](mailto:elizabeth_rowe@terc.edu)

[Jodi\\_asbell-clarke@terc.edu](mailto:Jodi_asbell-clarke@terc.edu)

[teon\\_edwards@terc.edu](mailto:teon_edwards@terc.edu)

Michael Eagle<sup>3</sup>

Carnegie Mellon University  
4902 Forbes Ave

Pittsburgh, PA 15213

[meagle@cs.cmu.edu](mailto:meagle@cs.cmu.edu)

Drew Hicks<sup>4</sup>

Tiffany Barnes<sup>5</sup>

Rebecca Brown<sup>6</sup>

NC State University

890 Oval Drive

Raleigh, NC 27606

[drew@drewhicks.com](mailto:drew@drewhicks.com)

[tbarnes@ncsu.edu](mailto:tbarnes@ncsu.edu)

[rabrown@ncsu.edu](mailto:rabrown@ncsu.edu)

## ABSTRACT

Building on prior work visualizing player behavior using interaction networks [1], we examined whether measures of implicit science learning collected during gameplay were significantly related to changes in external pre-post assessments of the same constructs. As part of a national implementation study, we collected data from 329 high school students playing an optics puzzle game, *Quantum Spectre*, and modeled their gameplay as an interaction network, examining errors hypothesized to be related to a lack of implicit understanding of the science concepts embedded in the game. Hierarchical linear modeling (HLM) showed a negative relationship between the science errors identified during gameplay and implicit science learning. These results suggest *Quantum Spectre* gameplay behaviors are valid assessments of implicit science learning. Implications for how gameplay data might inform classroom teaching in-game scaffolding is discussed.

## Keywords

Game-based learning, Interaction Networks, Implicit Science Learning, Hierarchical linear modeling

## 1. INTRODUCTION

As digital games become increasingly prevalent in today's society and are played by the majority of youth of all demographics [2], it behooves us to study how the energy and passion invested in gaming can be harnessed for productive purposes. Game-based learning interests education researchers and learning scientists because digital games uniquely engage learners and because their data logs can serve as input for innovative learning assessments [3]. Data logs generated through gameplay can be used to study players' in-game activity [4] and how game-based learning can be leveraged for classroom learning. Research shows that elements of gameplay can invoke complex thinking such as scientific inquiry [5] and may foster learning-related skills such as creativity and persistence [4].

This work examines complex behaviors of students solving optics puzzles in the educational game *Quantum Spectre*, using interaction networks. An *Interaction Network* is a complex network representation of all observed player-game interactions for a given problem or task in a game or tutoring system [6]. Regions of the network can be discovered by applying network clustering methods. These regions correspond to high-level student approaches to problems [7]. In this work, we used Interaction Networks as visualizations to analyze *Quantum Spectre* gameplay data and automated the coding of game states that correspond to incorrect applications of the game's core science concepts. Three types of errors were coded: two science errors (placement and rotation) and puzzle errors.

This paper reports HLM analyses that relate those coded game states to implicit science learning measured by external pre/post assessments. The analyses examine how game-based learning is a function of *what players do* in the game, not simply duration of gameplay or highest level reached. This information is useful for building an adaptive version of the game to scaffold players' implicit science learning and for informing teachers about important aspects of student competency.

## 2. IMPLICIT SCIENCE LEARNING

Polanyi argued that implicit knowledge (also called tacit knowledge) is foundational and a required element of explicit learning [8]. Implicit understandings are embodied and enacted through our interactions with the world around us, but may not yet be formalized or expressed verbally or textually. Vygotsky described similar abilities and understandings a learner brings to a learning situation that can be scaffolded by a teacher, environment, and tools [9]. Implicit misunderstandings (often called misconceptions) may get in the way of a learner's conceptual development [10, 11], particularly in the area of basic physics, such as Newton's Laws of Motion. The work of diSessa distinguishes between the intuitive knowledge that novices hold—a book will not fall through a table or a glowing filament is hot—from an expert understanding of these phenomena, explaining that while learners' behaviors may be guided by implicit understandings, the learner is not necessarily ready to express the related formalisms or question the ideas in a deeper sense [12].

Games promise to reveal implicit learning because they can be (a) “sticky”—meaning they encourage players to dwell in the phenomena and (b) they leave a digital trail that reveals the patterns the players used in their learning process. Several

researchers have used educational data mining techniques within an Evidence-Centered Design framework to develop stealth assessments that discern evidence of learning from the vast amount of click data generated by online science games such as *SimCityEDU* [13], *Physics Playground* [14], and *Surge* [15].

As players “level up” in a game, they typically deal with the mechanics in increasingly complex applications, building implicit knowledge about the underlying system. Because games allow players to fail, repeat, revise, and try again—recording what players do in the process—games may be powerful formative assessments of learning, and the strategies players build. The methods players use to tackle new challenges may demonstrate conceptual understanding that the learner may not express in other ways and that may not be measured by current external learning assessments [4, 16]. Careful alignment of game mechanics with learning and assessment mechanics [17] may reveal implicit learning and empower teachers and learners to help bridge game-based knowledge to other forms of learning.

In a classroom, teachers may be able to build on implicit game-based learning if they have the right information and tools to support students at key moments in the learning process. That may consist of real-time information, provided during class to know who is struggling and needs attention, or more reflective information after school to help plan lessons for the next day based on class gameplay [18]. Post-game debriefing and discussions connecting gameplay with classroom learning help students apply and transfer learning that takes place in games [19]. To exploit learning that happens in games, teachers need to build bridges between the students’ “aha” moments while playing [20] and the content being covered in the classroom.

### 3. QUANTUM SPECTRE

To examine implicit science game-based learning, we studied high school students playing a Physics-oriented game called *Quantum Spectre*. *Quantum Spectre* is a puzzle-style game, designed for play in browsers and on tablets (Figure 1).



**Figure 1:** *Quantum Spectre* Puzzle 21. Players must direct the laser beams to the matching colored targets using movable mirrors and other optical devices, selected from the inventory on the right.

Players use optical devices, such as lenses and mirrors, to guide colored laser beams to matching targets. The lenses and mirrors can be flat, convex, or concave and single or double-sided. All devices produce scientifically accurate results when interacting with the laser beams. When the laser beams in a puzzle reach the matching colored targets, the puzzle is solved (i.e., goal state is

reached) and the player is scored on the number of moves used. The player earns three stars if the puzzle has been solved in the optimal number of moves, two stars for a low number of extra moves, and one star for simply solving the puzzle. Regardless of their score, players can proceed onto the next level, but players can repeat earlier levels at any time to improve their performance.

The game is divided into 6 zones with 30 puzzles in each zone. In Zone 1 of *Quantum Spectre*, the puzzles focus on 2 key concepts:

- **The Law of Reflection**, or Angle of Incidence equals Angle of Reflection—When reflecting off of a smooth surface, the path of a ray of light (such as a laser beam) will make the same angle with the surface (relative to the normal) upon exit as it makes upon entry.
- **Slope**—Players can use the squares on the game grid and calculate the slope (rise over run) to figure out and/or predict the paths of laser beams and where to place items.

This study focuses on data from Puzzles 14-23 in Zone 1 of the game. At this point in gameplay, players have presumably mastered the game mechanic, and mastery of the puzzles typically requires an understanding of Slope and the Law of Reflection. Table 1 provides an overview of Puzzles 14-23. The number of goal states reflects the number of unique solutions (position-rotation combinations) for each puzzle.

**Table 1: *Quantum Spectre* Puzzles 14-23**

Game Level	# Mirrors	# Targets	# Optimal Moves	# Goal States
14	1	1	2	1
15	2	1	4	5
16	2	1	3	8
17	2	2	4	1
18	2	2	4	6
19	4	4	7	4
20	6	3	12	42
21	6	5	11	6
22	3	1	6	1
23	4	2	8	3

### 4. CLASSIFYING GAMEPLAY BEHAVIORS USING INTERACTION NETWORKS

To simplify the vast number of puzzle solution paths into a manageable group we could study, we used a method called Interaction Networks (INs). INs use a complex network data structure to represent players’ solutions as traces of game states and actions, with additional information such as edge labels (e.g., labels of player actions). This process involved 4 key steps [1]: creating a full IN for each puzzle, clustering player actions using laser shapes, classifying clusters for evidence of implicit science understanding, and automating coding of player actions.

#### 4.1 Create Full Interaction Network

To construct an IN, we collected the set of all solution attempts for that puzzle. Each interaction is defined as Initial State, Action, and Resulting State, from the start of the puzzle until the player

solves the puzzle or exits the system. A sample trace is shown in Figure 2. Player actions are represented as edges in the network.

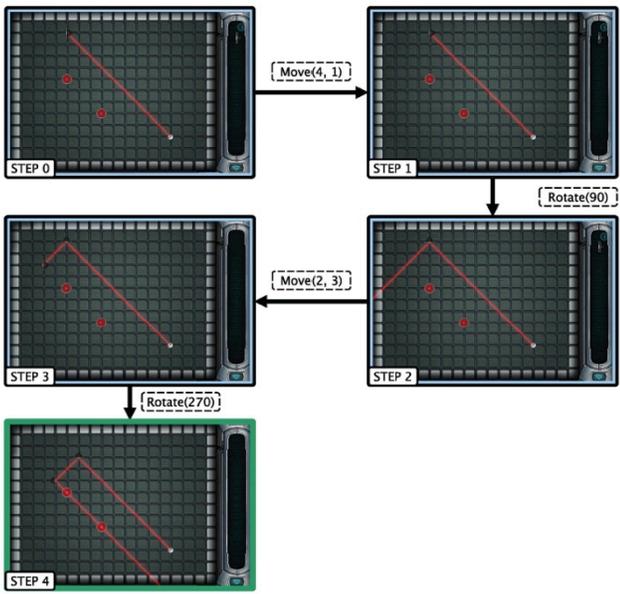


Figure 2: Sample trace of player actions in *Quantum Spectre* Puzzle 18 of Zone 1

Table 2 describes the complexity of the full interaction networks for Puzzles 14-23 for the full sample of students playing the game. The full IN of every state and every action taken was large, complex, and difficult to interpret in terms of player understanding.

Table 2: Interaction Networks in Puzzles 14-23

Game Level	# Players	Total # Moves	# Network Edges	# Unique States	# Laser Shapes
14	479	3003	462	164	5
15	473	3866	1009	484	10
16	462	3218	761	446	12
17	454	10878	1899	1067	21
18	439	10314	3458	1800	22
19	416	15389	7093	4550	330
20	384	10778	4947	2391	264
21	349	23080	13919	6261	696
22	282	3697	1500	1017	146
23	271	10529	6154	4138	364

## 4.2 Cluster States by Laser Shapes

Most puzzles have states in which different configurations of objects result in similar output. These states could be considered

equivalent since they show the same player proficiencies or errors, but a simple state representation would consider them as different states. In previous work using INs for games, it has been helpful to consider the output of a state as well as the position/orientation of objects in that state [7]. To group these equivalent states, we took a similar approach, using “laser shape” as part of our state representation to create Approach Maps. Approach Maps are a visual summary of the information contained in the interaction network [7]. This reduction is created by grouping similar states together based on how often students co-visit the states during their solution attempts. Here, the approach map consists of a list of targets hit by a laser of the appropriate color and a list of angles taken by that laser. This allows game states that represent similar errors to be effectively grouped together, as shown in Figure 3.

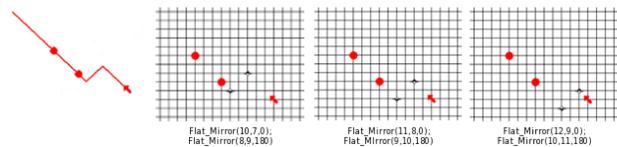


Figure 3: Using laser shape to group similar game states in Puzzle 18.

This approach preserves the relevant properties of a board state while ignoring distance traveled, which is not relevant to the game state.

## 4.3 Classify Player Actions for Implicit Science Understanding

A *Quantum Spectre* game designer who has a science education background, worked with a researcher to classify each laser shape into one of three categories:

- 1) *Correct move*—placement and rotation of the mirror are consistent with an eventual goal state
- 2) *Placement errors*—placement of the mirror in a location that does not match a goal state—may indicate a lack of understanding of slope.
- 3) *Rotation errors*—rotation of a mirror to an angle that does not match a goal state—may indicate a lack of understanding of the Law of Reflection.

As described elsewhere [1] using a subset of these data, the game designer and researcher also identified placements that were not consistent with a goal state but were more indicative of a lack of grasp of the puzzle mechanic than of a lack of science understanding. We labeled these *Puzzle errors*. For example, in puzzle shown in Figure 2, a correct solution requires players to use the two available mirrors to direct the laser through the two targets simultaneously. In Figure 4, player actions are consistent with someone who understands slope (i.e., they placed the mirror on the path of the laser) and the Law of Reflection (i.e., they rotated the mirror to reflect the mirror through the target). However, their actions are not going to let them solve this puzzle.

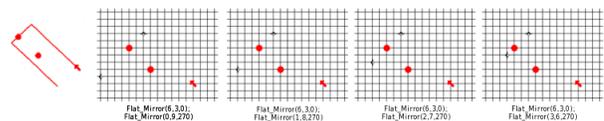


Figure 4: Sample Puzzle Errors in Puzzle 18.

## 4.4 Automated Coding of Individual Player Behaviors

Once all laser shapes had been coded and puzzle error placements identified, we automated the coding of individual player behaviors. Every player behavior was classified as a Placement Error, Rotation Error, or Puzzle Error (0=Not Present; 1=Present). These are mutually exclusive player behaviors. Player actions with none of these errors were classified as Correct. Figure 5 shows the distribution of player behaviors across each puzzle.

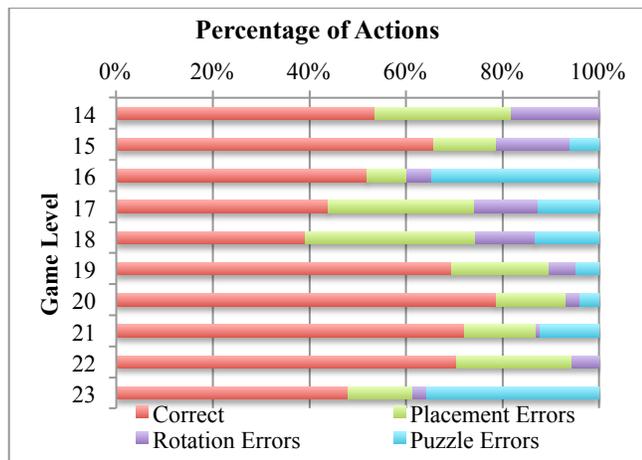


Figure 5: Error rates by puzzle level

The percentage of correct moves ranged from 39% in Level 18 to 79% in Level 20. Placement error rates range from 8% (Level 16) to 35% (Level 18). Rotation error rates were most common in earlier puzzles, 35% in Level 14 to 1% in Level 21. In two puzzles, Levels 14 and 22, no puzzle errors were possible. Puzzle errors in the remaining puzzles ranged from 4% (Level 20) to 36% (Level 23).

## 5. RESEARCH QUESTIONS & HYPOTHESES

In this paper, we examine the ways in which the extent of players' puzzle and science errors are related to changes in their performance on a pre-post assessment of slope and the Law of Reflection. We anticipated a negative relationship between placement errors, rotation errors, and pre-post assessment results—that is players who are demonstrating a lack of understanding of the science concepts in their gameplay will have smaller gains than players whose gameplay is consistent with an implicit understanding of slope and the Law of Reflection. Our anticipated relationship between puzzle errors and pre-post assessment results was less clear. It could be that puzzle errors interfere with their implicit learning of the science content. It could also be players who understand the science content are just as likely to make puzzle errors as players without that understanding, so there may be no relationship between the number of puzzle errors and pre-post assessment results.

## 6. METHODS

Teachers were assigned to one of three groups as part of a national *Quantum Spectre* implementation study. In Bridge classrooms, teachers encouraged students to play the game outside of class and used examples from the game as part of their science instruction. In Game Only classrooms, teachers encouraged students to play the game but provide no game examples during their science instruction. In Control classrooms, teachers and students did their normal science instruction with their students not knowing about

the game. This paper reports gameplay data from the 329 students in 29 classes (14 Bridge and 15 Game Only) that participated in the implementation study during the 2013-14 and 2014-15 academic years.

### 6.1 Sample

Because this study focuses on Puzzles 14-23 in Zone 1 of the game, 79 students were excluded from these analyses because they did not attempt Puzzle 14 of the game. The final sample of 329 high school science students included 132 females, 162 students in Bridge classrooms, 281 students in non-Honors/AP classrooms, and 249 students in classrooms where more than 75 percent of the students participated in the study.

### 6.2 Measures

This study collected gameplay log data, as described above, as well as pre-post assessment and student/classroom characteristics.

#### 6.2.1 Gameplay Metrics

To allow for the fact that students (a) used varying numbers of moves to solve the puzzles and (b) not all students completed Levels 14-23; the percentage of the total number of moves (actions) that were correct, placement errors, rotation errors, and puzzle errors was calculated. The mean error rate across all students was 19% placement errors, 7% rotation errors, and 12% puzzle errors. We used standardized (z-scores) error rates.

The total amount of time each student played *Quantum Spectre* and the highest level reached were also recorded. Previous analyses showed Puzzle 21 to have a high dropout rate [21], we analyzed whether or not players completing Puzzle 21 had any relationship to changes in pre-post assessment results. Among this sample, there was no significant difference in the percentage of students in Bridge and Game Only classrooms that reached Puzzle 22 ( $\chi^2=3.53$ , 1 d.f.,  $p=0.06$ ). Given the non-normal distribution of the amount of time students played *Quantum Spectre*, we categorized students as having played less than 1 hour, or 1 hour or more. Forty-one percent of students played 1 hour or more, this proportion did not vary among students in Bridge and Game Only classrooms ( $\chi^2=3.23$ , 1 d.f.,  $p=0.07$ ).

#### 6.2.2 Students & Classroom Characteristics

When completing the pre-assessment, students were asked to indicate their gender. We categorized class names (e.g., Honors Physics 101) obtained from teacher applications as being either Honors/AP classes or not. Seven of the 29 classes in this study were Honors/AP classes. Finally, we asked teachers the total number of students enrolled in each class. We calculated the percentage of the class with complete study information (e.g., complete consent/assent forms, pre-post assessments complete, and gameplay beyond Puzzle 1 in Zone 1). This ranged from 31 to 100 percent of each class, with the majority of classes (26) having more than half of the students participating.

#### 6.2.3 Assessments

Science content experts developed assessment instruments and tested them in a series of think-aloud interviews with 10 high school students. Each assessment contained 12 (pre) and 13 (post) questions that required minimal formalisms to complete. The pre- and post-assessments each included 3 items related to focal length that are not included in these analyses. Figures 6 and 7 are sample items for slope and the Law of Reflection, respectively. In Figure 6, students are asked which point (A-D) a line drawn through the two black points would hit. The item in Figure 7 asks students which letter each laser would hit.

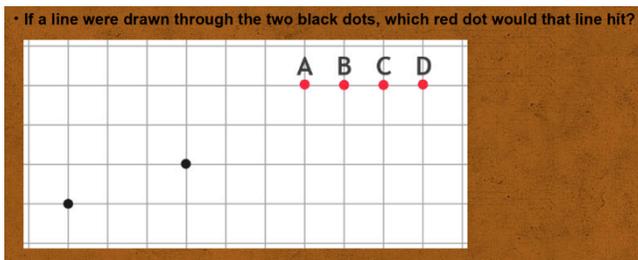


Figure 6: Sample Slope assessment item

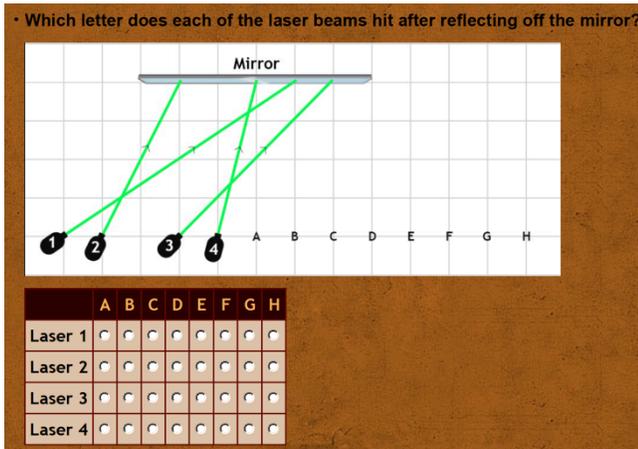


Figure 7: Sample Law of Reflection assessment item

These analyses are limited to the 9 pre and 10 post items focused on slope and the Law of Reflection. These pre- and post-assessment items had good internal consistency (Cronbach's alpha was 0.70 (pre) and 0.73 (post)). To account for the different number of items, we used the percentage of items answered correctly in the analyses. Students answered an average of 53 percent of the pre assessment items and 59 percent of the post assessment items correctly. Students in Bridge classrooms, however, answered significantly fewer questions correctly on both the pre- and post-assessment than students in Game Only classrooms ( $F=19.2, 1, 132 \text{ d.f.}, p<0.01$ ). On average, students in Bridge classrooms answered 48 percent of the pre-assessment and 55 percent of the post-assessment items. In contrast, students in Game Only classrooms answered 58 percent of the pre assessment items and 63 percent of the post assessment items correctly.

## 7. RESULTS

Using the SPSS MIXED linear models procedure, HLM analyses began with an unconditional 3-level model with students, classrooms, and teachers using Restricted Maximum Likelihood (REML) and unstructured covariances. In the 3-level model, seven percent of the variation was at the teacher level. Triple that proportion of the overall variation was attributable to the classroom level. A 2-level unconditional model with students nested within classrooms was estimated. In that model, a statistically significant 34 percent of the variance in the post-assessment was attributable to classroom level variation.

Sets of covariates were added to the unconditional HLM model in this order:

- Set 1. Pre-assessment score (standardized)
- Set 2. Study Group (Bridge or Game Only)
- Set 3. Student gender (1=Female)

Set 4. Classroom Level Characteristics: Whether or not they were enrolled in class in which more than half of the students completed the study (1=Yes); whether or not they were enrolled in an AP/Honors science class (1=Yes)

Set 5. In-game measures of implicit understanding—% Placement Errors, % Rotation Errors, and % Puzzle Errors (all standardized)

Set 6. Gameplay duration (>1 hour vs. not) and highest level reached (Level 22 vs. not)

Only statistically significant covariates were retained in the HLM model presented in this paper. Sets 3, 4, and 6 had no significant results, meaning student gender, Honors/AP status, gameplay duration and highest level reached were not significantly related to changes in pre-post assessment scores.

The model with the in-game measures of implicit understanding of slope and the Law of Reflection was a significantly better fit than the model without those measures ( $X^2(3 \text{ df}, N=317), 6.76, p<0.10$ ). The best-fitting HLM model, which accounts for 33 percent of the variation at the classroom level, is presented in Table 3. Overall, after accounting for students' performance on the pre-assessment, students who exhibited more Placement and Rotation errors while playing the game performed more poorly on the post than students with lower science error rates.

Table 3: Best-fitting HLM model

Parameter	Est.	Std Err	df	Sig.	95% Confidence Interval	
					Lower	Upper
Intercept	0.10	0.12	24	0.43	-0.15	0.35
Pre-Assessment <sup>1</sup>	0.35	0.05	320	0.00	0.26	0.45
Bridge (vs. Game Only)	-0.17	0.17	25	0.33	-0.52	0.18
%Placement Errors <sup>1</sup>	-0.08	0.05	304	0.09	-0.17	0.01
%Rotation Errors <sup>1</sup>	-0.17	0.05	320	0.00	-0.26	-0.07
%Puzzle Errors <sup>1</sup>	0.00	0.04	310	0.93	-0.09	0.08

<sup>1</sup>Standardized

The intercept coefficient represents the estimated outcome for male students who scored at the mean level of the pre-assessment, were in the Game Only group, were not in a Honors/AP class, and had mean levels of Placement and Rotation Errors. These students would score 0.07 standard deviations below the mean post-assessment score. The Pre-Assessment coefficient reflects the change in number of standard deviations of the post-assessment for every increase of 1 standard deviation on the pre-assessment. For every standard deviation increase on the pre-assessment, students would be expected to score 0.35 standard deviations higher on the post-assessment. Students in Bridge classes scored 0.17 standard deviations lower on the post-assessment than students in Game Only classes—a non-significant difference. There was no significant difference between Bridge and Game Only groups in their pre-post gains. This may be because Game

Only classroom instruction provided lab experiences with lasers that mirrored what Bridge classrooms did with *Quantum Spectre*, providing comparable experiences and similar gains.

Students whose placement or rotation error rate was one standard deviation above the mean, however, had post-assessment scores 0.08 and 0.17 standard deviations below the mean, respectively. There was no impact of puzzle errors. Interactions between study group (Bridge vs. Game Only) and gameplay errors were examined but none significantly improved the fit of the HLM model, suggesting the impact of these errors was the same across study groups.

## 8. DISCUSSION & IMPLICATIONS

Hierarchical linear modeling suggest a direct negative relationship between science-related gameplay errors and implicit science learning—players making errors consistent with a lack of implicit science understanding performed worse than players not making as many of those errors. Educators can use this information as a real-time, or reflective, formative assessment tool. This could be very useful in a class where students are playing a learning game, individually or in groups, while the teacher has an app that alerts them to which students are struggling and may need attention. A more comprehensive dashboard they can use after class might show them overall progress of their class and trends that inform how the next lessons are planned. Teachers might also use a dashboard to monitor their students' game-based learning as they play at home or with friends outside of class. The ability to validly infer implicit science learning from the digital records of game activity makes this all possible.

## 9. ACKNOWLEDGMENTS

We thank the teachers and students who participated in this study. This research was funded as part of a NSF DRK12 grant 1119144 to develop and study the Leveling Up games. We gratefully acknowledge the rest of the EdGE team: Erin Bardar, Barbara MacEachern, Jamie Larsen, and Katie Stokinger for their design and outreach efforts.

## 10. REFERENCES

- [1] Eagle, M., Rowe, E., Hicks, A., Brown, R., Barnes, T., Asbell-Clarke, J., & Edwards, T. (2015, October). Measuring implicit science learning using networks of player-game interactions. Presented at the annual ACM Symposium on Computer-Human Interaction in Play, London.
- [2] Lenhart, A., Purcell, K., Smith, A., & Zickuhr, K. (2010). *Social Media & Mobile Internet Use Among Teens and Young Adults*. Washington, DC: Pew Research Center.
- [3] National Research Council (2011). *Learning Science Through Computer Games and Simulations*. M.A. Honey and M.L. Hilton (Eds.), Wash., DC: National Academies Press.
- [4] Shute, V. & Ventura, M. (2013). *Stealth assessment: Measuring and supporting learning in video games*. MIT Press.
- [5] Steinkuehler, C., & Duncan, S. (2008). Scientific Habits of Mind in Virtual Worlds. *Journal of Science Education and Technology*, 17(6), 530–543.
- [6] Eagle, M., Peddycord, B., Hicks, A., Barnes, T. (2015, April). Exploring networks of problem-solving interactions. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge (LAK '15)*, Poughkeepsie, NY, USA, pp. 21-30.
- [7] Eagle, M., Barnes, T. (2014, July). Exploring differences in problem solving with data-driven approach maps. In proceedings of *Educational Data Mining (EDM2014)*, London, UK, pp. 76-83.
- [8] Polanyi, M. (1966). *The Tacit Dimension*. London: Routledge. (University of Chicago Press. ISBN 978-0-226-67298-4. 2009 reprint).
- [9] Vygotsky, L.S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, Mass.: Harvard University Press.
- [10] McCloskey, M. (1983). Intuitive Physics. *Scientific American*, 248(4), 122–130.
- [11] Minstrell, J. (1982). Explaining the “at rest” condition of an object. *The physics teacher*, 20(1), 10–14.
- [12] diSessa, A.A. (1993). Toward an Epistemology of Physics. *Cognition and Instruction*, 10(2/3), 105–225. doi: 10.2307/3233725.
- [13] GlassLab (2014). Psychometric Considerations In Game-Based Assessment. Institute of Play. From: <http://www.instituteofplay.org/work/projects/glasslab-research/>
- [14] Shute, V., Ventura, M. & Kim, J. (2013). Assessment and Learning of Qualitative Physics in Newton's Playground. *The Journal of Educational Research*, 106 (6.), 423–430, doi:10.1080/00220671.2013.832970.
- [15] Clark, D.B., Nelson, B., Chang, H., D'Angelo, C.M., Slack, K. & Martinez-Garza, M. (2011). Exploring Newtonian mechanics in a conceptually-integrated digital game: Comparison of learning and affective outcomes for students in Taiwan and the United States. *Computers and Education*, 57(3), 2178–2195.
- [16] Thomas, D., & Brown, J.S. (2011). *A New Culture of Learning: Cultivating the Imagination for a World of Constant Change*. Lexington, KY: CreateSpace.
- [17] Plass, J., Homer, B.D., Kinzer, C.K., Chang, Y.K., Frye, J., Kaczetow, W., Isbister, K., & Perlin, K. (2013). Metrics in Simulations and Games for Learning. In M. Seif El-Nasr, A. Drachen, & A. Canossa (Eds.), *Game Analytics: Maximizing the Value of Player Data* (694-730). Springer-Verlag.
- [18] Feng, M., & Heffernan, N.T. (2006). Informing teachers live about student learning: Reporting in the assistment system. *Technology Instruction Cognition and Learning*, 3(1/2), 63.
- [19] Lederman, L. C., & Fumitoshi, K. (1995). Debriefing the Debriefing Process: A new look. In D. C. K. Arai (Ed.), *Simulation and gaming across disciplines and cultures*. London: Sage Publications.
- [20] Andres, J.M.A.L., Andres, J.M.L., Rodrigo, M.M.T., Baker, R.S., & Beck, J.B. (2015) An investigation of eureka and the affective states surrounding eureka moments. To appear in *Proceedings of the 23rd International Conference on Computers in Education*.
- [21] Hicks, A., Eagle, M., Rowe, E., Asbell-Clarke, J., Edwards, T., & Barnes, T. (2016, April). Using game analytics to evaluate puzzle design and level progression in a serious game. Paper presented at the 6th international Learning Analytics & Knowledge conference, Edinburgh, U.K.

# Assessing Student-Generated Design Justifications in Engineering Virtual Internships

Vasile Rus, Dipesh Gautam,  
Department of Computer Science  
Institute for Intelligent Systems  
The University of Memphis  
Memphis, TN 38152  
{vrus,dgautam}@memphis.edu

Zachari Swiecki, David W.  
Shaffer  
Department of Educational Psychology  
University of Wisconsin-Madison  
Madison, WI 53706  
{swiecki,dws}@wisc.edu

Arthur C. Graesser  
Department of Psychology  
Institute for Intelligent Systems  
The University of Memphis  
Memphis, TN 38152  
graesser@memphis.edu

## ABSTRACT

Engineering virtual internships are simulations where students role play as interns at fictional companies, working to create engineering designs. To improve the scalability of these virtual internships, a reliable automated assessment system for tasks submitted by students is necessary. Therefore, we propose a machine learning approach to automatically assess student generated textual design justifications in two engineering virtual internships, *Nephrotex* and *RescuShell*. To this end, we compared two major categories of models: domain expert-driven vs. general text analysis models. The models were coupled with machine learning algorithms and evaluated using 10-fold cross validation. We found no quantitative differences among the two major categories of models, domain expert-driven vs. general text analysis, although there are major qualitative differences as discussed in the paper.

## Keywords

Virtual internships, machine learning, auto-assessment, epistemic frame theory

## 1. INTRODUCTION

In virtual internships, students play the role of interns in a virtual training environment. In engineering virtual internships, such as *Nephrotex* (NTX) and *RescueShell* (RS), students research and create multiple engineering designs [1]. As part of their design process, they regularly submit written work in the form of electronic engineering notebooks that are assessed by human judges. This human assessment is labor intensive, time consuming, and error-prone under certain circumstances such as time pressure. Furthermore, prior work has suggested that the reliability of human assessments can vary depending on the traits of the assessor, their experience, and the types of problems being assessed [14]. Thus, an automated assessment method that could provide efficiency in terms of time and cost as well as improved reliability is much needed. Our work presented here constitutes a step in this direction.

In the present study, we explored various models for automatically assessing notebooks in the engineering virtual internships NTX and RS. The content of these notebooks varies; however, in this study we focus on only one type of notebook in which students must justify their engineering designs by typing a short, free-text justification.

We have experimented with models that emulate an expert analysis of the student notebook entries as well as models derived from general textual analysis features. It should be noted that our work differs from previous attempts which rely on a semantic similarity approach, i.e. measuring how semantically close a student-generated response is to an ideal, expert-generated response as in [6].

The domain expert-driven models incorporate theoretically driven, content-based features identified by human experts such as “referencing any performance parameter such as cost”, which is a general design feature because it applies to all engineering designs in NTX and RS, or “indicating the power source”, a feature specific to the concrete task of designing an exoskeleton, which was the focus of the RS internship and not NTX. A challenge with the domain expert-driven models is that the features are specific to either the type of task, e.g. engineering design, or the concrete task itself, e.g. design an exoskeleton. This results in a scalability issue as these models must be redesigned manually by domain experts when moving to a new domain, new type of task, and/or a new concrete task. However, the net theoretical advantage of these domain expert-driven models is that they are tailored to the task at hand and therefore are expected to yield very good performance. These models also afford the ability to create automatic and tailored feedback to students given their task-specific diagnostic capabilities.

The other category of models that we used rely on general text analysis features inspired from previous work on automated essay scoring [2,5,13] and text analysis software tools such as Coh-Metrix [4] and LIWC [7]. For instance, in automated essay scoring the length in words of the essay, i.e. the number of all word occurrences or word tokens, is by far the best predictor of essay quality. Coh-Metrix is a software package that calculates the coherence of texts in terms of co-reference, temporal cohesion, spatial cohesion, structural cohesion, and causal/intentional cohesion. LIWC (Linguistic Inquiry and Word Count) uses a word count strategy to characterize texts along a number of dimensions that include standard language categories (e.g., articles, prepositions, pronouns), psychological processes (e.g., positive and negative emotion word categories), and traditional content dimensions (e.g., sex, death, home, occupation).

The key advantage of the general text analysis models is that they are generally applicable across types of tasks, specific tasks, and domains. In addition, the general text analysis features are relatively cost-effective and easy to derive from the data compared

to features derived by domain experts, which require (significantly more) human time and effort.

In this paper, we explore the predictive power of the two major categories of models mentioned above, domain-expert vs. general text analysis, in conjunction with a number of machine learning algorithms such as decision trees, naïve Bayes, Bayes Nets, and logistic regression. Furthermore, we employed an ensemble of classifiers approach in order to boost the performance of individual models. We conclude the paper with a qualitative assessment of the relative benefits of the proposed models for virtual internships by considering their predictive value, the labor involved in their development, and their ability to provide interpretable assessments for students.

## 2. BACKGROUND

We review in this section prior work on assessing students' open-ended responses with an emphasis on prior work in the area of educational technologies.

Automated essay scoring systems [2,5,13] have been developed for more than two decades as a way to tackle the costs, reliability, generality, and scalability challenges associated with assessing student generated open-ended responses to essay prompts. There are a number of systems available for automated essay scoring, some of which are commercial. It is beyond the scope of this paper to offer a thorough review of the work in this area. We limit ourselves to noting that the focus on automated essay scoring is on the argumentative power of an entire essay while in our case the focus is on required (design) items that must be present in paragraph-like justifications. This entails that style and higher-level constructs such as rhetorical structure are less important in our task as opposed to the essay scoring task and that factors that focus more on content measures are highly important. Given these differences and the fact that the two most predictive factors of essay quality are also content related, we included in our models the following two features: word count, i.e. total number of word occurrences or tokens in student justifications, and content word count, i.e. the total number of content word occurrences (nouns, verbs, adjectives, and adverbs).

Directly relevant to our study is previous work by Rus, Feng, Brandon, Crossley, and McNamara [8] who studied the problem of assessing student-generated paraphrases in the context of a writing strategy training tutoring system. One of the strategies in this tutoring system is paraphrasing. As the system is supposed to prompt students to paraphrase and then provide feedback on their paraphrases, Rus and colleagues collected a large corpus of student-generated paraphrases and analyzed them along several dozen linguistic dimensions ranging from cohesion to lexical diversity obtained from Coh-Metrix [4]. There are significant differences between their work and ours. First, we deal with justifications which can vary in length from a few words to a full paragraph as opposed to explicitly elicited paraphrases of target sentences. Second, we do use extra features to build our models besides the Coh-Metrix indices. Third, we assess the student generated justifications as acceptable or unacceptable (i.e., correct or incorrect). We could eventually investigate finer levels of correctness, e.g. on a scale from 1-5, which we plan to do as part of our future work.

Williams and D'Mello [15] worked on predicting the quality of student answers (as error-ridden, vague, partially-correct or correct) to human tutor questions, based on dictionary-based

dialogue features previously shown to be good detectors of cognitive processes (cf. [15]). To extract these features, they used LIWC (Linguistic Inquiry and Word Count; [6]), a text analysis software program that calculates the degree to which people use various categories of words across a wide array of texts genres. They reported that pronouns (e.g. I, they, those) and discrepant terms (e.g. should, could, would) are good predictors of the conceptual quality of student responses. Like Williams and D'Mello, we do use LIWC to analyze student notebooks' justifications. Furthermore, we employ expert-identified features and features from Coh-Metrix and automated essay scoring.

Prior work by Rus, Lintean, and Azevedo [9] investigated the performance of several automated models designed to infer the mental models of students participating in an intelligent tutoring system (ITS). The ITS was designed to teach students self-regulatory processes while they were learning about science topics such as the human circulatory system. Rus and colleagues used two methods, a content-based method and a word-weighting method, to derive features for their models. While our present work does not investigate models using word-weighting methods, we do investigate models using content-based features.

The content-based features used by Rus and colleagues included a taxonomy of relevant biology concepts derived by human experts, expert annotated pages of content from the ITS, and expert-generated paragraphs. In the present study, the content-based features, or domain-expert (DE) features, we used consist of discourse codes developed by human experts. Discourse codes indicate the presence or absence of specific concepts in student talk, or in this case, student written work. The DE features were developed through a grounded analysis of student design justifications collected from engineering virtual internships [3].

The learning that occurs in engineering virtual internships can be characterized by epistemic frame theory. This theory claims that professionals develop epistemic frames, or the network of skills, knowledge, identity, values, and epistemology that are unique to that profession [11]. For example, engineers share ways of understanding and doing (knowledge and skills); beliefs about which problems are worth investigating (values), characteristics that define them as members of the profession (identity), and a ways of justifying decisions (epistemology). In this study, we used epistemic frame theory to guide the development of the DE features. In prior work, elements of the engineering epistemic frame have been operationalized as discourse codes and used to assess engineering thinking in virtual internships [1]. In this study, the DE features we identified correspond to elements of the engineering epistemic frame that relate to justifying design decisions. The presence or absence of these features in a student's written work thus represents elements of the engineering epistemic frame that are present or lacking.

In sum, we used some of the features described by the above researchers in our work, such as word count, as well as novel features, e.g. features based on the engineering epistemic frame.

## 3. ENGINEERING VIRTUAL INTERNSHIPS

In this study, we examined student written work collected from the engineering virtual internships, *Nephrotex* (NTX) and *RescueShell* (RS). In NTX, students work in teams to design filtration membranes for hemodialysis machines, while in RS,

student teams design the legs of a mechanical exoskeleton used by rescue workers.

All interactions in virtual internships take place via a website in which students communicate with their teams using email and chat. During the internships, students research and create engineering designs in two cycles. In each cycle, students design five prototypes and later receive performance results for each prototype which they have to analyze and interpret.

During their design process, students submit records of their work via electronic notebook entries for each substantive task they complete, including summarizing research reports and justifying design decisions. The expectations of notebook entries are outlined in prompts, which students receive via email in the virtual internship website. Each notebook that students submit is divided into notebook sections, i.e., separate text fields for items that are defined by the email prompts. In this study, we analyzed notebook sections in which students provided justifications for their prototype design decisions.

Once students complete each notebook section, they submit the notebooks to trained human raters for assessment. In the fiction of the virtual internships, these raters play the role of more senior employees in the company who act as *mentors* to the students. The role of the mentors is to answer student questions and lead team discussions, in addition to assessing student work.

Once a mentor receives a notebook, they assess each section as acceptable or unacceptable using provided rubrics. The assessment system used by the mentors automatically generates pre-scripted feedback corresponding to the assessment given to each section. Currently, this feedback is generic in the sense that it does not respond to the particulars of a student's response. For example, an assessment of unacceptable on a notebook section requiring a summary generates feedback that (1) informs the student that the section was unacceptable, (2) reminds them of the content they were asked to summarize, and (3) points them to the documents they were asked to summarize. This automated feedback does not inform the student exactly why the section was rated as unacceptable. However, the mentor does have the option to compose specific feedback for the student if they wish.

Our work here moves us towards a more automated and student-tailored assessment and feedback mechanisms which could have significant impact on the economy of scaling virtual internships to all students, anytime, anywhere via Internet-connected devices.

## 4. EXPERIMENTS AND RESULTS

We describe first the data set we used in our experiments before presenting the experiments and results obtained with the models.

### 4.1 Data Set

In this study, we analyzed notebook sections from the NTX and RS virtual internships in which students justified their engineering design decisions. In these notebook sections, students were required to include the design input choices they selected—that is, their design specifications, and a justification explaining why this design was chosen for testing.

Mentors assessed these notebook entries as acceptable or unacceptable in real-time during the virtual internship using the following rubric:

1. Listed their design specifications

2. Included a justification referencing at least one design specification.

Acceptable justification may include:

1. Prioritizing attributes
2. Referencing internal consultant requests
3. The performance of a design specification on a specific attribute
4. Experimental justifications (e.g., holding design specifications constant)

To select data for this study, we randomly sampled 298 justification sections from 20 virtual internship sites, i.e. datasets corresponding to 20 schools where the virtual internships were implemented. Twelve were NTX sites and eight were RS sites. Of the 298 justifications sampled, 146 were from NTX and 152 were from RS. Students were given the same prompts for justification sections in NTX and RS. In addition, the same rubrics were used by raters in NTX and RS. Thus, we combined data from RS and NTX to train our models.

As described above, justification sections were originally assessed by mentors during the virtual internship in real time. The mentors were trained to assess notebook section, but they were not experts in the domain of engineering or the content of the virtual internships. In addition, they had to assess notebook sections under time constraints and while completing their other responsibilities as a mentor. For example, they could have to respond to student questions via chat while assessing. Thus, to obtain potentially more valid and reliable assessments for model training, the justification sections in this study were re-assessed by more experienced raters that did not face the constraints placed on the mentors. We found that the agreement between the human mentors and our experienced raters on the 298 student justifications we used in this work was  $\kappa = 0.271$ . This value is very low, indicating that mentors' assessments are not reliable, as we suspected.

Each justification section was re-assessed by two new raters, benchmark rater 1 (BE1) and benchmark rater 2 (BE2). BE1 had over two years of experience rating notebook sections from virtual internships and had contributed to the content development of both NTX and RS. BE1 was thus considered an expert rater for the purposes of this study. BE2 was a less experienced rater trained to assess justification sections. BE1 and BE2 assessed all 298 justification sections using the rubric above and agreed on one final judgement (acceptable or unacceptable) for each justification. Their inter-annotator reliability as measured by kappa was 0.767. Table 1 includes examples of notebook sections from NTX assessed as acceptable and unacceptable by the benchmark raters. About 73% of the instances in the data set were rated positively by the BEs. The distribution of positive and negative instances is shown in Table 2.

### 4.2 Feature Selection

As already mentioned, we focused on two major categories of models: models that rely on domain-experts (DE) versus models that rely on more general textual analysis features. We developed the DE features through a grounded analysis [3] of a sample of 98 justification sections. These features were developed by two researchers who re-assessed the sample and developed discourse codes corresponding to what they attended to while assessing. Next, we automated these codes using the *nCoder*, a tool for developing and validating automated discourse codes that relies

on authoring targeted regular expressions for each of the expert-identified codes [12]. These codes were included as features in our models (see Table 3 for descriptions).

**Table 1. Example of Acceptable and unacceptable notebooks from the virtual internship *Nephrotex***

Notebook entry	Assessment
<i>Design Specifications: PAM, Vapor, Negative Charge, 4 % Justification: This prototype was altered slightly from the original with this material by changing from 2% CNT to 4%. This is an attempt to increase reliability without hindering flux or blood cell reactivity.</i>	Acceptable
<i>Design Specifications: PAM, Vapor, Negative Charge, 2.0 Justification: These specifications ran best for PAM material</i>	Unacceptable

**Table 2. Distribution of human-ratings in the 298 instances.**

Human Rating	#Instances
Acceptable	217
Unacceptable	81
Total	298

The general textual analysis features were further divided by their source into the following three categories: features inspired from automated essay scoring (ES) research, features obtained with the automated tool for textual analysis Coh-Metrix, and features obtained with the automated tool for textual analysis LIWC. This categorization of the general textual analysis features is needed for several reasons. First, the various sources capture different aspects of a text. Second, this categorization allows us to conduct ablation studies in which we assess the contribution of each major category of features to solving the task at hand. It should be noted that there is overlap among the features from various groups/sources. For instance, the WC (LIWC), DESWC (Coh-Metrix), and Word\_Count (DE) features are all counts of white-spaces in a target text, i.e. justifications in our case. These features are slightly different from the token Count feature in the ES group which counts number of tokens after applying the Stanford tokenizer tool. Similar features will not end up in the same models if they correlate highly, as explained next.

Not all features have equal predictive power and having redundant or irrelevant features can decrease the performance of the models. Therefore, we had a feature selection step keeping features that have low correlation with each other ( $<.70$ ). When two features in a model had a correlation greater than  $.70$  of them was dropped. For instance, from the LIWC and Coh-Metrix groups of features the features selected via this process were: WC, SIXLTR, adverbs, verbs, DESSC, DESSL, DESSLd, PCNARz, PCCONNP (See Table 3 for descriptions). The feature selection step was needed given that we worked with various machine learning algorithms, some of which do not have a feature selection process linked to them, e.g. the stepwise variable selection in some regression implementations.

### 4.3 Results

We experimented with the proposed models in conjunction with a number of classification algorithms including decision trees, naïve Bayes, Bayes Nets, and logistic regression. We present here the

results obtained with the logistic regression classifier as it yielded the best results overall. The models were validated using 10-fold cross validation. Performance was measured using standard measures such as accuracy, false positive rate, precision, recall, F-measure, and kappa statistic. The false positive rate, the percentage of true negatives predicted as positives, is of special interest because it gives us an idea of how many justifications are deemed correct when in fact are not, by a particular method. That is, it indicates how many opportunities for feedback a specific method might miss as a justification deemed correct means there is no need for specific feedback to improve it. The evaluation results are shown in Table 4. We focus next on the most important model comparisons due to space constraints, e.g. we do not show results when combining two groups of features.

We started with models that included features from only one group, i.e. the individual feature group models shown in rows 1-4 in Table 4, selected the best such model and then added, sequentially, features from the other groups in batches, where each batch contained the selected features in one group. This procedure, also known as an ablation study in machine learning, allows to see what we gain if we add a group of features to a model that already contains feature from one or more groups. From Table 4, we infer that the ES and Coh-Metrix individual models are the best as they have slightly higher accuracy in prediction (85.23% for ES and 85.23% for Coh-Metrix) compared to other two individual feature groups. Also their kappas are the highest among the models with only one group of features.

In row 5, we show the results when combining all general text analysis features: ES, LIWC, and Coh-Metrix. As already mentioned before, we are directly interested in comparing the domain expert-driven model, derived from the DE features, with the model in row 5 that includes all the general text analysis features from the ES, LIWC, and Coh-Metrix groups. As we notice, these two qualitatively different models have very similar performance across all performance measures.

In addition to developing the above models from subsets of features, we used ensembles of 3 individual and combined models, respectively, in conjunction with a majority voting mechanism. For instance, if 2 or 3 out of 3 models predicted a justification as *accepted* then the final prediction for the instance was *accepted*. We experimented with voting in two different ways: (1) we used the best 3 models from the individual or combined groups of features; (2) we used the weakest 3 models obtained with any combinations of features from individual and combined groups of features; this latter case is based on results from statistics that show that combining weak classifiers should result, in general, in better performance relative to the performance of each of the weak classifiers. Both types of ensembles (weakest versus best) yielded in the best cases similar accuracies of  $\sim 86\%$  and similar performance across all the other performance measures. The false positive rate of the weakest combined model ensemble was lowest.

## 5. CONCLUSIONS

In this paper, we experimented with multiple models designed to automatically assess notebook sections from engineering virtual internships. In particular, we developed models to assess notebook sections in which students justified design decisions. All models performed very well with good and very good kappa scores (kappas scores of 0.6-0.8 are considered very good)

**Table 3. Descriptions of the some features used in the proposed models (not all shown due to space constraints).**

Features	Description
<b>LIWC</b>	
Word Count	<i>Word Count</i> (WC; Total number of words in text), <i>Token Count</i> (TC; Number of unique words in text),
Type Token Ratio	<i>Words &gt; 6 letters</i> (SIXLTR: total number of words greater than 6 letters) <i>Punctuations</i> <i>Ratio of TC and WC</i>
<b>Coh-Metrix</b>	
Lexical Component Counts	<i>DESPC</i> - Paragraph count, number of paragraphs; <i>DESSC</i> - Sentence count, number of sentences, <i>DESWC</i> - Word count, number of words
DESPL	<i>DESPL</i> - Paragraph length, number of sentences, mean; <i>DESPLd</i> - Paragraph length, number of sentences, standard deviation; <i>DESSLd</i> ; Sentence length, number of words, standard deviation;
Connectives Features	<i>PCCONNp</i> - the degree to which the text contains connectives such as adversative, additives and comparative connectives to express relations in the text.
Temporality Features	<i>PCTEMPz</i> - the temporality such as tense or aspect of the text; <i>SMTEMP</i> - temporal cohesion, measured by repetition score of tense and aspect
LDTTRa	Type token ratio of all words.
<b>Domain Expert (DE)</b>	
Exoskeleton Design Inputs	Control Sensor, Range of Motion, Power Source, Material, Actuator
Dialyzer Design Inputs	Process, Surfactant, Material, Carbon Nanotube Percentage
Attributes	Referencing any design attribute or performance parameter such as cost, reliability, etc.
Justification Features	<i>Balancing</i> - Justifying input choices by stating it made up for the weakness of another choice or by saying that another choice will balance out its weaknesses; <i>Client</i> - Justifying input choices by stating it would be good for the client or end user of the product; <i>Consultant.Requests</i> - Justifying input choices because the results meet or are expected to meet internal consultants' requests; <i>Evaluation</i> - Justifying input choices by evaluating the performance of the inputs
<b>Essay Scoring (ES)</b>	
Token Count	Count of word occurrences in the justification.
Content Word Count	Count of all content words (noun, adjective, verb, adverb) in the justification.

**Table 4. Performance evaluation results for various models.**

S.N.	Features	Accuracy	FP Rate	Precision	Recall	F-Measure	Kappa
1	ES	85.2349	0.2490	0.850	0.8520	0.8510	0.6181
2	LIWC	83.2215	0.2950	0.8270	0.832	0.8290	0.5591
3	Coh-Metrix	85.2349	0.2950	0.8480	0.8520	0.8460	0.5991
4	DE	83.2215	0.3020	0.8270	0.8320	0.8280	0.5555
5	ES+LIWC+Coh-Metrix	83.8926	0.2920	0.8340	0.8390	0.8350	0.5733
6	LIWC + DE + Coh-Metrix + ES	81.8792	0.3000	0.8150	0.8190	0.8170	0.5314

indicating that they are much better than chance predictions. Our results show that, in this context, the predictive value of models using only the general text analysis features is comparable to the predictive value of a model using only the DE features (a McNemar's test on paired nominal data revealed no significant difference between the two models' prediction).

In particular, the ES group of features is the best predictor of students' justifications quality. When other groups of features are added to the individual ES model, the results do not improve significantly. The fact that the ES features are so good is not

surprising. Word count, or essay length, which is one of the features in the ES group, is known as being the best predictor of essay quality in automated essay grading [6,10]. Also, the Coh-Metrix group of features are a good predictor of the quality of students' justifications.

It is important to note, however, that the predictive power of a model is only one dimension for evaluating the utility of automated assessment models in learning environments like virtual internships. We suggest that developmental cost and interpretability of the models are also valuable dimensions to

consider. Of the models presented above, those using only the general text analysis features have the lowest developmental cost. Moreover, these features are generally applicable across types of tasks, specific tasks, and domains. In contrast, models containing the DE features have a relatively high developmental cost because their features required the time and expertise of humans to develop. We do note that the DE features described in this paper were automated. Thus, they can readily be applied to more justification sections from engineering virtual internships. However, these DE features are specific to this context and are likely not generalizable outside of engineering virtual internships.

The utility of these automated assessment models lies in implementing them in real-time during a virtual internship where they will be used to assess student work and either generate automatic feedback or suggest feedback for human mentors to give. For the models using only the general text analysis features, any potential feedback would be in terms of features such as word count or “narrativity” of the text that are not directly related to the domain-relevant content of the text. Those models using DE features, however, could potentially generate domain-relevant feedback in terms of what DE features were present and absent in the text. For example, if a student’s justification section fails to relate their design decisions to the requests of the company’s internal consultants, that is, it lacks the “Consultant Requests” DE feature, feedback could be suggested to the mentor or provided automatically to the student informing them of this missing information and suggesting ways to include it. Thus, in terms of ease of interpretation, those models using only the general text analysis features have a relatively low ease of interpretation compared to those models that include the DE features.

In this context, we then suggest the use of the best predictive model to assess the overall quality of justifications in engineering virtual engineering internships, and subsequently use the DE-based model to identify potential domain-specific missing parts in an unacceptable justification in order to provide direct feedback to the student or at least make suggestions to human mentors regarding possible weak aspects of the justification. This approach balances the tradeoffs between generality and reliability versus domain and task specific diagnostic capabilities.

We plan to further improve the predictive power, generality, and diagnostic capabilities of our models. For instance, we are considering unsupervised methods to automatically detect domain specific codes that could be used as features in our DE models. Furthermore, we are considering unsupervised topic detection in student-generated justification as a way to generalize the applicability of our models to other domains and types of tasks.

## 6. ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation (DRL-0918409, DRL-0946372, DRL-1247262, DRL-1418288, DUE-0919347, DUE-1225885, EEC-1232656, EEC-1340402, REC-0347000), the MacArthur Foundation, the Spencer Foundation, the Wisconsin Alumni Research Foundation, and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison. The opinions, findings, and conclusions do not reflect the views of the funding agencies, cooperating institutions, or other individuals.

## 7. REFERENCES

- [1] Chesler, N. C., Ruis, A. R., Collier, W., Swiecki, Z., Arastoopour, G., & Shaffer, D. W. (2015). A novel paradigm for engineering education: Virtual internships with individualized mentoring and assessment of engineering thinking. *Journal of biomechanical engineering*, 137(2).
- [2] Dikli, S. (2006). An Overview of Automated Scoring of Essays. *Journal of Tech., Learning, and Assessment*, 5(1).
- [3] Glaser, B. G., & Strauss, A. L. (2009). The discovery of grounded theory: Strategies for qualitative research. Transaction Publishers.
- [4] Graesser, A. C., McNamara, D. S., Louwerse, M. M., and Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36, 2(2004), 193-202.
- [5] Leacock, C., and Chodorow, M. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37, 4(2003), 389-405.
- [6] Mohler, M., and Mihalcea, R. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the EACL* (Athens, Greece, March, 2009).
- [7] Pennebaker, J. W., Francis, M. E., and Booth, R. J. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71, (2001), 2001.
- [8] Rus, V., Feng, S., Brandon, R., Crossley, S., and McNamara, D.S. (2011). A Linguistic Analysis Of Student Generated Paraphrases. In R. C. Murray and P.M. McCarthy (Eds.), *Proceedings Of The 24th International Florida Artificial Intelligence Research Society Conference*. Menlo Park, CA: AAAI Press.
- [9] Rus, V., Lintean, M., Azevedo, R. (2009). Automatic Detection of Student Mental Models During Prior Knowledge Activation in MetaTutor. *Proceedings of the 2nd International Conference on Educational Data Mining*. Cordoba, Spain.
- [10] Rus, V., Niraula, N. (2012). Automated Detection of Local Coherence in Short Essays Based on Centering Theory", *CICling 2012*, March 11-17, IIT Delhi, India.
- [11] Shaffer, D. W. (2006). *How computer games help children learn*. Macmillan.
- [12] Shaffer, D.W., Borden, F., Srinivasan, A., Saucerman, J., Arastoopour, G., Collier, W., Ruis, A.R., & Frank, K.A. (2015). *The nCoder: A technique for improving the utility of inter-rater reliability statistics*. Epistemic Games Group Working Paper 2015-01. University of Wisconsin–Madison.
- [13] Shermis, M.D. & Burstein, J. (2003). *Automated Essay Scoring: A Cross Disciplinary Perspective*. Lawrence Erlbaum Associates, Mahwah (2003).
- [14] Tisi, J., Whitehouse, G., Maughan, S., & Burdett, N. (2013). *A review of literature on marking reliability research* (Report for Ofqual). Slough: NFER.
- [15] Williams, C., & D’Mello, S. (2010). Predicting student knowledge level from domain-independent function and content words. In *Intelligent Tutoring Systems* (pp. 62-71). Springer Berlin Heidelberg.

# Tensor Factorization for Student Modeling and Performance Prediction in Unstructured Domain

Shaghayegh Sahebi  
Intelligent systems Program  
University of Pittsburgh  
Pittsburgh, PA  
shs106@pitt.edu

Yu-Ru Lin  
School of Information  
Sciences  
University of Pittsburgh  
Pittsburgh, PA  
yurulin@pitt.edu

Peter Brusilovsky  
School of Information  
Sciences  
University of Pittsburgh  
Pittsburgh, PA  
peterb@pitt.edu

## ABSTRACT

We propose a novel tensor factorization approach, Feedback-Driven Tensor Factorization (FDTF), for modeling student learning process and predicting student performance. This approach decomposes a tensor that is built upon students' attempt sequence, while considering the quizzes students select to work with as its feedback. FDTF does not require any prior domain knowledge, such as learning resource skills, concept maps, or Q-matrices. The proposed approach differs significantly from other tensor factorization approaches, as it explicitly models the learning progress of students while interacting with the learning resources. We compare our approach to other state-of-the-art approaches in the task of Predicting Student Performance (PSP). Our experiments show that FDTF performs significantly better compared to baseline methods, including Bayesian Knowledge Tracing and a state-of-the-art tensor factorization approach.

## Keywords

Tensor factorization, student modeling, predicting students performance, learning analytics

## 1. INTRODUCTION

The growth of Massive Open Online Courses (MOOC) has rapidly increased the volume of data on students' education and learning behavior. This abundance of data calls for approaches that can automatically make sense of such data, and that remove the need for manual handling of such massive amounts of data. Predicting students performance and modeling student knowledge are two of the tasks that help researchers to understand such data. The goal in predicting student performance (PSP), is to estimate if a specific target student can handle a learning material successfully – for example, whether the student can succeed or fail at solving a specific quiz. Student knowledge modeling aims to quantify or infer a student's knowledge at each moment in time in each of the possible skills (or concepts) the student

may have. The set of skills are defined either manually or automatically based on the learning materials.

Understanding students' attempt data through PSP and student knowledge modeling encourages teachers to design better courses, allows for targeted personalization of course pace, and provides more accurate automatic learning material recommendation to students. Hence, a primary focus in educational data mining literature is on predicting student performance and student knowledge modeling. For example, Bayesian Knowledge Tracing was one of the pioneering approaches that could predict the success or failure of students in solving problems [1].

Recently, other approaches, such as factorization models, have been used for PSP. For example, Performance Factor Analysis (PFA) [5] is another approach to PSP and cognitive modeling. PFA takes into account the effects of the initial difficulty of the skills (knowledge components) and prior successes and failures of a student at learning the skills associated with the current item. These approaches require prior knowledge of the overall domain model – the association between skills and learning material.

More recent approaches have sought to overcome this limitation by using latent factor approaches. For example, Thai-Nghe et al. experimented on a context-aware factorization algorithm, based on collaborative filtering approaches, in the relevant recommender system literature [9]. Sahebi et al. studied various methods of the educational data mining field with matrix and tensor factorization approaches, from the recommender systems literature for PSP [7]. Lan et al. used quantized matrix completion to predict students' performance in SPARFA-Lite [4]. This method solves a convex optimization problem and gives a global optimum solution.

Tensors, or multi-dimensional arrays, have been used in the literature to represent data on student attempts [6]. One of the main reasons that tensors are a suitable representation for modeling educational data is their seamless integration ability and flexibility in representing multiple dimensions of the data, such as students, questions, time, and topic structure. Another reason for using tensors is their capability for decomposing interactions in multi-dimensional data.

While various tensor decomposition models and algorithms already exist in the literature [3], the potential for versa-

file modeling of tensors in the educational data mining field is under-explored. Although previous tensor factorization models that have been used in the literature have resulted in comparable performance in the task of PSP [6, 8], they are not tailored to educational data. More specifically, these models are built for purposes other than educational data mining (such as recommender systems), and thus do not consider the characteristics of educational data mining challenges.

One of these challenges is increases in student knowledge that occurs while they interact with learning material. As the students learn through quizzes, readings, and other learning resources, they incrementally learn the underlying skills that are present in these resources. Thus, this amount of knowledge increase for a student depends on the material that the student is interacting with. The current tensor factorization approaches that are used for PSP in the literature do not model this interaction.

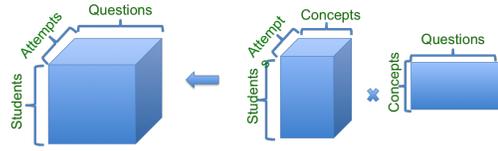
In this paper, we provide a solution to this problem by proposing a unique tensor factorization-based approach that can account for the constant learning of students. Our proposed tensor factorization model, called *feedback-driven tensor factorization*, directly models the increases in student knowledge by adding a feedback-based constraint on the previous student’s knowledge and the current learning material that a student is using. We compare our approach to Bayesian Knowledge Tracing and a baseline tensor factorization algorithm. Our experiments show the superior performance of our proposed approach, as compared to the baseline methods.

## 2. FEEDBACK-DRIVEN TENSOR FACTORIZATION (FDTF)

As mentioned in the introduction, the goal of our approach is to predict student performance while considering the fact that students are constantly learning. In order to achieve this goal, we represent student activities on learning material as a three-dimensional tensor  $\mathcal{Y}$ .

**Notations.** In this paper, tensors are represented by script letters, e.g.  $\mathcal{Y}$ ; Matrices are denoted by boldface capital letters, e.g.  $\mathbf{X}$ ; and vectors are represented by boldface lowercase letters, e.g.  $\mathbf{x}$ . In addition, we denote the  $i^{th}$  row of a matrix  $\mathbf{X}$  as  $\mathbf{X}_{i,:}$ , the  $j^{th}$  column as  $\mathbf{X}_{:,j}$ , and the entry  $(i, j)$  as  $\mathbf{X}_{i,j}$ .

Suppose that students are working with one resource type and are learning from it. To be more specific, suppose that  $m$  students are interacting with  $n$  quizzes, and that each student can have multiple attempts (at most  $l$ ) on each quiz. Then, we can represent the students’ attempt sequences on all quizzes as a tensor of size  $m \times n \times l$ . The  $k^{th}$  frontal slice of this tensor ( $\mathcal{Y}_{:, :, k}$ ) shows the success or failure of all students on all quizzes in their  $k^{th}$  attempt. To abbreviate, we use  $\mathcal{Y}_k$  to represent the  $k^{th}$  frontal slice of all tensors. Accordingly,  $\mathcal{Y}_{i, :, :}$  shows all the attempts of student  $i$  on all questions and  $\mathcal{Y}_{:, j, :}$  shows all attempts of all students on question  $j$ . We assume that each quiz consists of multiple (c) concepts (skills or knowledge components) and that the students should have some knowledge of these concepts in order to solve the quizzes that include such concepts. Some



**Figure 1: Phase 1: Decomposition of Student Performance into Student Knowledge and Concept-Map**

of the elements of  $\mathcal{Y}$  are unknown to us because not all of the students try all of the questions as many times. Based on these assumptions, we formulate the problem as a tensor factorization with two phases: the *prediction* phase and the *learning* phase.

In the prediction phase, we follow the assumption that students’ success or failure in quizzes depends on their knowledge and the concepts underlying those quizzes. In this phase, we decompose  $\mathcal{Y}$  into a tensor and a matrix: the tensor  $\mathcal{T}$  that shows the knowledge of students on the concepts at each of their attempts on the quizzes, and the matrix  $\mathbf{Q}$  that shows the concepts that are required to solve each quiz correctly. For each quiz  $j$ ,  $\mathbf{Q}_{:,j}$  shows the importance of each of the discovered concepts in it. Also,  $\mathcal{T}_{i,k,l}$  shows the knowledge of student  $i$  in concept  $k$  at the  $l^{th}$  attempt.

Based on this decomposition, we can estimate (predict) the unknown values of  $\mathcal{Y}$  using the multiplication of tensor  $\mathcal{T}$  and matrix  $\mathbf{Q}$ , as presented in Equation 1. Figure 1 gives an illustration of this decomposition.

$$\mathcal{Y} = \mathcal{T} \times \mathbf{Q} \quad (1)$$

We suppose that students learn by practicing the quizzes, and that the knowledge of students increases through this practice of the concepts. The learning phase of our tensor factorization approach models student learning, based on the quizzes that they choose to solve in each step. In order to do that, we construct a tensor  $\mathcal{X}$  that denotes when a student has or has not chosen to work on a specific problem at a specific time. Equation 2 shows how to build this tensor, based on  $\mathcal{Y}$ .

$$\mathcal{X}_{i,j,k} = \begin{cases} 1, & \text{if } \mathcal{Y}_{i,j,k} \text{ is observed} \\ 0, & \text{if } \mathcal{Y}_{i,j,k} \text{ is not observed} \end{cases} \quad (2)$$

In the learning phase, we assume that the amount of gained knowledge in each concept is a function of the student’s knowledge at the previous attempt, as well as the weight of concepts that are learned in the quiz that the student chooses to solve. Let  $f(\cdot)$  be such a function; then the gained knowledge at time  $t$  can be expressed as:

$$\mathcal{T}_t = f(\mathcal{T}_{t-1}, \mathcal{X}_t, \mathbf{Q})$$

Since we assume that knowledge of students grows over time, we should choose a monotonically increasing function for

$f(\cdot)$ . Also, to keep this knowledge increase from growing too large, this function should be bounded. Based on these assumptions, we model the knowledge growth of students as a logistic regression function that ranges between 0 (for no increase in the knowledge) to  $1 - \mathcal{T}_{t-1}$  (for a maximum increase in the knowledge). This allows us to have a bounded amount of knowledge that always stays between zero and one. To add to the flexibility of this function, and to account for different students' rate for learning from the quizzes, we add a factor  $\mu$  that controls the slope of the logistic regression function. The higher the learning rate ( $\mu$ ), the larger the knowledge increase and the faster the students reach a maximum state of knowledge. This increase can be seen in Equation 3.

$$\mathcal{T}_t = \mathcal{T}_{t-1} + \left( \frac{2(1 - \mathcal{T}_{t-1})}{1 + \exp(-\mu \mathcal{X}_t \mathbf{Q}')} - (1 - \mathcal{T}_{t-1}) \right), \quad (3)$$

which can be written as follows:

$$\mathcal{T}_t = 2\mathcal{T}_{t-1} + \frac{2(1 - \mathcal{T}_{t-1})}{1 + \exp(-\mu \mathcal{X}_t \mathbf{Q}')} - 1 \quad (4)$$

Based on this model, the more knowledgeable the student is in a concept, the less improvement she will obtain by practicing the same concepts again and again. The greatest increase in the student's knowledge happens when the student does not know the skills that are provided in the quiz. If we expand and simplify Equation 3, we achieve Equation 4. Since  $f(\cdot)$  is a monotonically increasing function, the estimated knowledge tensor ( $\mathcal{T}$ ) and domain model ( $\mathbf{Q}$ ) are both non-negative. This non-negativity is in accordance with assumptions in the educational domain: that the weight of each concept in each learning material cannot be negative and that the knowledge of students at any time and in any concept cannot be negative either.

Eventually, the matrix factorization includes solving Equations 1 and 4. Assuming that we have the values for  $\mathcal{X}_t$  and  $\mathbf{Q}$ , Equation 4 can be considered as a static update and we can only optimize Equation 1 iteratively and update the knowledge values in each iteration using Equation 4. To achieve this goal, we try to optimize for the least regularized estimation error of our observed tensor ( $\mathcal{Y}$ ) in Equation 5. Thus, our objective is to minimize the overall error, which is defined as:

$$\sum_{i=1}^t \|\mathcal{Y}_t - \mathcal{T}_t \mathbf{Q}\|^2 + \lambda (\sum_{i=1}^t \|\mathcal{T}_i\|^2 + \|\mathbf{Q}\|^2), \quad (5)$$

where  $\lambda$  is a regularization parameter. The last two terms are added to the error equation to regularize the values in tensor  $\mathcal{T}$  and matrix  $\mathbf{Q}$ . These two terms increase the sparsity of the knowledge and domain model by decreasing the values in these two factors, while preventing the factorization from being over-fit to the training data.

Since this method uses the iterative feedback loops and the two phases of prediction and learning, we name it Feedback-Driven Tensor Factorization (FDTF).

### 3. EXPERIMENTS

To assess the student performance prediction task, we compare the proposed FDTF model to a baseline tensor factorization algorithm that was introduced in previous rec-

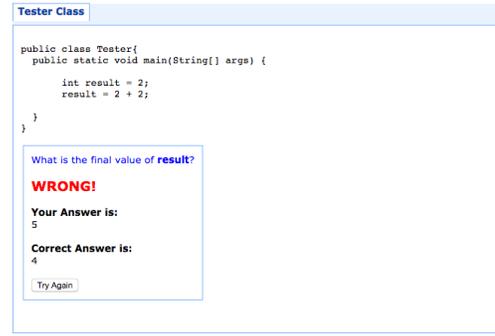


Figure 2: Screen-shot of QuizJet System

ommender system literature. This tensor factorization algorithm is called the Bayesian Probabilistic Tensor Factorization (BPTF) and models the temporal change of user interests on items [10]. We choose this model as a baseline because of its consideration for time sequencing and the common use of recommender systems algorithms in the educational data mining literature [7]. As our second baseline, we run the Bayesian Knowledge Tracing (BKT) algorithm on the data [1]. Since BKT requires a pre-defined set of concepts, we use the manually-labeled concepts that have been discovered by experts in this case.

The FDTF algorithm has two parameters that need to be tuned: the number of concepts ( $c$ ) and the learning rate of students ( $\mu$ ). We define these two parameters through cross-validation. Also, in our experiments, we set  $\lambda = 0.0001$ .

### 3.1 Dataset and Setup

We use student sequences of the QuizJet online self-assessment system to run our experiments [2]. This system produces parameterized Java quizzes based on a set of predefined templates. Hence, each student can repeat the same Java quiz, with different parameters, over and over again. The students submit their answer using a text box provided in the user interface and can receive immediate feedback. Figure 2 shows a screen-shot of this system in use.

The dataset was collected from the students who have taken a Java programming course from Fall 2010 to Spring 2013 (six semesters). The system was introduced in the class and students have voluntarily interacted with this system. The subject domain is organized by experts into 22 coherent topics. Each topic has several questions and each question is assigned to one topic. We use these sets of topics as the expert-labeled domain model in our experiments.

We experimented on 27,302 records of 166 students on 103 questions. The average number of attempts on each question is equal to three. Our dataset is imbalanced: the total number of successful attempts in the data equals 18,848 (69.04%) and the total number of failed attempts is 8454. We used a user-stratified 5-fold cross-validation to split the data so that the training set has 80% of the users (with all their records) randomly selected from the original dataset, while the remaining 20% of the users were retained for testing. In other words, 80% of students are in the training

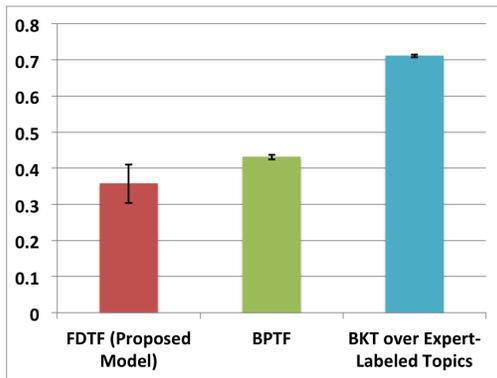


Figure 3: RMSE of Algorithms for Predicting Students Performance

set and we have all of their sequences. For the remaining students (20%) we use 20% of their data to predict the rest 80% of it. Eventually, we include  $80\% + 20\% * 20\% = 84\%$  of the whole dataset in the training set. We used the same set of data for all of the algorithms. We ran the experiments 3 times per stratification, and ended up with running each algorithm 15 times. The simple statistics of our dataset are shown in Table 1.

Table 1: Dataset Statistics

	Average	Min	Max
#attempts per sequence	3	1	50
#attempts per question	265	25	582
#attempts per student	165	2	772
#different students per question	87	7	142
#different questions per student	54	1	101

To find the best number of concepts ( $c$ ) in each of the automatic PSP algorithms, we use cross-validation.

### 3.2 Experimental Results

As explained in Section 3, we examine the prediction performance of the proposed FDTF algorithm and the baseline models BPTF and BKT with expert-labeled topics. We then compare the accuracy of these three approaches. Since the dataset is imbalanced with approximately 70% positive labels and 30% negative labels, we define predicted values that are greater than 0.3 as positive-label predictions and predicted values that are less than or equal to 0.3 as negative-label predictions. Figure 4 shows the accuracy of the mentioned algorithms. The red, green, and cyan bars represent the accuracy of FDTF, BPTF, and BKT. As we can see in this figure, although the accuracy of the baseline tensor factorization model (BPTF) is better than Bayesian Knowledge Tracing, it is significantly less than the accuracy of the proposed approach (FDTF). Eventually, FDTF performs significantly better than both of the baseline algorithms.

Although the task of predicting student performance is a binary classification task in this setting (predicting either failure or success for students), the Root Mean Squared Er-

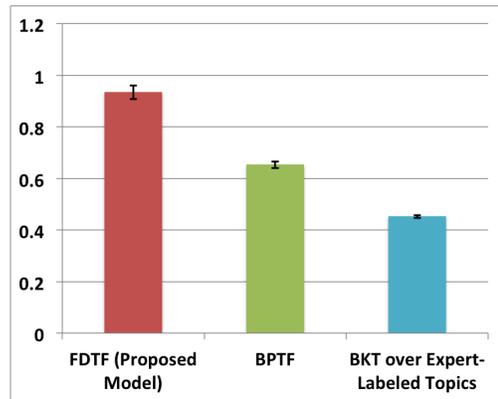


Figure 4: Accuracy of Algorithms for Predicting Students Performance

ror (RMSE) is traditionally used to evaluate this task in the literature. As a result, we compare the approaches based on the RMSE of approaches in addition to their accuracy. Figure 3 shows RMSE of these experiments for each of the approaches. Again, we can see that FDTF has a significantly better RMSE than both the BKT and BPTF algorithms.

These results show that, even though BKT adds the knowledge of topic-based domain model, the tensor factorization algorithms outperform it. Additionally, despite the facts that both BPTF and FDTF use the same data, model the student data as a tensor, and are temporal tensor factorization approaches, the proposed FDTF approach performs better than BPTF. These results show that explicitly modeling students' knowledge acquisition by considering their interactions with learning materials leads to better overall modeling of student knowledge, and thus provide a better overall prediction of student performance.

## 4. CONCLUSIONS AND FUTURE WORK

We proposed a novel tensor factorization model (FDTF) that can predict students' success or failure in future quizzes by explicitly modeling their knowledge acquisition during their interaction with learning materials. This approach does not require any expert or domain knowledge and can be automatically performed using students' historical attempt sequence. Our evaluations show that FDTF outperforms the predicting student performance approaches in the literature.

In future, we plan to explore the ability of the proposed approach in discovering the underlying domain model for the learning material, experiment on more diverse datasets, and compare our algorithm to other PSP and domain modeling approaches in the literature. We plan to improve our FDTF model to be able to model implicit feedback of students' activity, in addition to providing overall success and failure records.

The FDTF model has the potential to be used as a basis to recommend learning material to students. Also, it can help teachers discover domain models and edit or enhance learning materials, look up the concepts that students struggle to learn, and suggest appropriate learning activities.

## 5. ACKNOWLEDGMENTS

This work is partially supported by the Advanced Distributed Learning (ADL) Initiative (contract W911QY-13-C-0032).

## 6. REFERENCES

- [1] 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.
- [2] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Adaptive navigation support for parameterized questions in object-oriented programming. In *Learning in the Synergy of Multiple Disciplines*, pages 88–98. Springer, 2009.
- [3] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review - Society for Industrial and Applied Mathematics*, 51(3):455–500, 2009.
- [4] A. S. Lan, C. Studer, and R. G. Baraniuk. Quantized matrix completion for personalized learning. In *The 7th International Conference on Educational Data Mining (EDM)*, London, July 2014.
- [5] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis—a new alternative to knowledge tracing. In *14th International Conference on Artificial Intelligence in Education*, volume 2009, pages 531–538, 2009.
- [6] S. Sahebi, Y. Huang, and P. Brusilovsky. Parameterized exercises in java programming: using knowledge structure for performance prediction. In *The second Workshop on AI-supported Education for Computer Science (AIEDCS)*, pages 61–70. University of Pittsburgh, 2014.
- [7] S. Sahebi, Y. Huang, and P. Brusilovsky. Predicting student performance in solving parameterized exercises. In *Intelligent Tutoring Systems*, pages 496–503. Springer, 2014.
- [8] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. Matrix and tensor factorization for predicting student performance. In *the International Conference on Computer Supported Education*, pages 69–78. Citeseer, 2011.
- [9] N. Thai-Nghe, T. Horvath, and L. Schmidt-Thieme. Context-aware factorization for personalized student’s task recommendation. In *Proceedings of the International Workshop on Personalization Approaches in Learning Environments*, volume 732, pages 13–18, 2011.
- [10] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, volume 10, pages 211–222, 2010.

# Aim Low: Correlation-based Feature Selection for Model-based Reinforcement Learning

Shitian Shen  
Department of Computer Science  
North Carolina State University  
Raleigh, NC 27695  
ssh@ncsu.edu

Min Chi  
Department of Computer Science  
North Carolina State University  
Raleigh, NC 27695  
mchi@ncsu.edu

## ABSTRACT

We explored a series of feature selection methods for model-based Reinforcement Learning (RL). More specifically, we explored four common correlation metrics and based on them, we proposed the fifth one named Weighed Information Gain (WIG). While much existing correlation-based feature selection methods mostly explored high correlation by default, we explored two options: *High* vs. *Low*. The former selects the next feature that has the highest correlation measure with existing selected ones while the latter selects the one with the lowest correlations. The 10 correlation-based methods were compared against previous feature selection methods for model-based RL across several datasets collected from two vastly different intelligent tutoring systems. Our results showed that the 10 correlation-based methods significantly outperform all other methods across all datasets. Among the five correlation metrics, WIG performed best. Surprisingly, for each of correlation metrics, the low option significantly outperform its high correlation peer and thus it suggests that low correlation-based feature selection methods are more effective for model-based RL than high ones.

## 1. INTRODUCTION

Optimal decision making in complex interactive environments is challenging. In Intelligent Tutoring Systems (ITSs), for example, system's behaviors can be treated as a sequential decision process where at each step system selects an appropriate action from a set of alternatives. Each of these system decisions will affect the user's subsequent actions and performance. Its impact on outcomes cannot be observed immediately and the effectiveness of each decision is dependent upon the effectiveness of subsequent decisions. *Pedagogical strategies* are policies that are used to decide what system action to take next in the face of alternatives.

Reinforcement Learning (RL) is one of the best machine learning approaches for decision making in interactive envi-

ronments. RL focuses on inducing optimal policies on what action(s) an agent should take in any context that would maximize the agent's cumulative reward. While various RL approaches have shown promising, existing RL approaches tend to perform poorly when the interactive environment is complex in that many factors can impact desired outcomes yet not fully understood. Our general approach is to start from a collection of potentially relevant features and to apply *feature selection* methods to narrow them down to a compact and effective state representation. Many feature selection methods such as Least-squares temporal difference (LSTD) with lasso regularization [11], Monte-Carlo tree search algorithm [5] have successfully applied for RL. However, most of them are designed for model-free RL and we used model-based RL (Section 3).

In this paper, we proposed a series of correlation based feature selection methods by exploring different correlation metrics. Correlation-based methods have been widely used in supervised learning, where we use input state feature space  $X$  to predict output label  $Y$  and previous approaches mainly select the subsets of  $X$  with the *highest* correlation with the output label  $Y$  [8, 21]. However, for RL there is no output label  $Y$  and thus, to apply correlation-based feature selection methods directly to RL, we explored two options: *High* and *Low*. The former is to select the next feature that is the **most correlated (High)** with the selected ones while the latter option is to select the **least correlated (Low)** one. Theoretically speaking, choosing the most correlated feature may be effective since the selected feature is more likely to be related to decision making, however it may not make more contribution than the current selected feature set does. On the other hand, choosing the least correlated feature may raise the diversity of selected feature set and enrich the state representation, however it takes a risk of selecting irrelevant or noisy features.

In short, we explored both high and low options for five correlation metrics and resulted in 10 correlation-based methods. We compared them against an ensemble method, the methods involved in [3] referred as **RLPreviousFS** for the rest of paper, and the random feature selection method across several datasets collected from two vastly different ITSs: one is a data-driven logic tutor named Deep Thought and the other is a natural language physics tutor named Cordillera.

## 2. RELATED WORK

In general, existing feature selection for RL can be classified into three categories [6]: Filter, Wrapper and Embedded. Filter approaches can be seen as a preprocessing procedure in that it usually employs a ranking function so that either a fixed number of features with the highest rank or a feature set above a preset threshold value will be selected from the high-dimensional state space. This process is independent from the subsequent model learning process. For RL, the ranking function is generally based on which state feature subset would directly influence the rewards. For example, Morimoto et al. applied *kernel dimension reduction* to evaluate the conditional independence among state features and those with the most impacts on the next-time-step rewards are selected [14]. Hirotaka and Masashi [7] proposed a filter-type approach by directly evaluating the independence between immediate reward and state-feature sequences using conditional mutual information. However, it is not clear how their approach can be applied when immediate reward is not directly observable and only delayed reward is present.

Wrapper approaches search feature space and generate several candidate feature subsets, evaluate each subset using a learning algorithm, and then select the subset with the best performance. For example, Gaudel and Sebag applied Monte-Carlo tree search algorithm to generate candidate feature subsets and then evaluate the goodness of feature subset using the predefined score function [5]. In addition, Keller, et al applied LSTD to approximate value function, selected a feature subset by implementing *Neighborhood Component Analysis* to decompose approximation error, which can be used to evaluate the goodness of the feature subset [9]. Similarly, in *LSPI-FFS* Li, Williams and Balakrishnan also applied LSTD to approximate value function using linear model. They updated the parameters of the linear model through gradient descent and selected a feature subset with largest magnitude of weight [13].

Embedded approaches for RL conduct feature selection and policy induction process simultaneously. Kolter and Ng applied LSTD with *Lasso* regularization to approximate value function as well as to select effective feature subset [11]. Bach explored the penalization of approximation function by using *Multiple Kernel learning* (MKL)[2]. Wright, Loscalzo and Yu proposed *IFSE-NEAT*, the feature selection embedded in neuroevolutionary function, which approximates the value function, and features are selected based on their contributions to the evolution of topology of network[20].

In short, while much of prior research has done on feature selection for RL, most of them is for model-free RL. For Model-based RL, Chi et al. investigated 10 filter-based methods (**RLPreviousFS**) [3]. These methods were implemented to derive a set of various policies, where features are selected mainly based on the single feature performance and the covariance in training data. Their results showed there was no consistent winner among the ten feature selection methods and in some particular cases these methods performed no better than the random baseline method. Therefore, much research on feature selection for model-based RL is needed.

## 3. REINFORCEMENT LEARNING & MARKOV DECISION PROCESS

Generally speaking, RL can be divided into two categories: **model-free** and **model-based**. Model-free RL [4] typically uses samples to learn a value function, from which a policy is implicitly derived. Model-based RL, by contrast, first builds up a model from samples and then compute a policy based the model. Both approaches have their own strengths and weaknesses. Model-free methods are appropriate for domains where data collection is inexpensive and trivial. Model-based methods, on the other hand, are suitable when collecting data is expensive. Given the high cost of collecting training data in our task, we focused on model-based RL and used a Markov Decision Process (MDP) framework.

MDP is defined as a tuple  $\langle S, A, T, R \rangle$ .  $S$  denotes state space, which reflects the generalization of interactive environment; actions  $A$  are agent's possible behaviors; reward function  $R$  can be immediate or delayed feedback from environment respect to agent's behavior and  $R_{SS'}^a$  denotes the reward of transiting from state  $S$  to state  $S'$  by taking action  $a$ ; transition probabilities  $T$  are defined as  $T = \{p(S_j|S_i, A_k)\}_{i,j=1,\dots,m, k=1,\dots,n}$ , which is estimated from training corpus. More specifically,  $T_{SS'}^a = p(S'|S, a)$  denotes the probability of transiting from state  $S$  to state  $S'$  by taking action  $a$ .

Once the tuple  $\langle S, A, T, R \rangle$  is set, we transform the problem of inducing effective pedagogical strategies into computing an optimal policy in an MDP by dynamic programming approaches. More specifically, we calculate the value function  $V^\pi(S)$  under a policy  $\pi$  though Bellman equation[17], which is defined as:

$$\begin{aligned} V^\pi(S) &= E_\pi(R_t|S_t = S) \\ &= \sum_a \pi(S, a) \sum_{S'} T_{SS'}^a [R_{SS'}^a + \gamma V^\pi(S')] \end{aligned}$$

where  $\gamma$  is a constant called discount factor. The optimal value function can be estimated by

$$V^*(S) = \max_\pi V^\pi(S)$$

Then we can derive the optimal policy corresponding to the optimal value function  $V^*(S)$ . Here we used the toolkit developed by Tetreault and Litman [18]. Besides inducing an optimal policy, Tetreault, & Litman's toolkit also calculate the Expected Cumulative Reward (ECR) for the induced policy. The ECR of a policy is derived from a side calculation in the policy iteration algorithm: the V-values of each state, the expected reward of starting from that state and finishing at one of the final states. More specifically, the ECR of a policy  $\pi$  can be calculated as follows:

$$ECR_\pi = \sum_{i=1}^n \frac{N_i}{N_1 + \dots + N_n} \times V(s_i) \quad (1)$$

Where  $s_1, \dots, s_n$  is the set of all starting states and  $V(s_i)$  is the V-values for state  $s_i$ ;  $N_i$  is the number of times that  $s_i$  appears as a start state in the model and it is normalized by dividing  $\frac{N_i}{N_1 + \dots + N_n}$ . In other words, the ECR of a policy  $\pi$  is calculated by summing over all the initial start states in the

space and weighting them by the frequency with which each state appears as a start state. The higher the ECR value of a policy, the better the policy is supposed to perform.

In our application, we defined our action set  $A$  and reward function  $R$  in Section 5. However the state space  $S$  is not well-defined, where each state is a vector representation composed of a fixed number of state features  $F = \{F_1, F_2, \dots, F_p\}$ . Our approach is to apply various feature selection methods to narrow a wide set of feature space to a compact and effective subset that would model student learning process accurately.

## 4. METHODOLOGY

In this section, we first describe the five basic correlation metrics we used and then describe our general feature selection procedure. More specifically, we will describe our 10 correlation-based methods, the ensemble method, and finally briefly describe the RLpreviousFS methods.

### 4.1 Five Correlation Metrics

In order to quantize correlation among features, we used five correlation metrics. The first four are commonly used in supervised learning and here we will investigate whether they can be applied to RL. We proposed the fifth one, Weighted Information gain, by combining the four commonly used metrics and adapting them based on the characteristic our task and datasets. More specifically, we have:

1. Chi-squared (CHI)[22]: a statistical test used to identify whether the distribution of a categorical variable differ from the other one, which induces the independence between two variables. CHI is usually applied to evaluate the independence of two variables in mathematical statistics.
2. Information gain (IG)[12]: it measures the differ between the uncertainty of a variable  $Y$  and the uncertainty of  $Y$  given variable  $X$  as conditional information. It is calculated as:

$$IG(Y, X) = H(Y) - H(Y|X)$$

where  $H()$  is called entropy function, measure uncertainty of a variable. IG evaluates the certainty of variable  $Y$  obtained from variable  $X$ , which can be treated as one type of correlation between  $X$  and  $Y$ . IG has the bias towards the variable with a large number of distinct values.

3. Symmetrical certainty (SU)[21]: it is defined as:

$$SU(Y, X) = \frac{H(Y) - H(Y|X)}{H(X) + H(Y)}$$

SU evaluates the correlation between two variables by normalizing IG. SU compensates the weakness of IG and it is a symmetrical measurement, which treats a pair of variables symmetrically.

4. Information gain ratio (IGR)[10]: it's the ratio of information gain to the intrinsic information, which is the entropy of conditional information. IGR can be represented as:

$$IGR(Y, X) = \frac{H(Y) - H(Y|X)}{H(X)}$$

Comparing with IG, IGR takes the uncertainty of conditional information into account with purpose of removing bias of selecting variable with many distinct values. However, IGR is not a symmetrical measurement ( $IGR(X, Y) \neq IGR(Y, X)$ ).

5. Weighted Information gain (WIG): it is proposed as:

$$WIG(Y, X) = \frac{H(Y) - H(Y|X)}{(H(Y) + H(X))H(X)}$$

We propose WIG by combining IG, SU and IGR. Comparing with IGR, WIG normalized IG by considering the uncertainty of both variables  $X$  and  $Y$  and also compensate the weakness of IG. Comparing with SU, although WIG is not symmetrical measurement. Based on the above equation, WIG sets more weight for variable  $X$ . In our application, WIG is used for evaluating the correlation between current selected feature set  $Y$  with the new feature  $X$ .

For each of the five correlation metrics, we explored two options: High and Low, which resulted in 10 correlation-based methods named five High methods: CHI-high, IG-high, SU-high, IGR-high, WIG-high and five Low methods: CHI-low, IG-low, SU-low, IGR-low, and WIG-low. Our goal is to investigate which option is better: high vs. low and which of the five correlation metric performs the best.

### 4.2 Correlation-based Feature Selection

In this project, we followed a forward stepwise feature selection procedure in that: given current selected feature set, our correlation-based methods select the feature forwardly based on the five correlation metrics described above.

---

**Algorithm 1** Correlation-based Feature Selection Algorithm

---

**Require:**  $\Omega$ : Feature space;  $\mathcal{D}$ : Training data;  $\mathcal{N}$ : Maximum number of selected features  
**Ensure:**  $\mathcal{S}^*$ : Optimal feature set

- 1: **for**  $f_i$  in  $\Omega$  **do**
- 2:  $ECR_i \leftarrow \text{CALCULATE-ECR}(\mathcal{D}, f_i)$
- 3: **end for**
- 4: Add  $f^*$  with highest  $ECR$  to  $\mathcal{S}^*$
- 5: **while**  $\text{SIZE}(\mathcal{S}^*) < \mathcal{N}$  **do**
- 6: **for**  $f_i$  in  $\Omega - \mathcal{S}^*$  **do**
- 7:  $C_i \leftarrow \text{CALCULATE-CORRELATION}(\mathcal{S}^*, f_i, m)$
- 8: **end for**
- 9:  $\mathcal{F} \leftarrow \text{SELECTTOP}(C, 5, \text{reverse})$  ▷ Select top 5 features based on correlation metrics
- 10: **for**  $f_i$  in  $\mathcal{F}$  **do**
- 11:  $ECR_i \leftarrow \text{CALCULATE-ECR}(\mathcal{D}, \mathcal{S}^* + f_i)$
- 12: **end for**
- 13: Replace  $\mathcal{S}^*$  by  $\mathcal{S}^* + f_i$  with highest  $ECR$
- 14: **end while**

---

Algorithm 1 shows the concrete process of our correlation-based feature selection procedure. It contains three major

parts: in the first part (lines 1–4), it constructs MDPs for each single feature, induces a single-feature policy and calculates its  $ECR$ . Then the feature with highest  $ECR$  is added into current optimal feature set. In the second part (lines 6–9), it evaluates the correlations between current optimal feature set  $\mathcal{S}^*$  with other features  $f_i \in \Omega - \mathcal{S}^*$ , ranks the correlations, and then selects the top 5 highest ones for high correlations or the bottom 5 lowest ones for low correlations. They are selected to form a feature pool  $\mathcal{F}$ . In the third part (lines 10–13), several candidate feature sets are generated by combining current optimal feature set  $\mathcal{S}^*$  with each feature  $f_i \in \mathcal{F}$ . Then  $ECR$  for each candidate feature set can be evaluated by applying *Calculate-ECR* function. Current optimal feature set  $\mathcal{S}^*$  will be replaced by the candidate feature set with highest  $ECR$ . The algorithm will terminate until the size of optimal feature set reaches maximum number  $\mathcal{N}$ . The third part can be treated as the process of wrapper approach where several candidate feature sets are evaluated by the RL method. Therefore, our correlation-based methods are the combination of filter and wrapper approaches.

### 4.3 Ensemble Method

Our ensemble approach combines the 10 proposed correlation-based methods and 4 *RL-based* methods (Section 4.4), which are most effective methods among RLPPreviousFS. Its procedure is similar to that of correlation-based method except the second part (lines 6–9). The ensemble approach integrates the features generated from each method and generates a relatively big feature pool  $\mathcal{F}$ . The maximum size of  $\mathcal{F}$  is up to 70 but often smaller because of the overlapping feature sets. Note that it is still much larger than any of our 10 correlation-based methods which has 5 candidates for each step. After generating the feature pool, the ensemble method jumps to the third part (lines 10–13) of Algorithm 1. At each step, the ensemble method explores feature sets by adding the feature with maximum  $ECR$ .

### 4.4 RLPPreviousFS

Chi et. al [3] grouped RLPPreviousFS into three categories: 1) four RL-based methods; 2) two PCA-based method, which selects features with the high correlation with principle components; 3) four PCA&RL-based methods, which use RL-based methods to select features from a candidate feature set which is generated from PCA-based method. All three categories can be seen as the filter approaches.

## 5. TRAINING DATASETS

### 5.1 Two Deep Thought Datasets

Deep Thought (DT)[15] is a data-driven ITS. It is a rule-based system where students need to select different rules to complete logic proof problems. In DT, we focused on a problem level decision named problem solving (PS) vs. Worked Example(WE). More specifically, when starting the next training problem, the tutor will make a simple decision: “should it ask student to solve the next problem (PS), or should it provide an example to show the student how to solve the next problem (WE)”.

Our training dataset includes a total of 303 undergraduate CS students who used DT as part of class assignment in Fall 2014 and Spring 2015. The average amount of time spent in

the tutor was 416.60 minutes. To induce RL policies, a total of 134 features were extracted from the student-system log files. The reward function in DT dataset is calculated based on level score  $LevelScore_i$  where  $i \in [1, 6]$ . Particularly, we designed two type of reward: immediate and delay reward. Immediate reward is defined as  $R_i = LevelScore_i - LevelScore_{i-1}$  where  $i \in [1, 6]$ ,  $R_1 = LevelScore_1$ , it reflects the change of students’ performance level by level. Delayed reward is represented as  $R_{delay} = LevelScore_6 - LevelScore_1$ , which determines the change of students’ performance across all levels. For the convenience, we denote the two DT datasets with immediate reward as *DT-Immed* and that with delayed reward as *DT-Delay* respectively.

### 5.2 Six Cordillera Datasets

Cordillera [19] is a natural language tutoring system teaching college introductory physics. Different from DT tutor system, Cordillera requires students to input their answer by natural language free text. The data collection consists of the following stages: 1) background survey; 2) studying textbook and prerequisite materials, 3) taking a pretest; 3) training on Cordillera, 4) and taking a post test. Cordillera makes step-level decision: *Elicit/Tell (ET)*. The ET decision means “should the tutor system *elicit* the next problem-solving step for student, or should it *tell* student the instruction of next step directly”.

Our training corpus involves 64 students. In Cordillera, there are five primary Knowledge Components (KCs): Definition of Kinetic Energy (KE), Gravitational Potential Energy (GPE), Spring Potential Energy (PE), Total Mechanical Energy (TME), and finally Conservation of Total Mechanical Energy (CTME). In STEM domains such as math and science, it is commonly assumed that the relevant knowledge is structured as a set of independent but co-occurring KCs. A KC is “a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet” [19]. For the purposes of ITSs, these are the atomic units of knowledge. It is assumed that a tutorial dialogue about one KC (e.g., kinetic energy) will have no impact on the student’s understanding of any other KC (e.g., of gravity). This is an idealization, but it has served ITS developers well for many decades, and is a fundamental assumption of many cognitive models [1, 16]. Given the KCs’ independence assumptions, we will apply RL to induce KC-specific pedagogical strategies for each of the five primary KCs individually. Moreover some steps in Cordillera have mixed KC, thus we also apply RL to induce pedagogical policies irregardless of the KCs involved (denoted by Across). In short, we have a total of **six** Cordillera KC datasets, one per KC for the five primary KCs and one KC-general for the Across policy. Each of the KC datasets contains 50 state features and to induce RL-rules, we used the delayed reward defined as student Normalized Learning Gains (NLGs):  $NLG = \frac{Posttest - Pretest}{MaximumScore - Pretest}$ . Here *MaximumScore* is the maximum score a student can get and for both pretest and posttest, the maximum score is set to be 1.

## 6. EXPERIMENT & RESULT

To evaluate the effectiveness of induced policies, we set the maximum number of selected features to be 6 considering the size of our training datasets. In this section, we present

Table 1: The highest ECR Induced by Correlation-based Methods Across Eight Datasets

ITS	Data	CHI		IG		SU		IGR		WIG	
		High	Low	High	Low	High	Low	High	Low	High	Low
DT	Immed	55.89	129.82	53.87	95.81	53.87	95.81	53.87	95.81	59.04	<b>143.16*</b>
	Delay	8.89	12.56	8.89	12.58	10.73	12.58	8.94	<b>15.43*</b>	8.94	<b>15.43*</b>
Cordillera	KE	5.86	6.75	5.86	6.75	5.86	6.75	5.57	<b>7.64*</b>	5.57	7.62
	GPE	10.47	13.39	11.80	13.39	11.21	13.39	11.10	<b>17.23*</b>	10.82	<b>17.23*</b>
	SPE	12.67	17.17	12.67	14.88	12.67	<b>18.02*</b>	10.83	<b>18.02*</b>	10.27	<b>18.02*</b>
	TME	7.34	7.96	7.57	9.42	7.47	9.42	6.98	<b>10.04*</b>	6.40	<b>10.04*</b>
	CTME	23.01	32.71	24.01	31.22	24.31	31.22	23.01	<b>33.24*</b>	23.01	<b>33.24*</b>
	Across	1.77	2.26	1.77	2.26	1.77	<b>2.57*</b>	1.77	2.26	1.77	<b>2.57*</b>

Note: The best *ECR* among 10 methods for each dataset is highlighted by \*.

Table 2: Overall Evaluation Across Eight Datasets

	DT		Cordillera						
	Immed	Delayed	KE	GPE	SPE	TME	CTME	Across	
Low Correlation	<b>143.16</b>	<b>15.43</b>	<b>7.64</b>	<b>17.23</b>	<b>18.02</b>	<b>10.04</b>	<b>33.24</b>	2.57	
High Correlation	59.03	10.72	5.85	11.80	12.67	7.57	24.31	1.71	
Ensemble	127.79	12.61	7.33	16.40	16.95	9.12	32.06	<b>2.68</b>	
RLPreviousFS	60.28	12.56	6.17	14.41	11.90	7.15	24.60	2.03	
Random	8.53	7.62	4.26	7.34	10.52	4.78	22.02	1.20	

the experimental analysis of the correlation-based methods, the ensemble, the RLPreviousFS used in previous research, and random feature selection methods which is our baseline method.

## 6.1 Comparing correlation-based methods

In this section, we want to answer two questions:

- 1) which option is better for model-based RL: High vs. Low;
- 2) which of the five correlation metrics performs the best.

**High VS Low.** Table 1 shows the performance of the 10 correlation based methods across eight training datasets: two DT and six Cordillera datasets. The rows represent the eight datasets while columns represent the 10 correlation-based methods. Each cell in Table 1 shows the highest ECR of the policy generated from the corresponding correlation-based feature selection method on the corresponding dataset when the number of features varies from 1 to 6.

Table 1 shows that for each of five correlation metrics, the low correlation-based method significantly outperform its high correlation-based peer. For *DT-Immed* dataset, the *ECR* of WIG-low is 143.16, while *ECR* of WIG-High is only 59.04; the former is 140% higher than the latter. Similarly, the *ECRs* of CHI-low and CHI-High are: 129.82 vs. 55.89 and the former is 132% higher than the latter. The similar results is true across all five correlation metrics and across all eight datasets.

Moreover, the out-performance of the Low option over the High option seems to be more prominent on DT datasets than Cordillera datasets. For DT data, the average percent increase for the low correlation methods over the high correlation methods is 75.35%, the maximum percent increase is 142.48% and the minimum percent increase is 17.24%.

For Cordillera KC datasets, the average percent increase for the low correlation methods over the high ones is 35.15%, the maximum percent increase is 75.46% and the minimum percent increase is 8.45%. On average the low correlation methods outperform the high correlation peers by 45.2%.

To summarize, our results showed that the low correlation option is more suitable for the model-based RL than the high correlation option. It indicates that it is important to include a variety of features in the state representation for applying RL to induce pedagogical policies.

**Five Correlation Metrics.** In Table 1, for each of the eight datasets, we highlight the best ECR of the induced policies by \*. Table 1 shows that the WIG is the consistent winner in that it has the best ECR for all datasets except for *KE*. On the *KE* dataset, WIG-Low performance is slightly lower than the best policy: 7.62 for WIG-Low vs. the highest 7.64 for IGR-Low. Following WIG, IGR is the second best in that it has the highest ECR for six out of eight datasets. Note that WIG and IGR together produced all the best policies across all eight datasets and they overlapped on *DT-Delay*, *GPE*, *SPE*, *TME*, *CTME*. Except for WIG and IGR, the remaining three metrics only induced 2 best policies and both are found by SU-Low. In short, our proposed WIG performed the best among the five correlation metrics followed by IGR.

## 6.2 Overall Evaluation

Table 2 shows the overall comparison among all feature selection methods. With the purpose of simplicity, for the five low-correlation methods, the five high-correlation methods and the RLPreviousFS methods, we select the best one from each category. Thus, Table 2 will compare the five categories of feature selection methods: the best of the five Low-

correlations, the best of the five High-correlations, the ensemble, the best of RLPPreviousFS and the random method.

In Table 2, rows denote the five categories and columns show the eight datasets. Table 2 shows that as expected the random method performs the worst across all datasets. In addition, the best of the low correlation-based methods outperforms all other methods in all datasets except in the *Across* dataset, where the ensemble method performs slightly better than the best of the low correlation-based methods. On average, the best low correlation-based method increases over the best of RLPPreviousFS by 43.87% and over the ensemble method by 9.05%. In addition, the ensemble method improves over the best of RLPPreviousFS on average 36.46%. To summarize, we can rank the five categories of methods as Low correlation-based > Ensemble > High correlation-based, RLPPreviousFS  $\gg$  Random.

## 7. CONCLUSIONS & FUTURE WORK

In this paper, we proposed 10 correlation-based feature selection methods for model-based RL. Our result clearly showed that the low correlation-based methods are more effective than the ensemble, the high correlation-based, the RLPPreviousFS, and the random method. Among the five correlation-based metrics, our proposed WIG performed the best. WIG found the best policies across all eight datasets except that on KE, its performance is only slightly lower than the best one which is found by IGR.

While in supervised learning features associated with highest correlation are generally selected, for model-based RL selecting the next feature with lowest correlation is more effective. Moreover, it is surprising to see that the ensemble method only performed the best on one out of eight datasets. Given that the motivation for applying the ensemble method is that it can take the advantages of each method with purpose of achieving better results. Therefore, one of our future work is to explore other ways to make our ensemble method more effective.

## 8. ACKNOWLEDGEMENTS

This research was supported by the NSF Grant 1432156 "Educational Data Mining for Individualized Instruction in STEM Learning Environments".

## 9. REFERENCES

- [1] J. R. Anderson. *The architecture of cognition*. Cambridge, Mass. : Harvard University Press, 1983.
- [2] F. R. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, pages 105–112, 2009.
- [3] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 2011.
- [4] N. D. Daw. Model-based reinforcement learning as cognitive search: neurocomputational theories. *Cognitive search: Evolution, algorithms and the brain*, pages 195–208, 2012.
- [5] R. Gaudel and M. Sebag. Feature selection as a one-player game. In *ICML*, pages 359–366, 2010.

- [6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [7] H. Hachiya and M. Sugiyama. Feature selection for reinforcement learning: Evaluating implicit state-reward dependency via conditional mutual information. In *Machine Learning and Knowledge Discovery in Databases*, pages 474–489. Springer, 2010.
- [8] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.
- [9] P. W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 449–456. ACM, 2006.
- [10] J. T. Kent. Information gain and a general measure of correlation. *Biometrika*, 70(1):163–173, 1983.
- [11] J. Z. Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 521–528. ACM, 2009.
- [12] C. Lee and G. G. Lee. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information processing & management*, 42(1):155–165, 2006.
- [13] L. Li, J. D. Williams, and S. Balakrishnan. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *INTERSPEECH*, pages 2475–2478, 2009.
- [14] J. Morimoto, S.-H. Hyon, C. G. Atkeson, and G. Cheng. Low-dimensional feature extraction for humanoid locomotion using kernel dimension reduction. In *ICRA*, pages 2711–2716. IEEE, 2008.
- [15] B. Mostafavi, G. Zhou, C. Lynch, M. Chi, and T. Barnes. Data-driven worked examples improve retention and completion in a logic tutor. In *Artificial Intelligence in Education*, pages 726–729. Springer, 2015.
- [16] A. Newell. *Unified Theories of Cognition*. Harvard University Press; Reprint edition, 1994.
- [17] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.
- [18] J. R. Tetreault and D. J. Litman. A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication*, 50(8):683–696, 2008.
- [19] K. VanLehn, P. W. Jordan, and D. J. Litman. Developing pedagogically effective tutorial dialogue tactics: experiments and a testbed. In *SLaTE*. Citeseer, 2007.
- [20] R. Wright, S. Loscalzo, and L. Yu. Embedded incremental feature selection for reinforcement learning. Technical report, DTIC Document, 2012.
- [21] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.
- [22] M. F. Zibran. Chi-squared test of independence. *Department of Computer Science, University of Calgary, Alberta, Canada*, 2007.

# Personalization of Learning Paths in Online Communities of Creators

Mingxuan Sun\*

Division of Computer Science and Engineering  
Louisiana State University  
msun@csc.lsu.edu

Seungwon Yang

School of Library and Information Science  
Center for Computation and Technology  
Louisiana State University  
seungwonyang@lsu.edu

## ABSTRACT

In massive online communities of creators (OCOCs), one of the core challenges is to encourage users to learn to create original contents using basic components. Recommending the right learning components at the right time is critical for improving user engagement and has not been fully studied due to the unstructured nature of online communities. To address the problem, we propose in this paper a novel recommendation model which integrates Cox's survival analysis and collaborative filtering. Our model can incorporate factors such as user learning history and social engagements, which provides us insights in improving the personalized service. We apply our method to the user data from Scratch online platform and demonstrate the performance of the model.

## 1. INTRODUCTION

In recent years, the number of online learning communities (OCOCs) has increased exponentially as evidenced by successful platforms such as Scratch online<sup>1</sup>. These online communities offer flexible learning environment where users can create projects (e.g., games, art designs), share projects, and engage with like-minded users in the community. One of the goals is to foster learning programming concepts through developing and sharing projects among its users based on interactions in the community [11]. Previous studies [7] have found that creating and sharing projects is the gateway to other online social activities including commenting and following. However, only about 29% of Scratch users would like to share their projects and about half of them contribute no more than one project.

One way to improve user engagement is to track users' learning history and recommend contents tailored to each individual. For example, Scratch users learn to create projects by manipulating basic programming blocks such as "goto",

"change color", and "doIf". Each block is categorized in a certain Computational Thinking (CT) concept [6]. Users are expected to learn CT concepts such as "motion" by manipulating blocks such as "goto", "bounce", and "turn". Users may follow different learning paths over time. Based on programming blocks that each user has used in his/her previous projects, we can recommend particular blocks, concepts, or projects tailored to the individual. For instance, for users who are interested in animation projects with some basic motion blocks such as "goto", the system can recommend projects that have more advanced motion blocks such as "bounce".

In addition to what to recommend, when is a good time to recommend is another important factor to consider since suggesting blocks to users at the right time may influence learning effectiveness and efficiency. For example, if a user is still struggling with basic motion techniques such as "goto", it may not be a good idea to introduce a project or a more advanced programming concept such as "turn" or "direction". Our goal is to alleviate the high dropout rates in the early stage through personalization of the learning path.

In this paper, we propose a model to learn the probability of a user's exposure to a certain learning component at a particular time. The probability of exposure is estimated based on a collaborative filtering model, which recommends the user the items favored by the like-minded. The conditional probability of a user being exposed to a given item at a particular time is modeled by the Cox proportional hazard model from survival analysis.

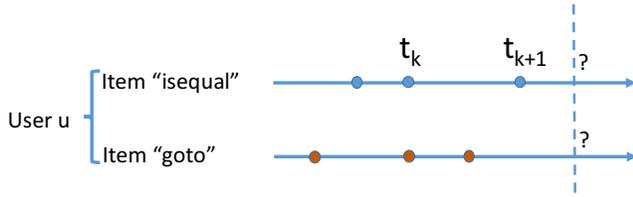
## 2. RELATED WORK

Early studies on learning behavior analysis for OCOCs have been based on case-studies evaluating learning process qualitatively [5, 12]. Other attempts [3, 7] have focused on clustering user behaviors based on types and volumes of users' online activities. A recent work by Yang et al. [14] modeled *informal learning trajectories* quantitatively as the growth of cumulative usage of programming blocks by each user.

Personalization approaches that are based on user behaviors have been widely studied in different types of Web services such as e-commerce. In e-commerce, most personalization approaches focus on recommending users the items that have been favored by like-minded users based on their purchase history. Traditional recommendation algorithms are memory based methods including vector similarity and correla-

\*Corresponding author.

<sup>1</sup><https://scratch.mit.edu/>



**Figure 1: Time-aware recommendation.** The occurrence time of user-item interaction is modeled using survival analysis. Our goal is to predict the most desired learning item  $i$  at a particular time  $t$  for each individual user  $u$ .

tion [2]. The state-of-the-art methods including the one that won the Netflix competition [9] are based on matrix factorization. The time factor in personalization services largely affects the user satisfaction of the service [13, 10]. Our contribution in this paper lies in that we incorporate both the Cox model and collaborative filtering to provide personalized recommendation for online learners.

### 3. METHOD

In OCOs, users create and share projects consisting of basic items such as programming blocks in Scratch. Each item belongs to a certain category. Based on user-item interaction histories, we would like to suggest items tailored to each user at a particular time. To achieve this goal, we propose to estimate the joint probability  $p(u, i, t) = p(t|u, i)p(u, i)$ , where  $p(u, i)$  is the probability of user  $u$  interacting with item  $i$  and  $p(t|u, i)$  is the conditional probability of user  $u$  interacting with item  $i$  at time  $t$ .

We model the occurrence time  $t$  of the event that user  $u$  interacts with item  $i$  using the Cox model in survival analysis. Survival analysis is used to estimate the probability of the occurrence of an event  $p(\text{event in } [t, t + \Delta t])$  such as when a patient fails to survive. In the online learning context, our task is to estimate the probability of the occurrence of exposing to a specific learning block for each user, which is  $p(t|u, i)$ . As shown in Figure 1, in the observed sequences of user-item interactions, a user builds a project with a set of items (e.g., “isequal” and “goto”) at time  $t_k$ . Then item  $i$  is used again in another project of the same user at time  $t_{k+1}$ . Let  $x_k$  be the covariates associated with user  $u$  at time  $t_k$ . We are interested in predicting the time gap  $t_{k+1} - t_k$ .

Let  $\lambda(t)$  denote the instantaneous rate of event happening at time  $t$  following the last event given the covariates  $x_k$ , that is  $\lambda(t) = P(T = t | T \geq t)$ . The Cox model assumes that the covariates only affect the magnitude of each individual hazard rates. Formally, for an individual observation with covariates  $x_k$ , the hazard at time  $t$  is:

$$\lambda(t) = \lambda_0(t) * \exp(x_k^T \beta), \quad (1)$$

where  $\lambda_0$  is the non-parametric baseline hazard function,  $x_k$  is the covariates, and  $\beta$  is the regression coefficient. The log likelihood of observing the occurrences is:

$$\log L = \sum_{k=1}^K \left\{ d_k \log \lambda(t_k) - \int_0^{t_k} \lambda(\tau) d\tau \right\}, \quad (2)$$

where  $d_k$  is a censor indicator, taking the value one if event

occurs at time  $t_k$  or the value zero if event does not occur till time  $t$  by the end of observation window. The parameters  $\beta$  and the baseline hazard  $\lambda_0$  can be estimated by maximizing the log partial likelihood with Breslow’s approximation [4].

We further estimate the probability  $p(u, i)$  of a user favoring a particular item (e.g., block) by adopting collaborative filtering (CF) recommendation algorithms. User interactions contain substantial information to improve recommendation accuracy. For example, in Scratch, users play with a set of programming blocks to develop a project. Therefore, the frequency of each type of block may indicate their preferences. Based on the previous learning history, the system can predict interesting blocks tailored to individual taste. Collaborative filtering methods focus on detecting users with similar preferences and recommending items favored by the like-minded. Algorithms range from similarity based CF methods [2] to matrix factorization based CF methods popularized by the Netflix Prize Competition [9].

Let  $r_{ui}$  denote the observed preference of user  $u$  for item  $i$ , where  $u = 1, 2, \dots, m$  and  $i = 1, 2, \dots, n$ . The pairs  $(u, i)$  are stored in the set  $O = \{(u, i) | r_{ui} \text{ is observed}\}$ . Since the observed ratings or event frequencies are very sparse, matrix factorization is used to learn latent features of both users and items in a lower dimensional space such that the product of each user-item pair can best approximate the ratings. Specifically, let  $\theta_u$  and  $v_i$  denote latent features for user  $u$  and items  $i$ , where  $\theta_u$  and  $v_i$  are  $k$ -dimensional vectors. The latent features can be estimated by minimizing a prediction loss function between the predicted ratings and true ratings of users. That is,

$$\min_{\Theta, V} \sum_{(u, i) \in O} (r_{ui} - \theta_u^T v_i)^2, \quad (3)$$

where  $\Theta = [\theta_1, \theta_2, \dots, \theta_m]$  is a  $k \times m$  matrix and  $V = [v_1, v_2, \dots, v_n]$  is a  $k \times n$  matrix. A gradient descent based method [9] can be used to estimate latent features. The probability of user favoring an item  $p(u, i)$  can be generated using a softmax function:

$$p(u, i) = \frac{\exp(r_{ui})}{\sum_{j=1}^n \exp(r_{uj})}, \quad (4)$$

## 4. EXPERIMENTAL RESULTS

We evaluate the model performance through two steps: time-to-return prediction and time-aware recommendation. In the first step, for every user-item interaction  $(u, i)$ , we estimate the probability of the next occurrence at time  $t$  and use the expected value of the time as the predicted time to return. In the second step, for each user  $u$  at a particular time  $t$ , we rank each item  $i$  by the joint probability  $p(u, i, t)$  and recommend top-K items. We present the experimental details including data collection, evaluation metrics, and competing baselines.

### 4.1 Data Collection

We apply our method to user data which was released in spring of 2014 from Scratch online<sup>2</sup>. Users can create a project by programming with basic components called blocks. Each block can be categorized into one or more CT concepts.

<sup>2</sup><https://llk.media.mit.edu/scratch-data/>

**Table 1: Covariate analysis for CT concept “conditionals”.** \*\*\*  $p < 0.001$ , \*\*  $p < 0.01$ , \*  $p < 0.05$ , .  $p < 0.1$

Covariate Name	Coefficient	P-Value
is.remix	0.190556	0.000593 ***
is.self.remix	-0.140119	0.062772 .
is.remixed	0.447668	2.44e-15 ***
like 2 or more	0.226432	0.001440 **
follow 2 or more	0.262599	0.000346 ***
comments 2 or more	0.478668	< 2e-16 ***
conditionals experience	0.332191	1.14e-08 ***
operators experience	-0.074161	0.236036
data experience	-0.157914	0.010259 *

We adopt the the mapping table from blocks to CT concepts as suggested in [6]. Users are encouraged to share their projects and interact with others by commenting projects, favoring projects, or following other users. For each user, the dataset includes the project details including block usage and timestamps. It also maintains tables of different types of social interactions including user follower-followed relationship and comments. The user history data collected from December 2011 to March 2012 are used to create the training and the testing datasets. Possible spam users who create more than 100 projects in a day are filtered out. The remaining data contains 22415 users and 170 learning blocks with 6 CT concepts. All user records observed during December 2011 to February 2012 are used to train the model through cross-validation and all user records during March 2012 are used for testing.

The following covariates are used to estimate the Cox model. Covariates related to user activity history include the number of days since registration and the gap since last login. User social interaction covariates include the number of projects liked, the number of friends followed, and the number of comments on projects. User project details include the number of projects created, the number of types of blocks, and the number of concepts. We collect user covariates on a daily basis and predict the days till the user’s next event. Users who had not been exposed to the event by the end of the time window were censored.

## 4.2 Performance Evaluation

In the first step “time-to-return prediction”, for every block pair  $(u, i)$ , we estimate the probability of the next occurrence at time  $t$  and treat the expected value of the time as the predicted time to return. Since the data are sparse, a direct estimation of a survival model for each block will be noisy. Instead, we train a Cox model for each CT concept using the interactions events of blocks belonging to that concept. To evaluate the performance, we predict the expected time from the learned density function and compute the Rooted Mean Square Error (RMSE) with respect to the true time. We compare the Cox model against the baselines including linear regression and decision tree regression. Smaller RMSE values indicate better performance.

The importance of covariates for predicting each individual user’s exposure to CT concepts “conditionals” and “data” are shown in Tables 1 and 2. Both tables show the covariates’ names, the regression coefficients and the significance scores. A positive regression coefficient for a vari-

**Table 2: Covariate analysis for CT concept “data”.** \*\*\*  $p < 0.001$ , \*\*  $p < 0.01$ , \*  $p < 0.05$ , .  $p < 0.1$

Covariate Name	Coefficient	P-Value
is.remix	0.15007	0.018629 *
is.self.remix	-0.18239	0.037235 *
is.remixed	0.52571	3.33e-16 ***
like 2 or more	0.23287	0.005221 **
follow 2 or more	0.20475	0.023510 *
comment 2 or more	0.54465	< 2e-16 ***
conditionals experience	0.03648	0.605257
operators experience	0.14616	0.041800 *
data experience	0.12105	0.076548 .

able implies a higher hazard if the value of the variable is high. Both tables show that the regression coefficients for the variable “is.remixed.bool” are positive. It indicates that if a user’s project is remixed by others, the hazard rate of observing the user’s next event will increase by a factor of  $exp(0.190556) - 1$  compared with the baseline hazard. On the contrary, a negative regression coefficient implies a lower hazard, which means the probability of user interacting with the blocks belonging to that concept will be smaller. The value of the coefficient is statistically significant at different significance levels. We only show the covariates with highest significant levels.

As shown in the tables, for both CT concepts “conditionals” and “data”, for users who share projects later remixed by others, it is more likely that these users will be back creating projects in the future. Interestingly, users who remix others’ projects will be more likely to create projects than those who remix their own projects. In addition, users who like two or more projects, who follow two or more friends, and who have two or more comments are more likely to create and share projects in the future than those who have no social interactions. This implies that social interactions help users to learn and share. In addition, we can see that users who have built blocks in the concept “conditional” are more likely to build blocks falling into the same concept. Interestingly, users who have built blocks in the concepts “operator” and “data” are more likely to build blocks in the concept “data”.

We then use the estimated model to predict the time to the next event in each CT concept. Table 3 displays the root mean square error (RMSE) for the return time prediction using the Cox model and baselines, respectively. For concepts “loops”, “conditionals”, “operators”, and “data”, the hazard based approach outperforms all the other baselines. For concept “event”, the hazard based approach performs very close to linear regression and both of them perform better than the others. All the baselines do not model the underlying temporal patterns in the observed sequences.

For the final step “time-aware recommendation”, suppose the testing event of user  $u$  occurs at time  $t$ , we compute the probability  $p(u, i, t)$  of the user favoring an item  $i$  at time  $t$  for each item  $i$  and rank among all items by probability. Ideally, the observed items that the user actually interacts with should appear on top positions. In information retrieval, we focus on the evaluation accuracy on top positions using several standard metrics including precision at  $k$  (P@k), Mean Average Precision (MAP) and Normalized Discounted

**Table 3: RMSE comparison for user return time prediction. Smaller values indicate better performance.**

	Loops	Events	Conditionals	Operators	Data
<b>Linear Regression</b>	9.13	<b>9.20</b>	8.94	8.79	8.68
<b>Decision Tree Regression</b>	9.33	9.41	9.13	9.00	8.80
<b>Cox model</b>	<b>9.04</b>	9.25	<b>8.63</b>	<b>7.97</b>	<b>7.62</b>

**Table 4: Comparison of recommendation accuracy.**

	P@1	P@3	P@5	MAP@20	NDCG@20
<b>NMF</b>	0.78	0.70	0.64	0.71	0.67
<b>SurvMF</b>	0.84	0.72	0.64	0.72	0.68

Cumulative Gain at  $k$ (NDCG) [8]. We compare with the state-of-the-art baseline non-negative matrix factorization (NMF) [1]. We follow the standard procedure in collaborative filtering to estimate the model using the user data in the training set and evaluate the performance of the prediction in the test set. Specifically, the user records observed before March 2012 are used to train and the user records in March 2012 are used to test. The data contains the rating of each user-block pair, where the rating corresponds to the categorization of event occurrences. The maximum rating is 6 for six or more event occurrences. At the time of the testing event, we compare the ranked list with ground truth. As shown in Table 4, since our method (SurvMF) integrates the survival model into the matrix factorization to capture the temporal dynamics of user-item interaction, it can achieve better performance.

## 5. CONCLUSIONS AND FUTURE WORK

In this work, we have focused on personalization of learning path in massive online communities of creators. One of the main challenges in online learning is high dropout rates in the early stage due to cognitive overload. To alleviate the problem, we propose a novel model integrating the Cox model and matrix factorization to recommend the right learning contents at the right time. The model can incorporate factors such as user learning history and social engagements. In addition, the latent features learned through matrix factorization further improves the recommendation accuracy. Empirical evaluations on real world data demonstrate the performance of our model.

## References

- [1] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.
- [2] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.
- [3] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–284. ACM, 2000.
- [4] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.
- [5] A. Dahotre, Y. Zhang, and C. Scaffidi. A qualitative study of animation programming in the wild. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '10*, pages 29:1–29:10, New York, NY, USA, 2010. ACM.
- [6] S. Dasgupta, W. Hale, A. Monroy-Hernández, and B. M. Hill. Remixing as a pathway to computational thinking. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, pages 1438–1449. ACM Press, 2016.
- [7] D. Fields, M. Giang, and Y. Kafai. Understanding collaborative practices in the scratch online community: Patterns of participation among youth designers. *To see the world and a grain of sand: Learning across levels of space, time, and scale: CSCW 2013 Conference Proceedings*, 2013.
- [8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.
- [9] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data*, 4(1):1–24, 2010.
- [10] J. Lehmann, M. Lalmas, E. Yom-Tov, and G. Dupret. Models of user engagement. In *User Modeling, Adaptation, and Personalization*, pages 164–175. Springer, 2012.
- [11] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, and Y. K. B. Silverman. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.
- [12] C. Scaffidi and C. Chambers. Skill progression demonstrated by users in the scratch animation environment. *Int. J. Hum. Comput. Interaction*, 28(6):383–398, 2012.
- [13] J. Wang and Y. Zhang. Opportunity model for e-commerce recommendation: right product; right time. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 303–312. ACM, 2013.
- [14] S. Yang, C. Domeniconi, M. Reville, M. Sweeney, B. Gelman, C. Beckley, and A. Johri. Uncovering trajectories of informal learning in large online communities of creators. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 131–140. ACM, 2015.

# Modeling Visitor Behavior in a Game-Based Engineering Museum Exhibit with Hidden Markov Models

Mike Tissenbaum  
UW–Madison  
225 North Mills St  
Madison, WI

miketissenbaum@gmail.com

Vishesh Kumar  
UW–Madison  
225 North Mills St  
Madison, WI

vishesh.kumar@wisc.edu

Matthew Berland  
UW–Madison  
225 North Mills St  
Madison, WI

mberland@wisc.edu

## ABSTRACT

Research has shown that supporting tinkering and exploration promotes a wide range of STEM related literacies. However, the open-endedness of tinkering environments makes it difficult to know whether learners' exploration is productive or not. This is especially true in museum spaces, where dwell times are short and facilitators lack a history of engagement with individual visitors. In response, this study uses telemetry data from Oztoc – an open-ended exploratory tabletop exhibit in which visitors embody the roles of engineers who are tasked with attracting and cataloging newly discovered aquatic creatures by building working electronic circuits. This data is used to build Hidden Markov Models (HMMs) to devise an automated scheme of identifying when a visitor is behaving productively or unproductively. Evaluation of our HMM was shown to effectively discern when visitors were productively and unproductively engaging with the exhibit. Using a Markov model, we identify common patterns of visitor movement from unproductive to productive states to shed light on how visitors struggle and the moves they made to overcome these struggles. These findings offer considerable promise for understanding how learners productively and unproductively persevere in open-ended exploratory environments and the potential for developing real time supports to help facilitators know how and when to best engage with visitors.

## Keywords

Learning analytics, museums, interactive tabletops modeling.

## 1. INTRODUCTION

While there is evidence that digitally-augmented museum spaces can enhance science learning [36, 11], there is increased interest in how less-structured, open-ended designs can support new forms of STEM-based (science, technology, engineering, and math) reasoning and collaboration [18, 19]. Tinkering, in particular, often characterized by playful, experimental, iterative styles of engagement, and iterative, investigative processes of learning and discovery, has shown considerable promise in helping novices develop engineering and computer science literacies [5, 26].

Tinkering is an ideal complement to the kinds of learner-centered constructivist pedagogy found in many hands-on science museums [1]; however, in the open-ended and exploratory tasks that typify tinkering, assessment and feedback is particularly difficult [8]. This is especially true in museum environments, as visitors often do not have the expertise or confidence to conduct the coherent, in-depth investigations required to answer their questions on their own [2]. As such, within open-ended environments there is a growing need to develop methods for understanding learners' tinkering and exploration.

Digitally mediated museum spaces, when properly instrumented, can capture data on visitors' tinkering and experimentation in

real-time (known as telemetry data), allowing researchers to identify and analyze temporal patterns in visitor interactions. We can then begin to investigate which patterns might be classified as productive (e.g., moving towards the broader learning goals of the exhibit) or unproductive (e.g., [23]). However, by their very nature, productive and unproductive states within open-ended tinkering activities are inherently difficult to classify.

One approach to understanding the state of a learner is through Markov Modeling [4]. Markov modeling is used to characterize patterns of sequential activity, but first-order Markov models only consist of sequences of known states, and we are often more interested in more complex relationships than just sequences of concrete data. One approach to finding hidden states in learners' activities is the use of Hidden Markov Models (HMM – [25]). Applying HMM to learning processes allows us to consider a learner as being in one of a fixed set of (“hidden”) states at any moment in time. These models, are particularly well suited for museums as individual visitors' states are particularly hard to capture and pre- and post-tests are problematic if we want to ensure a naturalistic setting [9]. In response, the paper advances a research trajectory in which we attempt to highlight productive and unproductive patterns of visitor interactions by mining their telemetry data from an interactive tabletop exhibit at a large urban interactive science museum. In particular, this research addresses the following questions: 1) *Can a Hidden Markov Model accurately predict if visitors are productively or unproductively engaged in an open-ended museum activity?* 2) *Can we identify the patterns of exploration and tinkering visitors engage in when they move from unproductive to productive states?*

## 2. BACKGROUND & PRIOR WORK

Within the context of this study, it is important to understand what we consider to be “productive” or “unproductive” patterns of practice. Within the learning sciences, there is interest in practices that can be considered productive for novices who are learning computer sciences and engineering [5]. With its focus on the *processes* of creative and improvisational exploration and making, tinkering is recognized as a means for developing a wide range of STEM literacies [22, 13]. Tinkering is predicated on engaging learners in activities centered on the use of scientific tools, processes, and phenomena to explore a problem space through experimentation, trial and error, and refinement [6, 10, 5].

With tinkering's focus on open exploration and learner-defined goals, understanding how and when a learner is engaged in productive tinkering is a challenge. For instance, making mistakes in “traditional” learning environments is often viewed as failure, but in tinkering environments, failure is not only tolerated but celebrated [26]. At their core, tinkering-focused environments enculturate the notion that learners should be allowed to persevere through initial struggles. However, it is not simply that learners

persist, but *why they are persisting* and *how they are persisting* [27]. With persistence, it is critical that learners actively move towards new solutions or problem conceptualizations, or they risk getting stuck in cycles of unproductive perseverance [23].

In museum settings, understanding when visitors are engaging in productive versus unproductive practices *and* having museum facilitators monitor these states is a challenge. This is especially true in open-ended exploratory exhibits in which multiple visitors can engage and leave at different times (rather than having well-defined beginning and end points) and can interact with the exhibit at multiple granularities (e.g., alone, in groups, or simultaneously with strangers). However, if we can develop ways for capturing visitors' hidden productive and unproductive states, we open up the possibility for understanding underlying patterns in their tinkering and learning and providing critical information to researchers, designers, and museum facilitators.

## 2.1 Tabletop Interfaces and Engineering

There is significant research into the role the “programming” environment plays in supporting novices in exploring and tinkering when learning computer science and engineering [20, 5]. Tangible engineering platforms, such as “snap together circuits” (e.g., *snaptcircuits.net*), allow novices to physically manipulate objects as they tinker and explore engineering concepts, providing clear feedback on their process (with pieces clearly fitting together, or lighting up when properly connected). Such interfaces can reduce learner overhead, freeing them to focus on exploration.

With their ability to support multiple visitors simultaneously and in promoting social interactions, interactive tabletops are increasingly used in science and engineering museum research [9, 1]. In general, interactive tabletops are well suited for supporting engineering practices as they promote greater co-awareness of peers' work [35], and can provide increased opportunities for others to monitor and provide feedback [20, 33]. The addition of tangible blocks (blocks that are recognized by the tabletop when placed on its surface) can further support visitors' engagement with engineering practices by allowing them to quickly try out ideas [16] and more generally explore and tinker.

While tabletops are great for supporting collaborative engineering learning, they can make it more difficult for museum explainers to know the state of tinkering of any one visitor. Similar to the problems teachers face with laptop lids [29], the flat surface of the multitouch tabletop can obscure visitors' interactions, forcing explainers to “hover” in order to know what visitors are doing. Even if explainers do hover, keeping track of multiple visitors' states manually (to know when and where they are needed) would be nearly impossible. In response, we need to develop models that can give us insight into visitor states, particularly in real-time.

## 2.2 Markov and Hidden Markov Models

A Markov decision process (MDP) is defined by its state set  $S$ , and transition probabilities  $P$  [41] – assuming identical actions between states, and identical rewards for each transition. This is represented as a graph, called a Markov Model, which depicts that given a state  $s$ , the probability of transitioning to any of the other states  $s'$  is  $T(s, s')$ . In a Markov model, transition probabilities are calculated given a sequence of user states. Calculating (and then visualizing) the likelihood of a transition between states has many potential uses: identifying optimal action sequences in Intelligent Tutoring Systems towards success and using these to provide hints to users [3]; or classifying and identifying common student errors and technical problems to reduce their occurrence [15].

Hidden Markov Models (HMMs), as their name suggests, are Markov Models of *hidden* states. These are not directly observed in the input sequences, but, rather, they exist as aggregated “descriptions” of a user's visible states or “action events” [17]. These have been used to classify users through their navigation or content access patterns [12] and characterize student behaviors in computer-based inquiry learning environments [17]. HMMs require: an input sequence of visible states; an initial transition table providing a starting estimate for the transition probabilities between the hidden states; and an emission table with the probabilities of each of the visible states given each hidden state. Initialization and verification for an HMM-based learning model is an important step, as inappropriate initialization might result in the model getting stuck in local minima [7]. After appropriate initialization via the transition and emission tables, the HMM labels each input state with the corresponding hidden states, and gives the transition probabilities between the hidden states.

## 3. DESIGNING AN OPEN-ENDED TABLETOP ENGINEERING EXHIBIT

### 3.1 The *Oztoc* Exhibit

In order to address our research goals, we are building upon an existing multitouch tabletop exhibit at a large urban science museum. The exhibit, named *Oztoc* [19], situates visitors as electrical engineers called in to help fictional scientists who have discovered an uncharted aquatic cave teeming with never-before documented species of aquatic life (Figure 1). The creatures who live in this cave are bioluminescent, and visitors are asked to help design and build glowing “fishing lures” to attract the “fish” so that scientists can better study them. Visitors place wooden blocks, which act as electrical components (i.e., batteries, resistors, Light Emitting Diodes or LEDs, and timers), on the interactive table to create simple circuits (which the table recognizes the blocks via fiducial symbols – see Figure 1).



Figure 1. Visitors assemble virtual circuits using wooden blocks that represent resistors (1), batteries (2), timers (3), and different colored LEDs (4). Visitors make circuit connections (depicted as lines on the tabletop - 5) by bringing the positive and negative terminals of the blocks (augmentations displayed by the table) in contact with one another. Creating a successful circuit (one that has the correct ratio of resistors, batteries, and LEDs) causes LEDs to glow and lures creatures attracted to it for cataloging.

*Oztoc*'s narrative aims to give learners a situated context in which to engage in engineering practices. To avoid many of the problems of other engineering and making exhibits [19], we wanted *Oztoc* to give visitors some freedom in choosing their own

goals (e.g., which types of fish to target) while still giving them a common set of materials and processes.

## 4. METHODOLOGY AND VISITORS

*Oztoc* is installed in an enclosed exhibit space just off the main floor of a large urban science center. A lollipop sign just outside the exhibit space indicates when videotaping will take place in the exhibit, allowing visitors to decide to enter or to return when data collection is not active. Researchers were present for technical support to museum staff only. Video data was collected via cameras placed in the exhibit space, audio from a boundary microphone, and telemetry data using the ADAGE system [31].

Visitors in this study come from a wide range of backgrounds and SES. Visitors were also multi-generational and came to the exhibit alone, as families, and in large groups.

### 4.1 Establishing Visitor Start and Stop Times

Unlike many other exhibits, *Oztoc* does not have pre-determined start and stop events (such as the beginning or end of a simulation or game) – it is a continual process in which visitors enter and leave, often at different times. Therefore, in order to accurately separate visitors’ sequences of activities, we developed a method for determining when visitors entered or exited the exhibit. Given all actions performed at each of the table’s four “zones” over a single day, we found that if a zone was inactive and empty over a set period of time – the “inactivity interval” (InI), the next event in that zone indicated a new visitor. We evaluated an InI ranging from 10-120 seconds, and the InI did not change significantly between 45-120 seconds. As such, we validated the 45-second InI with hand-labeled data. Our 45 second InI achieved full accuracy for the 2-hour sample of video data that we hand-labeled.

### 4.2 Coding Visitor Events

We needed to establish a granularity of the telemetry data that would allow us to understand the state of visitors’ tinkering at any moment. Based on previous research on visitors’ interactions with the exhibit [19], we chose to look at the events when visitors successfully created a circuit (denoted in the logs as *MakeCircuitCreate*). This state was particularly useful as a circuit was logged in ADAGE *even if the circuit “didn’t work”* (i.e., the LEDs were not supplied correct voltage), giving us insight into visitors’ process exploring different circuit configurations, solution states, and goals. By leveraging visitors’ histories at the table, we could mine for more complex relationships between their current circuit, previously made circuits, and those made by others at the table since their arrival. We then automatically coded each visitors’ *MakeCircuitCreate* event using four binary codes (see Table 1).

**Table 1.** Binary codes for *MakeCircuitCreate* events

Marker	Code	Description
Is the circuit complex?	S/C	The completed circuit has 3+ components
Does the circuit work?	N/W	The circuit successfully lights up
Is the circuit unique for self?	R/U	This is the first time the visitor has made this circuit
Is the circuit unique at the table?	E/O	No one else at the table has made a circuit with the same set of components

#### 4.2.1 Is the circuit complex? (coded S or C)

Earlier analysis of visitors’ interactions with the exhibit showed that most visitors (if they made *any* circuits) only made the basic three-component circuit (one LED, one resistor, and one battery) [34]. As such, the building of a complex (more than three component) circuit was a key indicator that visitors were trying out more complex configurations. If a circuit had three or less components we scored it an **S** (*indicating it was “simple”*), any circuit that had more than three components was scored a **C** (*indicating it was a complex circuit*). It is important to note that this code is not concerned with *whether or not the circuit works*, only the number of components used.

#### 4.2.2 Does the circuit work? (coded N or W)

Understanding the relationship between the individual components and making a working circuit is a critical factor in determining the success of an exploration. As such, each completed non-working circuit was coded with an **N** and each completed working circuit with a **W**.

#### 4.2.3 Is the circuit unique for self? (coded R or U)

Because problem solving through tinkering is characterized by exploration and iteration [26], we coded if a circuit created by a visitor was “unique” for them (i.e., had they constructed the exact same circuit earlier). A visitor who received a **W** on the *does the circuit work* code might seem to be engaging in productive tinkering; however, if they are simply repeating their first circuit over and over, this might indicate a failure to try out new ideas or expand their problem definition. To mark if a visitor’s circuit was unique we coded it with a **U**, if it was a repeat of a past circuit we assigned it an **R**.

#### 4.2.4 Is the circuit unique at the table?

Finally, *Oztoc* is designed to support visitors in collaborating with and building off others’ to advance their own exploration. This use of others’ constructed artifacts as a basis for one’s own work has been termed “echoing” and has been shown to be an important part in open-ended and exploratory tinkering [34]. We considered a circuit to be an echo if it had the same number of each component type (battery, resistors, and LEDs). If a visitor’s circuit echoed of one of their peers’, we assigned it an **E** (for echo); if the circuit was unique to the table, we assigned it an **O** (for original).

The process described above resulted in every *MakeCircuitCreate* event for each visitor receiving an easily interpretable four-digit code. For instance, a *MakeCircuitCreate* that was assigned a code of **SWRO** means that it was a simple (**S**), working circuit (**W**) that was a repeat of a past circuit made by the visitor (**R**), but had not been created by anyone else at the table since this visitor started playing (**O**). These codes provided a rich and detailed source of data for passing into a Hidden Markov Model to see if we could identify if visitors were productive or unproductive at any point during their engagement with the exhibit. Since the *MakeCircuitCreate* events were chronologically ordered and separated per visitor, we could further examine which created circuits led to important state shifts.

### 4.3 Coding for productive behaviors

Using the coded descriptions of the circuits created by the visitors, we wanted to make an HMM that identifies when a visitor was behaving “productively”, or not. For this purpose, building off of previous research [19], two members of the research team discussed and identified patterns of *MakeCircuitCreate* that were indicative of productive and unproductive tinkering.

One of the key patterns identified focuses on visitors trying out new circuit configurations to fix errors in their existing circuits or to develop new circuits (denoted by a **U** in codes). For instance, if a visitor attempted a few different non-working circuits – seen as a sequence of **SNUO**, **SNRO**, **SNUO**, **SNUO** (with the second circuit being a duplicate of a past circuit) – the sequence seems to indicate that while the visitor’s circuits do not work (indicated by the **Ns**), they are trying out new approaches and expanding their exploration. This sequence of activities was coded as productive behavior. If the visitor’s continued exploration results in cycle of repeated circuits coded with **Rs** (repeats) or did not eventually make a working circuit (coded with a **W**), we coded these actions as falling into unproductivity, as the visitor seems to have failed to figure out how to make a working circuit.

Similarly, a visitor might make a working circuit (indicated by a **W** in their circuit code) and repeat it over and over again (e.g., a series of circuits such as **SWRO**, **SWRO**, **SWRO**). This would seem to indicate that the visitor is repeating past success and is failing to consider new problem spaces or avenues for exploration.

A change of **SNRO** to **SNOE** – trying a new (**U** = self-unique) circuit that someone else on the table has made (**E** = table-echo), might be an attempt at gaining understanding by looking at what other visitors are doing – and was coded as productive depending on how many failed attempts the visitor had already made.

With this understanding, the first two authors first coded 200 circuit creates, and established reliability with 91% agreement. They then coded 644 of the (total of 3952) circuits made in player one’s zone (one of the four game quadrants) on the table.

#### 4.4 Training the Hidden Markov Model

We used our manually coded states to calculate appropriate values for the emission table for our HMM. The emission table was calculated by seeing how often a certain circuit code was marked as productive (or unproductive) as a proportion of all the circuits coded with the same hidden state. For instance, of all the circuits coded as productive, 5.6% of those were coded as **CWUO** and 6.49% were coded as **CWUE** (from the list of 16 circuit-codes), these values were then used to populate the HMM emission table.

We needed to identify when new visitors started playing at the table to ensure that the new visitors circuits were not considered as a continuation of earlier visitors. To do this we added events (a **0000** code) in the sequence of circuit-codes to signify new visitors. This brought up the question of whether the HMM should code new visitors as unproductive, productive, or another state altogether. To be able to show what state people tended to leave and begin at in the final transition table, we chose to make the visitor change a distinct state in our HMM even though it was not

a hidden state, and is equivalent to a direct observation.

We used Python’s `hmmlearn` package to create our HMM, which has the limitation of only looking for local optima in calculating the probabilities of transitioning from one hidden state to another. To account for this, different initial transition table values were tried. Results showed that the HMM stably converged to the final transition table (Figure 3).

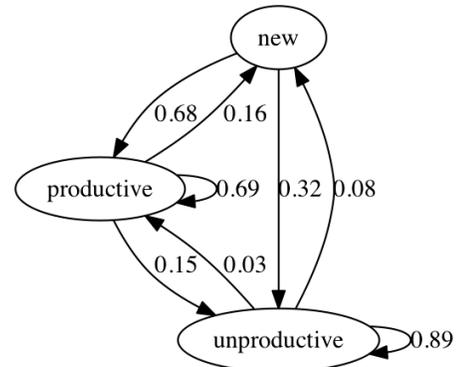


Figure 3. HMM for productive/unproductive states in Oztoc

## 5. FINDINGS

This study has two important findings, with the first finding acting as the scaffold for the second: First, the recognition of when visitors are engaged in productive or unproductive exploration; and second, the understanding of which sequences of events typically lead visitors from prolonged (at least three) consecutive unproductive states to a productive state.

### 5.1 Running HMM on Visitors’ Circuits

The result of the HMM’s final transition table revealed several interesting results (Figure 3). The HMM model shows that the probability of a new visitor beginning productively is 68%, versus 32% for beginning unproductively. Being unproductive appears to be a more stable state than being productive (89% versus 69%, respectively), and moving from unproductivity to productivity is also rarer than the reverse (3% versus 15%). The model also shows that the chances of leaving the table while being productive is higher than of leaving while unproductive (16% versus 8%).

To validate the predictive accuracy of the HMM’s classification we used a general agreement score, the calculated the area under the curve (AUC) of the model’s receiver operating characteristic (ROC) and Cohen’s Kappa as compared to our 644 hand-coded labels. Our HMM had 94% agreement, scored an ROC/AUC score of 0.79, and a Cohen’s Kappa of 0.59, which were

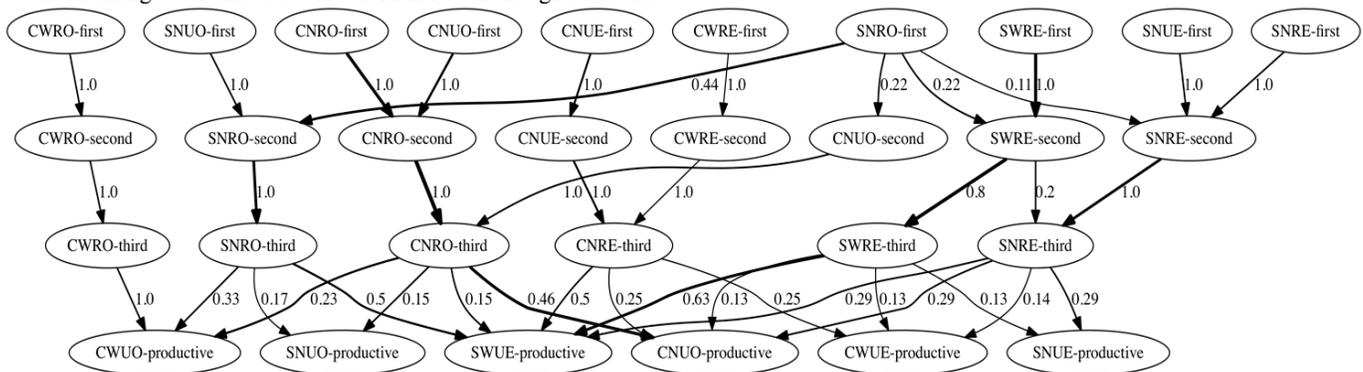


Figure 4. Markov model for visitors who transition from three consecutive unproductive states to a productive state

satisfactory measures to consider the HMM's coding reliable.

## 5.2 Developing Markov Models of Moving from Unproductive to Productive States

After the HMM tagged the circuits as productive or unproductive, we wanted to understand what patterns of activity preceded visitors becoming productive. We were particularly interested in sequences in which visitors struggled (had several unproductive moves) and then moved to a productive state. For this, we built a list of when a visitor had three consecutive unproductive circuits immediately followed by a productive circuit. We pruned the sequences that only happened once (as they were uninformative).

Once we had a list of the 4 step chains, we made a Markov model depicting the sequences of actions visitors followed when moving from unproductive to productive (Figure 4). This model also showed the likelihood that a visitor making a certain coded circuit would make another specific circuit next. The thickness of the lines between nodes indicates how many times a path occurred.

## 6. DISCUSSION

This paper outlined how the combination of Hidden Markov Models (HMMs) and Markov chains could be used to effectively predict when visitors were engaging productively or unproductively in an open-ended, exploratory museum exhibit. A closer examination of the HMM revealed several unexpected visitor behaviors. Visitors more often than not (68%) begin productively, but are less likely to stay productive (69%) than unproductive (89%) once in that state (Figure 3). The first finding is not entirely surprising, as our model considers open, thoughtful exploration as productive and it is hard to consider a visitor's "first move" as anything more than a first "exploratory step". This view is partially validated by the lower likelihood of staying productive – indicating many visitors fail to make thoughtful adjustments to their tinkering or explore new definitions of the problem space. This is compounded by instances where visitors make a successful circuit then "settle into" making the same circuit over and over. These findings are supported by the high percentage of visitors who either stay unproductive (89%) or leave the exhibit (8%). It should be noted that 69% is still a very high number of visitors staying productive and is probably further understated by the "first circuit" effect described above.

Another interesting finding is the high likelihood of leaving the table while being productive (16% compared to leaving the table while unproductive – 8%). On the surface this is surprising, as one would expect visitors to give up due to frustration more often than while 'succeeding'. The results may indicate that visitors who "figure out" multiple facets of the exhibit continue to engage productively until they leave – some of these effects have been covered in other research on this project [19]. Another possible explanation is that visitors started to engage in productive behaviors (such as trying something new that they had not done before or echoing the work of another visitor) that didn't immediately result in positive feedback from the system (e.g., capturing a fish) and they gave up.

When looking at the Markov model of unproductive to productive states we uncovered several interesting sequences (see Figure 4). For instance, unproductive circuits coded as **CNUO** (complex, not-working, unique, original) always went to **CNRO** (complex, not-working, repeated, original), followed by another **CNRO**, which finally led 15% of the time to a productive **SWUE** – a simple, working circuit that they had never made earlier, but had been made on the table in front of them by someone else! This is an interesting phenomenon – that a visitor, after some initial

failures at making working circuits with a high level of complexity, likely saw a simple working circuit made by someone else, and then switched to echoing that circuit. The ability to see the work of others helped them overcome their own unproductive exploration. We see similar patterns in the Markov chain sequences **SNUO** -> **SNRO** -> **SNRO** -> **SWUE**; and **SNUE** -> **SNRE** -> **SNRE** -> **SWUE**, highlighting the role that making the work of others engaged in parallel tasks visible can serve in helping visitors move from unproductive to productive states.

## 7. CONCLUSIONS AND NEXT STEPS

Tinkering and exploration are powerful ways for learners to engage in science and engineering practices [24]; however, supporting learners to productively engage in open-ended learning is inherently difficult, especially in museums [13]. Much of this has to do with the inherent chaos of the museum environment – hundreds (even thousands) of visitors interact with an exhibit in a day, coming and going at different times, and with different expectations and goals. For facilitators in exploratory exhibits, keeping track of the flow of participants and the state of their individual and collective tinkering efforts is nearly impossible.

This paper illustrates how data mining and analytics can help disambiguate the actions of visitors in such exhibits and uncover the hidden states of their tinkering. In addition to shedding light into how visitors productively and unproductively tinker, this work holds considerable potential for developing new ways to support facilitators. Knowing when and how visitors are engaging in unproductive exploration can help us develop complementary applications to help facilitators know when and how they are most needed. Knowing how visitors tend to move from unproductive to productive states can further guide us in developing strategies and scaffolds to help facilitators better engage with visitors.

While tablet applications have been used to provide added contextual information and alert museum facilitators about the visitors' interactions with exhibits in real-time [30], they have done so only using surface features, without understanding visitors' exploration 'states'. By uncovering the particular ways that a visitor is struggling, and understanding the subtle ways they can be "nudged" towards more productive exploration, there is the potential for dramatically influencing visitors' exploration and learning. By interceding at moments where visitors are struggling or are likely to give up, we may increase visitors dwell time, which has been shown to increase their collaboration with others, and domain learning [9]. In response, we are developing a tablet application that uses our models to support facilitators in real-time to understand how such applications compare to approaches that rely only on surface measures and unmodeled log data.

## 8. REFERENCES

- [1] Allen, S. (2002). Designs for learning: Studying science museum exhibits that do more than entertain. *Sci. Ed.* 88, S1 (2004), S17-S33.
- [2] Allen, S., & Gutwill, J. P. (2009). Creating a program to deepen family inquiry at interactive science exhibits. *Curator: The Museum Journal*, 52(3), 289-306.
- [3] Barnes, T., & Stamper, J. (2008, June). Toward automatic hint generation for logic proof tutoring using historical student data. In *Intelligent tutoring systems* (pp. 373-382). Springer Berlin Heidelberg.
- [4] Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6), 1554-1563.

- [5] Berland, M., Martin, T., Benton, T., Smith, C. P., & Davis, D. (2013) Using Learning Analytics to Understand the Learning Pathways of Novice Programmers. *Journal of the Learning Sciences* 22(4), 564-599.
- [6] Bevan, B., Gutwill, J. P., Petrich, M., & Wilkinson, K. (2015). Learning through STEM-rich tinkering: Findings from a jointly negotiated research project taken up in practice. *Science Education*, 99(1), 98-120.
- [7] Bicego, M., & Murino, V. (2004). Investigating hidden Markov models' capabilities in 2D shape classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2), 281-286.
- [8] Blikstein, P. (2013, April). Multimodal learning analytics. In Proceedings of the third international conference on learning analytics and knowledge (pp. 102-106). ACM.
- [9] Block, F., Hammerman, J., Horn, M., Spiegel, A., Christiansen, J., Phillips, B., ... & Shen, C. (2015, April). Fluid Grouping: Quantifying Group Engagement around Interactive Tabletop Exhibits in the Wild. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 867-876). ACM.
- [10] Dorn, B., & Guzdial, M. (2010). Learning on the job: Characterizing the programming knowledge and learning strategies of Web designers. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 703-712). New York, NY: ACM.
- [11] Feder, M. A., Shouse, A. W., Lewenstein, B., & Bell, P. (Eds.). (2009). *Learning Science in Informal Environments: People, Places, and Pursuits*. National Academies Press.
- [12] Fok, A. W., Wong, H. S., & Chen, Y. S. (2005, July). Hidden markov model based characterization of content access patterns in an E-Learning environment. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on* (pp. 201-204). IEEE.
- [13] Gutwill, J. P., & Allen, S. (2010). Facilitating family group inquiry at science museum exhibits. *Science Education*, 94(4), 710-742.
- [14] Gutwill, J. P., Hido, N., & Sindorf, L. (2015). Research to practice: Observing learning in tinkering activities. *Curator: The Museum Journal*, 58(2), 151-168.
- [15] Heathcote, E. A., & Prakash, S. (2007). What your learning management system is telling you about supporting your teachers: monitoring system information to improve support for teachers using educational technologies at Queensland University of Technology.
- [16] Hornecker, E., & Buur, J. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proc. CHI 2006*, ACM Press (2006), 437-446.
- [17] Jeong, H., Gupta, A., Roscoe, R., Wagster, J., Biswas, G., & Schwartz, D. (2008, June). Using hidden Markov models to characterize student behaviors in learning-by-teaching environments. In *Intelligent Tutoring Systems* (pp. 614-625). Springer Berlin Heidelberg.
- [18] Land, S. M. (2000). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3), 61-78.
- [19] Lyons, L., Tissenbaum, M., Berland, M., Eydt, R., Wielgus, L., & Mechtley, A. (2015, June). Designing visible engineering: supporting tinkering performances in museums. In *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 49-58). ACM.
- [20] Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). [The Scratch Programming Language and Environment](#). *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 4 (November 2010).
- [21] Martin, L. (2015). The promise of the Maker Movement for education. *Journal of Pre-College Engineering Education Research*, 5(1), 30-39.
- [22] Martinez, S. L., & Stager, G. (2013). Invent to learn: Making, tinkering, and engineering in the classroom.
- [23] McFarlin, D. B., Baumeister, R. F., & Blascovich, J. (1984). On knowing when to quit: Task failure, self-esteem, advice, and nonproductive persistence. *Journal of Personality*, 52(2), 138-155.
- [24] Petrich, M., Wilkinson, K., & Bevan, B. (2013). It looks like fun, but are they learning. *Design, make, play: Growing the next generation of STEM innovators*, 50-70.
- [25] Rabiner, L. R., & Juang, B. H. (1986). An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(1), 4-16.
- [26] Resnick, M., & Rosenbaum, E. (2013). Designing for tinkering. *Design, make, play: Growing the next generation of STEM innovators*, 163-181.
- [27] Ryoo, J. J., Bulalacao, N., Kekelis, L., McLeod, E., & Henriquez, B. (2015). Tinkering with "Failure": Equity, Learning, and the Iterative Design Process. In *annual FabLearn conference*. Palo Alto, CA: Stanford University.
- [28] Schweingruber, H., Keller, T., & Quinn, H. (Ed.). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*: National Academies Press, 2012.
- [29] Sharples, M. (2013). Shared orchestration within and beyond the classroom. *Computers and Education*, 69, 504-506.
- [30] Slattery, B., Lyons, L., Pazmino, P. J., Silva, B. L., & Moher, T. (2014). How interpreters make use of technological supports in an interactive zoo exhibit. In *11th International Conference of the Learning Sciences (ICLS 2014)*.
- [31] Stenerson, M. E., Salmon, A., Berland, M., & Squire, K. Adage: an open API for data collection in educational games. In *Proc, SIGCHI Play 2014*, ACM Press (2014), 437-438.
- [32] Sutton, R. & A. Barto. Reinforcement Learning: An Introduction, 1998, The MIT Press, Cambridge, MA.
- [33] Tissenbaum, M., Berland, M., & Lyons, L. (in review). CCLM Framework: Understanding Collaboration in Constructionist Tabletop Learning. *International Journal of Computer Supported Collaborative Learning*.
- [34] Wielgus, Tissenbaum & Berland (in review) Echoes paper
- [35] Xambó, A., Hornecker, E., Marshall, P., Jorda, S., Dobbyn, C., & Laney, R. (2013). Let's jam the reactable: Peer learning during musical improvisation with a tabletop tangible interface. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(6), 36.
- [36] Yoon, S. A., Elinich, K., Wang, J., Schoonveld, J. B., & Anderson, E. (2013). Scaffolding informal learning in science museums: How much is too much?. *Science Education*, 97(6), 848

# Learning Curves for Problems with Multiple Knowledge Components

Brett van de Sande

Pearson Education

brett.vandesande@pearson.com

## ABSTRACT

Learning curves have proven to be a useful tool for understanding how a student learns a given skill as they progress through a curriculum. A learning curve for a given Knowledge Component (KC) is a plot of some measure of competence as a function of the number of opportunities the student has had to apply that KC. Consider the case where each problem-solving step is recorded by, for instance, by an intelligent tutoring system. In this case, one normally assigns a unique KC to each problem-solving step and the construction of the associated learning curves is straightforward. On the other hand, many online homework systems only evaluate the student's final answer to a problem. In that case, the student has generally applied a number of KCs to find the answer and their performance on the problem is some composite of their mastery of all of the requisite KCs. In this paper, we propose a simple method for generating learning curves for multiple-KC problems that is independent of any particular theory of learning. In the case where there is only one KC per problem, the method reduces to the ordinary learning curves. We demonstrate this method using a set of artificially generated student data.

## Author Keywords

Learning Curves, Knowledge Components

## ACM Classification Keywords

I.2.6 Learning: Knowledge acquisition

## INTRODUCTION

The increased use of online homework systems and intelligent tutor systems (ITS) means that ever-increasing amounts of student log data is available for analysis. This data can be used to answer two important questions: what skills are students learning and how quickly are they learning them? To be more precise, we can equate skills with Knowledge components (KCs): small bits of information needed to solve a problem [11, 3]. KCs generally have some sort of pre-requisite

relations: For example, you cannot apply the area of a circle formula  $A = \pi r^2$  unless you first know the definition of "radius of a circle." However, aside from prerequisites, a KC can, by definition, be mastered independently from other KCs. This definition assumes that KCs are *context independent*. That is, the student's ability to apply that KC correctly or quickly does not depend on the particular problem the student is solving or the other KCs needed to solve that problem.

Since KCs are *defined* to have these properties, then it remains to be seen whether, and in what cases, they are a useful description of skill acquisition. One way to determine how well the KC picture is working is to examine the associated learning curves. If the curves are smooth, increasing/decreasing monotonically (depending on the measure of competence), and independent of context, then the KC picture is working.

Learning curves are a plot of some measure of mastery of a skill as a function of the number of opportunities that the student has had to apply that skill. Possible measures of mastery include:

- number of errors made before correctly applying the KC,
- time taken to correctly apply a KC,
- "assistance score," number of errors plus number of requests for help before completing a step, and
- "correctness", whether the student applied the KC correctly without any preceding errors or requests for help.

In the following, we will use "correctness" as our measure of competence for a given skill.

In a typical Intelligent Tutoring System (ITS), the student enters each problem-solving step into the tutor system. It is natural, in that case, to associate one KC with each student input and it is relatively straightforward to construct the associated learning curves. However, many online homework systems only require the student to enter their final answer to a problems into the system. In this case, a single input is the entire problem and it is natural to associate multiple KCs to each student input.

If multiple KCs are associated with a single input, then the construction of learning curves is more difficult. If the student gets the problem wrong, which KC is responsible? This is sometimes called the "assignment of blame problem" [7,

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single spaced.

Every submission will be assigned their own unique DOI string to be included here.

**Table 1. List of definitions and quantities**

$k, l, m$ : label representing a KC.

$t, u, v$ : label representing opportunity number for some KC.

$p$ : label representing an exercise.

$s$ : the student.

$P_{t,k}$  is a model parameter representing the probability that a student will apply KC  $k$  correctly on opportunity  $t$ .  $P_{t,k} \in [0, 1]$ .

$\xi_{s,p}$  is the model-given probability that student  $s$  will get problem  $p$  correct.

$C_{t,k}$  is the number of students in the dataset who correctly applied KC  $k$  on opportunity  $t$ .

$I(\mathbf{t}, \mathbf{k})$  is the number of students who got an exercise containing KCs  $\mathbf{k} = \{k_1, k_2, \dots\}$  incorrect where  $\mathbf{t} = (t_1, t_2, \dots)$  is a vector of corresponding opportunities. This exercise represents opportunity  $t_a$  for the student to apply KC  $k_a$ .

$\mathcal{T}_{s,p}$  is the set of KC, opportunity pairs such that problem  $p$  is opportunity  $t$  for student  $s$  to apply KC  $k$ .

6, 5]. In the following, a simple method is proposed which addresses the assignment of blame problem while making a minimum of theoretical assumptions, allowing one to construct learning curves for exercises with multiple KCs. Our strategy is to introduce a model where every point on each learning curve is identified as a model parameter. These model parameters, and their associated errors, are then determined by a maximum likelihood fit to student log data. In the case of a single KC per problem/step, this reduces to the usual learning curves.

### LEARNING CURVE MODEL

A number of studies have addressed the multiple-KC problem in the context of some model of learning, such as Bayesian Knowledge Tracing or Performance Factor Analysis [2, 4]. In the present work, our goal is simply to construct learning curves using a minimum number of model assumptions. Note that conventional learning curves themselves make two major assumptions:

1. They average over students. This corresponds to a model that does not have any student-specific parameters.
2. They ignore the problem context. This corresponds to a model that does not have any problem-specific parameters.

In fact, the construction of a learning curve is equivalent to fitting the student log data to a model containing a parameter representing each KC and step. In other words, if I define  $P_{t,k}$  as the probability that a student will correctly apply KC  $k$  at opportunity  $t$ , and determine  $P_{t,k}$  by fitting to the student log data, then plotting of  $P_{t,k}$  versus  $t$  is a learning curve for KC  $k$ .

This gives us a way forward in the multiple-KC case. We define a model having parameters  $\{P_{t,k}\}$ . The associated log-likelihood is

$$\log(\mathcal{L}) = \sum_{s,p \in \mathcal{C}_s} \log(\xi_{s,p}) + \sum_{s,p \in \mathcal{I}_s} \log(1 - \xi_{s,p}) \quad (1)$$

where  $s$  is the student,  $p$  is the problem,  $\mathcal{C}_s$  is the set of problems  $s$  got correct, and  $\mathcal{I}_s$  is the set of problems  $s$  got incorrect. Also,  $\xi_{s,p}$  is the model-given probability that student  $s$  will get problem  $p$  correct.

We will assume that the student must apply *all* of the associated KCs to solve a given exercise correctly. This is sometimes called a “conjunctive model” and is a good approach for typical K-12 math exercises [8]. This means that the total probability of success is the product of the KC probabilities:

$$\xi_{s,p} = \prod_{t,k \in \mathcal{T}_{s,p}} P_{t,k} \quad (2)$$

where  $\mathcal{T}_{s,p}$  is the set of KCs and opportunities such that problem  $p$  is opportunity  $t$  for student  $s$  to apply KC  $k$ .

To construct  $\mathcal{T}_{s,p}$ , one needs a list of KCs associated with each exercise  $p$ , sometimes referred to as the “Q-matrix” [10]. In this discussion, we will assume that the Q-matrix is known, perhaps determined by the problem author or a domain expert.

### Numerical Calculation

The likelihood given by Eqn. (1) is rather inconvenient for large numerical calculations. Instead, we will introduce variables that aggregate over student and exercise. Define  $C_{t,k}$  to be the number of students in the dataset who correctly applied KC  $k$  on opportunity  $t$ . Likewise, define  $I(\mathbf{t}, \mathbf{k})$  to be the number of students who got an exercise containing KCs  $\mathbf{k} = \{k_1, k_2, \dots\}$  incorrect where  $\mathbf{t}$  is a vector of associated opportunities. This exercise represents opportunity  $t_a$  for the student to apply KC  $k_a$ . Then, the log-likelihood can be written as

$$\log(\mathcal{L}) = \sum_{t,k} C_{t,k} \log(P_{t,k}) + \sum_{t,k} I(\mathbf{t}, \mathbf{k}) \log(1 - \Gamma(\mathbf{t}, \mathbf{k})) \quad (3)$$

where  $\Gamma(\mathbf{t}, \mathbf{k})$  is the probability that a student with opportunity vector  $\mathbf{t}$  will have success on a problem containing KCs  $\mathbf{k} = \{k_1, k_2, \dots\}$ . Following Eqn. (2),  $\Gamma(\mathbf{t}, \mathbf{k})$  is a product over the associated probabilities:

$$\Gamma(\mathbf{t}, \mathbf{k}) = \prod_a P_{t_a, k_a} \quad (4)$$

Note that the first term of Eqn. (3) has a much simpler form than the second term. This is due to our use of a conjunctive model. If a student gets an exercise “correct” then we know without ambiguity that they applied all of the associated KCs correctly. However, if they get a problem wrong, then it is not clear which KC is to blame and the associated probabilities must be considered jointly.

Let  $\{\hat{P}_{t,k}\}$  be the model parameters at the maximum likelihood point.  $\{\hat{P}_{t,k}\}$  can be found numerically by maximizing the log-likelihood, Eqn. (3) subject to the constraints

**Table 2. KC content of the artificial homework set. Students completed the first eight problems in the given order and the remaining problems in random order; they completed between 15 and 20 problems total.**

1	2	3	4	5	6	7	8	9	10
A	A	A	A	B	B	B	B	A	B
11	12	13	14	15	16	17	18	19	20
A	B	AB							

$0 \leq P_{t,k} \leq 1$ . For convenience, the *Mathematica* function **FindMaximum**, was used to calculate the maximum of  $\log(\mathcal{L})$ . However, any optimization algorithm that enforces constraints and uses information about the gradient of the function should work as well.

### Error analysis

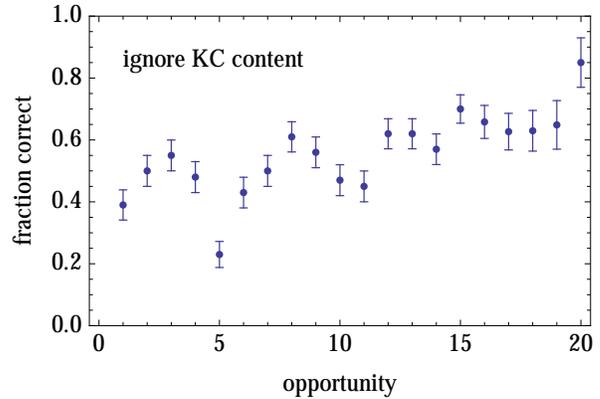
It is important to calculate the standard errors associated with the model parameters. Unlike the single KC per problem case, the model parameters may be strongly correlated and the errors can have unexpected values. In addition, the error analysis can elucidate any cases where the model parameter cannot be determined from the data (we will discuss this further in the conclusion).

Before finding the errors, we need to examine the the maximum likelihood point and identify any parameters that lie on the boundaries  $\hat{P}_{t,k} = 0$  or 1. The likelihood function  $\mathcal{L}$  is not stationary in these parameters at the maximum likelihood point, so the error analysis cannot be applied to them; they should be not be included in the Hessian matrix below, Eqn (5). In practice, this should not a significant issue, since  $\hat{P}_{t,k} = 0$  or 1 typically occurs when there are just a few student problem-solving instances for a given  $t$  and  $k$ .

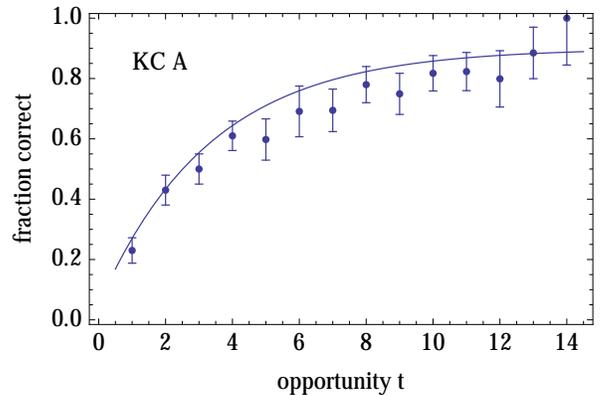
For a maximum likelihood fit, the standard errors associated with the model parameters can determined using the following procedure [1, 9]. First, we find the Hessian matrix associated with  $P_{t,k} = \hat{P}_{t,k}$ . The matrix elements of the Hessian are given by

$$\frac{\partial^2 \log(\mathcal{L})}{\partial P_{t,k} \partial P_{u,l}} \Big|_{P_{v,m} = \hat{P}_{v,m}} = - \frac{1}{\hat{P}_{t,k} \hat{P}_{u,l}} \sum_{\mathbf{t}, \mathbf{k}} \frac{I(\mathbf{t}, \mathbf{k}) \Gamma(\mathbf{t}, \mathbf{k})}{(1 - \Gamma(\mathbf{t}, \mathbf{k}))^2} \Big|_{P_{v,m} = \hat{P}_{v,m}}. \quad (5)$$

To find the standard error associated with each of the model parameters  $\hat{P}_{t,k}$ , we invert the negative of the Hessian matrix and take the square root of the diagonal elements. If this process fails (the Hessian matrix is singular), it is a signal that some of the model parameters cannot be uniquely determined from the given log data. Similarly, if the Hessian matrix is nearly singular, then the associated standard errors will be very large. This will single out any model parameters that cannot be determined from the data.



**Figure 1. Learning curve for the artificial homework set where we assume each problem has the same single KC. Note the jump after opportunity 4 due to the fact that the first four and second four problems have different KCs.**



**Figure 2. Learning curve for KC A. The solid line is the model used to generate the student data and the points with error bars represent the learning curve determined from the student data using our procedure. Note that the error bars for the last few opportunities are larger, due to student attrition.**

### APPLICATION TO STUDENT DATA

To illustrate how this model works, we will generate an artificial student performance dataset. Consider a homework assignment of 20 problems that exercise two KCs, *A* and *B* as detailed in Table 2. We assume that students progress through the first 8 problems in the given order, but solve the remaining 12 problems in random order, completing between 15 and 20 problems. We assume that student mastery for the KCs is given by the functions  $P_{t,A} = 0.9 - 0.85e^{-0.3t}$  and  $P_{t,B} = 0.85 - 0.45e^{-0.1t}$ ; see Figures 2 and 3. We use this model to generate a set of outcomes,  $\mathcal{C}_s$ ,  $\mathcal{I}_s$ , and  $\mathcal{T}_{s,p}$ , for 100 students.

If we ignore the KC content of the problems, we can plot a naïve learning curve for this student data; See Fig. 1. We see a discontinuity at  $t = 4$  due to the change in actual KC content of the problems. The last problems are more difficult, since they involve two skills and so the student performance on them is suppressed.

Next, we use our procedure to generate learning curves and associated errors for this dataset. The results are plotted in

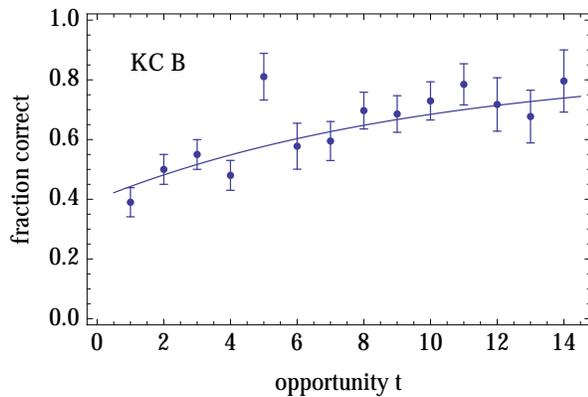


Figure 3. Learning curve for KC B. The high value at  $t = 5$  is a statistical fluctuation: as we increase the number of students, the model parameters will converge to the solid line.

Figs. 2 and 3. As expected, they agree well with the model used to generate the student data. This shows that our method is working. Note that the error bars can vary considerably from point to point.

### CONCLUSION

The primary goal of the approach developed here is to plot learning curves for cases where there are problems (or problem steps) involving multiple KCs. In practice, we find our method to be numerically robust (no problems with local maxima).

However, there is one case where it may fail: if there is a KC that always appears along with another KC for several problems and all the students in the dataset solve nearly the same ordered sequence of problems, then there is no way distinguish between the two KCs for one or more value of  $t$ . This will result in a Hessian matrix that is not positive-definite and the matrix inversion will fail. We believe that this situation will rarely arise in practice, since most datasets involve students in multiple courses, and students are generally not forced to solve problems in a specific order.

In this work, we focused on a “conjunctive model” for combining KCs, as this is likely the most appropriate model for typical math and science exercises. Although the basic strategy we present here could be applied to other models (disjunctive, compensatory) for combining KCs, the details of the associated numerical calculation would look rather different.

Obviously, the next step is to apply this approach to real student data. This would require a set of exercises that have been tagged with multiple KCs, where the mix of KCs vary significantly from exercise to exercise. In addition, the student activity would have to fairly heterogeneous, with different students taking different paths through the exercises.

### ACKNOWLEDGMENTS

This work was supported by Pearson education. I would like to thank Ilya Golden for reviewing the manuscript.

### REFERENCES

1. Edwards, A. W. F. *Likelihood*. Johns Hopkins University Press, 1992.
2. Gong, Y., Beck, J., and Heffernan, N. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In *Intelligent Tutoring Systems*, V. Aleven, J. Kay, and J. Mostow, Eds., vol. 6094 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010, 35–44.
3. Koedinger, K. R., Corbett, A. T., and Perfetti, C. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Sci.* 36, 5 (2012), 757–798.
4. Koedinger, K. R., Pavlik, P. I., Stamper, J., Nixon, T., and Ritter, S. Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing. In *Proceedings of the 3rd International Conference on Educational Data Mining* (2010), 91–100.
5. Nwaigwe, A., and Koedinger, K. R. The Simple Location Heuristic is Better at Predicting Students’ Changes in Error Rate Over Time Compared to the Simple Temporal Heuristic. In *Proceedings of the 4th International Conference on Educational Data Mining* (Eindhoven, the Netherlands, 2011), 71–80.
6. Nwaigwe, A., Koedinger, K. R., Vanlehn, K., Hausmann, R., and Weinstein, A. Exploring alternative methods for error attribution in learning curves analysis in intelligent tutoring systems. *Frontiers in Artificial Intelligence and Applications* 158 (2007), 246.
7. Ohlsson, S. Towards Intelligent Tutoring Systems that Teach Knowledge Rather than Skills: Five Research Questions. In *New Directions in Educational Technology*, E. Scanlon and T. O’Shea, Eds., no. 96 in Nato ASI Subseries F. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
8. Pardos, Z. A., Beck, J. E., Ruiz, C., and Heffernan, N. T. The Composition Effect: Conjunctive or Compensatory? An Analysis of Multi-Skill Math Questions in ITS. In *Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings*, UNC-Charlotte, Computer Science Dept. (Montreal, Canada, June 2008), 147–156.
9. Pawitan, Y. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford University Press, June 2001.
10. Tatsuoka, K. K. Rule Space: An Approach for Dealing with Misconceptions Based on Item Response Theory. *Journal of Educational Measurement* 20, 4 (Dec. 1983), 345–354.
11. VanLehn, K. The Behavior of Tutoring Systems. *Int. J. Artif. Intell. Ed.* 16, 3 (Jan. 2006), 227–265.

# A Nonlinear State Space Model for Identifying At-Risk Students in Open Online Courses

Feng Wang and Li Chen  
Department of Computer Science  
Hong Kong Baptist University, Hong Kong  
{fwang, lichen}@comp.hkbu.edu.hk

## ABSTRACT

How to identify at-risk students in open online courses has received increasing attention, since the dropout rate is unexpectedly high. Most prior studies have focused on using machine learning techniques to predict student dropout based on features extracted from students' learning activity logs. However, little work has viewed the dropout prediction problem as a sequence classification problem in the consideration that the dropout probability of a student at the current time step can be likely dependent on her/his engagement at the previous time step. Therefore, in this paper, we propose a nonlinear state space model to solve this problem. We show how students' latent states at different time steps can be learned via this model, and demonstrate its outperforming prediction accuracy relative to related methods through experiment.

## Keywords

At-risk students; Dropout prediction; Open online courses, Nonlinear state space model

## 1. INTRODUCTION

With the advent of open online courses, such as MOOC websites Edx, Coursera, Khan Academy, high quality education can easily be accessed by students at low cost. However, although many thousands of participants have enrolled on the online courses, their dropout rate is extremely higher than expected. As reported in [8], the average dropout rate of current MOOCs is approximately 75%.

Identifying at-risk students by predicting their dropout probability thus becomes timely important, given that early prediction can help instructors provide proper support to those students to retain their learning interests. To address this issue, some researchers focused on extract features from students' learning activities (such as watching videos, working on assignments, and posting in or viewing discussion forums) for building machine learning models (like support vector

machine (SVM) [9] and logistic regression (LG) [14]). However, they rarely considered that students' learning activities across different time steps (e.g., weeks) might be interrelated and take different weights in making the prediction. For instance, recent activities could be more important to reflect students' engagement degree. If a student actively engages with a course in the current week, it is more likely that s/he will continue to engage with this course in the coming week. Otherwise, if s/he becomes inactive, it may infer that her/his interest in the course is decreased. Recently, though some approaches, such as the one based on Hidden Markov Model (HMM) [2] and that based on Recurrent Neural Network (RNN) [12], have been proposed to model students' states over time, they still suffer from some issues: 1) the estimation of next state depends only on the current state; 2) the estimated states are deterministic that would lead to error propagation in the estimation procedure; 3) the parameters of their models are time-invariant.

In our work, we focus on predicting whether a student will have activities in the coming week. We particularly formulate this issue as *sequential classification* problem, and develop *Nonlinear State Space Model* (NSSM) [1] to solve it. Essentially, NSSM has several advantages. Firstly, it can be used to discover a student's latent state (i.e., *engagement pattern*) to characterize the student's intention to perform certain activities. The student's dropout probability is then computed based on the state estimated for that time. Secondly, relative to HMM and RNN, NSSM takes into account all of the current and previous states to estimate next state. It can also accommodate uncertainty given that the state in NSSM is a set of random variables with *multivariate Gaussian distribution*. Thirdly, the parameters in NSSM are time varying (i.e., being different at different time steps), which makes it more flexible to model students' dynamics.

In short, this paper has two main contributions: 1) we implement Nonlinear State Space Model (NSSM) to address the dropout prediction problem, which particularly models students' latent states varying over time; 2) we conduct experiment to compare our method with related ones including logistic regression (LG), simultaneously smoothed logistic regression (LR-SIM), and RNN with long short-term memory cell (LSTM). It shows that our method is more accurate in identifying at-risk students who tend to drop out.

In the remainder, we first describe related work in Section 2, and then present our methodology in Section 3. In Section 4,

we give experimental results. In Section 5, we conclude our work and indicate its future directions.

## 2. RELATED WORK

High dropout rate that popularly exists in current MOOCs has driven some researchers to investigate the issue of identifying at-risk students who are likely to quit. They have considered different features to build the prediction model, such as those extracted from clickstream data (e.g., watching a lecture video, posting to discuss forums, submitting an assignment) [2, 5, 6, 9, 14], quiz performance [5, 6, 14], centrality of students in discussion forums [15], and sentiments of discussion forum posts [4].

As for prediction model, some studies have applied support vector machines (SVM) [9], logistic regression (LG) [14], survival analysis techniques like Cox proportional hazard model [15], and probabilistic soft logic (PSL) [13]. However, their common limitation is that they assume a student’s dropout probabilities at different time steps are independent, which limits the approach’s applicability in practice as usually a student’s state at one time can be influenced by her/his previous state.

Alternatively, [6] extended logistic regression model to smooth the dropout probabilities across weeks with the aim to minimize the difference of succeeding predicted probabilities between weeks. [2] used Hidden Markov Model (HMM) to model student’s actions over time, which encodes their behaviour features into a set of mutually exclusive discrete states. [12] adopted Recurrent Neural Network (RNN) model with long short-term memory (LSTM) cells, which is able to encode features into continuous states. However, though RNN may be advantageous against HMM, it inherently suffers from error propagation phenomenon because the estimation of current state depends only on the estimated previous state.

In comparison, in our model, the uncertainty of estimated states is considered by representing the state as random variables drawing from a multivariate Gaussian distribution. What’s more, we adopt extended Kalman filter and smoother for state estimation so as to take into account all observed activities in sequence, which makes it different from, and potentially more effective than, HMM and RNN where only states at two consecutive time steps are related.

## 3. OUR METHODOLOGY

### 3.1 Problem Statement

As mentioned above, our goal is to estimate the probability that a student stops engaging with a course in the coming week, given her/his learning activities up to the current time step.

The temporal prediction of dropout probability requires us to assemble some features<sup>1</sup> for expressing time-varying behavior of students. Therefore, we extract 28 typical features for each week  $t$ , denoted as  $N$  dimensional vector  $\mathbf{x}_{i,t} \in \mathbb{R}^N$ ,

<sup>1</sup>Prior to model training, these features are normalized to have mean 0 and variance 1, and the normalization parameters (mean, standard deviation) are used for normalizing the testing set.

by considering the seven types of activity<sup>2</sup>. The summarization of these temporal features is listed in Table 1.

**Table 1: List of features derived from each student’s learning activities by the week  $t$**

Features	Description
$x_1$	The average number of activities per week by the week $t$ .
$x_2$	The total number of activities in week $t$ .
$x_3$	The average number of sessions per week by the week $t$ . <sup>3</sup>
$x_4$	The total number of sessions in week $t$ .
$x_5$	The average number of active days per week by the week $t$ . <sup>4</sup>
$x_6$	The total number of active days in week $t$ .
$x_7$	The average time consumption per week by the week $t$ .
$x_8$	The total time consumption in week $t$ .
$x_9 - x_{15}$	The average number of 7 different types of activity per week by the week $t$ .
$x_{16} - x_{22}$	The total number of 7 different types of activity in week $t$ .
$x_{23} - x_{25}$	The average number of videos watched, wiki viewed and problem attempted per session by the week $t$ respectively.
$x_{26} - x_{28}$	The average number of videos watched, wiki viewed and problem attempted per session in week $t$ respectively.

In consequence, we obtain a sequence  $(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \dots, \mathbf{x}_{i,n_i})$  for each student  $i$  across  $n_i$  weeks, as well as the corresponding sequence of dropout labels  $(y_{i,1}, y_{i,2}, \dots, y_{i,n_i})$ . Here  $n_i$  represents the number of weeks during which student  $i$  has engaged with the course. Formally, for current week  $t$ , if there are activities associated to student  $i$  in the coming week, her/his dropout label in the week  $t$  is assigned  $y_{i,t} = 0$ , otherwise  $y_{i,t} = 1$ . We can then treat the dropout prediction task as a *sequential classification* problem, for which the student’s latent states evolving over time are not observable directly. As illustrated in Figure 1, as the course progresses, given the student  $i$ ’s features  $\mathbf{x}_{i,t}$  for the current week  $t$ , and his/her previous state  $\mathbf{s}_{i,t-1}$ , we want to estimate the student’s current state  $\mathbf{s}_{i,t}$  and whether s/he will continue engaging with the course in the coming week  $y_{i,t}$ .

### 3.2 Nonlinear State Space Model (NSSM)

Specifically, we employ a nonlinear state space model (NSSM) with continuous value states to summarize all the information about a student’s past behavior. Formally, let the vector  $\mathbf{s}_{i,t} \in \mathbb{R}^K$  ( $K \ll N$ ) be the latent state of student  $i$  in the  $t$ -th week, which depends on the observed explanatory features  $\mathbf{x}_{i,t}$  and her/his previous state  $\mathbf{s}_{i,t-1}$ , as follows:

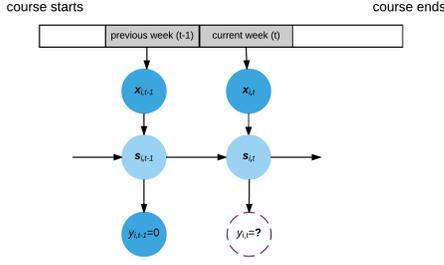
$$\mathbf{s}_{i,t} = \mathbf{F}\mathbf{s}_{i,t-1} + \mathbf{G}\mathbf{x}_{i,t} + \mathbf{w}_{i,t} \quad (1)$$

in which the matrix  $\mathbf{F} \in \mathbb{R}^{K \times K}$  transforms the previous state into the current state, the matrix  $\mathbf{G} \in \mathbb{R}^{K \times N}$  transforms the observed features to reflect the current state, and

<sup>2</sup>The seven types of activity consist of watching lecture videos, working on course’s problems, accessing course’s modules, accessing course’s wiki, posting or viewing course’s forum, navigating through courses, and closing course page.

<sup>3</sup>The minimal elapsed time between two separate sessions is set as 60 minutes.

<sup>4</sup>The day that has at least one activity is treated as an active day.



**Figure 1: The illustration of MOOCs dropout prediction problem and the graphical state space model. The dark blue signifies an observed variable and the light blue signifies a latent variable.**

$\mathbf{w}_{i,t}$  represents a diffusion variable which follows a multivariate Gaussian with mean  $\mathbf{0}$  and covariance  $\mathbf{Q}_{i,t}$  (i.e.,  $\mathbf{w}_{i,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{i,t})$ ). Note that the dimension of the state vector  $K$  is usually smaller than the dimension of feature vector  $N$ . This hyperparameter  $K$  controls the complexity of the model, and requires manual tuning to determine its optimal value.

In our work, we aim to infer the dropout probability  $\pi_{i,t}$  for student  $i$  in week  $t$ , which can be represented as logistic regression

$$\pi_{i,t} = \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t} + \beta_t^T \mathbf{x}_{i,t}) \quad (2)$$

$$= \frac{1}{1 + \exp(-\mathbf{h}_t^T \mathbf{s}_{i,t} - \beta_t^T \mathbf{x}_{i,t})} \quad (3)$$

where  $\mathbf{h}_t \in \mathbb{R}^{K \times 1}$  and  $\beta_t \in \mathbb{R}^{N \times 1}$  are two vectors of coefficients for current state variable  $\mathbf{s}_{i,t}$  and input feature  $\mathbf{x}_{i,t}$  respectively. In this model, the non-stationary of student dynamic is captured by time-evolving state variable  $\mathbf{s}_{i,t}$ , and time-varying parameters  $\mathbf{h}_t$  and  $\beta_t$ .

### 3.3 Expectation Maximization

With the nonlinear state space model described in Eqn. 1 and Eqn. 2, we design an Expectation-Maximization (EM) algorithm (see Algorithm 1) that iterates between state estimation (E-step) and parameter estimation (M-step) [11]. The E-step makes use of extended Kalman filter and smoother to estimate states, and the M-step re-estimates the parameters by maximizing the likelihood of all observed data, in which the state variables of student are replaced by their posteriori values from the extended Kalman smoother.

#### 3.3.1 Expectation Step

In the expectation step, the expected mean of student state  $\mathbf{s}_{i,t}$  and its covariance  $\mathbf{P}_{i,t}$  are obtained using the extended Kalman filter and smoother. Specifically, given student  $i$ 's entire  $t-1$  weeks' observation sequence  $D_i^{(t-1)} = \{(\mathbf{x}_{i,1}, y_{i,1}), (\mathbf{x}_{i,2}, y_{i,2}), \dots, (\mathbf{x}_{i,t-1}, y_{i,t-1})\}$ , the posterior mean and covariance of student state  $\mathbf{s}_{i,t-1}$  are supposed to be represented by  $E(\mathbf{s}_{i,t-1} | D_i^{(t-1)}) = \mathbf{s}_{i,t-1}^{(t-1)}$  and  $Cov(\mathbf{s}_{i,t-1} | D_i^{(t-1)}) = \mathbf{P}_{i,t-1}^{(t-1)}$  respectively. The predicted student state  $\mathbf{s}_{i,t}$  and its covariance  $\mathbf{P}_{i,t}^{(t-1)}$  for  $t = 1, 2, \dots, n_i - 1, n_i$  can then be defined

**Algorithm 1** EM algorithm for estimating latent student state and model parameters.

- 1: Initialize each student's starting state  $\mathbf{s}_{i,0}$  and model parameters  $\Phi = \{\mathbf{F}, \mathbf{G}, \mathbf{h}_t, \beta_t\}$
- 2: **repeat**
- 3:   **procedure E-step:**
- 4:     **Extended Kalman filter:** For  $t = 1, 2, \dots, n_i - 1, n_i$ , correct the student state  $\mathbf{s}_{i,t}$  and its covariance  $\mathbf{P}_{i,t}$  by using Eqn. 10 and Eqn. 11 respectively.
- 5:     **Extended Kalman smoother:** For  $t = n_i, n_i - 1, \dots, 2, 1$ , smooth the predicted student state  $\mathbf{s}_{i,t}^{(t)}$  and covariance  $\mathbf{P}_{i,t}^{(t)}$  by using Eqn. 13 and Eqn. 14 respectively.
- 6:   **end procedure**
- 7:   **procedure M-step:**
- 8:     Update parameters of the model  $\Phi$  via equations from Eqn. 17 to Eqn. 20.
- 9:   **end procedure**
- 10: **until** converged

as:

$$\mathbf{s}_{i,t}^{(t-1)} = \mathbf{F}\mathbf{s}_{i,t-1}^{(t-1)} + \mathbf{G}\mathbf{x}_{i,t} \quad (4)$$

$$\mathbf{P}_{i,t}^{(t-1)} = \mathbf{F}\mathbf{P}_{i,t-1}^{(t-1)}\mathbf{F}^T + \mathbf{Q}_{i,t} \quad (5)$$

By following the extended Kalman filtering, the nonlinear function  $\sigma(\cdot)$  can be approximated by its Taylor series expansion as follows:

$$\begin{aligned} \pi_{i,t} &= \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t} + \beta_t^T \mathbf{x}_{i,t}) \\ &\approx \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \beta_t^T \mathbf{x}_{i,t}) + \mathbf{A}_{i,t}^T (\mathbf{s}_{i,t} - \mathbf{s}_{i,t}^{(t-1)}) \end{aligned} \quad (6)$$

where

$$\begin{aligned} \mathbf{A}_{i,t} &\triangleq \frac{\partial \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t} + \beta_t^T \mathbf{x}_{i,t})}{\partial \mathbf{s}_{i,t}} \\ &= \sigma \left( \mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \beta_t^T \mathbf{x}_{i,t} \right) \\ &\quad \left( 1 - \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \beta_t^T \mathbf{x}_{i,t}) \right) \mathbf{h}_{i,t} \end{aligned} \quad (7)$$

The one-step ahead prediction  $\pi_{i,t}^{(t-1)}$  for the dropout probability is computed as:

$$\pi_{i,t}^{(t-1)} = \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \beta_t^T \mathbf{x}_{i,t}) \quad (8)$$

For the sake of simplicity, we set the state noise covariance as  $\mathbf{Q}_{i,t} = q_{i,t} \mathbf{I}$ , where the state noise variance  $q_{i,t}$  is computed via:

$$q_{i,t} = \max\{\mu_{i,t}^{(t)} - \mu_{i,t}^{(t-1)}, 0\} \quad (9)$$

in which  $\mu_{i,t}^{(\cdot)} = \pi_{i,t}^{(\cdot)}(1 - \pi_{i,t}^{(\cdot)})$ . After receiving a new observation  $(\mathbf{x}_{i,t}, y_{i,t})$ , the predicted state  $\mathbf{s}_{i,t}^{(t-1)}$  in Eqn. 4 and covariance  $\mathbf{P}_{i,t}^{(t-1)}$  in Eqn. 5 will be updated as:

$$\mathbf{s}_{i,t}^{(t)} = \mathbf{s}_{i,t}^{(t-1)} + \mathbf{K}_{i,t} \left( y_{i,t} - \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(t-1)} + \beta_t^T \mathbf{x}_{i,t}) \right) \quad (10)$$

$$\mathbf{P}_{i,t}^{(t)} = (\mathbf{I} - \mathbf{K}_{i,t} \mathbf{A}_{i,t}) \mathbf{P}_{i,t}^{(t-1)} \quad (11)$$

in which  $\mathbf{K}_{i,t}$  is the Kalman gain computed according to [3]:

$$\mathbf{K}_{i,t} = \mathbf{P}_{i,t}^{(t-1)} \mathbf{A}_{i,t}^T \left( \mathbf{A}_{i,t} \mathbf{P}_{i,t}^{(t-1)} \mathbf{A}_{i,t}^T + \mathbf{Q}_{i,t} \right)^{-1} \quad (12)$$

It is worth noting that the predicted state  $\mathbf{s}_{i,t}^{(t)}$  and covariance  $\mathbf{P}_{i,t}^{(t)}$  in Kalman filter are estimated based on the observation  $D_i^{(t)}$  up to week  $t$ . We take advantage of extended

Kalman smoother to smooth the estimated states by considering the entire sequence of the student's observations  $D_i^{(n_i)}$ . The smoothed states could hence be more accurate than the filtered ones. Specifically, the student state  $\mathbf{s}_{i,t-1}^{(n_i)}$  and covariance  $\mathbf{P}_{i,t-1}^{(n_i)}$  for  $t = n_i, n_i - 1, \dots, 1$  are recursively smoothed as:

$$\mathbf{s}_{i,t-1}^{(n_i)} = \mathbf{s}_{i,t-1}^{(t-1)} + \mathbf{J}_{i,t-1} \left( \mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(t-1)} - \mathbf{G}\mathbf{x}_{i,t-1} \right) \quad (13)$$

$$\mathbf{P}_{i,t-1}^{(n_i)} = \mathbf{P}_{i,t-1}^{(t-1)} + \mathbf{J}_{i,t-1} \left( \mathbf{P}_{i,t}^{(n_i)} - \mathbf{P}_{i,t}^{(t-1)} \right) \mathbf{J}_{i,t-1}^T \quad (14)$$

where  $\mathbf{J}_{i,t-1}$  is the smoothing gain defined as:

$$\mathbf{J}_{i,t-1} = \mathbf{P}_{i,t-1}^{(t-1)} \mathbf{F}^T \left( \mathbf{P}_{i,t}^{(t-1)} \right)^{-1} \quad (15)$$

Note that the initial values  $\mathbf{s}_{i,n_i}^{(n_i)}$  and  $\mathbf{P}_{i,n_i}^{(n_i)}$  for the smoother are the final estimates of the filter.

### 3.3.2 Maximization Step

At the maximization step, given the observed data  $D$  of  $N$  students, the likelihood is defined as

$$\begin{aligned} \mathcal{L}(D|\Phi) &= \sum_{i=1}^N \sum_{t=1}^{n_i} y_{i,t} \log(\sigma(\mathbf{h}_{i,t}^T \mathbf{s}_{i,t}^{(n_i)} + \beta_t^T \mathbf{x}_{i,t})) \quad (16) \\ &+ (1 - y_{i,t}) \log(1 - \sigma(\mathbf{h}_{i,t}^T \mathbf{s}_{i,t}^{(n_i)} + \beta_t^T \mathbf{x}_{i,t})) \\ &- \frac{1}{2} \sum_{i=1}^N \sum_{t=1}^{n_i} (\mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t})^T \mathbf{Q}_{i,t}^{-1} (\mathbf{s}_{i,t}^{(n_i)} \\ &- \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t}) - \frac{1}{2} \sum_{i=1}^N \sum_{t=1}^{n_i} \log|\mathbf{Q}_{i,t}| \end{aligned}$$

By using the posterior hidden state variables  $\mathbf{s}_{i,t}^{(n_i)}$  from Kalman smoother, the optimal parameters  $\Phi = \{\mathbf{G}, \mathbf{F}, \mathbf{h}_t, \beta_t\}$  can be obtained by maximizing the likelihood defined in Eqn. 16. We then apply the gradient based method L-BFGS [10] to update model parameters by using the following derivation formulas respectively:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{F}} = - \sum_{i=1}^N \sum_{t=1}^{n_i} \left( \mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t} \right) \mathbf{Q}_{i,t}^{-1} \mathbf{s}_{i,t-1}^{(n_i)} \quad (17)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{G}} = - \sum_{i=1}^N \sum_{t=1}^{n_i} \left( \mathbf{s}_{i,t}^{(n_i)} - \mathbf{F}\mathbf{s}_{i,t-1}^{(n_i)} - \mathbf{G}\mathbf{x}_{i,t} \right) \mathbf{Q}_{i,t}^{-1} \mathbf{x}_{i,t} \quad (18)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{h}_t} = \sum_{i=1}^N \sum_{t=1}^{n_i} \left( y_{i,t} - \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(n_i)} + \beta_t^T \mathbf{x}_{i,t}) \right) \mathbf{s}_{i,t}^{(n_i)} \quad (19)$$

$$\frac{\partial \mathcal{L}}{\partial \beta_t} = \sum_{i=1}^N \sum_{t=1}^{n_i} \left( y_{i,t} - \sigma(\mathbf{h}_t^T \mathbf{s}_{i,t}^{(n_i)} + \beta_t^T \mathbf{x}_{i,t}) \right) \mathbf{x}_{i,t} \quad (20)$$

**Initialization of the EM Algorithm:** The initial value of parameters  $\Phi$  should be chosen with care, otherwise the EM algorithm may not converge. In our experiment, the matrix  $\mathbf{G}$  is initially set as the transform matrix resulted from principle component analysis (PCA) algorithm [7], and the matrix  $\mathbf{F}$  is assigned to be an identity matrix.

## 4. EXPERIMENT

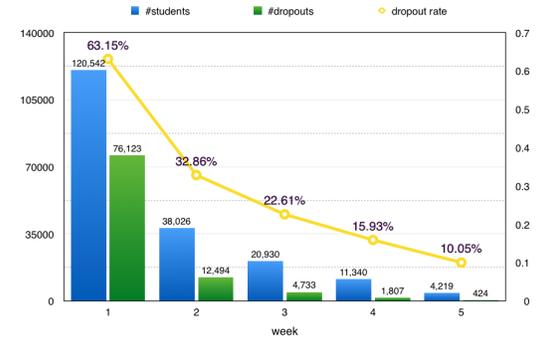
In order to evaluate the performance of our proposed model, we conducted an experiment on a real-life dataset.

### 4.1 Dataset

We use a data set collected from xuetangX<sup>5</sup>, one of the largest MOOC platforms in China. This dataset was released for KDD CUP 2015<sup>6</sup>. The dataset, as shown in Table 2, includes 79,186 students each of whom enrolled on at least one course among the whole set of 39 courses. Each enrollment is associated with a log of the student's activities including watching lecture videos, working on course's problems, accessing course's modules, and so on. Totally, there are 8,157,277 activity logs and the longest lifetime of enrollment is 5 weeks.

**Table 2: Statistics of xuetangX dataset for the experiment**

Item	Statistical description
# courses	39
# students	79,186
# enrollments	120,542
# activity logs	8,157,277
# longest lifetime of enrollment	5 weeks



**Figure 2: The number of students, number of dropouts, and the dropout rate in different weeks.**

As shown in Figure 2, we observe that 76,123 students dropped out in the first week. Another observation is that the longer the student has engaged with the course, the less likely s/he quit the course. For example, the dropout rate of students who have engaged with the courses for 5 weeks is 10.05% vs. 63.15% for 1 week.

### 4.2 Evaluation Metrics

Due to the class imbalance phenomenon, we use Area Under the Receiver Operating Characteristics Curve (AUC) as the evaluation metric, as it is invariant to imbalance. Concretely, AUC measures how likely a classifier can correctly discriminate between positive and negative samples. An AUC of 1 indicates perfect discrimination whereas 0.5 corresponds to a classifier that guesses randomly.

<sup>5</sup><http://www.xuetangx.com>

<sup>6</sup><http://www.kddcup2015.com>

### 4.3 Compared Methods

We compared our model with related methods:

- Logistic Regression (LG) [14]: In this method, a logistic regression classifier is trained to make dropout prediction for each week. Specifically, for a student  $i$  in week  $t$ , his/her dropout probability is computed as the logistic function of the weighted sum of input features  $\mathbf{x}_{i,t}$ :

$$p(y_{i,t}|\mathbf{x}_{i,t}, \mathbf{w}_t) = \frac{1}{1 + \exp(-y_{i,t}\mathbf{w}_t^T \mathbf{x}_{i,t})} \quad (21)$$

where  $\mathbf{w}_t = [w_{t1}, w_{t2}, \dots, w_{tN}]^T$  is the weight vector to be learned. The objective function for week  $t$  is

$$\mathcal{L}(\mathbf{w}_t) = \sum_{i \in N_t} \log(1 + \exp(-y_{i,t}\mathbf{w}_t^T \mathbf{x}_{i,t})) + \frac{\lambda_1}{2} \|\mathbf{w}_t\|^2 \quad (22)$$

where  $N_t$  is the set of students who engage with the course in week  $t$  and  $\lambda_1 > 0$  is the regularization parameter for  $\mathbf{w}_t$ .

- Simultaneously Smoothed Logistic Regression (LR-SIM) [6]: It extends the logistic regression by smoothing the predicted dropout probabilities across consecutive weeks. In this model, a regularization term is added into the objective function to minimize the difference of the predicted probabilities between two adjacent weeks, such as  $\mathbf{w}_t^T \mathbf{x}_{i,t}$  and  $\mathbf{w}_{t-1}^T \mathbf{x}_{i,t-1}$ . A new feature space  $\mathbf{x}'_{i,t}$  is introduced, which has  $T \times N$  dimensions ( $T$  is the total number of weeks), with the  $t$ -th component having  $N$  features corresponding to the features in the original feature space  $\mathbf{x}_{i,t}$  for week  $t$ , and other  $T - 1$  components corresponding to zeroes. Then, a single weight vector  $\mathbf{w}$  is introduced, which also has  $T \times N$  dimensions corresponding to  $\mathbf{x}'_{i,t}$ . The final objective function is defined as:

$$\mathcal{L}(\mathbf{w}) = \sum_{i \in N_t} \sum_{t=1}^{n_i} \log(1 + \exp(-y_{i,t}\mathbf{w}^T \mathbf{x}'_{i,t})) + \frac{\lambda_1}{2} \|\mathbf{w}\|^2 + \lambda_2 \sum_{t=2}^T \sum_{i \in N_{t,t-1}} \|\mathbf{w}^T \mathbf{x}'_{i,t} - \mathbf{w}^T \mathbf{x}'_{i,t-1}\|^2 \quad (23)$$

where  $N_{t,t-1}$  is the set of students who engage with the course in both weeks  $t$  and  $t - 1$ , and  $\lambda_2 > 0$  is the regularization parameter for the difference of the resulted dropout probabilities between two adjacent weeks.

- RNN with Long Short-Term Memory Cell (LSTM) [12]: It uses a recurrent neural network (RNN) model with long short-term memory (LSTM) architecture to train a sequence classifier model that produces temporal prediction. Similar to our proposed model, given the student's week-by-week features and dropout labels  $\{(\mathbf{x}_{i,t}, y_{i,t}), 1 \leq t \leq n_i\}$ , the LSTM model is applied to estimate the student state, which can then be used to predict the student's future actions.

Note that we did not compare with Hidden Markov Model (HMM) based method [2] because it can be treated as a special case of RNN by representing student state as discrete variable. For all the compared models, we used the same set of features as input (see Table 1).

### 4.4 Results and Discussion

The main hyperparameter to determine the NSSM model's performance is the dimensionality of student state  $K$  (see Eqn. 1). We compared the performance of NSSM in terms of AUC with varying dimension of latent state  $K$ , and observed that the optimal value of  $K$  in most cases is 12. Therefore, in our experiment, we set  $K$  as 12 to train the NSSM model.

#### 4.4.1 Single Course

In this setting, we trained a separate model for each course. To get sufficient data for training, we only consider the popular courses that include more than 5,000 students. After filtering, 6 popular courses are used in this experiment. As students may enroll in a course at different time steps, we select 70% students who enrolled in the course in early period as the training data, and remaining 30% students as the testing data.

	LR	LR-SIM	LSTM	NSSM
Week 1	0.812	0.886	0.891	<b>0.900</b>
Week 2	0.819	0.876	0.887	<b>0.891</b>
Week 3	0.807	0.854	0.861	<b>0.870</b>
Week 4	0.768	0.778	0.786	<b>0.796</b>
Week 5	0.673	0.679	0.689	<b>0.702</b>

**Table 3: Performance comparison of LR, LR-SIM, LSTM and NSSM in terms of average AUC on 6 popular courses.**

Table 3 presents the average AUC scores across weeks by testing different models. The results indicate that the models that consider dependence between consecutive weeks, such as LR-SIM, LSTM and NSSM, achieve higher AUC score than the baseline LR model without this consideration. For example, for the first week, the AUC score of NSSM is 0.9, which is 10.8% improvement relative to that of LR model. Furthermore, we can see that the methods that model the student's states over time (i.e., LSTM and NSSM) achieve higher AUC than LR and LR-SIM in most cases. More notably, our proposed model NSSM performs consistently better than LSTM, suggesting that the student states estimated by NSSM is more predictive than those by LSTM. We can also observe that the accuracy during early weeks is higher than that of later weeks by most of models. This implies that the dropout prediction task may become harder with increasing lifetime of engagement, as there might be various hidden reasons that cause a student to quit the course.

#### 4.4.2 Across Courses

In this setting, we are interested in evaluating whether the proposed model trained on some courses can serve other courses as well, for which we randomly select 70% courses for training and remaining 30% for testing. In this experiment, we use all of the student data from the training courses to train the model.

Table 4 shows the performance comparison. Same conclusions can be made as in the previous Section 4.4.1. Specifically, from this table, we can observe that our proposed model NSSM still outperforms the other models (e.g., LR, LR-SIM and LSTM) across different weeks. For example,

	LR	LR-SIM	LSTM	NSSM
Week 1	0.835	0.933	0.936	<b>0.936</b>
Week 2	0.911	0.915	0.915	<b>0.919</b>
Week 3	0.868	0.872	0.867	<b>0.871</b>
Week 4	0.782	0.784	0.785	<b>0.789</b>
Week 5	0.655	0.662	0.673	<b>0.686</b>

**Table 4: Performance comparison of LR, LR-SIM, LSTM and NSSM in terms of AUC on new courses across weeks.**

for the first week, the AUC score of NSSM is 0.686, which is 12% improvement relative to that of LR model. Furthermore, we can see that the improvement from NSSM with regard to LSTM is slight, and the relative improvement during later weeks is larger than that of early weeks (e.g., +5.1% during week 4 vs +4.4% during week 2). This observation implies that the NSSM has the potential to make better dropout predictions for students who have longer lifetime of engagement than LSTM. In addition, as these results are predictions made for students from new courses, we can conclude that our proposed model is capable of making better dropout prediction in new courses, in comparison with other models.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we have focused on identifying at-risk students in online courses by making dropout prediction. We particularly take advantage of nonlinear state space model (NSSM) because it can discover a student’s latent state to characterize the student’s intention to perform certain activities. We conducted experiment on a real-world dataset, which demonstrates that our proposed model achieves higher prediction accuracy than related methods. We also showed that the NSSM model trained on data from some courses can make dropout prediction for students in new courses.

However, because the extended Kalman filter and smoother we used in this paper may not be an optimal parameter estimator, the difference between NSSM and LSTM is slight. Therefore, in the future, we will exploit other advanced algorithms (e.g., Unscented Kalman filter) to estimate the parameters in our nonlinear state space model. For the second future direction, as the experiment presented in this paper is limited to xuetangX dataset, we plan to evaluate our proposed model on datasets collected from other MOOC platforms, such as Edx and Coursera.

## 6. ACKNOWLEDGMENTS

This research work was supported by Hong Kong GRF ECS/HKBU211912 and partially supported by ITF ITS/271/14FX.

## 7. REFERENCES

- [1] H. J. Andrew. *Stochastic Processes and Filtering Theory*. Academic Press, Inc., New York and London, 1970.
- [2] G. Balakrishnan. Predicting student retention in massive open online courses using hidden markov models. Master’s thesis, EECS Department, University of California, Berkeley, May 2013.
- [3] M. Y. Byron, K. V. Shenoy, and M. Sahani. Derivation of extended kalman filtering and smoothing equations. Technical report, 2004.
- [4] D. S. Chaplot, E. Rhim, and J. Kim. Predicting student attrition in moocs using sentiment analysis and neural networks. In *Proc. of the 2015 AIED Workshop on Intelligent Support for Learning in Groups*, pages 7–12, 2015.
- [5] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and Best Practices in and around MOOCs*, 7:7–16, 2014.
- [6] J. He, J. Bailey, B. I. P. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, pages 1749–1755, 2015.
- [7] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.
- [8] K. Jordan. Mooc completion rates: The data. Available at: <http://www.katyjordan.com/MOOCproject.html>. [Accessed: 04/02/2016], 2016.
- [9] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting mooc dropout over weeks using machine learning methods. In *Proc. of the 2014 EMNLP Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65, 2014.
- [10] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.
- [11] W. Mader, Y. Linke, M. Mader, L. Sommerlade, J. Timmer, and B. Schelter. A numerically efficient implementation of the expectation maximization algorithm for state space models. *Applied Mathematics and Computation*, 241:222–232, 2014.
- [12] F. Mi and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Proc. of the 2015 ICDM Workshop on Data Mining for Educational Assessment and Feedback*, pages 256–263, November 2015.
- [13] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé, III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Proc. of the 28th AAAI Conference on Artificial Intelligence*, pages 1272–1278, 2014.
- [14] C. Taylor, K. Veeramachaneni, and U.-M. O’Reilly. Likely to stop? Predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*, 2014.
- [15] 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 *Proc. of the 2013 NIPS Data-Driven Education Workshop*, volume 11, 2013.

# Transactivity as a Predictor of Future Collaborative Knowledge Integration in Team-Based Learning in Online Courses

Miaomiao Wen, Keith Maki, Xu Wang, Steven P Dow, James Herbsleb and Carolyn Rose  
Carnegie Mellon University  
Pittsburgh, USA  
{mwen,kmaki,xuwang,spdown,jdh,cprose}@cs.cmu.edu

## ABSTRACT

To create a satisfying social learning experience, an emerging challenge in educational data mining is to automatically assign students into effective learning teams. In this paper, we utilize discourse data mining as the foundation for an online team-formation procedure. The procedure features a deliberation process prior to team assignment, where participants hold discussions both to prepare for the collaboration task and provide indicators that are then used during automated team assignment. We automatically assign teams in a way that maximizes average observed pairwise transactivity exchange within teams, whereas in a control condition, teams are assigned randomly. We validate our team-formation procedure in a crowdsourced online environment that enables effective isolation of variables, namely Amazon's Mechanical Turk. We compare group knowledge integration outcomes between the two team assignment conditions. Our results demonstrate that transactivity-based team assignment is associated with significantly greater knowledge integration ( $p < .05$ , effect size 3 standard deviations).

## 1. INTRODUCTION

Although there are typically thousands of students in a Massive Open Online Course (MOOC), social isolation is still the norm in the current generation of MOOCs. However, there is evidence that many students would prefer to have more social engagement in that context. Recent research shows that a quarter of learners want to meet new people in their courses; and another 20% of learners in typical MOOCs want to take their courses with friends or colleagues [17]. To satisfy learners' social needs, there is growing interest in enabling group learning in MOOC learning contexts. Recent emerging platforms like NovoEd<sup>1</sup> and cMOOCs are designed with team-based learning or social interaction at center stage. Additionally, many recent xMOOCs are adopting

<sup>1</sup><https://novoed.com>

team-based learning features (e.g., in EdX<sup>2</sup>). There is accumulating evidence that social interaction is associated with enhanced commitment to the course [11], which has the potential to address one of MOOC critics' biggest concerns, namely high attrition rates [18]. However, how to automatically assign students to effective MOOC learning groups is still an open question [12, 25, 20]. Methods for mining educational data have been used to optimize instruction or feedback for individuals [21]. In this paper we explore how a form of educational data mining (namely, mining of discussion behavior) can be used to optimize the experience of collaborative learners through the support of effective team formation.

Algorithms for group assignment typically bring together students based on learning style, personality or demographic information. For team assignments based on such algorithms, student information must be collected and then provided to the algorithm [9]. Because of the paucity of available student personal information in MOOCs, designing a team-formation process that relies on mining of discussion data to fill in missing information would be a valuable contribution. Moreover, research identifying valuable evidence for effective team formation is needed since recent work shows that forming teams based on typical demographic features, e.g. gender and time zone, does not significantly improve teams' engagement and success in MOOCs [25]. In an online interaction, demographic information about learners is only relevant to the extent that it influences how those students come across and interact with others. Thus, observation of behavior and interaction between students may be a better source of insight for assigning students to groups in which they will function well as a team. This provides an excellent opportunity for data mining technology to make a contribution in support of valued learning processes. The alternative to automated assignment is self-selected teams. When a student population is large, which is usually the case in MOOCs, it is difficult for students to navigate through a list of students or teams to find a team that fits. Previous work has shown that many self-selected teams fail in team-based MOOCs [23]. As an alternative to both of these approaches, we design a practical group-formation procedure through which participants are organized into small groups

<sup>2</sup>[https://courses.edx.org/courses/course-v1:McGillX+GROOCx+T3\\_2015](https://courses.edx.org/courses/course-v1:McGillX+GROOCx+T3_2015),  
<https://www.edx.org/course/medicinal-chemistry-molecular-basis-drug-davidsonx-d001x-1>

based on the data mined from their participation processes in the course. This procedure uses a deliberation process, where participants hold discussions in preparation for the collaboration task; teams are then automatically assigned based on features of their interaction during deliberation.

In recent years there has been increasing interest in mining discourse data for insights into learning processes [7], for understanding factors associated with attrition in MOOCs [16], and for building models to trigger dynamic support for collaborative learning [11]. In this paper, we mine students' collaborative process to collect information for automatic team assignment. In particular, we automatically identify an important property of discourse, transactivity, from students' discussion. Transactivity is known to be higher within groups where there is mutual respect [5] and a desire to build common ground [14]. Previous studies showed that high transactivity groups are associated with higher learning [22], higher knowledge transfer [13], and better problem solving [5]. Prior work has demonstrated success at automatic detection of transactivity and relevant discussion constructs [14]. Because of the social underpinnings of transactivity, it is reasonable to hypothesize that automated detection of transactivity could form the basis for an automated group assignment procedure in online learning contexts. In this paper, we combine text-mining and algorithm-based team formation; We study whether by grouping individuals with a history of engaging in more transactive communication during a pre-collaboration deliberation can help them achieve more effective collaboration in their teams. Simply stated, our research question is:

*Can evidence of transactive discussions during deliberation inform the formation of more successful teams?*

As a step towards effective team-based learning in MOOCs, in this paper, we explore the team-formation process in an experimental study conducted in an online setting that enables effective isolation of variables, namely Amazon's Mechanical Turk (MTurk). While crowd workers likely have different motivations from MOOC students, their remote individual work setting without peer contact resembles today's MOOC setting where most students learn in isolation [6]. This allows us to test the causal connection between variables in order to identify principles that later we will test in an actual MOOC. A similar approach was taken in prior work to inform design of MOOC interventions for online group learning [6]. We designed a collaborative knowledge integration task where participants work together on writing an energy proposal for a city. This knowledge integration task is modeled after ones used in earlier collaborative learning studies [4]. The participants in our study will be referred to as students throughout the paper.

## 2. METHODS

Our experimental study is designed as a validation of a team-formation paradigm. In this paradigm, we attempt to offer teams a running start in their collaboration work by starting them with individual work, which they then discuss as a community. In addition to providing the basis for assignment to teams, the community engagement prior to team formation provides students with a breadth of exposure to different perspectives relevant to the group work. Based

on the interactions displayed during this community discussion, students are automatically assigned to teams. The students then enter their teams for the bulk of their group work. We test a transactivity-maximization team-formation method. Instead of grouping students high in transactivity into teams and students low in transactivity together, the team assignment algorithm maximizes the average amount of transactive communication within all the teams through a constraint satisfaction algorithm.

## 2.1 Experimental Paradigm

### 2.1.1 Collaboration Task Description

For the team task, we designed a highly-interdependent collaboration task that requires negotiation in order to create a context in which effective group collaboration would be necessary for task success. The task is comparable to a course project where a student team writes a proposal collaboratively. We used a Jigsaw paradigm, which has been demonstrated as an effective way to achieve a positive group composition and is associated with positive group outcomes [4]. In a Jigsaw task, each student is given a portion of the knowledge or resources needed to solve the problem, but no one has enough to complete the task alone. Following the Jigsaw paradigm, each member of the team was given special knowledge of one of the four energy sources, and was instructed to represent the values associated with their energy source in contrast to the rest, e.g. coal energy was paired with an economical energy perspective. The team collaborative task was to select a single energy plan and write a proposal arguing in favor of the group decision with respect to the associated trade-offs, meaning team members needed to negotiate a prioritization among the city requirements with respect to the advantages and disadvantages they were cumulatively aware of. The set of potential energy plans was constructed to reflect different trade-offs among the requirements, with no plan satisfying all of them perfectly. This ambiguity created an opportunity for intensive exchange of perspectives. The collaboration task is shown in Figure 1.

### 2.1.2 Experimental Procedure

We designed a four-step process for the task:

*Step 1: Preparation.* In this step, each student was asked to provide a nickname, which would be used in the deliberation and collaboration phases. To prepare for the Jigsaw task, each student was randomly assigned to read an instructional article about the pros and cons of a single energy source. Each article was approximately 500 words, and covered one of four energy sources (coal, wind, nuclear, and hydro power). To strengthen their learning and prepare them for the proposal writing, we asked them to complete a quiz reinforcing the content of their assigned article. The quiz consisted of 8 single-choice questions, and feedback including correct answers and explanations was provided along with the quiz.

*Step 2: Pre-task.* In this step, we asked each student to write a proposal to recommend one of the four energy sources (coal, wind, nuclear, and hydro power) for a city given five requirements, e.g. "The city prefers a stable energy". After each student finished this step, their proposal was automatically posted in a forum as the start of a thread with the title "[Nickname]'s Proposal".

In this final step, you will work together with other Turkers to recommend a way of distributing resources across energy types for the administration of City B. City B requires 12,000,000 MWh electricity a year from four types of energy sources: coal power, wind power, nuclear power and hydro power. We have provided 4 different plans to choose from, each of which emphasizes one energy source as primary. Your team needs to negotiate which plan is the best way of meeting your assigned goals, given the city's requirements and information below.

City B's requirements and information:

1. City B has a tight yearly energy budget of \$900,000K. Coal power costs \$40/MWh. Nuclear power costs \$100/MWh. Wind power costs \$70/MWh. Hydro power costs \$100/MWh.
2. The city is concerned with chemical waste. If the main energy source releases toxic chemical waste, there is a waste disposal cost of \$2/MWh.
3. The city is a famous tourist city for its natural bird and fish habitats.
4. The city is trying to reduce greenhouse gas emissions. If the main energy source releases greenhouse gases, there will be a "Carbon tax" of \$10/MWh of electricity.
5. The city has several large hospitals that need a stable and reliable energy source.
6. The city prefers renewable energy. If renewable energies generate more than 30% of the electricity, there will be a renewable tax credit of \$1/MWh for the electricity that is generated by renewable energies.
7. The city prefers energy sources whose cost is stable.
8. The city is concerned with water pollution.

	Energy Plan				Cost	Waste disposal cost	Carbon tax	Renewable tax credit	Total
	Coal	Wind	Nuclear	Hydro					
<b>Plan 1</b>	40%	20%	20%	20%	\$840,000K	\$14,400K	\$48,000K	\$9,600K	\$892,800K
<b>Plan 2</b>	20%	40%	20%	20%	\$912,000K	\$0	\$0	\$11,000K	\$901,000K
<b>Plan 3</b>	20%	20%	40%	20%	\$984,000K	\$14,400K	\$0	\$9,600K	\$988,800K
<b>Plan 4</b>	20%	20%	20%	40%	\$984,000K	\$0	\$0	\$11,000K	\$973,600K

**Figure 1: This figure displays the collaborative task as it was presented to the students. In addition to the task statement, they had a chat interface and a shared document space to work in.**

*Step 3: Deliberation.* In this step, students joined a threaded forum discussion akin to those available in many online environments. Each proposal written by the students in the Pre-task (Step 2) was displayed for students to read and comment on. Each student was required to write at least five replies to the proposals posted by the other students. To encourage the students to discuss transactively, the task instruction for this step included the request to, when replying to a post, "elaborate, build upon, question or argue against the ideas presented in that post, drawing from the argumentation in your own proposal where appropriate."

*Step 4: Collaboration.* In the collaboration step, team members in a group were first gathered for synchronous interaction and then directed to a shared document space to write a proposal together to recommend one of four suggested energy plans based on a city's eight requirements. Students in the same team were able to see each other's edits in real time, and were able to communicate with each other using a synchronous chat utility on the right sidebar. The collaborative task was designed to contain richer information than the individual proposal writing task in Step 2.

### 2.1.3 Outcome Measures

We evaluated team success using two types of outcomes, namely objective success through quantitative task performance (i.e., the quality of the integrated proposal, which indicates collaborative knowledge integration [3]) and process measures, as well as subjective success through a group satisfaction survey. The quantitative task performance measure was an evaluation of the quality of the proposal produced by the team. The goal of evaluating the team knowledge integration process is to distinguish instances when students are

making statements based on reasoning from simply repeating what they have read. In particular, the scoring rubric defined how to identify the following elements for a proposal: (1) Which requirements were considered; (2) Which comparisons or trade-offs were made; (3) Which additional valid desiderata were considered beyond stated requirements; (4) Which incorrect statements were made about requirements. Positive points were awarded to each proposal for correct requirements considered, comparisons made, and additional valid desiderata. Negative points were awarded for incorrect statements. We measured *Team Knowledge Integration* by the total points assigned to the team proposal, i.e. team proposal score. Two PhD students who were blind to the conditions applied the rubric to five proposals (a total of 78 sentences) and the inter-rater reliability was good ( $Kappa = 0.74$ ). The two raters then coded all the proposals.

We used the *length of chat discussion* during teamwork as a measure of team process in the Collaboration step. On average the longer discussions referred to more substantive issues.

**Group Experience Satisfaction** was measured using a four item group experience survey administered to each student after the Collaboration step. The survey was based on items used in prior work [19, 6]. In particular, the survey instrument included items related to:

- Satisfaction with team experience.
- Satisfaction with proposal quality.
- Satisfaction with the group communication.
- Perceived learning through the group experience.

Each of the items was measured on a 7-point Likert scale.

#### 2.1.4 Control Variables

Intuitively, students who display more effort in the Pre-task might perform better in the collaboration task, so that level of effort is an important control variable. We used each student's individual Pre-task proposal length as a control variable for Individual Performance. Analogously, we used each group's average group member Pre-task proposal length as a control variable for the group knowledge integration analyses.

#### 2.1.5 Transactivity Annotation, Prediction, and Measurement

To enable us to use counts of transactive contributions as evidence to inform an automated group assignment procedure, we needed to automatically judge whether a reply post in the Deliberation step was transactive or not using machine learning. A transactive contribution displays the author's reasoning and connects that reasoning to material communicated earlier. Two example posts illustrating the contrast are shown below:

- Transactive  
*"Nuclear energy, as it is efficient, it is not sustainable. Also, think of the disaster probabilities".*
- Non-transactive  
*"I agree that nuclear power would be the best solution".*

Using a validated and reliable coding manual for transactivity from prior work [14], an annotator previously trained to apply that coding manual annotated 426 reply posts collected in pilot studies we conducted in preparation for the studies reported in this paper. Each of those posts was annotated as either "transactive" or "non-transactive". 70% of them were transactive.

Automatic annotation of transactivity has been reported in the Computer Supported Collaborative Learning literature. For example, researchers have applied machine learning using text, such as chat data [15] and transcripts of whole group discussions [2]. We trained a Logistic Regression model with L2 regularization using a set of features consisting of single word features (i.e., unigrams) as well as a feature indicating the post length [10]. We evaluated our classifier with a 10-fold cross validation and achieved an accuracy of 0.843 and a 0.615 Kappa. Given the adequate performance of the model, we used it to predict whether each reply post in the Deliberation step was transactive or not.

To measure *the amount of transactive communication* between two students in the Deliberation step, we counted the number of times a pair of their posts in a same discussion thread were transactive; or one of them was a thread starter and the other student's reply was transactive.

## 2.2 Transactivity Maximization Grouping

The Transactivity Maximization teams were formed so that the average amount of transactive discussion observed in the Deliberation step among the team members in the team

was maximized. A Minimal Cost Max Network Flow algorithm was used to perform this constraint satisfaction process [1]. This network flow algorithm tackles resource allocation problems with constraints. In our case, we need to satisfy the Jigsaw constraint. At the same time, the minimal cost part of the algorithm maximized the transactive communication that was observed among the group members during the Deliberation step. The algorithm finds an approximately optimal grouping within  $O(N^3)$  ( $N$  = number of students) time complexity. A brute force search algorithm, which has an  $O(N!)$  time complexity, would take too long to finish in real time.

Our algorithm can achieve an approximately optimal solution in an admissible time. Instead of maximizing the pair-wise accumulated transitivity post count, we approximate the solution by maximizing the accumulated transitivity post count between two adjacent pairs of users. The algorithm can be generalized to form teams of any size. In our experiment, the team size is 4. We build a directed weighted graph based on students' discussion network. Then we use the successive shortest path algorithm to find a sub-optimal, but nevertheless substantially better than random grouping [1]. The algorithm greedily finds a flow with minimum cost until there is no remaining flow in the network, as outlined in Algorithm 1.

---

#### Algorithm 1 Successive Shortest Paths for Minimum Cost Max Flow

---

```

 $f(v_1, v_2) \leftarrow 0 \forall (v_1, v_2) \in E$ 
 $E' \leftarrow a(v_1, v_2) \forall (v_1, v_2) \in E$ 
while  $\exists \Pi^* \in G' = (V, E')$ 
  s.t.  $\Pi^*$  a minimum cost path from S to D do
    for each  $(v_1, v_2) \in \Pi^*$ 
      if  $f(v_1, v_2) > 0$  then
         $f(v_1, v_2) \leftarrow 0$ 
        remove  $-a(v_2, v_1)$  from  $E'$ 
        add  $a(v_1, v_2)$  to  $E'$ 
      else
         $f(v_1, v_2) \leftarrow 1$ 
        remove  $a(v_1, v_2)$  from  $E'$ 
        add  $-a(v_2, v_1)$  to  $E'$ 
      end
    end
  end

```

---

#### 2.2.1 Experimental Manipulation

In our study, students participated in a deliberative discussion as a community in a threaded discussion forum prior to being assigned to teams automatically. We investigated how the nature of the experience in that context may contribute to the success of the teams. We made use of a Jigsaw paradigm in the team assignment of teams in both the experimental and control conditions. In the experimental condition, which we termed the Transactivity Maximization condition, we additionally applied a constraint that preferred to maximize the extent to which students assigned to the same team had participated in automatically detected transactive exchanges in the deliberation. In the control condition, which we termed the Random condition, apart from enforcing the Jigsaw constraint, teams were formed by random assignment. In this way we tested the hypothesis that observed transactivity is an indicator of potential for effective

tive team collaboration. We ran the study in 10 separate batches, with 5 batches in each condition. In each batch, all the students in that batch were assigned to teams using the Random strategy or all the students were assigned to teams using the Transactivity Maximization strategy. The average level of amount of transactivity during the deliberation stage was not significantly different between batches. Thus we can test if the team-formation method can predict future collaborative knowledge integration. All the steps and instructions of the task were identical for the two conditions, as described in 2.1.2.

### 2.3 Participants

Participants were recruited on MTurk with the qualifications of having a 95% acceptance rate on 1,000 tasks or more. Each student was only allowed to participate once. A total of 246 students participated in the experiment, the students who were not assigned into groups or did not complete the group satisfaction survey were excluded from our analysis. The experiment lasted on average 35.9 minutes. We included only teams of 4 students in our analysis. There were in total 27 Transactive Maximization teams and 27 Random teams, with no significant difference in attrition between conditions ( $\chi^2(1) = 1.46, p = 0.23$ ). The dropout rate of students in Random groups was 27%. The dropout rate of students in Transactivity Maximization groups was 19%.

## 3. RESULTS

As a manipulation check, we compared the average amount of transactivity observed among teammates during the deliberation between the two conditions using a t-test. The groups in the Transactive Maximization condition ( $M = 12.85, SD = 1.34$ )<sup>3</sup> were observed to have had significantly more transactive exchanges during the deliberation than those in the Random condition ( $M = 7.00, SD = 1.52$ ) ( $p < 0.01$ ), with an effect size of 3.85 standard deviations, demonstrating that the maximization was successful in manipulating the average experienced transactive exchange within teams between conditions.

*Teams that experienced greater transactivity during deliberation demonstrate better team knowledge integration.*

To assess whether the Transactivity Maximization condition resulted in more effective teams, we tested for a difference between group-formation conditions on Team Knowledge Integration. We built an ANOVA model with Grouping Criteria (Random, Transactivity Maximization) as the independent variable and Team Knowledge Integration as the dependent variable. Average team member Pre-task proposal length was again the covariate. There was a significant main effect of Grouping Criteria ( $F(1,52) = 6.13, p < 0.05$ ) on Team Knowledge Integration such that Transactivity Maximization teams ( $M = 11.74, SD = 0.67$ ) demonstrated significantly better performance than the Random groups ( $M = 9.37, SD = 0.67$ ) ( $p < 0.05$ ), with an effect size of 3.54 standard deviations, which is a large effect. Effect size is measured in terms of Cohen's  $d$ .

Across the two conditions, observed transactive communication during deliberation was significantly correlated with Team Knowledge Integration ( $r = 0.26, p < 0.05$ ). This

<sup>3</sup>SD is short for standard deviation in this paper.

also indicated teams that experienced more transactive communication during deliberation demonstrated better Team Knowledge Integration.

*Teams that experienced greater transactivity during deliberation demonstrate more intensive interaction within their teams.*

In the experiment, students were assigned to teams based on observed transactive communication during the deliberation step. Assuming that individuals that were able to engage in positive collaborative behaviors together during the deliberation would continue to do so once in their teams, we would expect to see evidence of this reflected in their observed team process. Group processes have been demonstrated to be strongly related to group outcomes in face-to-face problem solving settings [24]. Thus, we should consider evidence of a positive effect on group processes as an additional positive outcome of the experimental manipulation.

In order to test whether such an effect occurred, we built an ANOVA model with Grouping Criteria (Random, Transactivity Maximization) as the independent variable and length of chat discussion during teamwork as the dependent variable. There was a significant effect of Grouping Criteria on length of discussion ( $F(1,45) = 9.26, p < 0.005$ ). Random groups ( $M = 20.00, SD = 3.58$ ) demonstrated significantly shorter discussions than Transactive Maximization groups ( $M = 34.52, SD = 3.16$ ), with an effect size of 4.06 standard deviations.

### Survey results

For each of the four aspects of the group experience survey, we built an ANOVA model with Grouping Criteria (Random, Transactivity Maximization) as the independent variable and the survey outcome as the dependent variable. Team ID and assigned energy condition (Coal, Wind, Hydro, Nuclear) were included as control variables nested within condition. There were no significant effects on Satisfaction with team experience or with proposal quality. However, there was a significant effect of condition on Satisfaction with communication within the group ( $F(1,112) = 4.83, p < 0.05$ ), such that students in the Random teams ( $M = 5.12, SD = 1.7$ ) rated the communication significantly lower than those in the Transactivity Maximization teams ( $M = 5.69, SD = 1.51$ ), with effect size 0.38 standard deviations. Additionally, there was a marginal effect of condition on Perceived learning ( $F(1,112) = 2.72, p = 0.1$ ), such that students in the Random teams ( $M = 5.25, SD = 1.42$ ) rated the perceived benefit to their understanding they received from the group work lower than students in the Transactivity Maximization teams ( $M = 5.55, SD = 1.27$ ), with effect size 0.21 standard deviations. Thus, with respect to subjective experience, we see advantages for the Transactivity Maximization condition, but the results are weaker than those observed for the objective measures. Nevertheless, these results are consistent with prior work where objectively measured learning benefits are observed in high transactivity teams [8].

## 4. DISCUSSION

In this paper we presented an experiment to address our research question regarding the extent to which benefit could be achieved by selecting teams based on evidence of trans-

active exchange observed during the deliberation. We designed an automatic team-formation process that combines discourse data mining and algorithm-based team formation. Here we found that teams formed such that observed transactive interactions between team members in the deliberation was maximized displayed objectively better knowledge integration than teams assigned randomly. On subjective measures we see a significant positive impact of transactivity maximization on perceived communication quality and a marginal impact on perceived enhanced understanding, both of which are consistent with what we would expect from the literature on transactivity where high transactivity teams have been demonstrated to produce higher quality outcomes and greater learning [22]. These results provide positive evidence in favor of a design for a team-formation strategy in two stages: Individuals first participate in a pre-teamwork deliberation activity where they explore the space of issues in a context that provides beneficial exposure to a wide range of perspectives. Individuals are then grouped automatically through a transactivity detection and maximization procedure that uses communication patterns arising naturally from community processes to inform group formation with an aim for successful collaboration.

This research was supported in part by funding from Google and the Gates foundation.

## 5. REFERENCES

- [1] R. K. Ahuja and J. B. Orlin. A fast and simple algorithm for the maximum flow problem. *Operations Research*, 37(5):748–759, 1989.
- [2] H. Ai, R. Kumar, D. Nguyen, A. Nagasunder, and C. P. Rosé. Exploring the effectiveness of social capabilities and goal alignment in computer supported collaborative learning. In *Intelligent Tutoring Systems*, pages 134–143. Springer, 2010.
- [3] M. Alavi and A. Tiwana. Knowledge integration in virtual teams: The potential role of kms. *Journal of the American Society for Information Science and Technology*, 53(12):1029–1037, 2002.
- [4] E. Aronson. *The jigsaw classroom*. Sage, 1978.
- [5] M. Azmitia and R. Montgomery. Friendship, transactive dialogues, and the development of scientific reasoning. *Social development*, 2(3):202–221, 1993.
- [6] D. Coetzee, S. Lim, A. Fox, B. Hartmann, and M. A. Hearst. Structuring interactions for large-scale synchronous peer learning. In *CSCW*, 2015.
- [7] M. Dascalu, S. Trausan-Matu, D. S. McNamara, and P. Dessus. Readerbench: Automated evaluation of collaboration based on cohesion and dialogism. *International Journal of Computer-Supported Collaborative Learning*, 10(4):395–423, 2015.
- [8] R. De Lisi and S. L. Golbeck. Implications of piagetian theory for peer learning. 1999.
- [9] R. Decker. Management team formation for large scale simulations. *Developments in Business Simulation and Experiential Learning*, 22, 1995.
- [10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [11] O. Ferschke, I. Howley, G. Tomar, D. Yang, and C. Rose. Fostering discussion across communication media in massive open online courses. In *CSCL*, 2015.
- [12] K. Ghadiri, M. H. Qayoumi, E. Junn, P. Hsu, and S. Sujitparapitaya. The transformative potential of blended learning using mit edx 6.002 x online mooc content combined with student team-based learning in class. *environment*, 8:14, 2013.
- [13] G. Gweon. *Assessment and support of the idea co-construction process that influences collaboration*. PhD thesis, Carnegie Mellon University, 2012.
- [14] G. Gweon, M. Jain, J. McDonough, B. Raj, and C. P. Rosé. Measuring prevalence of other-oriented transactive contributions using an automated measure of speech style accommodation. *International Journal of Computer-Supported Collaborative Learning*, 8(2):245–265, 2013.
- [15] M. Joshi and C. P. Rosé. Using transactivity in conversation for summarization of educational dialogue. In *SLaTE*, pages 53–56, 2007.
- [16] 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.
- [17] R. F. Kizilcec and E. Schneider. Motivation as a lens to understand online learners: Toward data-driven design with the olei scale. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(2):6, 2015.
- [18] C. Kulkarni, J. Cambre, Y. Kotturi, M. S. Bernstein, and S. Klemmer. Talkabout: Making distance matter with small groups in massive classes. In *CSCW*, 2015.
- [19] I. Lykourantzou, A. Antoniou, and Y. Naudet. Matching or crashing? personality-based team formation in crowdsourcing environments. *arXiv preprint arXiv:1501.06313*, 2015.
- [20] S. MacNeil, C. Latulipe, B. Long, and A. Yadav. Exploring lightweight teams in a distributed learning environment. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 193–198. ACM, 2016.
- [21] C. Piech, J. Huang, A. Nguyen, M. Phulsuksombati, M. Sahami, and L. Guibas. Learning program embeddings to propagate feedback on student code. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1093–1102, 2015.
- [22] S. D. Teasley, F. Fischer, A. Weinberger, K. Stegmann, P. Dillenbourg, M. Kapur, and M. Chi. Cognitive convergence in collaborative learning. In *International conference for the learning sciences*, pages 360–367, 2008.
- [23] M. Wen, D. Yang, and C. P. Rosé. Virtual teams in massive open online courses. *Proceedings of Artificial Intelligence in Education*, 2015.
- [24] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688, 2010.
- [25] Z. Zheng, T. Vogelsang, and N. Pinkwart. The impact of small learning group composition on student engagement and success in a mooc. *Proceedings of Educational Data Mining*, 7, 2015.

# Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation

Kevin H. Wilson<sup>\*</sup>, Yan Karklin<sup>\*</sup>, Bojian Han<sup>†</sup>, Chaitanya Ekanadham<sup>\*</sup>  
<sup>\*</sup>Knewton, Inc. New York, NY    <sup>†</sup>Carnegie Mellon University, Pittsburgh, PA  
{kevin,yan,chaitu}@knewton.com    bojianh@andrew.cmu.edu

## ABSTRACT

Estimating student proficiency is an important task for computer based learning systems. We compare a family of IRT-based proficiency estimation methods to Deep Knowledge Tracing (DKT), a recently proposed recurrent neural network model with promising initial results. We evaluate how well each model predicts a student's future response given previous responses using two publicly available and one proprietary data set. We find that IRT-based methods consistently matched or outperformed DKT across all data sets at the finest level of content granularity that was tractable for them to be trained on. A hierarchical extension of IRT that captured item grouping structure performed best overall. When data sets included non-trivial autocorrelations in student response patterns, a temporal extension of IRT improved performance over standard IRT while the RNN-based method did not. We conclude that IRT-based models provide a simpler, better-performing alternative to existing RNN-based models of student interaction data while also affording more interpretability and guarantees due to their formulation as Bayesian probabilistic models.

## Acknowledgements

Many thanks to Siddharth Reddy, David Kuntz, Kyle Hausmann, and Celia Alicata for discussions of this work and help editing the manuscript.

## Keywords

Item Response Theory, Recurrent Neural Nets, Bayesian Models of Student Performance, Deep Knowledge Tracing

## 1. INTRODUCTION

A key challenge for computer-based learning systems is to estimate a student's proficiency based on her previous interactions with the system. Accurate estimation of proficiency

enables more efficient diagnosis and remediation of her weaknesses and more effective advancement of her knowledge frontier. Proficiency estimates can also provide the student or teacher with actionable information to improve student outcomes when reported as analytics [21].

Two classical families of methods for estimating proficiency are Item Response Theory (IRT) [8, 13] and Bayesian Knowledge Tracing (BKT) [2]. IRT essentially amounts to structured logistic regression (see Section 2.1), estimating latent quantities corresponding to student ability and assessment properties such as difficulty. BKT does not capture assessment properties but employs a *dynamic* representation of student ability. A growing body of recent work has focused on modeling various structural properties of students and assessments in an attempt to combine the advantages of IRT and BKT, for instance [14, 15, 11, 5, 10, 12, 3]). In a recently proposed method known as Deep Knowledge Tracing (DKT) [16], a recurrent neural network was trained to predict student responses and was shown to outperform the best published results ([15]) on the publicly available ASSISTments data set [4] by about 20 percentage points with respect to the AUC metric described in Section 4.

To investigate DKT's advantage over traditional models, we compared a standard one parameter IRT model, two extensions of that model, and DKT on three data sets (two are publicly available and one is proprietary) on a realistic online prediction task that is typically required by computer-based learning systems (see Section 4), and which was consistent with the evaluation task employed in [16].<sup>1</sup> We reproduce the results of [16] on the ASSISTments data set, but find that proper accounting for duplicate data negates the claimed performance gains. For the two larger data sets, computational tractability hampered our ability to train DKT on fine-grained content labels, while training IRT-based models scaled to handle them. Moreover, the IRT-based models' best tractable performance matches or outperforms DKT's best tractable performance on all data sets, with a hierarchical extension of IRT performing the best in all cases. We conclude that for these data sets, IRT-based models provide simple, better-performing alternatives to DKT while also affording more interpretability and guarantees due to their formulation as Bayesian probabilistic models.

<sup>\*</sup>Contributed equally to the work.

<sup>†</sup>Performed initial coding and analysis while at Knewton.

<sup>1</sup>Code for the IRT and DKT models, as well as instructions for reproducing our results, can be found at [github.com/Knewton/edm2016](https://github.com/Knewton/edm2016).

## 2. MODELS OF STUDENT RESPONSES

In this section we set notation and describe the models we compare. Throughout, we will represent the student response data  $D$  as a set of tuples  $(s, i, r, t)$  indicating the student, item, correctness, and time of each response. In this paper, time will be indexed by interaction index (rather than wall clock time).

### 2.1 Item Response Theory (IRT)

Item Response Theory (IRT) is a standard framework for modeling student responses dating back to the 1950s [8, 13]. A single number, called the *proficiency* or *ability*, represents a student’s knowledge state during the course of completing several assessments. It is assumed that this proficiency is not changing during this examination.<sup>2</sup>

The model assumes that many students have completed a test of dichotomous items and assigns each student  $s$  a proficiency  $\theta_s \in \mathbb{R}$ . A key innovation of IRT is to model variation across different items. In its simplest form, the *one-parameter model*, each item  $i$  is assigned a parameter  $\beta_i$ , representing the *difficulty* of the item. The probability that a student  $s$  answers item  $i$  correctly is given by  $f(\theta_s - \beta_i)$ , where  $f$  is some sigmoidal function.

When  $f$  is the logistic function, this corresponds to (structured) logistic regression, where the factors for a response to an item are indicators for students and items. We use a variant of this model known as 1PO (one-parameter ogive) IRT, where the link function  $f(x) = \Phi(x)$  is the cumulative distribution function of the standard normal distribution<sup>3</sup>. The maximum likelihood solution of  $\{\theta_s, \beta_i\}$  is underdetermined<sup>4</sup>; we take a Bayesian approach and regularize the solution of  $\{\theta_s, \beta_i\}$  by imposing independent standard normal prior distributions over each  $\theta_s$  and  $\beta_i$ .

#### 2.1.1 Learning

To train the parameters on student response data, we maximize the log posterior probability of  $\{\theta_s, \beta_i\}$  given the response data (the set of response correctnesses  $\{r : (s, i, r, t) \in D\}$ , each of which is 0 or 1). Assuming independent, standard normal priors on each  $\theta_s, \beta_i$ , the log posterior is:

$$\begin{aligned} \log P(\{\theta_s\}, \{\beta_i\} | D) = & \\ & \sum_{(s,i,r,t) \in D} r \log f(\theta_s - \beta_i) + (1-r) \log(1 - f(\theta_s - \beta_i)) \\ & - \frac{1}{2} \sum_s \theta_s^2 - \frac{1}{2} \sum_i \beta_i^2 + C. \end{aligned} \quad (1)$$

We maximize this objective with respect to the parameters using standard second-order ascent methods to obtain the maximum a posteriori (MAP) estimate of each parameter.

### 2.2 Hierarchical IRT (HIRT)

<sup>2</sup>For an in depth discussion of IRT and a review of related literature see [17], especially Chapter 5.

<sup>3</sup>The ogive yields nearly identical results to the commonly used logistic link function, but allows closed-form posterior computation in the temporal IRT model described in Sec. 2.3

<sup>4</sup>For example, the response predictions are invariant when adding a constant offset to the  $\{\theta_s\}$ ’s and  $\{\beta_i\}$ ’s.

In many situations, including each of our data sets, the assessment items may have structure that can inform predictions of student responses. For example, groups of items may assess the same topic, resulting in item properties that are more similar within groups than across them. Alternatively, items may be derived from common templates. Templates, often found in math courses, look like “What is  $x + y$ ?” and a particular instantiation is generated by choosing values for  $x$  and  $y$ . For example, the ASSISTments data set contains several *problems*, many of which are with the same *template*, many of which in turn assess a single *skill*.

We can augment the IRT model to incorporate knowledge about item groups, resulting in a hierarchical IRT model (HIRT). Each item  $i$  is associated with a group  $j(i)$  whose difficulty is distributed normally around a per-group mean  $\mu_{j(i)}$ :  $\beta_i \sim N(\mu_{j(i)}, \sigma^2)$ . Each  $\mu_j$  is in turn distributed according to the hyperprior  $\mu_j \sim N(0, \tau^2)$ . This reflects the belief that the difficulty of items in the same group should be similar. The degenerate cases provide some intuition: the limit  $\sigma \rightarrow 0$  is the same model as 1PO IRT where we consider the items in the group to be the same item, and the limit  $\tau \rightarrow 0$  is equivalent to a 1PO IRT model with no groupings.

#### 2.2.1 Learning

Learning is done similarly to Bayesian IRT (section 2.1), except that we ascend the *modified* log posterior probability

$$\begin{aligned} \log P(\{\theta_s\}, \{\beta_i\}, \{\mu_t\} | D) = & \\ & \sum_{(s,i,r,t) \in D} r \log f(\theta_s - \beta_i) + (1-r) \log(1 - f(\theta_s - \beta_i)) \\ & - \frac{1}{2} \sum_s \theta_s^2 - \frac{1}{2\sigma^2} \sum_i (\beta_i - \mu_{j(i)})^2 - \frac{1}{2\tau^2} \sum_j \mu_j^2 + C. \end{aligned} \quad (2)$$

We maximize this objective with respect to  $\{\theta_s, \beta_i, \mu_j\}$ .

### 2.3 Temporal IRT (TIRT)

1PO IRT and HIRT assume each student’s knowledge state remains constant over time. However, in a setting where a student may be acquiring (or forgetting) knowledge over a period of time (e.g., while interacting with a tutoring system), we can extend this model by modeling each  $\theta_s$  as a stochastic process varying over time (see for example [5]). We adopt the approach described in [3], modeling the student’s knowledge as a Wiener process:

$$P(\theta_{s,t+\tau} | \theta_{s,t}) = e^{-\frac{(\theta_{s,t+\tau} - \theta_{s,t})^2}{2\gamma^2\tau}} \quad \forall s, t, \tau. \quad (3)$$

In other words, the change in student  $s$ ’s knowledge state between time  $t$  and a future time  $t + \tau$  (expressed as  $\theta_{s,t} - \theta_{s,t+\tau}$ ) is normally distributed about 0 with variance  $\gamma^2\tau$  where  $\gamma$  is a parameter controlling the “smoothness” with which the knowledge state varies over time.

#### 2.3.1 Learning

We fit the parameters according to the procedure described in [3]. Estimating the entire trajectory  $\vec{\theta}_{s,t}$  for each student simultaneously with item parameters is very expensive and

difficult to do in real-time. To simplify the approach, we learn parameters in two stages:

1. We learn the  $\beta_i$  according to a standard 1PO IRT model (see Section 2.1.1) on the training student population and freeze these during validation.
2. For each response of each student in the held-out validation population, we predict this response according to a temporal IRT model given the student's previous responses, as described below. For further details of the validation procedure, see Section 4.

For the second step, we combine the approximation:

$$P(\{(s', i, r, t') \in D : s' = s, t' \leq t\} | \theta_{s,t}) \approx \prod_{(s', i, r, t') \in D : s' = s, t' \leq t} P((s', i, r, t') | \theta_{s,t}) \quad (4)$$

with (3), integrating out previous proficiencies of the student to get a tractable approximation of the log posterior over the student's current proficiency given previous responses:

$$\log P(\theta_{s,t} | D) \approx \sum_{\substack{(s', i, r, t') \in D \\ s' = s, t' \leq t}} [r \log f(\tilde{\alpha}_{t'}(\theta_{s,t} - \beta_i)) + (1 - r) \log(1 - f(\tilde{\alpha}_{t'}(\theta_{s,t} - \beta_i)))] \quad (5)$$

where  $\tilde{\alpha}_{t'} = (1 + \gamma^2(t - t'))^{-1/2}$ . The  $\tilde{\alpha}_t$ 's are essentially discounting the relative effect of older responses when estimating the current proficiency. See [3] for details.

## 2.4 Deep Knowledge Tracing (DKT)

Recently, a recurrent neural network was used to predict student responses [16]. Such architectures have seen enormous success in applications to a wide range of other domains (e.g., image processing [6], speech recognition [7], and natural language processing [20]).

In this model, the input vectors are representations of whether the student answered a particular question correctly or incorrectly at the previous time step, and the output vectors are representations of the probability, over all the questions in the question bank, that a student will get the question correct at the following time step. In [16], the authors propose using a one-hot vector  $\vec{x}_{s,t} \in \mathbb{R}^{2I}$  to represent the response of a student  $s$  (on item  $i$ ) at time  $t$ . Here  $I$  is the total number of items and the first  $I$  slots represent answering correctly and the remaining  $I$  slots represent answering incorrectly. Output vectors  $\vec{y}_{s,t} \in \mathbb{R}^I$  are vectors of probabilities, where the  $i$ th element of  $\vec{y}_{s,t}$  is the model's predicted probability that student  $s$  would answer item  $i$  correctly at time  $t + 1$ .

We use a model with one hidden layer, of dimension  $H$ , which is fully connected<sup>5</sup> to both the input and output layers, as well as recurrently to itself. This model is able to capture temporal effects (via the recurrent component of the network) and remains flexible enough to describe non-trivial relationships between items.

<sup>5</sup>Note that in [16], an LSTM network was used in addition to the RNN described here, and the performance of the two networks was comparable.

### 2.4.1 Learning and Parameter Choices

In order to make learning tractable, we reduced the dimensionality of the input by projecting the  $\vec{x}_{s,t} \in \mathbb{R}^{2I}$  to a lower dimensional space  $\mathbb{R}^C$  using a random projection matrix  $c : \mathbb{R}^{2I} \rightarrow \mathbb{R}^C$ , as was done in [16]. We used batch gradient ascent with dropout [18], and chose the input dimensionality  $C$  and the hidden dimensionality  $H$  by sweeping these parameters on a data set that was held out from the data used for training and cross-validation.

The predictions are given by the following equations:

$$\vec{h}_{s,t+1} = g(W_{hh}\vec{h}_{s,t} + W_{xc}c(\vec{x}_{s,t}) + \vec{b}_h) \quad (6)$$

$$\vec{y}_{s,t+1} = \phi(W_{hy}\vec{h}_{s,t+1} + \vec{b}_y) \quad (7)$$

Here,  $g$  and  $\phi$  are the logistic and arctangent functions, respectively. The parameters of the model  $W_{hh}, W_{xc}, W_{hy}, \vec{b}_h, \vec{b}_y$  are fit by optimizing the cross-entropy of the responses with the predicted probabilities (which is equivalent to the log likelihood if these probabilities were produced via a generative probabilistic model):

$$\sum_{(s,i,r,t) \in D} r \log y_{s,t,i} + (1 - r) \log(1 - y_{s,t,i}) \quad (8)$$

Stochastic gradient ascent with minibatches of students on the unrolled RNN, coded using Theano [1], was used to optimize this objective function.

## 3. DATA SETS

In order to test these models, we used three data sets, two publicly accessible and one proprietary. Each of these data sets comes from a system in which students interact with a computer-based learning system in a variety of educational settings (e.g., interspersed with classroom lectures, offline work, etc.).

### 3.1 ASSISTments

This data set comes from the ASSISTments product, an online platform which engages students with formative assessments replete with scaffolded hints. Most assessments are templated, and each problem is aligned with one, several, or none of the skills that the product is attempting to teach.

The data set [4] is divided in two parts, the "skill builder" set associated with formative assessment and the "non skill builder" set associated with summative assessment. All of our results are reported on the "skill builder" data set as we expect a stronger temporal signal from formative assessment than from summative assessment. This was also the evaluation data set for [16].

In preprocessing the data, we associated items not aligned with a skill to a designated "dummy" skill, as was done in [16]. We chose to discard rows duplicating a single interaction (represented by a unique `order_id` value), a step we do not believe was taken by [16]. These duplicate rows arise when a single interaction is aligned with multiple skills. Without removing these duplicates, models that process all skills simultaneously, including DKT and the IRT variants used in this paper, will see the same student interaction several times in a row, essentially providing these models

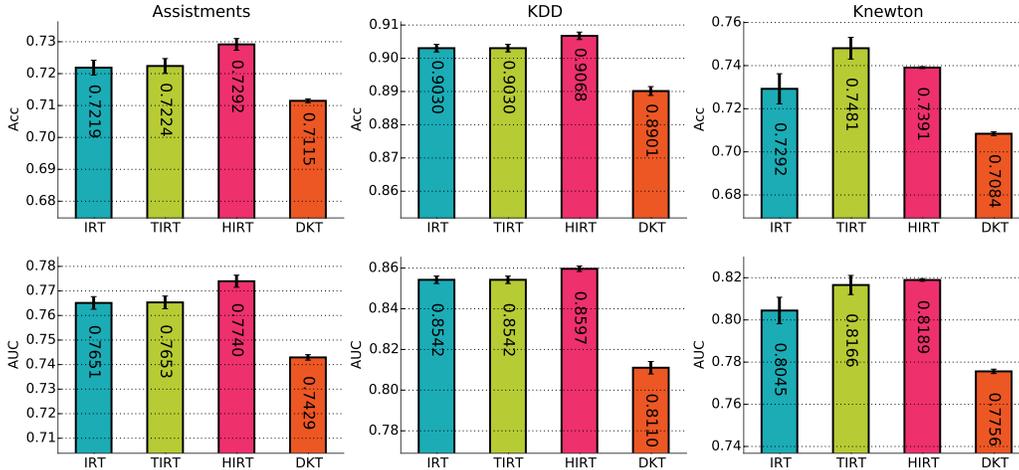


Figure 1: Summary of results across models and metrics. Error bars represent the standard error of measure of the metric across five folds. For TIRT, parameter selection yielded  $\gamma^2 = 0.01$  for ASSISTments,  $\gamma^2 = 0$  for KDD (making it identical to IRT), and  $\gamma^2 = 100.0$  for Knewton. For HIRT, parameter selection yielded  $\sigma^2 = 0.125$  and  $\tau^2 = 0.5$  for ASSISTments,  $\sigma^2 = 0.5$  and  $\tau^2 = 0.25$  for KDD, and  $\sigma^2 = 0.25$  and  $\tau^2 = 0.125$  for Knewton. For DKT,  $C = 50$ ,  $H = 100$ , and the probability of dropout is 0.25 for all models.

access to the ground truth when making their predictions. This can artificially boost prediction results by a significant amount (see Section 5), as these “duplicate” rows account for approximately 25% of the rows. Indeed, we observed that the performance gains of DKT are negated when these duplicates are removed (see Section 5). Note that typical BKT-based approaches are not susceptible to this artificial boost, since they usually split the data by skill and train separate models.

After pre-processing, the data set consisted of 346,740 interactions for 4,097 users on 26,684 items arising from 815 templates and 112 skills. The overall percent correct was 64.54%.

### 3.2 KDD Cup

In 2010, the PSLC DataShop released several data sets derived from Carnegie Learning’s Cognitive Tutor in (Pre-)Algebra from the years 2005–2009 [19]. We used the largest of the “Development” data sets, labeled “Bridge to Algebra 2006–2007.”

One distinct difference between Carnegie Learning’s product and ASSISTments is that Carnegie Learning provides much finer representations of the concepts assessed by an individual item. In particular, Carnegie Learning is built around scaffolded, formative assessment, where each *step* a student takes to answer a *problem* is counted as a separate interaction, with each step potentially assessing different skills (called Knowledge Components (KCs) in the data set). Note that this “Problem  $\rightarrow$  Step” structure provides a hierarchy which HIRT (Section 2.2) can exploit.

Like ASSISTments, any particular interaction may assess zero or more skills. We follow the same methodology as we

did in Section 3.1, arbitrarily but consistently retaining only one of the skills after preprocessing, and associating items not associated with any skills with a designated “dummy” skill.

After pre-processing, the data set retained 3,679,198 interactions for 1,146 users on 207,856 steps arising from 19,355 problems and 494 KCs. The overall percent correct was 88.82%.

### 3.3 Knewton

Data was collected from a variety of educational products integrated with Knewton’s adaptive learning platform and used in various classroom settings across the world. These products vary with respect to the educational content used (disciplines spanned math, science, and English language learning) as well as the way in which students are guided through the content. For example, students may take an initial assessment and then be remediated on areas needing improvement. In other products, students start from the beginning and work toward a predefined goal set by the teacher. In all of these settings, Knewton receives data about each interaction (the  $(s, i, r, t)$  tuple of Section 2). We utilized approximately 1M responses of 6.3K randomly sampled students on 105.6K questions spanning roughly 4 months. Students who worked on fewer than 5 questions total were excluded. After pre-processing, student history lengths ranged from 5 to 3.2K responses. The overall percent correct of these responses is 54.6%.

## 4. EVALUATION METHODOLOGY

### 4.1 Parameter Selection

For each data set, 20% of students were first set aside for parameter selection, which we performed as follows:

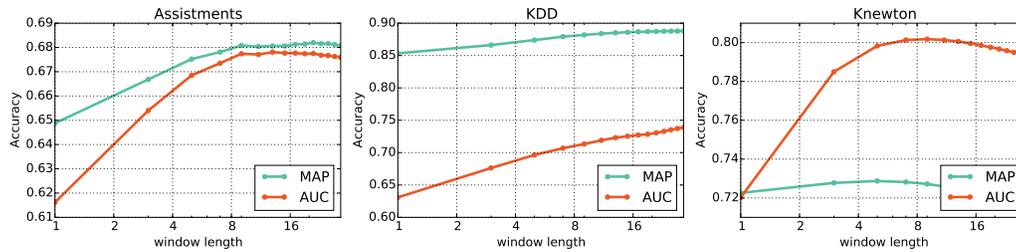


Figure 2: Accuracy metrics for the three data sets computed using a rolling window of previous responses, as a function of window length. Response accuracy is computed by predicting correct if the majority of responses in the window are correct.

	IRT	HIRT	tIRT	DKT*
ASSISTments	problem_id	template_id → problem_id	problem_id	template_id
KDD	Step Name	Problem Name → Step Name	Step Name	KC
Knewton	item_id	concept_id → item_id	item_id	concept_id

Table 1: Item labels yielding best results for each model and data set. For HIRT, the first label specifies the difficulty mean grouping identifier, and the second the item identifier.

- For IPO IRT there were no parameters to select.
- For HIRT, we swept values of the variances  $\tau^2$  and  $\sigma^2$  of the group means and item difficulties respectively, including regimes ( $\tau^2$  small) which made the model mathematically equivalent to IPO IRT.
- For TIRT, we swept the temporal smoothness parameter  $\gamma^2$ , including the regime ( $\gamma^2$  small) which made the model mathematically equivalent to IPO IRT.
- For DKT, we swept the compression dimension  $C$  (the dimension of the space to which the input was projected using a random matrix), the hidden dimension  $H$ , the dropout probability  $p$ , and the step size of our gradient ascent.

## 4.2 Online prediction accuracy

We use an evaluation method we call *online response prediction* which matches that of [16]. Students are first split into training and testing populations. Each model is first trained on the training population and the model parameters that are not student-level (item parameters for IRT-based models, weights for neural networks) are frozen. Then for each time  $t > 1$  in each testing student’s history, we train the student-level parameters in the model on the first  $t - 1$  interactions of the student history and allow it to compute the probability that the  $t$ ’th response is correct. This process mirrors the practical task that must be completed by an ITS.

We report two different metrics for comparing the predicted correctness probabilities with the observed correctness values. Accuracy (Acc) is computed as the percent of responses in which the correctness coincides with the probability being greater than 50%. AUC is the Area Under the ROC Curve of the probability of correctness for each response.

We use five-fold cross validation (by partitioning the students) on the 80% of the data set remaining after parameter

selection (Section 4.1), averaging the Acc and AUC metrics over five different splits of the student population.

## 5. RESULTS AND DISCUSSION

Table 1 enumerates the fields chosen in each data set to identify items and item groups (for HIRT only) that yielded the computationally tractable model with the best results. Note that for the IRT-based models, our validation scheme (Section 4.2) estimates a single number  $\theta_{st}$  for each student at each point  $t > 1$  of the validation. For computational reasons, it was not feasible to evaluate DKT on fine-grained labels in KDD and Knewton (for ASSISTments, fine-grained labels were tractable but yielded worse results), whereas all IRT variants were able to process data at the finest levels.

We trained and validated each of the three models on each of the three data sets as described in Sec. 4. The results on our evaluation task are summarized in Figure 1. The results clearly indicate that simple IRT-based models do as well or significantly better than DKT across all data sets.

The fact that HIRT is the best-performing model across the board (except for MAP accuracy on the Knewton dataset where TIRT slightly outperforms it) suggests that grouping structure is useful information to exploit when predicting student responses. Indeed, the HIRT model does have access to strictly more information than the other models in that it has both the item and group identifier associated with each interaction. While the DKT model does have the ability to infer item relationships from data, our results indicate that building in this knowledge is more advantageous in a variety of educational settings. One potential area to explore is in learning a hierarchical model purely from the data, which could profit from the structured Bayesian framework without requiring prior information or expert labels.

The temporal IRT model yielded higher accuracy on the Knewton dataset, but not on the other two data sets. To understand these effects, we investigated the degree to which temporal structure in the data affects predictive performance

by looking at how a naive “windowed percent correct” (predict the student will answer the  $t$ th question correctly if they answer at least half of the previous  $w$  questions correctly) model performs as a function of window length  $w$  (Figure 2). The Knewton data set has a clear optimal window length – integrating over windows too short or too long degraded performance, which is indicative of nontrivial temporal structure. However, for the ASSISTments and KDD data sets, longer window lengths perform equal or better than shorter window lengths, suggesting that static models would do just as well in these cases. Indeed, this would explain why TIRT does more or less the same as baseline 1PO IRT on ASSISTments and KDD but shows significant improvement on the Knewton data set. However, it does not explain why DKT logs regardless of the amount of temporal structure.

Finally, we note that our DKT results in Figure 1 contradict those of [16] on the ASSISTments data set, which reported an AUC of 0.86. We believe this is due to data cleaning issues, specifically the issue of removing duplicates so as not to artificially boost online prediction accuracy, as discussed in Section 3.1. Indeed, we were able to reproduce the performance reported in [16] when applying our RNN implementation on the raw data set (with duplicates left in).

Other recent work [9] points out that the specific method of computing AUC in [16] also significantly affects the reported performance relative to BKT-based models, and further demonstrates that BKT-based models can perform just as well as DKT on a variety of data sets.

## 6. CONCLUSION

Our results indicate that simple IRT-based models equal or outperform DKT on a variety of data sets, suggesting that incorporating domain knowledge into structured Bayesian models comprises a promising area of future research for modeling student interaction data.

In our experience, structured models were easier to train and required less parameter tuning than DKT. Moreover, the computational demands of DKT hampered our ability to fully explore the parameter space, and we found that computation time and memory load were prohibitive when training on tens of thousands of items. These issues could not be mitigated by reducing dimensionality without significantly impairing performance. Further work on discriminative models is necessary to bridge this gap, but currently, IRT-based models seem superior both in terms of performance and ease of use, making them suitable candidates for real-world applications (e.g. intelligent tutoring systems, recommendation systems, or student analytics).

A promising avenue of research could explore combining the advantages of structured Bayesian models with those of large-scale discriminative models, which have provided superior performance in several other domains, particularly in the large-data regime. A crucial challenge for structured models is how to accommodate the diversity of educational settings from which the data are collected (different content, different classroom environments, etc.) while retaining the structure that drives predictive power and interpretability.

## 7. REFERENCES

- [1] BERGSTRA, J., ET AL. Theano: a CPU and GPU math expression compiler. In *SciPy 2010*.
- [2] CORBETT, A., AND ANDERSON, J. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 4 (1995), 253–278.
- [3] EKANADHAM, C., AND KARKLIN, Y. T-SKIRT: Online estimation of student proficiency in an adaptive learning system. *Machine Learning for Education Workshop at ICML* (2015).
- [4] FENG, M., HEFFERNAN, N., AND KOEDINGER, K. Addressing the assessment challenge with an online system that tutors as it assesses. In *User Modeling, Adaption, and Personalization*, G.-J. Houben, G. McCalla, F. Pianesi, and M. Zancanaro, Eds. 2010, pp. 243–266.
- [5] GONZALEZ-BRENES, J., HUANG, Y., AND BRUSILOVSKY, P. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *EDM 2014*.
- [6] GREGOR, K., ET AL. DRAW: A recurrent neural network for image generation. In *ICML 2015*.
- [7] HINTON, G., ET AL. Deep neural networks for acoustic modeling in speech recognition.
- [8] HULIN, C. L., AND DRASGOW, F. Item Response Theory. In *Handbook of Industrial and Organizational Psychology*, S. Zeck, Ed., vol. 1. American Psychological Association, 1990, pp. 577–636.
- [9] KHAJAH, M., LINDSEY, R. V., AND MOZER, M. C. How deep is knowledge tracing? In *EDM 2016*.
- [10] KHAJAH, M. M., HUANG, Y., GONZÁLEZ-BRENES, J. P., MOZER, M. C., AND BRUSILOVSKY, P. Integrating knowledge tracing and item response theory. *Personalization Approaches in Learning Environments* (2014), 7.
- [11] LAN, A. S., STUDER, C., AND BARANIUK, R. G. Time-varying learning and content analytics via sparse factor analysis. In *KDD 2014*.
- [12] LEE, J. I., AND BRUNSKILL, E. The Impact on Individualizing Student Models on Necessary Practice Opportunities.
- [13] LORD, F. M. *A Theory of Test Scores*. No. 7 in Psychometric Monograph. Psychometric Corporation, 1952.
- [14] PARDOS, Z. A., AND HEFFERNAN, N. T. Modeling individualization in a Bayesian Networks implementation of knowledge tracing. In *User Modeling, Adaption, and Personalization*, P. D. Bra, A. Kobsa, and D. Chin, Eds. 2010, pp. 255–266.
- [15] PARDOS, Z. A., AND HEFFERNAN, N. T. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In *User Modeling, Adaption, and Personalization*, J. A. Konstan, R. Conejo, J. L. Marzo, and N. Oliver, Eds. 2011, pp. 243–254.
- [16] PIECH, C., BASSEN, J., HUANG, J., GANGULI, S., SAHAMI, M., GUIBAS, L., AND SOHL-DICKSTEIN, J. Deep Knowledge Tracing. In *NIPS 2015*.
- [17] RUPP, A. A., TEMPLIN, J., AND HENSON, R. A. *Diagnostic Measurement: Theory, Methods, and Applications*. Guilford Press, 2010.
- [18] SRIVASTAVA, N., ET AL. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.
- [19] STAMPER, J., NICULESCU-MIZIL, A., RITTER, S., G.J GORDON, G., AND KOEDINGER, K. Challenge data sets from KDD Cup 2010. [ps1cdatashop.web.cmu.edu/KDDCup/downloads.jsp](http://ps1cdatashop.web.cmu.edu/KDDCup/downloads.jsp).
- [20] VINYALS, O., ET AL. Grammar as a foreign language. In *NIPS 2015*.
- [21] WILSON, K. H., AND NICHOLS, Z. The Knewton Platform: A General-Purpose Adaptive Learning Infrastructure.

# Going Deeper with Deep Knowledge Tracing

Xiaolu Xiong, Siyuan Zhao, Eric G.

Van Inwegen, Joseph E. Beck

Worcester Polytechnic Institute

100 Institute Rd

Worcester, MA 01609

508-831-5000

{xxiong, szhao, egvaninwegen,

josephbeck}@wpi.edu

## ABSTRACT

Over the last couple of decades, there have been a large variety of approaches towards modeling student knowledge within intelligent tutoring systems. With the booming development of deep learning and large-scale artificial neural networks, there have been empirical successes in a number of machine learning and data mining applications, including student knowledge modeling. Deep Knowledge Tracing (DKT), a pioneer algorithm that utilizes recurrent neural networks to model student learning, reports substantial improvements in prediction performance. To help the EDM community better understand the promising techniques of deep learning, we examine DKT alongside two well-studied models for knowledge modeling, PFA and BKT. In addition to sharing a primer on the internal computational structures of DKT, we also report on potential issues that arise from data formatting. We take steps to reproduce the experiments of Deep Knowledge Tracing by implementing a DKT algorithm using Google's TensorFlow framework; we also reproduce similar results on new datasets. We determine that the DKT findings don't hold an overall edge when compared to the PFA model, when applied to properly prepared datasets that are limited to main (i.e. non-scaffolding) questions. More importantly, during the investigation of DKT, we not only discovered a data quality issue in a public available data set, but we also detected a vulnerability of DKT at how it handles multiple skill sequences.

## Keywords

Knowledge tracing, deep learning, recurrent neural networks, student modeling, performance factors analysis, data quality

## 1. INTRODUCTION

Deep Learning (DL) is an emerging approach within the machine learning research community. A series of deep learning algorithms have been proposed in recent years to move machine learning systems toward the discovery of multiple levels of representation and they already had important empirical successes in a number of traditional AI applications such as computer vision and natural language processing [8]. Much more recently, Google's deep learning networks [7] beat a top human player at the game of Go. Most research in deep learning (e.g. Google's deep learning algorithm) has been focused on the studies of artificial neural networks.

Deep knowledge tracing (DKT), the recent adoption of recurrent neural nets (RNNs) in the area of educational data mining, achieved dramatic improvement over well-known Bayesian Knowledge Tracing models (BKT) and the results of it have been

demonstrated to be able to discover the latent structure in skill concepts and can be used for curriculum optimization [1].

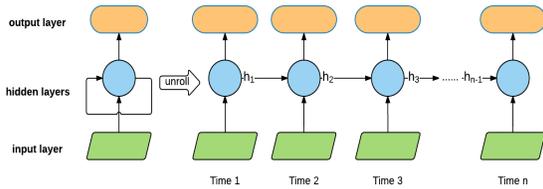
Driven by both noble goals (testing the reproducibility of scientific findings) and some selfish ones (how did they do so much better at predicting student performance?!), we set out to take the theories, algorithms, and code from the DKT paper and apply them ourselves to the same data and more data sets. As to the goal of reproducing the findings, we were motivated by studies discussing the importance of reproducibility [5]. In addition to applying DKT to the same data, we also tested the algorithm on a different ASSISTments dataset (which covers data in 2014-2015 school year), as well as the one of data sets from KDD Cup 2010. In our experiments with the original DKT algorithm, we uncovered three aspects of the ASSISTments 2009-2010 data set that, when accounted for, drastically reduce the effectiveness of the DKT algorithm. These can broadly be summarized as 1). an error in reporting the data (wherein rows of data were randomly duplicated). 2). an inconsistency of skill tagging, and 3). the use of information ignored by PFA and BKT. We will discuss these three inconsistencies and their impacts on the prediction accuracies in section 3.

## 2. DEEP KNOWLEDGE TRACING AND OTHER STUDENT MODELING TECHNIQUES

When describing neural networks, the use of 'deep' conventionally refers to the use of multiple processing layers; the 'Deep' in DKT refers to the recurrent structure of the network and the 'depth' of information over time. This family of neural nets represents latent knowledge state, along with its temporal dynamics, using large vectors of artificial neurons, and allows the latent variable representation of student knowledge to be learned from data rather than hard-coded.

Typical RNNs suffer from the now famous problems of vanishing and exploding gradients, which are inherent to deep networks. Figure 1 shows an unrolled RNN; there are loops at hidden layers, allowing information to be retained; this is the 'depth' of an RNN. When building a deep neural net, the standard activation functions, and cumulative backpropagation error signals either shrink rapidly or grow out of bounds. i.e., they either decay or grow exponentially ('vanish' or 'explode'). Long short-term memory (LSTM) model [14] is introduced to deal with the vanishing gradient problem; it also achieves remarkable results on many previously un-learnable tasks. LSTM, a variation of recurrent neural networks, contains LSTM units in addition to regular RNN units. LSTM units have two unique gates: forget and input gates

to determine when to forget previous information, and which current information is important to remember.



**Figure 1. An unrolled Recurrent Neural Network (RNN)**

The idea behind LSTM is simple. Some of the units are called constant error carousels (CEC). Each CEC uses an activation function  $f$ , the identity function, and has a connection to itself with fixed weight of 1.0. Due to  $f$ 's constant derivative of 1.0, errors backpropagated through a CEC cannot vanish or explode but stay the same magnitude. CECs are connected to several nonlinear adaptive units needed for learning nonlinear behavior. Weight changes of these units often profit from error signals, which propagate far back in time through CECs. CECs are the main reason why LSTM nets can learn to discover the importance of (memorize) events that happened thousands of discrete time steps ago while previous RNNs routinely fail in cases of minimal time lags of 10 steps. LSTM learns to solve many previously unlearnable DL tasks and clearly outperformed previous RNNs on tasks both in terms of reliability and speed [1].

In the DKT algorithm, at any time step, the input to RNNs is the student performance on a single problem of the skill that the student is currently working on. Since RNNs only accept fixed length of vectors as the input, we used one-hot encoding to convert student performance into fixed length of vectors whose all elements are 0s except for a single 1. The single 1 in the vector indicates two things: which skill was answered and if the skill was answered correctly. This data presentation draws a clear distinction between DKT and other student modeling methods, such as Bayesian Knowledge Tracing and Performance Factor Analysis.

The Bayesian Knowledge Tracing (BKT) model [10] is a 2-state dynamic Bayesian network where student performance is the observed variable and student knowledge is the latent data. The model takes student performances and uses them to estimate the student level of knowledge on a given skill. The standard BKT model is defined by four parameters: initial knowledge and learning rate (learning parameters) and slip and guess (mediating parameters). The two learning parameters can be considered as the likelihood the student knows the skill before he even starts on an assignment (initial knowledge,  $K_0$ ) and the probability a student will acquire a skill as a result of an opportunity to practice it (learning rate). The guess parameter represents the fact that a student may sometimes generate a correct response in spite of not knowing the correct skill. The slip parameter acknowledges that even students who understand a skill can make an occasional mistake. Guess and slip can be considered analogous to false positive and false negative. BKT typically uses the Expectation Maximization algorithm to estimate these four parameters from training data. Based on the estimated knowledge, student performance at a particular practice opportunity can be calculated except the very first one, which only applies the value of  $K_0$ .

Skills vary in difficulties and amount of practices needed to master, so values for four BKT parameters are skill dependent. This leads to one major weakness of BKT [11]: it lacks the ability to handle multi-skill questions since it works by looking at the historical observation of a skill and cannot accommodate all skills simultaneously. One simple workaround is treating the multiple skill combination as a new joint skill and estimate a set of parameters for this new skill. Another common solution to this issue is to associate the performance on multiple skill questions with all required skills, by listing the performance sequence repeatedly [12]. This makes the model see this piece of evidence multiple times for each one of required skills. As a result, a multiple skill question is multiple single skill questions.

Another popular student modeling approach is the Performance Factors Analysis Model (PFA) [9]. PFA is a variant of learning decomposition, based on a reconfiguration of Learning Factor Analysis. Unlike, BKT, it has the ability to handle multiple skill questions. Briefly speaking, it uses the form of the standard logistic regression model with the student performance as the dependent variable. It reconfigures LFA (Learning factors analysis) [13] on its independent variables, by dropping the student variable and replaces the skill variable with question identity. This model estimates parameters for each item's difficulty and also two parameters for each skill reflecting the effects of the prior correct and incorrect responses achieved for that skill. Previous work that compares KT and PFA have shown that PFA to be the superior one [11]. One reason is the richer feature set that PFA can utilize and the fact that learning decomposition models are ensured to reach global maxima while the typical fitting approach of BKT is no guarantee of finding a global, rather than a local maximum.

Beside the theoretical comparison of DKT, BKT, and PFA, we can also compare them visually by looking at the differences between them in terms of inputs data. Consider a simple scenario that a student answers two questions from two skills each, Tables 1-3 compare different training data formats for these three modeling methods under that same scenario of student responses.

**Table 1. An example of BKT's training data**

Model ID	Skill ID	Response Sequence
1	A	1,0
2	B	0,1

**Table 2. An example of PFA's training data**

Index ID	Skill ID	Prior Correct	Prior Incorrect	Difficulty	Correct
1	A	0	0	0.7	1
2	A	1	0	0.75	0
3	B	0	0	0.6	0
4	B	0	1	0.65	1

**Table 3. An example of DKT’s training data**

Index ID	One-hot encoding
1	1,0,0,0
2	0,0,1,0
3	0,0,0,1
4	0,1,0,0

### 3. METHODOLOGY AND DATA SETS

#### 3.1 Implementation of Deep Knowledge

##### Tracing in Tensorflow

The original version of DKT (Lua DKT<sup>1</sup>) was implemented in Lua scripting language using Torch framework and its source code has been released to the public. In order to have a comprehensive understanding of the DKT model, we decided to replicate and implement DKT model in Python and utilize Google’s TensorFlow API [3] to help us with building neural networks. TensorFlow is Google Brain’s second generation machine learning interface; it is flexible and can be used to express a wide variety of algorithms.

Our implementation of DKT in TensorFlow (TensorFlow DKT<sup>2</sup>) can be described as a directed graph, which is composed of a set of nodes. The graph represents a data flow computation, with extensions for allowing certain nodes to maintain and update persistent state and for branching and looking control, this is crucial for allowing RNN nodes to work on sequential data. In the directed graph, each node has zero or more inputs and zero or more outputs and represents the instantiation of an operation. An operation represents an abstract computation. In our implementation of DKT model, we adapted the loss function of the original DKT algorithm. It has 200 fully-connected hidden nodes in the hidden layer. To speed up the training process, we used mini-batch stochastic gradient descent to minimize the loss function. The batch size for our implementation is 100. For one batch, we randomly select data from 100 students in our training data. After the batch finishes training, 100 students in the batch are removed from the training data. We continue to train the model on next batch until all batches are done. Just as in the original Lua implementation, Dropout [4] was also applied to the hidden layer to avoid over-fitting.

### 4. DATA SETS

#### 4.1 ASSISTments 2009-2010 Data Set

The original DKT paper conducted one of three of experiments using the ASSISTments 2009-2010 skill builder data set [16]. This data set was gathered from ASSISTments’ skill builder problem sets, in which a student achieves mastery by working on similar (often isomorphic) questions until they can correctly answer  $n$  right in a row (where  $n$  is usually 3). After mastery, students do not commonly rework the same skill. This dataset contains 525,535 rows of student responses; there are 4,217 student ID’s and 124 skills. Lua DKT achieved an AUC of 0.86

<sup>1</sup> <https://github.com/chrispiech/DeepKnowledgeTracing>

<sup>2</sup> <https://github.com/siyuanzhao/2016-EDM>

and noticeably outperformed BKT (AUC = 0.67) on this data set. However, during our investigation on the DKT source code and application, we believe we discovered three issues that have unintentionally inflated the performance of Lua DKT. These issues are:

##### 4.1.1 Duplicated records

To our surprise and dismay, we found that the ASSISTments 2009-2010 data set has a serious issue of quality: large chunks of records are duplications that should not be there for any reason (e.g. see records of order id 36369610). These duplicated records have the same information but only differ on the “opportunity” and “opportunity\_original”; these two features record the number of opportunities a student has practiced on a skill and the number of practices on main problems of a skill respectively. It is impossible to have more than one ‘opportunity’ count for a single order id. This is definitely an error in the data set and these duplicated records should not be used in any analysis or modeling studies. We counted there are 123,778 rows of duplications out of 525,535 in the data set (23.6%). The existence of duplicated data is an avoidable oversight and ASSISTments team has acknowledged this error on their website. All new experiments in this work and following discussions exclude data of these duplications.

##### 4.1.2 Mixing main problems with scaffolding problems

A mastery learning problem set normally contains over a hundred of main problems, and each main problem may have multiple associated scaffolding problems. Scaffolding problems were designed to help students acquire an integrated set of skills through processes of observations and guided practice; they are usually tagged with different skills and have different designs from the main problems. Because of the difference in usage, scaffolding questions should not be treated as the same as main problems. Student modeling methods such as BKT and PFA exclude scaffolding features. The experiment conducted by Lua DKT did not filter out scaffolding problems. This means that Lua DKT had the advantage of additional information; thus, the prediction results cannot be compared fairly with BKT. There are 73,466 rows of records of scaffolding problems.

##### 4.1.3 Repeated response sequences with different skill tagging (Duplication by skill tag)

The 2009-2010 skill builder dataset was created as a subset of the 2009-2010 full dataset. The full dataset from 2009-2010 includes student work from both skill builder assignments (where a student works until a mastery threshold is reached) and more traditional assignments (where a student has a fixed number of problems). Any problem (or assignment) can be tagged with any number of skill tags. Typically, problems have just one skill tag; they seldom are tagged with two skills; they are very rarely tagged with three or more. Depending on the design of the content creator, a problem set may have multiple skill tags; many assignments - especially skill builders - will have the same skill tag for all problems. When the full dataset was decomposed into only mastery style assignments, the problems, and assignments that were tagged with multiple skills were included with a single tag, but repeated for each skill. This means that the sequence of action logs from one student working on one assignment was now repeated once per skill. For models such as RNNs that operate over sequences of vectors and memory on the entire history of

previous inputs, the issue of duplicated sequences is going to add additional weight onto the duplicated information; this will have undesired effects on RNN models.

For an example, suppose we have a hypothetical scenario that a student answers two problems which have been tagged with skill “A” and “B”; he answers first one correctly and the next one incorrectly. Table 4 shows the data set where responses have been repeated on skill “A” and “B”. This format of data can be used in BKT models since BKT can build two models for skill “A” and “B” separately. When applying this sequential data set to DKT, we believe DKT can recognize the pattern when a problem tagged with skill “B” follows a problem tagged with “A”; the skill “B” problem has an extremely high chance to repeat skill “A” problem’s response correctness. Note that skill ID can be mapped to skill names, but the order of skill ID is completely arbitrary.

**Table 4. An example of repeated multiple-skill sequence**

Index ID	Skill ID	Problem ID	Correctness
1	A	3	1
1	B	3	1
2	A	4	0
2	B	4	0

One approach to change the way of how multiple-skill problems are handled is to simply use the combination of skills as a new joint skill. Table 5 shows the data set which uses a joint skill of A and B. In this case, DKT no longer has access to repeated information. PFA and BKT can also adapt this format of data too.

**Table 5. An example of joint skills on multiple-skill problems**

Index ID	Skill ID	Problem ID	Correctness
1	A, B	3	1
2	A, B	4	0

**Table 6. Three variants of ASSISTments 2009-2010 Data set**

	09-10 (a)	09-10 (b)	09-10 (c)
Has duplicated records	No	No	No
Has scaffolding problems	Yes	No	No
Repeated multiple-skill sequences	Yes	Yes	No
Joint skills from multiple-skill	No	No	Yes

In order to understand the impact of having scaffolding problems and two approaches to dealing with multiple-skill problems, we generate three different data sets (namely 09-10 (a), 09-10 (b), 09-

10 (c)) derivate from the ASSISTments 2009-1010 data set, as summarized in Table 6.

## 4.2 ASSISTments 2014-2015 Data Set

Even without the issue of duplicate rows, 2009-2010 skill builder set has lost its timeliness and certainly cannot represent the latest student data in an intelligent tutoring system. So we gathered another data set that covers 2014-2015 school years’ student response records [16]. In this experiment, we randomly selected 100 skills from this year’s data records. This data set contains 707,944 rows of records; each record represents a response to a main problem in a mastery learning problem set. Each problem set has only one associated skill and we take caution to make sure there is no duplicated row in this data set. We suspect this new data set contains different information that covers student learning patterns, item difficulties and skill dependencies.

## 4.3 KDD Cup 2010 Data Set

Our last data set comes from the Cognitive Algebra Tutor 2005-2006 Algebra system [6]. This data was provided as a development dataset in the KDD Cup 2010 competition. Although both ASSISTments and Cognitive Algebra Tutor involve using mathematics skills to solve problems, they are actually rather different from each other. ASSISTments serves primarily as computer-assisted practice for students’ nightly homework and review lessons while the Cognitive Tutor is part of an integrated curriculum and has more support for learners during the problem-solving process. Another difference in terms of content structure is that the Cognitive Tutor presents a problem to a student that consists of questions (also called steps) of many skills. The Cognitive Tutor uses Knowledge Tracing to determine when a student has mastered a skill. A problem in the tutor can consist of questions of different skills, once a student has mastered a skill, as determined by KT, the student no longer needs to answer questions of that skill within a problem but must answer the other questions which are associated with the un-mastered skills. The number of skills in this dataset is substantially larger than the ASSISTments dataset [15]. One issue of using KDD data on PFA is how to estimate item difficulty feature. In this work, we use a concatenation of problem name and step name. However many such pairs are only attempted by 1 student and the difficulty values of these items are either 1.0 or 0.0, leading to both overfitting and data leakage. To fix that, we replace difficulty values of these items with skills’ difficulty information. Filtering out rows with missing values resulting in 607,026 rows of data with students responded correctly at 75.5% of the time. This KDD data set has 574 students worked on 436 skills in mathematics. The complete statistic information of five data sets can be found in Table 7.

**Table 7. Data set statistics**

	# records	# Students	# Skills
09-10 (a)	401,757	4,217	124
09-10 (b)	328,292	4,217	124
09-10 (c)	275,459	4,217	146
14-15	707,944	19,457	100
KDD	607,026	574	436

## 5. RESULTS

Student performance predictions made by each model are tabulated and the accuracy was evaluated in terms of Area Under Curve (AUC) and the square of Pearson correlation ( $r^2$ ). AUC and  $r^2$  provide robust metrics for evaluation predictions where the value being predicted is either a 0 or 1 also represents different information on modeling performance. An AUC of 0.50 always represents the scored achievable by random chance. A higher AUC score represents higher accuracy.  $r^2$  is the square of Pearson correlation coefficient between the observed and predicted values of dependent variable. In the case of  $r^2$ , it is normalized relative to the variance in the data set and it is not directly a measure of how good the modeled values are, but rather a way of measuring the proportion of variance we can explain using one or more variables.  $r^2$  is similar to root mean squared error (RMSE) but is more interpretable. For example, it is unclear whether an RMSE of 0.3 is good or bad without knowing more about the data set. However, an  $r^2$  of 0.8 indicates the model accounts for most of the variability in the data set. Neither AUC nor  $r^2$  method is a perfect evaluation metric, but, when combined, they account for different aspects of a model and provide us a basis for evaluating our models.

Experiments on every data set have been 5-fold student level cross-validated and all parameters are learned from training data. We used EM to train BKT and the limit of iteration was set to 200. Besides the number of hidden nodes and the size of mini-batch parameters we have discussed, we set the number of epochs of DKT to 100.

The cross-validated model predictions results are shown in Table 8 and Table 9. As can be seen, DKT clearly outperforms BKT on all data sets, but the results are no longer overwhelmingly in favor of DKT (both implementations). Note that Lua DKT implementation which we can access uses regular RNN nodes; TensorFlow DKT uses LSTM nodes.

**Table 8. AUC results**

	Torch DKT	TensorFlow DKT	PFA	BKT
09-10 (a)	0.79	0.81	0.70	0.60
09-10 (b)	0.79	0.82	0.73	0.63
09-10 (c)	0.73	0.75	0.73	0.63
14-15	0.70	0.70	0.69	0.64
KDD	0.79	0.79	0.71	0.62

**Table 9.  $r^2$  results**

	Lua DKT	TensorFlow DKT	PFA	BKT
09-10 (a)	0.22	0.29	0.11	0.04
09-10 (b)	0.22	0.31	0.14	0.07
09-10 (c)	0.14	0.18	0.14	0.07
14-15	0.10	0.10	0.09	0.06
KDD	0.21	0.21	0.10	0.05

On the ASSISTments data sets, average DKT prediction performance across two implementations is better than PFA and it is not affected by removing scaffolding, as we change dataset from 09-10 (a) to 09-10 (b). On the other hand, PFA's performance increases from 0.70 to 0.73 in AUC and 0.11 to 0.14 in  $r^2$  ( $p \leq 0.05$ ), we believe that removing scaffolding helps reducing noise from data and provides PFA with a dataset with lower variance. When we switch to dataset 09-10 (c) where multiple skills were combined into joint skills, the performance of DKT suffers a noticeable hit, average AUC and average  $r^2$  drop from 0.81 to 0.74 and from 0.30 to 0.18 respectively. This observation confirms our suspicion on repeated response sequence inflating the performance of DKT models. On the 09-10 (c) dataset and 14-15 dataset where no repeated response sequences and scaffolding problems, we notice that PFA performs as well as DKT.

A deeper way of looking at the impact of repeated response sequences on data set 09-10 (b) is splitting the prediction results into two, the predictions of leading records and repeated data points. We see that predictions on repeated data points (e.g. skill "B" problems in Table 4) have nearly perfect performance metrics (AUC = 0.97,  $r^2 = 0.74$ ). On the other hand, the leading records (e.g. skill "A" problems in Table 4) have much lower prediction results (AUC = 0.77,  $r^2 = 0.23$ ). That said, we also notice these numbers are still higher than 09-10 (c)'s results, which uses joint skill tags to avoid repeated sequences. One can explain this as making DKT to model skills individually can cause data duplications but it also can have benefits on building skill dependencies over time and use such information to make better predictions.

On the KDD dataset, the performance results of two DKT implementations are definitely better than both BKT and PFA ( $p \leq 0.05$ ). There are a few possible reasons for this performance gap between PFA and DKT. First of all, as we have mentioned, we have to adjust item difficulty values for many problems in order to avoid overfitting and data leakage, which leads to the lower predictive power of that feature and lower PFA performance. Another possible explanation of DKT is winning on KDD data set is that DKT can better exploit step responses. The structure of KDD data set made it is difficult to distinguish "main problems" and "scaffolding problems", thus PFA is unable to have a more unified data set for this part of the experiment. That said, the advantage of DKT shows its power on complicated and realistic data sets.

## 6. DISCUSSION AND CONTRIBUTION

Within this paper, we have compared two well-studied knowledge modeling methods with the emerging Deep Knowledge Tracing algorithm. We have compared these models in terms of their power of predicting student performance in 5 different data sets. Contrary to our expectation, the DKT algorithm did not achieve overwhelmingly better performance when compared to PFA model on ASSISTments data sets when they are properly prepared. DKT appears to perform much better on KDD dataset, but we believe this is due to PFA model undermined by inaccurate item difficulty estimation.

A second interesting finding is that when DKT is fed repeated response sequences derived from the transformation of problems tagged with multiple skills, the overall performance of DKT is certainly better than PFA and BKT. Our explanation is that DKT's implementation backbone, RNNs, has the power of

remembering exact patterns of sequential data and could thus inflate prediction performance on responses tagged with multiple skills and repeated per skill. More discussion and special attention are required when handling multiple skill problems in DKT algorithm.

Last, but not least, during the investigation of DKT, we discovered an issue in data quality arising from duplicated information in a publicly available data set. The duplication issues (caused by unclear transformational rules and some other as-of-yet-to-be-ascertained cause) allowed us a natural experiment to examine the impact of duplications on the robustness of these algorithms. These discoveries (the data duplications and their subsequent impact) should serve as a reminder of the importance of data preprocessing and transformation procedures in the work of knowledge discovery and data mining. Or, put another way, while we advance new algorithms and fine tune their parameters, we should also consider (and, if possible, report on) the robustness of the algorithms to common data glitches.

## 7. FUTURE WORK AND CONCLUSION

There are several directions for further research in the area of DKT modeling. Prior work [2] has shown that the use of context-dependent RNN language model improved the performance in the task of the Wall Street Journal speech recognition task. More features like student features (e.g. prior knowledge, completion rates, time on learning, etc.), and content features (problem difficulty, skill hierarchies, etc.) may be available and could be used. A context-dependent DKT implementation could be created by adding an extra input vector containing these features.

Another open area for future work is that DKT and other deep learning algorithms are not restricted to one kind of output or application. It is also possible that we could apply deep learning algorithms on other modeling challenges such as wheel spinning, mastery speed, and affect detection.

In conclusion, our work here focuses on a primitive investigation of DKT and aims to provide us deeper insight on how DKT works. Overall, this paper suggests that DKT remains a promising approach to modeling student knowledge; however, we see that data which contains problems tagged with multiple skills has to be dealt carefully in DKT modeling. But, considering that this implementation of DKT: a) only relied on the sequences of student responses (just as BKT does) and no other information on skills and problems and b) performs substantially better than BKT and as good as PFA, we believe that DKT has great potential to outperform other methods when it utilizes more features.

## 8. ACKNOWLEDGEMENTS

We thank multiple current NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428 & DRL-1031398), the US Dept. of Ed (IES R305A120125 & R305C100024 and GAANN), and the ONR.

## 9. REFERENCES

[1] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems* (pp. 505-513).

[2] Mikolov, T., & Zweig, G. (2012, July). Context dependent recurrent neural network language model. In *SLT* (pp. 234-239).

[3] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., & Ghemawat, S. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. White paper, Google Research.

[4] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.

[5] Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251), aac4716.

[6] Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G.J., & Koedinger, K.R. (2010). Algebra I 2005-2006. Challenge data set from KDD Cup 2010 Educational Data Mining Challenge. Find it at <http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp>

[7] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.

[8] Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009, June). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 41-48). ACM.

[9] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis-A New Alternative to Knowledge Tracing. Online Submission.

[10] 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.

[11] Gong, Y., Beck, J. E., & Heffernan, N. T. (2010, June). Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent tutoring systems* (pp. 35-44). Springer Berlin Heidelberg.

[12] Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic bulletin & review*, 7(2), 185-207.

[13] Cen, H., Koedinger, K., & Junker, B. (2006, June). Learning factors analysis—a general method for cognitive model evaluation and improvement. In *Intelligent tutoring systems* (pp. 164-175). Springer Berlin Heidelberg.

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

[15] Pardos, Z. A., & Heffernan, N. T. (2011). KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization* (pp. 243-254). Springer Berlin Heidelberg.

[16] ASSISTments Data. (2015). Retrieved March 07, 2016, from <https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010>

# Boosted Decision Tree for Q-matrix Refinement

Peng Xu  
Polytechnique Montreal  
peng.xu@polymtl.ca

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

## ABSTRACT

In recent years, substantial improvements were obtained in the effectiveness of data driven algorithms to validate the mapping of items to skills, or the Q-matrix. In the current study we use ensemble algorithms on top of existing Q-matrix refinement algorithms to improve their performance. We combine the boosting technique with a decision tree. The results show that the improvements from both the decision tree and Adaboost combined are better than the decision tree alone and yield substantial gains over the best performance of individual Q-matrix refinement algorithm.

## 1. INTRODUCTION

A Q-matrix, as proposed by Tatsuoka (Tatsuoka, 1983), is a term commonly used in the literature of psychometrics and cognitive modeling that refers to a binary matrix which shows a correspondence between items and their latent attributes. Items can be questions or exercises proposed to students, and latent attributes are skills needed to succeed these items. Usually, a Q-matrix is defined by a domain expert. However, this task is non trivial and there might be errors, which in turn will result in erroneous diagnosis of students knowledge states (Rupp & Templin, 2008; Madison & Bradshaw, 2015). Therefore, better means to validate a Q-matrix is a highly desirable goal.

A fair number of algorithms have emerged in the last decade to validate an expert given Q-matrix based on empirical data (see for eg. recent work from Chen, Liu, Xu, & Ying, 2015; de la Torre & Chiu, 2015; Durand, Belacel, & Goutte, 2015). Desmarais, Xu, and Beheshti (2015) showed that Q-matrix refinement algorithms can be combined using an ensemble learning technique. They used a decision tree and the results show a substantial and systematic performance gain over the best algorithm, in the range of 50% error reduction for real data, even though the best algorithm is not always the same for different Q-matrices.

The encouraging the results obtained by combining the out-

put of Q-matrix refinement algorithms leads us to pursue further along the line of using ensemble learning, or meta-learning techniques. In particular, a common approach is to use boosting with a decision tree algorithm. This is the approach explored in the current study.

## 2. THREE TECHNIQUES TO Q-MATRIX VALIDATION

Our approach relies on meta-learning algorithms whose principle in a general way is to combine the output of existing algorithms to improve upon the individual or average results. In our case, the approach combines a decision tree trained on the output of Q-matrix validation algorithms with boosting, a weighted sampling process in the training of the decision tree to improve its accuracy. In this section, we first describe the Q-matrix validation techniques before describing the decision tree and boosting algorithms.

### 2.1 minRSS

The first Q-matrix refinement technique that serves as input to the decision tree is from Chiu and Douglas (2013). We name this technique minRSS. The underlying cognitive model behind minRSS is the DINA model(see De La Torre, 2009).

For a given Q-matrix, there is a unique and ideal response pattern for a given a student skills mastery profile. That is, if there are no slip and guess factors, then the response pattern for every category of student profile is fixed. The difference between the real response pattern and the ideal response pattern represents a measure of fit for the Q-matrix, typically the Hamming distance. Chiu and Douglas (2013) considered a more refined metric. The idea is if an item has a smaller variance (or entropy), then it should be given a higher weight in measure of fit. A first step is to compute the ideal response matrix for all possible student profile, and then to find the corresponding student profiles matrix  $A$  given observed data. First, a squared sum of errors for each item  $k$  can be computed by

$$RSS_k = \sum_{i=1}^N (r_{ik} - \eta_{ik})^2$$

where  $r$  is the real response vector while  $\eta$  is the ideal response vector, and  $N$  is the number of respondents. Then, the worst fitted item (highest  $RSS$ ) is chosen to update its correspondent q-vector. Given all permutations of the skills for a q-vector, the q-vector with the lowest  $RSS$  is chosen to

replace the original one. The Q-matrix is changed and the whole process repeated, but the previously changed q-vector is eliminated from the next iteration. The whole procedure terminates when the *RSS* for each item no longer changes. This method was shown by Wang and Douglas (2015) to yield good performance under different underlying conjunctive models.

## 2.2 maxDiff

Akin to minRSS, the maxDiff algorithm relies on the DINA model. De La Torre (2008) proposed that a correctly specified q-vector for item  $j$  should maximize the difference of probabilities of correct response between examinees who have all the required attributes and those who do not. A natural idea is to test all q-vectors to find that maximum, but that is computationally expensive. De La Torre (2008) proposed a greedy algorithm that adds skills into a q-vector sequentially. Assuming  $\delta_{jl}$  represents the difference to maximize, the first step is to calculate  $\delta_{jl}$  for all q-vectors which contains only one skill and the one with biggest  $\delta_{jl}$  is chosen. Then,  $\delta_{jl}$  is calculated for all q-vectors which contains two skills including the previously chosen one. Again the q-vector with the biggest  $\delta_{jl}$  is chosen. This whole process is repeated until no addition of skills increases  $\delta_{jl}$ . However, this algorithm requires knowing slip and guess parameters of the DINA model in advance. For real data, they are calculated by EM (Expectation Maximization) algorithm (De La Torre, 2009).

## 2.3 ALSC

ALSC (Conjunctive Alternating Least Square Factorization) is a common matrix Factorization (MF) algorithm. Desmarais and Naceur (2013) proposed to factorize student test results into a Q-matrix and a skills-student matrix with ALSC.

ALSC decomposes the results matrix  $\mathbf{R}_{m \times n}$  of  $m$  items by  $n$  students as the inner product two smaller matrices:

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

where  $-\mathbf{R}$  is the negation of the results matrix ( $m$  items by  $n$  students),  $\mathbf{Q}$  is the  $m$  items by  $k$  skills Q-matrix, and  $-\mathbf{S}$  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. As such, the model of equation (1) is analogous to the DINA model without a slip and guess parameter.

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:

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

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

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

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.

## 3. DECISION TREE

The three algorithms for Q-matrix refinement described in the last section are to be combined to yield with a decision tree to obtain an improved refinement recommendation, and further improved by boosting. We describe the decision tree before moving on to the boosting method.

Decision tree is a well-know technique in machine learning and it often serves as an ensemble learning algorithm to combine individual models into a more powerful model. It uses a set of feature variables (individual model predictions) to predict a single target variable (output variable). There are several decision tree algorithms, such as ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), CART (Breiman, Friedman, Stone, & Olshen, 1984). We used `rpart` function from the R package of the same name (Therneau, Atkinson, & Ripley, 2015). It implements the CART algorithm. This algorithm divides the learning process into two phases. The first phase is for feature selection, or tree growing, during which the feature variables are chosen sequentially according to *Gini impurity* (Murphy, 2012). Then in the second phase, the pruning phase, deep branches are split into wider ones to avoid overfitting.

A decision tree is a supervised learning technique and therefore requires training data. To obtain training data of sufficient size, Desmarais et al. (2015) use synthetic data from Q-matrices generated by random permutations of the perturbed Q-matrix. Since the ground-truth Q-matrix of synthetic data is known, it becomes possible to generate training data containing the class label. The training set for decision tree can take this form:

Target	Algorithm target prediction			Other factors
	minRSS	maxDiff	ALSC	...
1	1	0	1	...
0	0	1	0	...
...	...	...	...	...

The other factors considered to help the decision tree to improve prediction are the number of skills per row (SR), number of skills per column (SC). Moreover, a feature named *stickiness* is introduced and makes a critical difference. It measures the rigidity of cells under each validation methods. Stickiness represents the rate of a given algorithm's false positives for a given cell of a Q-matrix. The rate is measured by "perturbating" in turn each and every cell of the Q-matrix, and by counting the number of times the cell is a false positive. The decision tree can use the stickiness factor as an indicator of the reliability of a given Q-matrix refinement algorithm suggested value for a cell. Obviously, if a cell's stickiness value is high, the reliability of the corresponding algorithm's refinement will be lower.

## 4. BOOSTING

The current work extends the idea of using a decision tree with another meta-learning technique named boosting.

Boosting (Schapire & Freund, 2012) serves as a meta-learning technique for lifting a base learner. It operates on weights of the loss function terms. For a training set of  $N$  samples

and a given loss function  $L$ , the global loss is

$$Loss = \sum_{i=1}^N L(y_i, f(x_i))$$

Different ways of choosing loss function yield different boosting algorithm. The most famous algorithm for boosting is Adaboost (Freund & Schapire, 1997), which is especially set for binary classification problem and uses exponential loss.

In our case, the base learner is the decision tree. Adaboost trains the decision tree for several iterations, but with a different weighted training data for each iteration. That is, each time a decision tree is trained, the wrongly predicted data records in the current iteration will be assigned higher weights in the computation of the loss function for the next training iteration of the decision. The final output of Adaboost is a **sgn** function (*sign function*) of a weighted sum of all “learners” trained in the whole procedure (the decision tree with different weights vectors).

For a training set of  $N$  samples, the whole procedure for Adaboost is shown below (Murphy, 2012):

```

Initialize  $\omega_i = 1/N$ 
for  $i = 1$  to  $M$  do
  Fit the classifier  $\phi_m(x)$  to the training set using weights  $w$ 
  Compute  $err_m = \frac{\sum_{i=1}^N \omega_i I(\hat{y}_i \neq \phi_m(x_i))}{\sum_{i=1}^N \omega_i}$ 
  Compute  $\alpha_m = \log[(1 - err_m)/err_m]$ 
  set  $\omega_i \leftarrow \omega_i \exp[\alpha_m I(\hat{y}_i \neq \phi_m(x_i))]$ 
end for
return  $f(x) = \text{sgn}(\sum_{m=1}^M \alpha_m \phi_m(x))$ 

```

In which  $M$  is the number of iterations (10 in our experiment),  $\omega_i$  is the weight for  $i$ -th data,  $I(\cdot)$  is the indicator function,  $\hat{y}_i \in \{1, -1\}$  is the class label of training data, and  $\phi_m(x)$  is the decision tree model in our case.

Boosting has had stunning empirical success (Caruana & Niculescu-Mizil, 2006). More detailed explanation and analysis of boosting can be found in Bühlmann and Hothorn (2007) and Hastie, Tibshirani, and Friedman (2009). The Adaboost algorithm was implemented in this experiment to improve the results obtained by Desmarais et al. (2015). The results are reported in section 7.

## 5. METHODOLOGY AND PERFORMANCE CRITERION

To estimate the ability of an algorithm to validate a Q-matrix, we perturbate a “correct” Q-matrix and verify if the algorithm is able to recover this correct matrix by identifying the cells that were perturbed while avoiding to classify unperturbed cells as perturbed. In this experiment, only one perturbation is introduced. For synthetic data, the “correct” matrix is known and is the one used in the generation of the data. For real data, we assume the expert’s is the correct one, albeit it may contain errors.

Table 1: Q-matrix for validation

Name	Number of			Description
	Skills	Items	Cases	
QM1	3	11	536	Expert driven from (Henson, Templin, & Willse, 2009)
QM2	3	11	536	Expert driven from (De La Torre, 2008)
QM3	5	11	536	Expert driven from (Robitzsch, Kiefer, George, & Uenlue, 2015)
QM4	3	11	536	Data driven, SVD based

In order to use a standard performance measure, we define the following categories of correct and incorrect classifications as the number of:

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

We give equal weight to perturbed and unperturbed cells and use the F1-score, or F-score for short. The F-score is defined as

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

In which *precision* is calculated by the model accuracy on non-perturbed cell while *recall* is calculated by the model accuracy on perturbed cell.

## 6. DATASET

For the sake of comparison, we use the same datasets as the ones used in Desmarais et al. (2015). Table 1 provides the basic information and source of each dataset.

## 7. RESULT

The results of applying Adaboost over the decision tree (DT) are reported in table 2 for synthetic data and Table 3 for real data. The individual results of each algorithm are reported (minRSS, maxDiff, and ALSC), along with the decision tree (DT) and the boosted decision tree (BDT). Different improvement over baselines are reported as:

- **DT %Gain**: the Decision Tree (DT) improvement over the **best** of the three individual algorithm (minRSS, maxDiff, ALSC)
- **BDT %Gain**: Boosted Decision Tree improvement over the DT performance, which corresponds to the gain we get from boosting.

Let us focus on the F-Score which is the most informative since it combines results of the perturbed and non perturbed

Table 2: Results for synthetic data

QM	Individual			Ensemble		
	minRSS	maxDiff	ALsC	DT (%Gain)	BDT (%Gain)	
Accuracy of perturbed cells						
1	0.809	0.465	0.825	0.946 (69.4%)	<b>0.951</b> (9.2%)	
2	0.069	0.259	0.359	0.828 (73.2%)	<b>0.903</b> (43.5%)	
3	0.961	0.488	0.953	<b>1.000</b> (99.7%)	<b>1.000</b> (0.0%)	
4	0.903	0.489	0.853	0.956 (54.3%)	<b>0.971</b> (33.9%)	
$\bar{X}$	0.685	0.425	0.747	0.933 (74.2%)	<b>0.956</b> (21.7%)	
Accuracy of non perturbed cells						
1	0.970	0.558	0.387	<b>0.990</b> (65.1%)	<b>0.990</b> (0.0%)	
2	0.987	0.529	0.431	0.989 (20.5%)	<b>0.996</b> (59.1%)	
3	0.950	0.258	0.736	0.994 (88.9%)	<b>1.000</b> (100.0%)	
4	0.966	0.559	0.391	0.997 (92.2%)	<b>0.998</b> (19.2%)	
$\bar{X}$	0.968	0.476	0.486	0.993 (65.3%)	<b>0.996</b> (49.4%)	
F-score						
1	0.882	0.507	0.527	0.968 (72.4%)	<b>0.970</b> (7.4%)	
2	0.128	0.348	0.392	0.902 (83.8%)	<b>0.947</b> (46.1%)	
3	0.955	0.337	0.831	0.997 (93.5%)	<b>1.000</b> (100.0%)	
4	0.934	0.522	0.536	0.976 (64.0%)	<b>0.984</b> (33.6%)	
$\bar{X}$	0.725	0.429	0.571	0.961 (78.4%)	<b>0.975</b> (46.4%)	

cells of the Q-matrix. For synthetic data, the error reduction of boosting over the gain from the decision tree is substantially improved for all Q-matrices. The range of improvement is from 7% to 100%. For real data, two of the four Q-matrices show substantial improvements of around 40%, whereas the other two show no improvements, even a decrease of 8.7% for Q-matrix 3 which is characterized by a single skill per item. However, let us recall that we assume the expert Q-matrices are correct, which may be over optimistic. Violation of this assumption could negatively affect some of the Q-matrices scores for real data.

Note that QM3 has an inconsistent 100% gain from boosting with synthetic data compared to a small loss is obtained with real data. The value of 100% should be taken cautiously because the F-score difference is measured close to the boundary of 1 and therefore the result of only a few cases in our sample. Nevertheless, the fact that a very high F-score is obtained for synthetic data compared to real data does raise attention and might be related to the fact that it is the only single skill per item matrix.

## 8. DISCUSSION

This study shows that the gain obtained from combining the output of multiple Q-matrix refinement algorithms with a decision tree can be further improved with boosting. The results for synthetic data show an F-score error reduction from boosting over the DT score of close to 50% on average for all four Q-matrices, and a 18% reduction for real data. Compared with the score of the three individual refinement algorithms, minRSS, maxDiff, and ALsC, the combined ensemble learning of decision tree is very effective.

Table 3: Results for real data

QM	Individual			Ensemble		
	minRSS	maxDiff	ALsC	DT (%Gain)	BDT (%Gain)	
Accuracy of perturbed cells						
1	0.485	0.167	0.515	<b>0.758</b> (50.0%)	<b>0.758</b> (0.0%)	
2	0.345	0.093	0.564	0.618 (12.5%)	<b>0.764</b> (38.1%)	
3	0.212	0.091	0.364	<b>0.818</b> (71.4%)	<b>0.818</b> (0.0%)	
4	0.394	0.111	0.576	0.576 (0.0%)	<b>0.818</b> (57.1%)	
$\bar{X}$	0.359	0.115	0.505	0.692 (33.5%)	<b>0.789</b> (23.8%)	
Accuracy of non perturbed cells						
1	0.435	<b>0.670</b>	0.418	0.606 (-19.4%)	0.606 (0.0%)	
2	0.875	0.929	0.110	0.956 (37.9%)	<b>0.966</b> (21.4%)	
3	0.661	<b>0.830</b>	0.219	0.785 (-26.2%)	0.752 (-15.1%)	
4	0.520	<b>0.889</b>	0.148	0.546 (-308.7%)	0.658 (24.7%)	
$\bar{X}$	0.623	<b>0.829</b>	0.224	0.723 (-79.1%)	0.746 (8.0%)	
F-score						
1	0.459	0.267	0.461	<b>0.673</b> (39.4%)	<b>0.673</b> (0.0%)	
2	0.495	0.168	0.184	0.751 (50.6%)	<b>0.853</b> (40.9%)	
3	0.321	0.164	0.273	<b>0.801</b> (70.7%)	0.784 (-8.7%)	
4	0.448	0.198	0.235	0.560 (20.3%)	<b>0.730</b> (38.5%)	
$\bar{X}$	0.431	0.199	0.288	0.696 (45.25%)	<b>0.760</b> (17.8%)	

However, we find strong differences between the Q-matrices. For eg., QM2 benefits of improvements close to 50% (QM2), while QM1 has a null improvement for real data and only 7.4% for synthetic data. In that respect, the boosting does not provide a gain that is as systematic as the one obtained from the DT which is positive for all matrices.

An important advantage of the meta-learning approach outlined here is that it can apply to any combination of algorithms to validate Q-matrices. Future work could look into combining more than the three algorithms of this study, and add new algorithms that potentially outperform them. And if the current results generalize, we would expect to make supplementary gains over any of them.

Moreover, the Q-matrices used in this research are quite small in size. The performance of boosted decision tree on larger Q-matrix and larger dataset would also be of interest.

However, besides the Q-matrix-based algorithms mentioned above, there are other frameworks for knowledge tracing or domain modeling, especially when dealing with dynamic data. For example, there are Learning Factor Analysis (Cen, Koedinger, & Junker, 2006), Weighted CRP (Lindsey, Khajah, & Mozer, 2014), HMM-based Bayesian Knowledge Tracing (Corbett & Anderson, 1994; Lindsey et al., 2014) and other HMM-based models (González-Brenes, 2015). Comparison with these frameworks are also left to future work.

## References

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

- Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 477–505.
- Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on machine learning* (pp. 161–168).
- Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis—a general method for cognitive model evaluation and improvement. In *Intelligent tutoring systems* (pp. 164–175).
- Chen, Y., Liu, J., Xu, G., & Ying, Z. (2015). Statistical analysis of q-matrix based diagnostic classification models. *Journal of the American Statistical Association*, 110(510), 850–866.
- Chiu, C.-Y., & Douglas, J. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. *Journal of Classification*, 30(2), 225–250.
- 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.
- De La Torre, J. (2008). An empirically based method of q-matrix validation for the dina model: Development and applications. *Journal of educational measurement*, 45(4), 343–362.
- De La Torre, J. (2009). Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1), 115–130.
- de la Torre, J., & Chiu, C.-Y. (2015). A general method of empirical q-matrix validation. *Psychometrika*, 1–21.
- Desmarais, M. C., & Naceur, R. (2013). A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In *Artificial intelligence in education* (pp. 441–450).
- Desmarais, M. C., Xu, P., & Beheshti, B. (2015). Combining techniques to refine item to skills q-matrices with a partition tree. In *Educational data mining 2015*.
- Durand, G., Belacel, N., & Goutte, C. (2015). Evaluation of expert-based q-matrices predictive quality in matrix factorization models. In *Design for teaching and learning in a networked world* (pp. 56–69). Springer.
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1), 119–139.
- González-Brenes, J. P. (2015). Modeling skill acquisition over time with sequence and topic modeling. In *Aistats*.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer.
- Henson, R. A., Templin, J. L., & Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2), 191–210.
- Lindsey, R. V., Khajah, M., & Mozer, M. C. (2014). Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in neural information processing systems* (pp. 1386–1394).
- Madison, M. J., & Bradshaw, L. P. (2015). The effects of q-matrix design on classification accuracy in the log-linear cognitive diagnosis model. *Educational and Psychological Measurement*, 75(3), 491–511.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1), 81–106.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning* (Vol. 1). Morgan Kaufmann.
- Robitzsch, A., Kiefer, T., George, A. C., & Uenlue, A. (2015). Cdm: Cognitive diagnosis modeling [Computer software manual]. Retrieved from <http://CRAN.R-project.org/package=CDM> (R package version 4.5-0)
- Rupp, A. A., & Templin, J. (2008). The effects of q-matrix misspecification on parameter estimates and classification accuracy in the dina model. *Educational and Psychological Measurement*, 68(1), 78–96.
- Schapire, R. E., & Freund, Y. (2012). *Boosting: Foundations and algorithms*. MIT press.
- Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of educational measurement*, 20(4), 345–354.
- Therneau, T. M., Atkinson, B., & Ripley, B. (2015). rpart: Recursive partitioning. r package version 4.1-10. *Computer software program retrieved from http://CRAN.R-project.org/package=rpart*.
- Wang, S., & Douglas, J. (2015). Consistency of nonparametric classification in cognitive diagnosis. *Psychometrika*, 80(1), 85–100.

# Individualizing Bayesian Knowledge Tracing. Are Skill Parameters More Important Than Student Parameters?

Michael V. Yudelson  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213 USA  
+1 (412) 268-5595  
yudelson@cs.cmu.edu

## ABSTRACT

Bayesian Knowledge Tracing (BKT) models were in active use in the Intelligent Tutoring Systems (ITS) field for over 20 years. They have been intensively studied, and a number of useful extensions to them were proposed and experimentally tested. Among the most widely researched extensions to BKT models are various types of individualization. Individualization, broadly defined, is a way to account for variability in students that are working with the ITS that uses BKT model to represent and track student learning. One of the approaches to individualizing BKT is to split its parameters into per-skill and per-student components. In this work, we are proposing an approach to individualizing BKT that is based on Hierarchical Bayesian Models (HBM) and, in addition to capturing student-level variability in the data, weighs the contribution of per-student and per-skill effects to the overall variance in the data.

## Keywords

Student models of practice, Bayesian knowledge tracing, hierarchical Bayesian models, skill vs. student parameterization.

## 1. INTRODUCTION

Bayesian Knowledge Tracing (BKT) is one of the most popular student modeling techniques in the field of Intelligent Tutoring Systems (ITS). It has been in active use for over two decades and has been confirmed to be the modeling approach researchers can rely on.

Over the years, a large number of extensions to the standard BKT were proposed and tested in posthoc analyses as well as experimentally. Among the most widely researched additions to BKT is the ability to account for students' individual traits. It has been confirmed in the area of modeling student learning in general and in the case of BKT that accounting for student-level variability in the data could benefit the model's statistical goodness of fit, as well as potentially improve the generalizability of the model.

Known approaches could be separated into three groups. The first group, binary multiplexing of the initial skill mastery probability based on the student characteristics, for example, the correctness of the first response (Pardos & Heffernan, 2010). This method has been proven to benefit the overall student model quality, and the implementation of this approach was a runner-up in the 2010 KDD Cup data mining challenge. The second group, fitting BKT parameters not across students for a particular skill, but for a student/skill pair (Lee & Brunskill, 2012). This approach has not been evaluated for predictive correctness. The third group, are the methods separating BKT parameters into per-student and per-skill components (Corbett & Anderson, 1995; Yudelson et al., 2013).

The two approaches from the third group were shown to improve model fits reliably.

While the BKT individualization approaches mentioned above were successful in one way or the other, are arguably yet to achieve a sufficient flexibility and rigor of the available parameterization devices. In this paper, we propose and investigate an individualized Bayesian Knowledge Tracing that, on top of refining certain aspects of its predecessor (Yudelson et al., 2013), draws on the flexibility of the Hierarchical Bayesian Models' representation to capture relative weight of student-level and skill-level variability in the learning data as defined by respective parameters. Also, we empirically explore the possibility of clustering student-level factors via mixes of Gaussian distributions.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 outlines the methods. Section 4 describes the data we used for this investigation. Section 5 talks about the results. Finally, Section 6 closes with a few discussion points.

## 2. RELATED WORK

### 2.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) is a probabilistic framework (Corbett & Anderson, 1995) it is used to assess student progress with a unit of knowledge often referred to as skill. Upon correct or incorrect action, an estimate of student mastery of skill(s) is re-computed. Computationally, BKT is a Hidden Markov Model with two hidden states, representing whether a particular skill is un-mastered or mastered. Observations of student performance on opportunities to practice a skill are binary: a student either solves a problem step correctly or not (due to error or because of a hint request). While students might go through dozens of attempts to get a particular step correct, traditionally, only students' first attempts are considered for updating skill mastery estimates.

There are four skill parameters used in BKT: initial probability of knowing the skill a priori –  $p(L_0)$  (or  $p-init$ ), probability of student's knowledge of a skill transitioning from not known to known state after an opportunity to apply it –  $p(T)$  (or  $p-learn$ ), probability to make a mistake when applying a known skill –  $p(S)$  (or  $p-slip$ ), and probability of correctly applying a not-known skill –  $p(G)$  (or  $p-guess$ ). Given that parameters are set for all skills, the formulae used to update student knowledge of skills are as follows. The initial probability of student  $u$  mastering skill  $k$  is set to the  $p-init$  parameter for that skill Equation (1a). Depending on whether the student  $u$  applied skill  $k$  correctly or incorrectly, the conditional probability is computed either using Equation (1b) or Equation (1c). The conditional probability is used to update the

probability of skill mastery according to Equation (1d). To compute the probability of student  $u$  applying the skill  $k$  correctly on an upcoming practice opportunity one uses Equation (1e).

$$p(L_1)_u^k = p(L_0)^k \quad (1a)$$

$$p(L_{t+1}|obs = correct)_u^k = \frac{p(L_t)_u^k \cdot (1 - p(S)^k)}{p(L_t)_u^k \cdot (1 - p(S)^k) + (1 - p(L_t)_u^k) \cdot p(G)^k} \quad (1b)$$

$$p(L_{t+1}|obs = wrong)_u^k = \frac{p(L_t)_u^k \cdot p(S)^k}{p(L_t)_u^k \cdot p(S)^k + (1 - p(L_t)_u^k) \cdot (1 - p(G)^k)} \quad (1c)$$

$$p(L_{t+1})_u^k = p(L_{t+1}|obs)_u^k + (1 - p(L_{t+1}|obs)_u^k) \cdot p(T)^k \quad (1d)$$

$$p(C_{t+1})_u^k = p(L_{t+1})_u^k \cdot (1 - p(G)^k) + (1 - p(L_{t+1})_u^k) \cdot p(G)^k \quad (1e)$$

## 2.2 Introducing Student-Level Factors to the Bayesian Knowledge Tracing

Having student-level parameters is a regular feature of models of student learning and learning performance. The logistic regression based Rasch model (van der Linden & Hambleton, 1997) that captures test item complexity and its extension –the Additive Factors Model (Cen et al., 2008) both include a parameter to account for variability in the student a priori abilities. Including student-level parameters in these models helps both the fit as well as the interpretability of the models overall.

There were a few attempts to introduce student-specific parameters to otherwise skill-only standard BKY. The original work on BKT (Corbett & Anderson, 1995) discussed fitting skill-level and student-level parameters on respective slices of the data to later combine and apply the two in the context of each student-skill pair. As a result, the correlation of expected and observed within-student accuracies was higher for the thus individualized model.

Another approach to individualization suggests the multiplexing probability of initial skill mastery ( $p-init$ ) based on student cohort (Pardos & Heffernan, 2010). Based on the correctness of the first student’s response, the appropriate skill  $p-init$  is set to the lower or higher predetermined constant. This prior-per-student model outperforms standard BKT on a significant fraction of problem sets authors considered.

According to yet another approach (Lee & Brunskill, 2012), BKT parameters were fit within each student-skill pair’s data slice and not across skills or students. Authors did not discuss on the goodness of fit of their individualized models, however. Their primary focus was on whether the individualized model when deployed in an intelligent tutoring system, would schedule fewer or more problems to be solved as compared to standard BKT model. The conclusion was that a considerable fraction of students, as judged by individualized model, would have received a significantly different amount of practice problems.

Finally, another individualization approach that we would be

using for comparison in this work suggests something akin to the original discussion of the BKT individualization (Yudelson et al., 2013). Student and skill components of BKT parameters are fit one set after the other using a coordinate gradient descent procedure with an active parameter set maintained throughout the process. In addition to improved fits, BKT models individualized this way were shown to lead to optimized problem-sequences leading to saving students some efforts.

Overall, there is enough evidence that introducing student-level parameters to BKT benefits the fit of the model and could optimize student learning experience.

## 2.3 Introducing Item-Level Factors to the Bayesian Knowledge Tracing

Recently, a noticeable amount of work focused on addressing item-level variability in BKT models. Pardos & Heffernan (2011) presented their KT-IDEM model that features special nodes that capture item difficulties and, together with skill-level latent variables are influencing the student performance.

In the approach Huang and colleagues took (Huang et al., 2015), it is possible to address not just items, but even item level features, adding parameters in a way it is done in regression analysis. In another work (Khajah et al., 2014), authors are discussing merging an IRT model and BKT model. This approach resulted in an HBM that combines features of both. It is worth to note that the latter two use Markov Chain Monte Carlo methods to fit their models.

## 3. METHODS

Our objective is to introduce further improvements to the approach to individualizing BKT and draw comparisons to regular BKT as well its original version in terms of statistical fitness as well as and to attempt to judge the plausibility of their respective student-level parameters.

### 3.1 Individualized BKT Model via Parameter-Splitting

Individualization of the BKT that was proposed in (Yudelson et al., 2013) prescribes to put every individualized parameter in the context of a particular student that works on a particular skill. In this context,  $p-init$ ,  $p-learn$ ,  $p-slip$ , and  $p-guess$  parameters have two components: a per-skill component and a per-student component. The two are combined using a pairing function shown in Equation 2a. Here, components are first converted from probability scale to log-odds scale using logit function (Equation 2b), added, and the sum is converted back to the probability scale using sigmoid function (Equation 2c). An individualized model, where all per-student components are equal to 0.5 (0 on the log-odds scale) is equivalent to the standard BKT model.

$$f(P_k^i, P_u^i) = S(l(P_k^i) + l(P_u^i)) \quad (2a)$$

$$l(p) = \ln\left(\frac{p}{1-p}\right) \quad (2b)$$

$$S(x) = \frac{1}{1 + e^{-x}} \quad (2c)$$

Fitting of such individualized BKT (iBKT) model is done by computing gradients of the log-likelihood function given individual student/skill data samples with respect to every iBKT parameter (Levinson et al., 1983). On every odd run, gradients are aggregated across skills to update skill component of the parameters. On every even run, the gradients are aggregated across students to update respective student components. This

block-coordinate descent is performed until all parameter values stabilize up to a pre-set tolerance criterion. An active set of parameter components is maintained to fit only those that still haven't stabilized. An extended discussion of the method, as well as derived formulas for the gradients is given in the original publication of this work (Yudelson et al., 2013).

The standard and individualized model described above we implemented in the tool called `hmm-scalable`. The tool has a suite of solvers, including the classical BKT Expectation Maximization solver for standard BKT, as well as a set of stochastic and conjugate gradient descent solvers. `Hmm-scalable` is freely available on GitHub repository<sup>1</sup> of the International Educational Data Mining Society (standard BKT models only).

### 3.2 Individualized BKT via Hierarchical Bayesian Model

We have also implemented the BKT as well as the iBKT approach described above in the form of a Hierarchical Bayesian Model (HBM). HBMs allow for a more universal and flexible way of representing iBKT. The HMB BKT just like the `hmm-scalable` BKT had  $4N$  parameters, where  $N$  is the number of skills. In the iBKT models, both `hmm-scalable` and HBM version, only the  $p$ -*init* and  $p$ -*learn* were individualized. Thus, the number of parameters in the `hmm-scalable` version of iBKT was  $4N+2M$ , where  $M$  is the number of students. HBM version of the iBKT treated per-student parameters as being drawn from Gaussian distributions and had 4 hyper-parameters: mean and standard deviation for student-level  $p$ -*init* and  $p$ -*learn*. While we did not specifically check or prove this, but intuitively, confining a parameter to the bounds of a particular distribution serves as a form of regularization and, theoretically, could improve the generalizability of the model. Although iBKT models  $4N+2M$  had parameters, the per-student and per-skill parameters, when combined using the pairing function from Equation 2a, could result in up to  $2N+2NM$  in-context parameters.  $P$ -*guess* and  $p$ -*slip* were not individualized ( $2N$ ),  $2NM$  represents all possible combinations of students and skills for  $p$ -*init* and  $p$ -*learn*.

$$f(P_k^i, P_u^i, W_0, W_k, W_u, W_{uk}) = S(W_0 + W_k l(P_k^i) + W_u l(P_u^i) + W_{uk} l(P_u^i) l(P_k^i)) \quad (3)$$

The main contribution of this paper is to not only mix per-student and per-skill parameters together but to weight each component of the mixture in an attempt to define whether either one has a larger impact on the resulting in-the-context parameter value. We have taken Equation (2a) and changed into Equation (3). Here we have the bias term ( $W_0$ ), the weights for the per-skill and per-student components ( $W_k$  and  $W_u$  respectively), and also the interaction term for the two with the weight ( $W_{uk}$ ). The  $W$  weights are drawn from Gaussian distribution. Each of them is constrained to  $[0, 1]$ , and the sum is fixed at 2. We have used the same  $W$  weights for mixing both  $p$ -*init* and  $p$ -*learn*. Thus, we have 8 additional hyperparameters and this new model, that we will refer to as iBKT-W HBM, has  $4N+2M+4$  parameters and 12 hyper-parameters. If  $\{W_0, W_k, W_u, W_{uk}\}$  weights were set to  $\{0, 1, 1, 0\}$  respectively, the model would be equivalent to the iBKT HBM model.

When exploring the per-student parameter values if the iBKT-W HBM model, we have noticed that, in spite of being drawn from

the Gaussian distribution, the actual distribution has a hint of being binomial (cf. Figure 1). It is especially visible for the distribution of the per-student values of  $p$ -*init*. In order to address this phenomenon, we have created yet another HBM model, that we will call iBKT-W-2G HBM, where the per-student  $p$ -*init* and  $p$ -*learn* parameters will be drawn from a mixture of 2 Gaussian distributions. In this new model, there are 4 means of the Gaussian distributions (2 for per-student  $p$ -*init* and 2 per-student for  $p$ -*learn*), 2 variances (1 for per-student  $p$ -*init* and 1 per-student for  $p$ -*learn*) instead of 4 as in iBKT-W HBM. The membership in one or the other mixture is modeled by a 2-parameters categorical distribution based on Dirichlet(1,1) distribution. Thus, there are, just as before,  $4N+2M+4$  parameters, while the number of hyperparameters is 16. Table 1 summarizes the information about parameters of all of the models we have considered in this work.

HBM versions of the three iBKT models are not supported by `hmm-scalable`. To build them we used BUGS language (Lunn et al., 2009) implemented as `rjags` package in R (Plummer, 2016). As opposed to `hmm-scalable`, that uses a form of exact inference, BUGS models were build using the Gibbs Sampler implemented in the `rjags` package.

To fit HBM iBKT models we used 10 chains running in parallel for the duration of 500 iterations. Unfortunately, it is not possible whether a model fit using a Gibbs sampler has converged. It is, however, possible to say whether it did not. In our experimental runs, we have confirmed there were no signs that the models failed to converge. Each model took roughly 1 hour to finish.

**Table 1. Model parameters and hyper-parameters. Number of skills –  $N$ , number of students –  $M$**

Model	Parameters	Hyper-parameters
Majority Class	0	0
Standard BKT <code>hmm-scalable</code>	$4N$	0
Standard BKT JAGS	$4N$	0
iBKT <code>hmm-scalable</code> *	$4N+2M$	0
iBKT HBM*	$4N+2M$	4
iBKT-W HBM *	$4N+2M+4$	12
iBKT-W-2G HBM *	$4N+2M+4$	16

\* for all iBKT models we only individualize  $p$ -*init* and  $p$ -*learn*.

## 4. DATA

We used the data from the KDD Cup 2010 Educational Datamining Challenge<sup>2</sup>. The data was donated by Carnegie Learning Inc., a publisher of mathematics curricula and a producer of intelligent tutoring system – Carnegie Learning’s Cognitive Tutor – for middle school, high school, and college. The KDD Cup 2010 datasets are quite large. Algebra dataset has close to 10 million student transactions, and pre-algebra dataset has a little over 20 million transactions.

Although computational capabilities of the `hmm-scalable` tool allow fitting BKT and iBKT models within minutes, R

<sup>1</sup> <https://github.com/IEDMS/standard-bkt>

<sup>2</sup> <http://pslcdatashop.web.cmu.edu/KDDCup>

implementation of the Gibbs Sampler and the BUGS language are not as scalable. Because of that, we have selected a subset of the pre-algebra dataset, namely, a sample where students worked on Linear Inequalities unit. This sample consisted of 66,307 transactions of 336 students. This sample only contained transactions labeled with the skills that the Carnegie Learning’s Cognitive Tutor tracks. There were 30 skills that the unit on linear inequalities taught.

From the rich feature set of the data we took four columns: success at first attempt at a problem step (student activity is blocked and sequenced into working on individual problem steps and BKT traditionally only looks at the first attempt; anonymous student id; concatenation of curriculum unit, section, and problem (was not necessary for our analyses, but required by `hmm-scalable`); and relevant skill(s) practiced at that particular step.

## 5. RESULTS

### 5.1 Model Fits

The results of statistical fitness of the models we have discussed are in Table 2. There we list four fitness metrics, the Deviance Information Criterion (van der Linde, 2005), root mean squared error, Accuracy and area under ROC curve ( $A'$ ). DIC is a metric based on log-likelihood. It is often used for Bayesian model selection. Accuracy is a point measure of how often the model guesses the correct response (here whether the student was correct or incorrect). RMSE goes a little further by quantifying how close the each prediction is to the correct classification of a correct or incorrect response. The area under the ROC curve is a measure of how well the model can tell the classes or responses apart. As the name suggests, it is a curve metric, without a working point, like accuracy (with which a 0.5 threshold is often used).

As we can see in Table 2, the majority class model performance is low as expected  $A'$  is at 0.50 (as it should be), accuracy is about 72%. There are usually more correct responses in the Carnegie Learning’s Cognitive Tutor data since the tutor breaks problems into steps and guides students towards the correct solution.

As we move down in Table 2, we can see that model accuracies start improving. Standard BKT models outperform Majority Class. There is a small advantage of the HBM model fit using R implementation of JAGS over the `hmm-scalable`. iBKT models (here we only individualize  $p$ -init and  $p$ -learn) are a further improvement of the fit, again, with a small advantage for the HBM version of the model. The weighted version of the iBKT (iBKT-W) is only implemented as an HBM and, again, shows an improvement overall (in terms of DIC, RMSE, and  $A'$ ).

**Table 2. Performance of the models**

Model	DIC	RMSE	Acc.	$A'$
Majority Class		0.52516	0.7242	0.5000
Standard BKT <code>hmm-scalable</code>	66230	0.40571	0.7561	0.7649
Standard BKT HBM	65347	0.40299	0.7569	0.7728
iBKT <code>hmm-scalable</code> *	64215	0.39376	0.7680	0.7990
iBKT HBM *	63644	0.39287	0.7692	0.7992
iBKT-W HBM *	63587	0.39236	0.7687	0.8005
iBKT-W-2G HBM *	63412	0.39252	0.7689	0.8005

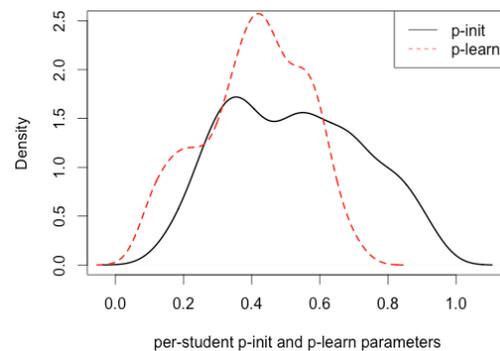
\* for all iBKT models we only individualize  $p$ -init and  $p$ -learn.

In addition to observing model fits, we have performed one round of 3-fold item-stratified cross-validation to verify whether the differences between the iBKT model fit by `hmm-scalable` and the iBKT-W model fit by JAGS become more visible. Although the fit metrics deteriorated a bit, the partial order of the models regarding the goodness of fit did not change.

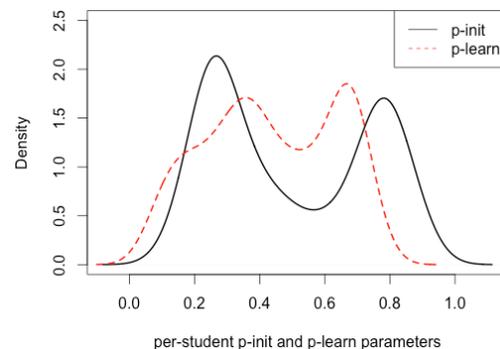
### 5.2 Per-Skill and Per-Student Parameters

When we plotted the densities of per-student  $p$ -init and  $p$ -learn parameters for the weighted iBKT, we have noticed that the distributions had a hint of bimodality, especially the distribution of per-student  $p$ -init (rf. Figure 1). Given that the HBM is drawing parameter values from a Gaussian distribution, the bi-modality is quite pronounced. To check our intuition, we have constructed a modified version of the weighted iBKT where per-student  $p$ -init and  $p$ -learn are mixtures of two Gaussians. The new model, iBKT-W-2G, did not show improvement in fit statistics, except for DIC. However, the distributions of the corresponding per-student  $p$ -init and  $p$ -learn were visibly bimodal (rf. Figure 6). The two means for the  $p$ -init parameters are 0.280 and 0.786. The two means for the  $p$ -learn parameters are 0.277 and 0.630.

The weights for pairing the per-student and per-skill parameters for both of the weighted iBKT models are given in Table 3. Both the bias weight  $W_0$  and interaction  $W_{uk}$  seem to be sufficiently small. Although there is no exact agreement between the two models, in both the weight of the per-skill parameters ( $W_k$ ) are two to three times smaller than that of per-student parameters ( $W_u$ ).



**Figure 1. Density plots for per-student  $p$ -init and  $p$ -learn parameters of iBKT-W HBM model.**



**Figure 2. Density plots for per-student  $p$ -init and  $p$ -learn parameters of iBKT-W-2G HBM model.**

**Table 3. Skill-student weights in iBKT-W models**

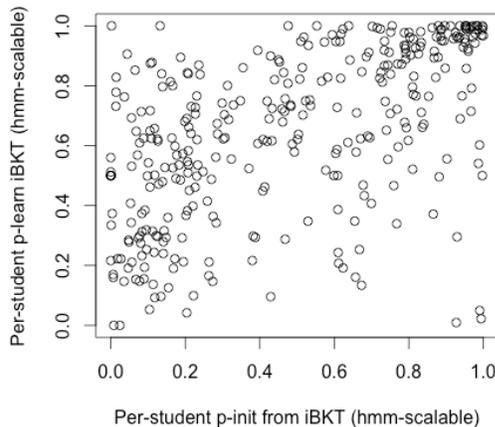
Model	$W_0$	$W_k$	$W_u$	$W_{uk}$

iBKT-W HB	0.012	0.565	1.420	0.004
iBKT-W-2G HBM	0.019	0.700	1.274	0.007

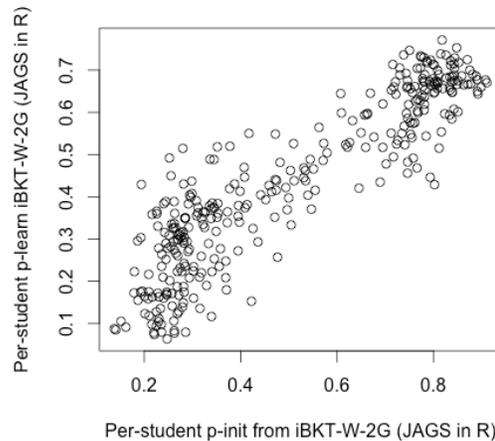
### 5.3 Extra Look At Per-Skill and Per-Student Parameters

In an attempt to investigate the differences between iBKT model fit using `hmm-scalable` and the iBKT-W-2G fit using JAGS, we have plotted the per-student  $p-init$  and  $p-learn$  parameters for both. The respective plots are in Figure 3 and Figure 4. As we can see in Figure 3, where per-student parameters of iBKT `hmm-scalable` model are plotted, correlation of  $p-init$  and  $p-learn$  is mid-range and is equal 0.55. Notably, a tangible portion of students, as estimated by the model, have low  $p-init$  and high  $p-learn$  parameters. If we interpret  $p-init$  as student's overall prior preparation and  $p-learn$  as student's overall rate of learning, these would be the students that came in with the low level of knowledge and quickly caught up. Using the same logic, there are also a few students that came in with high prior knowledge but suffered from low learning rate.

The plot of per-student  $p-init$  and  $p-learn$  parameters of iBKT-W-2G HBM model is entirely different (rf. Figure 8). The correlation is very high – 0.90. Although the student points are lined up almost linearly, it is possible to discern two clusters (lower left, and upper right) that roughly correspond to two mixed Gaussians represented by a categorical node in the model. Here, there are effectively no students in the upper left or bottom right corners of the graph. Namely, those arriving with lower preparation, but the high rate of learning, or, vice-versa, high preparation, but the lower rate of learning. The former is unfortunate since the unprepared students that can quickly close the gap are, arguably, the most desired ones since they make the application that assisted them (e.g., Carnegie Learning's Cognitive Tutor) shine.



**Figure 3. Scatter plot of per-student  $p-init$  (x-axis) and  $p-learn$  (y-axis) from iBKT model fit by `hmm-scalable`. The correlation between the two is 0.55 (significant at 0.001 level).**



**Figure 4. Scatter plot of per-student  $p-init$  (x-axis) and  $p-learn$  (y-axis) from iBKT-W-2G model fit by JAGS in R. The correlation between the two is 0.90 (significant at 0.001 level).**

## 6. DISCUSSION

### 6.1 Small Differences in Statistical Fits

Arguably the most pressing question about comparing the `hmm-scalable`-fit iBKT model and the HBM models is why the differences in statistical accuracy are so small. Given that some of the changes in per-student parameters are quite large (rf. Figures 3 and 4), we are to expect more pronounced differentiation, especially since the fitting method and parameterization changed.

We would like to refer to an earlier work where we examined alternative parameterizations of a logistic regression model of student math learning (Yudelson et al., 2011). As we have found there, despite virtually no difference in statistical fit, the parameter values and especially their interpretability improved. We did not estimate the interpretability of the parameter values of the HBM models, however, the relative distribution of the iBKT-W-2G HBM per-student parameters is, arguably, more realistic than that of the iBKT `hmm-scalable`.

Besides, as we were able to show in (Yudelson & Ritter, 2015), the absence of a *tangible* difference in statistical fit between two models may, none the less, correspond to considerable variance in assigned practice when the models compared are deployed in the actual system and used for knowledge tracking and problem selection.

### 6.2 What Do The Gaussians Mixtures Represent?

We have followed the trace of the possible bi-modal distributions of per-student  $p-init$  and  $p-learn$  parameters in the iBKT-W and constructed iBKT-W-2G model where per-student parameters are represented as mixtures of 2 Gaussian distributions with the same standard deviation.

To reverse-engineer the fuzzy mixture variable that *clusters* students we have attempted to correlate it with a set of student performance metrics. These included: overall number of problems solved, time spent, hints requested (both on the first attempt at a step and overall), errors committed (both on the first attempt at a step and overall), percent correct (both on the first attempt at a step and overall), time spent per problem, errors committed and hints requested per problem. None of them correlated with the fuzzy mixture variable reliably. It is likely that the resulting

clustering represents some latent student factor, we just could not interpret it.

### 6.3 Weighting Per-Skill and Per-Student Parameters

We have tried more models than the two HBM iBKT-W's we reported. The models included those individualizing *p-init* and *p-learn* separately or together, with weighting or without, mixing 1, 2, or 3 Gaussians (18 variants overall) – in all cases per-student parameter component weight was two-to-three times larger than that of per-skill components. One explanation for that could be possible over-fitting. There are 336 students and 30 skills. Even though the model is hierarchical and both per-skill and per-student parameter values are regularized, they are an order of magnitude more per-student values. To confirm or disconfirm the over-fitting hypothesis we would have to perform multiple sample-and-fit rounds where the number of students is equal to the number of skills.

## 7. ACKNOWLEDGMENTS

The author would like to give special thanks to Mr. Christopher MacLellan for introducing him to the BUGS language, Dr. Ilya Goldin for sharing his draft of a single-skill BKT implementation in BUGS, and Dr. Kenneth R. Koedinger for useful feedback while this work took shape.

## 8. REFERENCES

- [1] Cen, H., Koedinger, K.R., Junker, B.: Comparing Two IRT Models for Conjunctive Skills. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 796–798. Springer, Heidelberg (2008)
- [2] 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)
- [3] Fawcett, T. (2006) An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861–874.
- [4] Huang, Y., Gonzalez-Brenes, J. P., and Brusilovsky, P. (2015) The FAST toolkit for Unsupervised Learning of HMMs with Features. In: The Machine Learning Open Source Software Workshop at the 32nd International Conference on Machine Learning (ICML-MLOSS 2015).
- [5] Khajah, M., Wing, R. M., Lindsey, R. V., & Mozer, M. C. (2014) Incorporating latent factors into knowledge tracing to predict individual differences in learning. In J. Stamper, Z. Pardos, M. Mavrikis, & B. M. McLaren (Eds), Proceedings of the 7th International Conference on Educational Data Mining (pp. 99-106).
- [6] Lee, J.I., Brunskill, E.: The Impact on Individualizing Student Models on Necessary Practice Opportunities. In: Yacef, K., Zaïane, O.R., Hershkovitz, A., Yudelso, M., Stamper, J.C. (eds.) Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012), pp. 118–125 (2012)
- [7] Levinson, S.E., Rabiner, L.R., Sondhi, M.M.: An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. *Bell System Technical Journal* 62(4), 1035–1074 (1983)
- [8] van der Linde, A. (2005). DIC in variable selection. *Statistica Neerlandica*, 59: 45-56.
- [9] van der Linden, W.J., Hambleton, R.K.: Handbook of Modern Item Response Theory. Springer, New York (1997)
- [10] Lunn D, Spiegelhalter D, Thomas A, Best N. (2009) The BUGS project: Evolution, critique, and future directions. *Statistics in Medicine*, 28:3049-67.
- [11] Pardos, Z.A., Heffernan, N.T.: Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010)
- [12] Pardos, Z. & Heffernan, N. (2011) KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Konstant et al. (eds.) Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP). (pp. 243-254), Girona, Spain. Springer.
- [13] Plummer, M. (2016). rjags: Bayesian Graphical Models using MCMC. R package version 4-5. <https://CRAN.R-project.org/package=rjags>
- [14] Yudelso, M., Koedinger, K., Gordon, G. (2013) Individualized Bayesian Knowledge Tracing Models. In: Lane, H.C., and Yacef, K., Mostow, J., Pavlik, P.I. (eds.) Proceedings of 16th International Conference on Artificial Intelligence in Education (AIED 2013), Memphis, TN. LNCS vol. 7926, (pp. 171–180).
- [15] Yudelso, M., Pavlik, P.I., Koedinger, K.R. (2011) User Modeling – a Notoriously Black Art. In J.A. Konstan, R. Conejo, J.L. Marzo, and N. Oliver (Eds.) Proceedings of the 19th International Conference on User Modeling Adaptation and Personalization (UMAP 2011), Girona, Spain, (pp. 317-328).
- [16] Yudelso, M. & Ritter, S. (2015) Small Improvement for the Model Accuracy – Big Difference for the Students. In: Industry Track Proceedings of 17th International Conference on Artificial Intelligence in Education (AIED 2015), Madrid, Spain.

# Deep Learning + Student Modeling + Clustering: a Recipe for Effective Automatic Short Answer Grading

Yuan Zhang, Rajat Shah, and Min Chi  
North Carolina State University  
{yzhang93, rshah6, mchi}@ncsu.edu

## ABSTRACT

In this work we tackled the task of Automatic Short Answer Grading (ASAG). While conventional ASAG research makes prediction mainly based on student answers referred as Answer-based, we leveraged the information about questions and student models into consideration. More specifically, we explore the Answer-based, Question, and Student models individually, and subsequently in various combined and composite models through feature engineering. Additionally, we extend the exploration of machine learning methods by utilizing Deep Belief Networks (DBN) together with other five classic classifiers. Our experimental results show that our proposed feature engineering models significantly out-performed the conventional Answer-based model and among the six machine learning classifiers, DBN is the best followed by SVM, and Naive Bayes is the worst.

## 1. INTRODUCTION

Developing effective Computer-based assessment has been increasingly gaining its importance over years and it is widely believed that open-ended problems are more effective to access student knowledge than multiple choices. The former require students to generate free text and communicate their responses and thus student answers are relatively immune to test-taking shortcuts like eliminating improbable answers. On the other hand, grading student's free text answers is often time-consuming and challenging. Therefore, much research has focused on how to automatically grade student free text answers. Generally speaking, research to date has concentrated on two sub-tasks: grading student essays, which includes checking the style, grammar, and coherence of an essay [13], and grading student short answers [16, 18, 19], which is the focus of this work. More formally, [7] defined short answers as those: 1) in the form of natural language; 2) requiring students to recall external knowledge that is not provided by the question; 3) of which the length ranges between one phrase to one paragraph; 4) focusing on the correctness of the content rather than the style; and 5) and are closed, which means that the answers have to match the specific facts corresponding to

questions. The goal of this work is to explore effectiveness of various Machine Learning (ML) approaches on Automatic Short Answers Grading (ASAG). An ASAG system is one that automatically classify student answers into, correct or incorrect, based on the referred correct one(s).

Much of the prior research on ASAG is answer-based which involves applying various Natural Language Processing (NLP) techniques to extract a wide variety of text-based features directly from student answers. These features include various measurements of text similarities between student answers and the referred correct ones. Often time, the shorter the student answers, the harder to classify them into correct or incorrect because the limited text provides fewer lexical features. Many classic NLP approaches such as bag-of-words or keyword matching often fail to work. For example, Table 1 shows an example of student short answer extracted from our training corpus. In this example, using text similarities alone would fail to recognize that the student's answer is correct because it looks quite different from the referred correct answer.

**Table 1: An Example of Student Short Answer.**

---

**Tutor:** Why are there no potential energies involved in this problem?  
**Student:** There is no second object that is massive and can have gravitational energy. (**Correct**)  
**Correct Answer:** Because the rock is the only object in the system, there are no potential energies involved.

---

On the other hand, information about question and student knowledge can be handily used to improve the effectiveness of existing answer-based ASAG model. For example, in the example above if we know that the question is about "potential energy" and the student's knowledge on "potential energy" is very high, it is more likely that the student will answer the question correctly even though his/her answer looks quite different from the correct one. Thus in this paper we will investigate whether the effectiveness of ASAG can be further improved if we leverage question model, student model, or both into the answer-based model. To the best of our knowledge, this is the first comprehensive study exploring the effectiveness of feature space from all three models on the task of ASAG. For simplicity reasons, in the following we will refer the three models as Answer(Ans), Question(Ques), and Student (Stu) models respectively.

Prior research on ASAG has explored several classic ML classifiers such as Naïve Bayes and Decision Tree. In re-

cent years, Deep Belief Network (DBN) [5] has been successfully implemented and applied in a wide variety of real-world tasks [15,17]. DBN enables the automatic extraction of representative features via an unsupervised pre-training and it can learn the latent complex relationship among features. Given the potential complex connections among the features from Ans, Ques and Stu models, we investigated on leveraging DBN to exploit the more discriminative feature space to facilitate automatic grading. As far as we know, this is the first study to apply DBN to the task of ASAG.

To summarize, we investigated on improving ASAG by utilizing DBN together with five classic ML methods and by extending existing answer-based approaches to leverage a wide range of state features which are either based on or generated from Ans, Ques, and Stu models.

## 2. RELATED WORK

Popular Natural Language Tutors like AutoTutor [11] and BEETLE II [12] have extensively studied how to automatically understand student Natural Language inputs so that the system can respond to student's responses adaptively. Pulman and Sukkarieh used manually crafted patterns in the part-of-speech tagged answers for pattern matching with the correct answer [19]. Their approach is question-specific in that they applied Naïve Bayes and Decision Tree to automatically generate patterns for each question using a set of marked answers. Results showed their approach can achieve an average accuracy of 84%.

Mohler and Mihalcea developed an unsupervised approach using Knowledge-based and Corpus-based text-to-text similarity measures [18]. They used Latent Semantic Analysis coupled with domain specific corpus built from Wikipedia. Their resulted measures outperformed other similarity measures in that the former obtained Pearson correlation  $r = 0.463$  between the computer assigned grades and average of human assigned grades.

Recently, Microsoft's Power Grading [2] took a semi-automated approach based on the observation that similar answers get similar grades. Thus, instead of directly grading student answers, Power Grading builds a hierarchy of short-answer clusters and lets human grader either grade the entire cluster with same score or manipulate the clusters as needed. Inspired by their work and promising results, we borrowed some of the features such as length and tf-idf from previous research into this work.

Our approach differed from previous research in that: 1) unlike relying solely on answer-based methods, we explored features from Ans, Ques and Stu models individually and combined; 2) our models are trained across all questions, that is, it is question-general instead of building question-specific classifiers in previous research; 3) previous approaches mainly involved two or three ML methods while we used a total of six including the state-of-the-art DBN together with five other traditional ML approaches.

## 3. METHODS

In this section, we will briefly describe the features involved in this study and the ML classifiers applied. For the latter, we will focus on DBN.

## 3.1 State Features

To investigate the impact of state features on the task of ASAG, we compare the effectiveness of various features from Ans, Ques and Stu models *individually* and *combined*. We also *composite* new features generated within or across different models.

### 3.1.1 Answer (Ans) Model

In [7], Burrows et al. identified two categories of answer-based approaches: corpus-based approaches are based on mapping the concepts in student answers to those in the reference correct answers [16], while alignment-based approaches are based on clustering student answers by some quality similarity estimates among student answer representations regardless of the correct answers. Our Ans model includes both corpus-based features and alignment-based ones.

Based on [2] and [18], we defined five Ans-based features by measuring the text similarity between student answer and the correct answer(s). The latter consist of the referred correct answer and the correct answers generated by students. More specifically, we have:

- *length difference*: the length difference (in words) between the student and the correct answers.
- *max-matched idf*: the maximum value of idf of matched words in a student answer. The idf of each word is calculated based on the Bag-Of-Word(BOW) generated from the word-answer matrix. This is a good measure to reflect whether prominent keywords in correct answers show up in the student answer.
- *cosine similarity* is calculated using tf-idf vectors of the student answer and the referred correct answers.
- *weighted text similarity*: Wu & Palmer similarity is a knowledge-based measure for text similarity [18], which is based on word similarities. More specifically, we formalize the text similarity between the student answers  $s$  and the correct answers  $c$  as sentences. We construct a domain specific word list  $d$  for the specific domain by assigning higher weight to domain specific words. Then the text similarity is calculated by weighting the similarities of general words  $sim_w(s, c)$  and those of domain specific words  $sim_d(s, c)$ .
- *Latent Semantic Analysis* (LSA, Landauer and Dumais, 1997): is a computational method which aims to represent a corpora of natural text using the latent subspace. This subspace reflects the weight of each word in each answer so that similar correct answers share similar weight vector of words.

### 3.1.2 Question (Ques) Model

In domains such as math and science, it is commonly assumed that the relevant knowledge is structured as a set of independent but co-occurring Knowledge Components (KCs). A KC is "a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet" [21].

In many Intelligent Tutoring Systems (ITs) such as Cordillera, completion of a tutor question requires students to apply multiple KCs. By including KCs in our model, we wish to guide the learning process in distinguishing between different

types of questions. Moreover, utilizing KCs is helpful for exploiting the homogeneity among questions. The central idea of Ques model is to build a *Q-matrix* to represent the relationship between individual questions and KCs. Q-matrices are typically encoded as a binary 2-dimensional matrix with columns representing KCs and rows representing questions. Previous researchers have focused on the task of generating or tuning Q-matrices based upon a dataset [1, 20]. For the present work we employ a static Q-matrix manually generated from domain experts.

Additionally, for each question we also include a feature named *questionDifficulty*. It has consistently been selected as one of the important features in our previous work on exploring various state features for modeling student learning [9]. *questionDifficulty* is defined as difficulty level of a question and its value is roughly estimated from the training corpus based on the percentage of answers that were correct on the question in the training dataset.

### 3.1.3 Student (Stu) Model

Student modeling is an important component for any interactive e-learning environment so that the system can adapt its behaviors based on student needs and knowledge [3]. There are many techniques for generating student models and among them, Bayesian Knowledge Tracing (BKT) [10] is the most widely used. Fundamentally, the BKT model can be seen as a Hidden Markov Model with two hidden states: learned and unlearned. They are defined based on whether a student has mastered the target knowledge or not. BKT keeps a running assessment of the probability that a student is in the learned state based on the student's past history of performance (e.g. *correct*, *incorrect*). BKT assumes that student learning process is a Markov Chain in that at each time  $t+1$ , the probability of a student has learned the knowledge  $p^{t+1}$  is only dependent on his learning state at time  $t$ .

Our Stu model used the outputs of the BKT, that is the probability that a student is in the learned state after answering  $n$  questions, denoted as  $p(S^n = \textit{learned})$  as state features. Moreover, our Stu model is KC-specific in that for each of domain KCs, our model will include one probability of being in the learned state on the corresponding KC in the Stu model. Our goal is to use these KC specific probabilities to predict whether the student will answer the next question correctly. Additionally, we also included student KC-specific pretest scores which measures student initial incoming competence.

Therefore, our final Stu model includes a combination of KC-specific learning probabilities calculated from BKT and the student KC-specific pretest scores.

### 3.1.4 Composite Feature Space

In this part we will explore state features representing the underlying connections between the Ques and the Stu models. As described above, KCs are involved in both Ques and Stu models and thus we hypothesized that a student's performance on a problem should depend on the KCs involved in the problem and the student's performance on corresponding KCs. Hence, we conduct the Cartesian product (CP) using the Ques and Stu models. Additionally, we applied the clustering on the Stu model based on their learn-

ing states and pretest scores. Compared with the original features in the Stu model, using student clustering can be seen as more compact representation. Here we used Gaussian Mixture Model, which is a type of soft-clustering methods. Similarly, we hypothesized that the students with similar patterns in Stu clusters may have similar performance on certain types of questions and thus we also conduct the Cartesian product using the student clustering features and Ques vector.

## 3.2 Six Classifiers

Prior research on ASAG successfully explored several classic ML methods which included: Naive Bayes (NB), Logistic Regression (LR), Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM). In recent years deep learning model has been widely used in computer vision and image processing. In this paper, we will compare Deep Belief Networks (DBN) [5] against those five classic ML methods. Given the space constraints, we only briefly describe DBN in the following paragraphs.

DBN is one of the most widely implemented deep learning models. Through the unsupervised pre-training in the first stage, DBN is able to extract the latent features that are more representative than the original input features. Given the input features, DBN first utilizes the stacked Restricted Boltzmann Machine (RBM) layers to automatically extract the high-level features. After the feature extraction in pre-training phase, the weights in these layers are then folded into neural networks for supervised training. Since the capacity of feature extraction mainly lies in the pre-training phase, we now present the mechanism of RBM.

RBM is a restricted version of Markov Random Field. It consists of two layers of variables, visible units  $V$  and hidden units  $H$ . From the perspective of feature extraction,  $V$  stands for the original feature inputs and  $H$  denotes the extracted feature representation. The joint distribution of  $V$  and  $H$  is defined by an energy-based probabilistic model, as follows:

$$P(V, H) = \frac{\exp(-E(V, H))}{Z}, \quad (1)$$

$$Z = \sum_{V, H} \exp(-E(V, H))$$

where the energy function  $E(V, H)$  is defined to be:

$$E(V, H) = -V^T W H - B^T V - C^T H. \quad (2)$$

In the above equation,  $W$  denotes the weights between  $V$  and  $H$ . Specifically,  $W_{i,j}$  represents the weight between  $V_i$  and  $H_j$ , and  $B$ ,  $C$  denote the biases for visible units and hidden units, respectively. The denominator  $Z$  serves as the normalizer for the probability distribution.

Given that each unit of  $V$  or  $H$  is independent with other units in the same layer, the conditional distribution is fully factorial and can be easily derived. Due to the intractability of gradient computation brought by the factor  $Z$ , the training of RBM (i.e., pre-training phase) follows the Contrastive Divergence algorithm [14], which executes  $K$  steps of alternating Gibbs sampling to approximate the gradient. The details can be found in [4].

## 4. DATA DESCRIPTION

Our training corpus was collected from Cordillera [8, 21], a Natural Language ITS that teaches students introductory college physics. The domain consists of a subset of the physics work-energy domain, which is characterized by eight primary KCs including Kinetic Energy, Gravitational Potential Energy, Spring Potential Energy, and so on. In Cordillera, students interact with tutor by means of natural language entries, and currently the Natural Language understanding module in Cordillera is using human interpreters referred as the language understanding wizard [6]. The *only* task performed by the human wizards is to match student answers to the closest response from a list of potential correct or incorrect responses.

Our training corpus involves 158 students. The data collection consists of the following stages: 1) background survey; 2) studying textbook and prerequisite materials, 3) taking a pretest; 3) training on Cordillera, 4) and taking a post test. In total there are 482 different questions involved in the training corpus and it takes students roughly 4-9 hours to complete the training. Our training corpus includes sequences of tutorial dialogue interactions between students and Cordillera, one sequence per student, and the average number of Cordillera-student interactions is more than 280 per student. For each interaction in a sequence, it consists of a tutor question, a student answer to the question, and two output labels *correct* or *incorrect* based on human wizards inputs. Thus, *these human manually generated binary labels function as ground truth in our training corpus.*

Based on the definition in [7], our training corpus included **16228** short answers selected from a total of 27868 dialogues. The average length of student answers in our corpus is **7.6** words. **61.66%** of training corpus is labeled as “correct” while the rest are labeled as “incorrect”. A series of standard natural language pre-processings including stop word removal, tokenization, punctuation removal and word correction, have been conducted on our training corpus. Additionally, we also conducted domain-specific pre-processing, which includes expanding acronyms to their full forms and removing quantitative questions with equations.

## 5. EXPERIMENTS

To evaluate the effectiveness of various features from Ans, Ques, and Stu models individually, combined, and/or composite features generated from these three models, we use two ubiquitously implemented classifiers - LR and SVM in Experiment 1. Then in Experiment 2, we will compare DBN against five classic ML classifiers on the best feature model produced in Experiment 1.

### 5.1 Experiment 1: Exploring Feature Space

For Ans model, we use the five Ans features described in 3.1.1. For Ques model, we include 9 Ques features (one is *questionDifficulty* and the other eight are KC-based Q-matrix features, one feature per KC) and for Stu model, we include 16 Stu features (8 KC-based learning parameters and 8 KC-based pretest scores). Generally speaking, our Experiment 1 can be divided into three stages:

In stage 1, we compare the Ans, Ques, and Stu model individually. Our goal is to investigate whether either Ques or

Stu model will be more effective than Ans model for ASAG. In stage 2, we will compare different ways of combining the three basic models. Our results from stage 1 show that Ans-based model alone performs better than either Ques or Stu model (depicted in Section 6.1.1) and thus we mainly explore whether to include the Ques and/or Stu models to the Ans-based model in stage 2. Finally, in stage 3, we will compare different ways of generating new features from the three models (depicted in Section 3.1.4) together with the best model learned from stage 2, which is AQS. Table 2 summarize the types of feature models we explored in each stage.

**Table 2: Feature Representations.**

Feature	Abbr.	Construction
<b>Stage 1</b> Basic	<i>A(ns)</i>	Ans Model
	<i>S(tu)</i>	Stu Model
	<i>Q(ues)</i>	Ques Model
<b>Stage 2</b> Combined	<i>AS</i>	A + S
	<i>AQ</i>	A + Q
	<i>AQS</i>	A + Q + S
<b>Stage 3</b> Composite	<i>CF1</i>	<i>AQS</i> + SC (Student Clustering)
	<i>CF2</i>	<i>AQS</i> + SC + CP(Q,S)
	<i>CF3</i>	<i>AQS</i> + SC + CP(Q,SC)

\* CP denotes Cartesian Product.

To quantitatively evaluate the effectiveness of different feature models, we train LR and SVM with 10-fold cross-validation (CV). LR is widely adopted as the prediction model in industry for its efficiency and robustness. On the other hand, SVM is one of the most popular classifier due to its effectiveness and the capability to incorporate different kernels. Here we adopt RBF kernel for our SVM models.

### 5.2 Experiment 2: Six Classifiers

In Experiment 2, we evaluate six classifiers with 10-fold cross-validation using the best feature model from Experiment 1, CF3. The six classifiers are NB, LR, DT, ANN, SVM and DBN. As for the DBN, we build three hidden layers, with 74, 34, 10 hidden units respectively and the learning rate is set to be 0.01.

Among the six classifiers, NB assumes the state features are conditionally independent given the output label while the other models do not have such strong assumption and thus are able to combine multiple features to make predictions. Since there exist latent connections among our extracted features, we expect that NB would perform poorly compared to other models. While all five remaining classifiers can make use of combined features to explore latent connections among features, their approaches are different: LR only linearly combines features; DT synthesizes the features at different branches to make predictions; the hidden layers in ANN and the kernel function of SVM can effectively achieve the non-linear feature mapping; while SVM and ANN utilize the relatively fixed pattern for feature combination, DBN enables the extraction of more representative features via a separate unsupervised pre-training procedure. Although the best model CF3 already contains composite features, we expect the DBN can further leverage the latent connections among features that cannot be manually captured in CF3.

## 6. RESULTS

Five widely used measures, Accuracy, Area Under the Curve (AUC), Precision, Recall and F-measure are used to evaluate how well various classifiers performed. For precision, recall and F-measure, we treat incorrect answers as the positive class because it is more important for the system to know when the student answer is incorrect.

### 6.1 Experiment 1: Exploring Feature Space

In the following, we will report our results from each stage listed in Table 2. Given that  $A(ns)$  (Ans model) is the fundamental model studied in previous research, it will be our baseline model for comparisons across three stages.

#### 6.1.1 Stage 1: Three Basic Models

We first compare Ans, Ques and Stu model separately and Table 3 shows the 10-fold cross-validation results. In Table 3, the best performance of corresponding classifier with respect to each measure is in bold and the best value of each measure is marked \*.

Table 3: Performance of Basic Models.

Classifier	Evaluation	A	S	Q
LR	Accuracy	<b>0.646</b>	0.616	0.633
	AUC	<b>0.589</b>	0.499	0.548
	Precision	<b>0.564</b>	0.025	0.425
	Recall	0.342	0.001	<b>0.548</b>
	F-measure	0.426	0.002	<b>0.478</b>
SVM	Accuracy	<b>0.728*</b>	0.540	0.636
	AUC	<b>0.654*</b>	0.546	0.567
	Precision	<b>0.830*</b>	0.422	0.551
	Recall	0.331	<b>0.572*</b>	0.271
	F-measure	0.474	<b>0.486*</b>	0.364

\* The majority class is 0.617.

\* '\*' is for the highest value of each measure across all models.

Table 3 shows that all three models beat the majority class baseline (0.617) except for the case of applying SVM on Stu model. As expected, when using either LR or SVM, Ans model outperforms Stu and Ques models on Accuracy, AUC and precision. For the other two measures, Stu model provides the best Recall and F-measure when using SVM and Ques model yields the best Recall and F-measure when using LR. Moreover, when comparing LR and SVM, Table 3 shows that SVM classifier seems to be more effective than LR in that the highest values of five measures are all generated by SVM, marked \*. More specifically, for Ans model, SVM outperforms LR on all the measures except Recall; for Stu model, SVM outperforms LR on every measure except for Accuracy; finally, for Ques model, SVM outperforms LR on three out of five measures, the exceptions are recall and F-measure.

Overall, while the Ans model generate the best Accuracy, AUC and Precision, the best Recall and F-measure are generated using either the Ques model for LR or the Stu model for SVM. Therefore, we expect combining the Ques and Stu model with Ans model would result in more effective models.

#### 6.1.2 Stage 2: Three Combined Models

To test the effectiveness of combining multiple features, we show the 10-fold CV performance of A, AQ, AS and AQS by applying LR and SVM respectively in Table 4.

Table 4: Performance of Combined Features.

Classifier	Evaluation	A	AQ	AS	AQS
LR	Accuracy	0.646	0.719	0.712	<b>0.768</b>
	AUC	0.589	0.696	0.690	<b>0.753</b>
	Precision	0.564	0.656	0.663	<b>0.737</b>
	Recall	0.342	0.591	0.576	<b>0.671*</b>
SVM	F-measure	0.426	0.621	0.616	<b>0.703</b>
	Accuracy	0.728	0.784	0.777	<b>0.822*</b>
	AUC	0.654	0.731	0.733	<b>0.781*</b>
	Precision	0.830	0.880	<b>0.881*</b>	0.876
SVM	Recall	0.331	0.505	0.513	<b>0.615</b>
	F-measure	0.474	0.641	0.649	<b>0.723*</b>

Table 5: Performance of Composite Features.

Classifier	Evaluation	A	CF1	CF2	CF3
LR	Accuracy	0.646	0.786	0.802	<b>0.810</b>
	AUC	0.589	0.769	0.784	<b>0.794</b>
	Precision	0.564	0.736	0.764	<b>0.774</b>
	Recall	0.342	0.692	0.707	<b>0.720</b>
SVM	F-measure	0.426	0.713	0.734	<b>0.746</b>
	Accuracy	0.728	0.835	0.830	<b>0.848*</b>
	AUC	0.654	0.799	0.824	<b>0.850*</b>
	Precision	0.830	<b>0.887*</b>	0.778	0.769
SVM	Recall	0.331	0.649	0.795	<b>0.859*</b>
	F-measure	0.473	0.750	0.787	<b>0.811*</b>

\* CF1 AQS + Student Clustering (SC).

\* CF2 AQS + SC + Cartesian product(Ques, Stu).

\* CF3 AQS + SC + Cartesian product(Ques, SC).

It is observed that by adding either Ques or Stu model into Ans model, the effectiveness of resulted models is greatly improved on each of five measures. For example, the Accuracy increases from 0.646 for Ans model to 0.719 for AQ model, and 0.712 for AS model under LR. We can observe the same pattern when SVM is applied. For both LR and SVM classifier, it seems that AQ and AS have comparable performance.

AQS, the combination of all three models, outperforms either AQ or AS for both LR and SVM on all five measures except on Precision by SVM where AS has a slightly higher value (0.881) than AQS (0.876). Therefore, it suggests that Stu and Ques model indeed contribute different information to ASAG task. Similarly, across three models, Table 4 shows that the SVM classifier seems to be more effective than LR in that the best of each of the five measures (those marked \*) are generated by SVM except for Recall where the best value 0.671 is generated by LR on AQS model.

#### 6.1.3 Stage 3: Three Composite Models

Given that AQS performs as the best model in Stage 2, we explore whether the effectiveness of classifiers can be further improved by adding composite features. Table 5 shows the performance of CF1, CF2 and CF3.

Tables 4 and 5 show that CF1 is more effective than AQS on every measure when using SVM and on four out of five measures except on Precision using LR. It suggests that the using student clustering can indeed further improve the performance of either LR and SVM.

The improvement from CF1 to CF2 and CF3 mainly stems from the power of Cartesian product. Furthermore, the difference between CF2 and CF3 lies in the different choices of features used for Cartesian product. The result shows that there exists stronger association between the latent student clusters and Ques model than that between Stu model and Ques model. Overall, SVM outperforms LR throughout CF1

to CF3 in that the best of five measures (those marked \*) are all generated by SVM in Table 5.

To summarize, the performance of SVM dominates LR when using individual feature models, combined models, and composite models. With only one exception, the best of each of the five measures (those marked \*) are all generated by SVM across all three stages. Finally across the nine models, the best model for both LR and SVM is CF3 in that CF3 is more effective than the other eight models on every measures using LR and on four out five measures except on Precision using SVM. Therefore, CF3 is selected for Experiment 2.

## 6.2 Experiment 2: Six Classifiers

Table 6 shows the performance of the six ML classifiers on CF3:  $AQS + SC + CP(Q,SC)$  using 10-fold cross-validation. From the results, we draw the first conclusion that NB falls behind other classifiers with a large margin of 18% except on Recall. As expected, LR, DT, ANN, SVM and DBN outperform NB in all the evaluations due to the capacity of combining features and NB's strong independent assumption. Table 6 shows that DBN yields the highest Accuracy, AUC, Precision and F-measure while SVM reaches the best recall value of 0.859 closely followed by DBN. For AUC and F-measure, we have the values in the increasing order for NB, LR, DT, ANN, SVM, and DBN. Overall, our results suggest that DBN performs the best among the six classifiers followed by SVM and NB performs the worst.

Table 6: Comparing the Six Classifiers

Evaluation	NB	LR	DT	ANN	SVM	DBN
Accuracy	0.631	0.810	0.825	0.837	0.848	<b>0.850*</b>
AUC	0.667	0.794	0.813	0.827	0.850	<b>0.890*</b>
Precision	0.511	0.774	0.775	0.791	0.769	<b>0.830*</b>
Recall	0.823	0.720	0.765	0.784	<b>0.859*</b>	0.838
F-measure	0.631	0.746	0.770	0.787	0.811	<b>0.834*</b>

## 7. CONCLUSION

In this paper we tackled the task of ASAG through feature engineering and exploration of better ML approaches such as DBN. For feature engineering, we utilized two other models: Ques and Stu models and explored various combined and composite feature representation. Our results showed that by utilizing the composite features, we obtain an AUC improvement of around 35% and 30% and F-measure improvement of around 75% and 72% on LR and SVM respectively as compared with using Answer-based features only. The comparisons among different classification models shows that DBN outperforms all other methods on Accuracy, AUC, Precision and F-measure. On Recall, DBN performs slightly worse than SVM. Furthermore, the experiment has led to some interesting observations: (1) The clustering of student, as a more compact representation, leads to more discriminative features when combined with question features using Cartesian product. (2) While SVM results in better Accuracy, the composite feature representation brings less improvement on SVM than LR probably because we used RBF kernel in our SVM models which allows the classifier to operate in an infinite-dimension of feature space.

## 8. ACKNOWLEDGMENTS

This research was supported by the NSF Grant 1432156 "Educational Data Mining for Individualized Instruction in STEM Learning Environments".

## 9. REFERENCES

- [1] T. Barnes. The q-matrix method: Mining student response data for knowledge. 2005.
- [2] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 2013.
- [3] J. E. Beck and B. P. Woolf. Using a learning agent with a student model.
- [4] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2009.
- [5] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 2007.
- [6] N. O. Bernsen and L. Dybkjaer. *Designing Interactive Speech Systems: From First Ideas to User Testing*. Springer-Verlag New York, Inc., 1997.
- [7] S. Burrows, I. Gurevych, and B. Stein. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 2015.
- [8] R. Carolyn. Tools for authoring a dialogue agent that participates in learning studies. *Artificial Intelligence in Education: Building Technology Rich Learning Contexts that Work*, 158:43, 2007.
- [9] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 2011.
- [10] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 1994.
- [11] S. K. D'Mello, B. Lehman, and A. Graesser. A motivationally supportive affect-sensitive autotutor. In *New perspectives on affect and learning technologies*. Springer, 2011.
- [12] M. O. Dzikovska, A. Isard, P. Bell, J. D. Moore, N. Steinhauser, and G. Campbell. Beetle ii: an adaptable tutorial dialogue system. In *Proceedings of the SIGDIAL 2011 Conference*, pages 338–340. Association for Computational Linguistics, 2011.
- [13] D. Higgins, J. Burstein, D. Marcu, and C. Gentile. Evaluating multiple aspects of coherence in student essays. In *HLL-NAACL*, 2004.
- [14] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 2002.
- [15] G. B. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR*, 2012.
- [16] C. Leacock and M. Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 2003.
- [17] A.-r. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. In *NIPS*, 2009.
- [18] M. Mohler and R. Mihalcea. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th EACL*, 2009.
- [19] S. G. Pulman and J. Z. Sukkarieh. Automatic short answer marking. In *Proceedings of the second workshop on Building Educational Applications Using NLP*.
- [20] K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 1983.
- [21] K. VanLehn, P. W. Jordan, and D. J. Litman. Developing pedagogically effective tutorial dialogue tactics: experiments and a testbed. In *SLaTE*. Citeseer, 2007.