SHORT PAPERS

Spectral Bayesian Knowledge Tracing

Mohammad Falakmasir
University of Pittsburgh
210 South Bouquet Street,
Pittsburgh, PA 15213
(412) 624-5755
falakmasir@pitt.edu

Michael Yudelson
Carnegie Learning, Inc.
437 Grant St.
Pittsburgh, PA 15219
(412) 690-2442
myudelson@
carnegielearning.com

Steve Ritter
Carnegie Learning, Inc.
437 Grant St.
Pittsburgh, PA 15219
(412)-690-2442
sritter@
carnegielearning.com

Ken Koedinger
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
(412)-268-7667
koedinger@cmu.edu

ABSTRACT
Bayesian Knowledge Tracing (BKT) has been in wide use for modeling student skill acquisition in Intelligent Tutoring Systems (ITS). BKT tracks and updates student's latent mastery of a skill as a probability distribution of a binary variable. BKT does so by accounting for observed student successes in applying the skill correctly, where success is also treated as a binary variable. While the BKT served the ITS community well, representing both the latent state and the observed performance as binary variables is, nevertheless, a simplification. In addition, BKT as a two-state and two-observation first-order HMM is prone to noise in the data. In this paper, we present work that uses feature compensation and model compensation paradigms in an attempt to conceptualize a more flexible and robust BKT model. Validation of this approach on the KDD Cup 2010 data shows a tangible boost in model accuracy well over the improvements reported in the literature.

Keywords
Cognitive model of student practice, Bayesian Knowledge Tracing.

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 used for 20 years now, and it has served the educational community well. Among the major weaknesses of BKT are the non-identifiability of the parameters, parameter degeneracy [1], and, in general, susceptibility to the noise in the naturally-occurring data. BKT is, by definition, a first-order Hidden Markov Model (HMM) with a binary latent variable representing student knowledge and a binary observed variable indicating student performance. While representing latent student knowledge as a binary variable with *known* and *unknown* states has been widely accepted by the Intelligent Tutoring Community (ITS), it is, no doubt, a simplification. Accounts of the need for a larger number of latent states can be found in the literature, including but not limited to the work of Aleven et al. [2].

Practical issues occur in other fields where first-order HMMs are used intensively (e.g., speech recognition, handwriting recognition, etc.). In these fields, it is common to adopt various compensation measures including model compensation and feature compensation [3]. In this paper, we are applying both compensation paradigms to create a variant of BKT – Spectral BKT – in an attempt to overcome some of BKT's shortcomings. Spectral BKT uses spectral observations – *n*-grams of the consecutive original unary observations of correct and incorrect skill application. It also relies on an extended set of latent states. While a number of Spectral BKT configurations can be conceived, we constructed and empirically tested a setup with eight spectral observations (3-grams of original observations) and four states. To validate the Spectral BKT approach uses an openly available KDD Cup 2010 data set of the 2008-2009 Carnegie Learning's Cognitive Tutor data. The resulting improvement is well above all reported in the literature.

2. RELATED WORK
2.1 Bayesian Knowledge Tracing
Bayesian Knowledge Tracing (BKT) was introduced by Corbett and Anderson [4] in 1995. The standard BKT model assumes that student knowledge of a particular skill is an unobserved binary latent variable that changes based on the binary correctness of the observed student performance. Standard BKT has 4 parameters. Probability of knowing skill a priori (pInit), probability of learning the skill after each opportunity to apply it (pLearn), probability of making a mistake when applying an already known skill (pSlip), and probability of luckily producing a correct response when the skill is not known (pGuess). The probability of knowledge decay (pForget) is assumed to be zero in standard BKT. In general, a HMM with two states and two observations that has a total of 10 numeric values would be said to have 5 parameters (last value in every row is redundant). However, since forgetting is set to zero, BKT is assumed to have 4 parameters.

A large volume of work has been published on fitting BKT models and its variations. Wang and Beck [6] introduced two hierarchical factors into BKT to account for and compare class and student level parameter variability. Xu and Mostow [7] blend BKT approach with logistic regression and create an LR-DBN model that is capable of addressing multiple skill coding for a single step (something that BKT technically doesn't, due to conditional independence assumptions). González-Brenes et al. [8] generalized BKT model to address a feature-rich context addressing multiple skills per step, temporal features, and expert knowledge. Another work of Pardos and Heffernan is an extension of BKT call KT-IDEM [9]. It addressed item variance in the data via introducing item difficulty observable nodes.

2.2 Empirical Problems of BKT
A noticeable portion of the work on BKT models is devoted to discussing problems researchers face when fitting them to the data. Baker et al. [1], when talking about the contextual estimation of guess and slip parameters in BKT, stipulate that their model is less prone to the BKT model degeneracy. What is often meant by

degeneracy are the cases when probabilities of slipping and guessing assume unjustifiably high values, and this often calls for the use of parameter caps. BKT model degeneracy is the artifact of the known issue in HMM called label switching [10]. The issue is made more convoluted by the fact that forgetting is not allowed to vary in BKT and is set to zero.

Work by Beck and Chang [11] discusses an example of yet another problem of BKT – identifiability. There often exists a range of parameter value sets that result in the same likelihood given the data it's estimated on. Falakmasir and colleagues [12] have encountered the same problem in their previous work on the Spectral Learning approach to fitting BKT models. In that work, the formulation of the best-fitting parameter search problem was transformed into the spectral space, where a global optimum of the objective function is guaranteed to be reached. When translating the spectral solution back to the HMM space, the authors had to define a heuristic to pick the most *plausible* parameters from an infinite set of equally good parameter sets.

2.3 Theoretical Issues with BKT

Arguably, it's the two Markov assumptions and the setup of the BKT that result in its known shortcomings. First, is the Limited Horizon Assumption states that the probability of being in a state at time t depends only on the state at time t-1. This kind of HMM is called a first-order HMM since it only has a *memory* of one previous time slice. Second Markovian assumption is the Stationary Process Assumption that the conditional distribution over the next state given the current state does not change over time. Given the fact that BKT has only one parameter to capture state transition, student learning rate is forced to remain constant.

Both, the limited *memory*, and the constant learning rate are simplifications and one can easily construct a case for a more flexible representation of skill learning. For example, between the *unknown* state and *known* state there can be states that capture the preliminary stage of learning when the student having just seen one or two problems is mostly guessing. Before transitioning to the known state, the skill could be in the state that often results in slips since student's knowledge is not strong enough. Another likely reason for BKT's limitations is sensitivity to noise. In BKT, Gaussian noise is assumed for the latent (knowing the skill) and the observed variables. However, when dealing with naturally occurring data, the signal to noise ratio might drop considerably. As a result, one might arrive at degenerate model parameters.

There are two main approaches to handling noise in HMM: feature compensation and model compensation. In feature compensation, the noisy traits (for example, observations) are enhanced to remove the effect of the noise. In model compensation, the original models are mapped into a new model that can be learned from the noisy observations. It has been empirically established that feature compensation is simpler and more efficient to implement, but model compensation has the potential for the greater robustness [3].

3. SPECTRAL BKT

In this work, we are attempting to combine feature compensation and model compensation to overcome the shortcomings of the standard BKT that assumes an ideal noise-free environment and is represented by a first-order HMM. We address feature compensation by changing the way we treat the observations. Instead of a single observation, we are considering n-grams – sequences of consecutive observations for the skill, where next n-gram observation inherits n-1 atomic observations from the previous one. In NLP, 3-grams are often successfully used for feature compensation and we have empirically found that 3-grams work sufficiently well while 2-grams do not. From the information-theoretic point of view, the entropy rate of Hidden Markov Processes with two states proved to have at most second order behavior (captured by second-order HMM) [13]. This means that if we consider the data to be generated by a relatively noise-free naturally-occurring process and that the skills are fine-grained enough, we only need to look at 3-grams of the observations in order to find the true model. One may use n-grams with n greater than 3. However, the computations involved would grow exponentially. Figure 1 shows how the original sequence of observations is encoded into 3-grams.

The model compensation is addressed by adding two intermediate states between the *unknown* and *known* to the original BKT. Once the new observations are defined, the new model that we will call Spectral BKT (due to the use of spectral observations) can be treated as a first order HMM for the purposes of fitting the parameters.

In Spectral BKT, state 1 is the *known* state and state 4 is the *unknown* state. States 2 and 3 we leave unlabeled at this point. Like in the standard BKT, once the student is in the *known* state we assume no un-learning. Moreover, the probability of going from *the unknown* state directly to *the known* state is zero. Finally, once the knowledge transitions from the *unknown* state, there's no return. Given these assumptions, the sparsity structure changes the number of state transition parameters from 1 in standard BKT to 6 in Spectral BKT. By enforcing the sparsity structure in our transition matrix, we guarantee the forward progressing from unknown to known in each iteration and prevent the EM algorithm from learning degenerate models. We assume no further sparsity in any of the 4 priors and 4*7=28 values of the observation matrix, we have (4-1)+6+(7-2)*4=37 parameters in this particular Spectral BKT conceptualization.

The transformation of the original data for fitting the new Spectral BKT is fairly simple (rf. Figure 1). However, when we talk about model predictions, the Spectral BKT produces probability distributions over 8 3-gram observations and one has to make special arrangements to convert them to 2 (probability of correct and of incorrect) in order to compare it with the standard BKT algorithm fairly. First, we ordered the spectral observations from 000 to 111 linearizing a partial order heuristic (rf. Figure 2a). According to this heuristic a spectral observation 011 is the second best indication of success after observation 111. Spectral observation 101 is third best with, potentially, a careless slip in the middle. Spectral observations 001 and 110 were a judgment call. We have placed 001 before 110, assuming it is an early indicator of learning, and 110 is a premature indicator of learning with a failure in the end.

When mapping 8 values to binary success and failure, we came up with three rules. A *regular* rule splits 8 probabilities exactly in half and sums of the two groups are the new probability of correct and incorrect (third column in Figure 2a). The *regular* rule can also be interpreted as looking at the third bit of each 3-gram A *strict* rule is more stringent about which observation probabilities are counted toward success. A *relaxed* rule is more. Since our Spectral BKT produces a first 8-probability predictions starting with the third original observation (due to the use of 3-grams), we have also devised mapping of the 8 probabilities to produce predictions for the first two observations. These mappings are given in Figure 2b,c and reference the spectral observations from Figure 2a. For example, if the observed data contained observations 0, 1, and 0, and the Spectral BKT prediction of

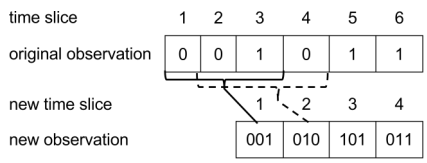


Figure 1. New *n*-gram observations

	Spectral obs.	Regular	Strict	Relaxed
1	111	1	1	1
2	011	1	1	1
3	101	1	1	0
4	001	1	1	0
5	110	0	1	0
6	100	0	0	0
7	010	0	0	0
8	000	0	0	0

(a)

	Spectral obs.	Regular	Strict	Relaxed
1,5	11*	1	1	1
2,7	01*	1	0	1
3,6	10*	0	0	1
4,8	00*	0	0	0

(b)

	Spectral obs.	Regular
1,3,5,6	1**	1
2,4,7,8	0**	0

(c)

Figure 2. Mapping spectral observations from a distribution over 8 probabilities to 2: a) predicting starting with a 3rd original observation when two prior observations are available. b) & c) predicting original observations 2 and 1.

correctness was {0, 0, 0.1, 0.1, 0.1, 0.2, 0.2, 0.3}, then, according to the regular mapping rules, probabilities of correct for the three observations would have been 0.4, 0.3, and 0.2.

4. DATA

To validate our models we used data from KDD Cup 2010 donated by Carnegie Learning, Inc. and available for downloading at http://pslcdatashop.web.cmu.edu/KDDCup. Of the two datasets available we chose Bridge to Algebra. This dataset contains about 20 million transactions belonging to over 6 thousand students working on nearly 150 sections of mathematics curriculum practicing around 1650 skills. The dataset contains information about curriculum context (unit and section the student is in), problem context (problem name and problem step name), cognitive skill labels, timing, as well as correctness of the first attempt to solve the problem step and assistance information (number of hints requested and number of errors). The KDD Cup 2010 is currently the largest freely available collection of learner data. That and the fact that this data was collected by Carnegie Learning's Cognitive Tutor that uses BKT model makes it a good candidate for testing the Spectral BKT. According to the custom of the Carnegie Learning's Cognitive Tutor, skills were considered unique within each curriculum section even if the skill label repeated across several sections. Also, we have treated an absence of the skill (a null skill) as a special skill.

5. MODEL VALIDATION

For the purposes of training the models, we have transformed the original data with unigram observations into a dataset with 3-gram observations. We ran 10-fold student-based and item-based cross-validations that each produced a set of predictions for the transformed 3-gram data. To fit and cross-validate the models we used the `hmmsclbl` tool – a C/C++ utility specially developed to work with large data sets and successfully used in [5] (available for download at http://github.com/IEDMS/standard-bkt). Standard BKT outputs two predictions per data row – probability of correct application of the skills in question and the probability of incorrect application. Spectral BKT works with 8 spectral observations and its predictions come in the form of probability distributions of 8 values per row of the predicted data. Spectral BKT models predictions were mapped from the 8-values onto the 2-value probability distribution schema in Figure 2. The summary of the cross-validation results for the training dataset is listed in Table 1. Here we list the performance of standard BKT next to the performance of Spectral BKT model. We only list results the *relaxed* 3-gram-to-unigram mapping, since *regular* and *strict* mapping performed worse. We tested several solver algorithms `hmmsclbl` supports, including EM and stochastic gradient descent. EM gave a consistently better performance, but the margin was small: within 1% in accuracy and 0.03 in RMSE.

When running student-stratified cross-validation, we were repeatedly *hiding* the full data belonging to 10% of the students. In item-stratified cross-validation, the transactions belonging to problems that we intended to *hide* could appear in individual students' data in arbitrary locations. For the purposes of item-stratification, we have marked the data of 10% of the items as unobserved but accounted for the opportunity to apply skills.

Standard BKT model has 4 parameters per skill. Spectral BKT model, as per our conceptualization of the transition matrix, has 37. The number of parameters being an order of magnitude higher, the AIC and BIC metrics that penalize for that go up 3% and 9% (item-stratified cross-validation). In the case of student-stratified cross-validation, both AIC and BIC are decreasев by 21% and 13%. Accuracy and RMSE in case of Spectral BKT improve a lot. To the best of our knowledge, the overall accuracy of BKT or its variations was never reported to be above 90% on the dataset we used and Spectral BKT hits an impressive 92%.

Recall that we had to *back-predict* the predictions of Spectral BKT for student skill opportunities one and two due to the use of 3-gram observations. For this purpose, in Table 1 we list the additional accuracy and the RMSE values for student skill opportunity 1 alone (7% of the data), opportunity 2 alone (6% of the data), and opportunity 3 and further (87% of the data). To no surprise, the first opportunity prediction of Spectral BKT is slightly worse than the one of standard BKT by a margin in the

Table 1. Comparison of cross-validation results for standard BKT and Spectral BKT

					All opportunities		Opportunity 1		Opportunity 2		Opportunity 3+	
Model	**Par/skill**	**CV**	**AIC**	**BIC**	**Acc.***	**RMSE**	**Acc.**	**RMSE**	**Acc.**	**RMSE**	**Acc.**	**RMSE**
BKT	4	item	15380089	15478083	0.8609	0.3293	0.7469	0.4112	0.8099	0.3731	0.8740	0.3181
Spectral BKT	37	item	15853433	16758456	0.9208	0.2472	0.7472	0.4146	0.8915	0.2940	0.9337	0.2289
BKT	4	student	13947080	14045074	0.8659	0.3153	0.7435	0.4108	0.8126	0.3665	0.8799	0.3020
Spectral BKT	37	student	11553442	12458465	0.9196	0.2405	0.7469	0.4130	0.8897	0.2937	0.9325	0.2210

* For a reference, the majority class accuracy of predicting correct response for every row is 0.8569.

third digit of both accuracy and RMSE, both around 74% and 0.41 respectively. On the second opportunity prediction, Spectral BKT has a decisive edge of almost 9% and 0.08 in RMSE. On the third opportunity and further, Spectral BKT has a comfortable advantage of around 5% in accuracy and 0.09 in RMSE.

6. DISCUSSION

The performance of the Spectral BKT demonstrated a tangible improvement over standard BKT and only with an incremental change in the underlying computations. We attribute the boost in predictive performance to the several factors. First, feature compensation via considering 3-grams of original observations allows for a more stable estimate of the learning process. In a sequence of responses {0,1,0,1,1}, the third value of 0 would be treated a potential slip by the standard BKT. At the same time, Spectral BKT would consider it, as a part of the first triple {0,1,0} to be the *noisy guessing*, and then, in the second triple {1,0,1}, as part of the *noisy slipping*. Finally, in the third triple {0,1,1}, 0 would be considered to be a part of noise-free learning pattern. The fact that there are more than 2 states allows Spectral BKT to represent an intermediate configuration of student learning in addition to just known or unknown. As a result, Spectral BKT is able to deal with the noise in the observations better.

The interpretation of a new conceptualization of the process of learning remains an open question. Having agreed on that state 1 is the known state and state 4 is the unknown state, we could offer several hypotheses of what the remaining middle states are. The first hypothesis relates to the linear view of the stages of mastering the skill. When a student just started learning and only seen a few problems, their knowledge is overly specific, and they would end up guessing and failing a lot. We can call this state 3 – *too-specific*. Once the student sees more problems and starts to generalize the knowledge, they would still occasionally slip due to over-generalization. We can call this state 2 – *too-general*.

Our second hypothesis is related to a publication by Aleven and colleagues [2]. In this work, authors study the metacognitive behavior of students by administering two types of tutors. First, the cognitive tutor that implements a mastery learning approach. Second, the meta-cognitive tutor used a previously created model of effective and ineffective help-seeking behavior in order to study the effect of different meta-cognitive traits on learning. Authors conclude that the use of the standard BKT model with two states might be limiting the capability of the meta-cognitive tutor to offer effective help due to lack of intermediate states between the *known* state and *unknown* state that might give us a better insight into student behavior. In the light of the work by Aleven et al., the progression of the states could be reflecting an interaction of binary latent capturing skill mastery (*known*, *unknown*) with the binary latent capturing effective use of meta-cognitive strategies (2 mastery states * 2 metacognitive states = 4 overall states). To address this hypothesis, one might consider step durations (available in the original dataset) or design and run a focused investigation like the one in [2].

In our work, we used 3-grams of original binary observations, giving us 8 new spectral observations and we also used 4 states. This particular setup can be changed in the search of a better Spectral BKT model. Increasing the number of states could be potentially beneficial. However, one must be careful, for as the number of states grows, the chance to observe relevant patterns of binary observations drops and the Spectral BKT might be under-defined and this could have problems with performing on unseen data whether the patterns missing from the training set are present. When there are fewer states that there are spectral observations, the states serve the aggregation role. We empirically tried configurations of Spectral BKT with 2 states and 4 bigram spectral observations that did not result in an improvement over standard BKT, as well as a configuration with 8 states and 16 4-gram spectral observations that did not result in a tangible improvement over the configuration we discussed in this paper.

7. ACKNOWLEDGMENTS

This work is partially supported by the Andrew Mellon Pre-doctoral Fellowship and extends a project initiated during the Pittsburgh Science of Learning Center's Summer School at Carnegie Mellon University. We extend our special thanks to Dr. Geoffrey J. Gordon from Machine Learning Department at Carnegie Mellon University for insightful comments.

8. REFERENCES

[1] Baker, R.S.J.d., Corbett, A.T., Aleven, V. (2008) Improving Contextual Models of Guessing and Slipping with a Truncated Training Set. EDM 2008, pp.67-76.

[2] Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence and Education*, 16, 101-128.

[3] Gales, M. & Young, S. (2007) The Application of Hidden Markov Models in Speech Recognition. *Foundations and Trends in Signal Processing*, 1(3): 195-304.

[4] Corbett, A.T. & Anderson, J.R. (1995) Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

[5] Yudelson, M., Koedinger, K., & Gordon, G. (2013) Individualized Bayesian Knowledge Tracing Models. AIED 2013, pp. 171-180.

[6] Wang, Y. & Beck, J. (2013) Class vs. Student in a Bayesian Network Student Model. AIED 2013, pp. 151-160.

[7] Xu, Y. & Mostow, J. (2012). Comparison of methods to trace multiple subskills: Is LR-DBN best? EDM 2012, pp.41-48.

[8] González-Brenes, J.P., Huang, Y., & Brusilovsky, P. (2014). General Features in Knowledge Tracing: Applications to Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. EDM 2014, pp.84-91.

[9] Pardos, Z. & Heffernan, N. (2011) KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. UMAP 2011, pp. 243-254.

[10] Redner,R. & Walker,H. (1984) Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev., 26, 195-239.

[11] Beck, J. & Chang, K.-m. 2007. Identifiability: A fundamental problem of student modeling. UM 2007, 137–146.

[12] Falakmassir, M.H., Pardos, Z.A., Gordon, G.J., & Brusilovsky, P.L. (2013) A Spectral Learning Approach to Knowledge Tracing. EDM 2013, pp. 28-34.

[13] Zuk, O., Kanter, I., Domany, E., & Aizenman, M. (2006) Taylor series expansions for the entropy rate of hidden Markov processes. ICC 2006, pp.1598-1604.

Direct estimation of the minimum RSS value for training Bayesian Knowledge Tracing parameters

Francesc Martori
ASISTEMBE
IQS Universitat Ramon Llull
Barcelona, Spain
francesc.martori@iqs.edu

Jordi Cuadros
ASISTEMBE
IQS Universitat Ramon Llull
Barcelona Spain
jordi.cuadros@iqs.edu

Lucinio González-Sabaté
ASISTEMBE
IQS Universitat Ramon Llull
Barcelona, Spain
lucinio.gonzalez@iqs.edu

ABSTRACT
Student modeling can help guide the behavior of a cognitive tutor system and provide insight to researchers on understanding how students learn. In this context, Bayesian Knowledge Tracing (BKT) is one of the most popular knowledge inference models due to its predictive accuracy, interpretability and ability to infer student knowledge. However, the most popular methods for training the parameters of BKT have some problems, such as identifiability, local minima, degenerate parameters and computational cost during fitting. In this paper we address some of the issues of one of these training models, BKT Brute Force. Instead of finding the parameter values that provide the lowest Residual Sum of Squares (RSS), we estimate this minimum RSS value from some a priori known values of the skill. From there we perform some preliminary analysis to improve our knowledge of the relationship between the RSS, from BKT-BF, and the four BKT parameters.

Keywords: Bayesian Knowledge Tracing · BKT Brute Force · RSS modeling

1. INTRODUCTION

1.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [1] is a student model used to infer a student's knowledge given their history of responses to problems, which it can use to predict future performance. Using students' responses to questions, which are tagged with the skills that the instructor wants the students to learn, the model tells the probability a student has mastered a skill.

BKT is a two state Hidden Markov Model, these states being the one in which the student knows a given skill, and the one where the student does not. The "knowledge" state is absorbent, implying that the student will not forget the skill once it is learned. To calculate the probability that a student knows the skill given their performance history, BKT needs to know four probabilities:

L_0, the probability a student knows the skill before attempting the first problem,

T, the probability a student, who does not currently know the skill, will know it after the next practice opportunity, that is the transition probability at each practice opportunity,

G, the probability a student will answer a question correctly despite not knowing the skill,

S, the probability a student will answer a question incorrectly despite knowing the skill.

According to this model, knowledge affects performance (mediated by the guess and slip rates), and knowledge at one time step affects knowledge at the next time step: if a student is in the unknown state at time t, then the probability they will be in the "knowledge" state at time t+1 is P(T). Usually, a separate BKT model is fit for each skill and only the first attempt at each question is taken for each student.

1.2 Bayesian Knowledge Tracing – Brute Force

Bayesian Knowledge Tracing – Brute Force [2] (BKT-BF) is an algorithm to estimate the values for the BKT parameters. It is a simple brute force algorithm, where a grid of possible values is set so that for each combination of parameters, a RSS value is obtained. At the end, the combination of values resulting in the lowest Residual Sum of Squares (RSS) value for a skill is the one that will be used in BKT.
In BKT-BF, the RSS is calculated as follows:

$$RSS = \sum_{i}^{students} \sum_{t=1}^{dim} \left(O_{i,t} - C_{i,t}\right)^2 \quad \text{eq. 1}$$

Where:

$O_{i,t}$ is $\{0,1\}$ depending on the student's answer to a given question,

students is the number of different students who faced any question of a given skill,

dim is the number of different questions that are tagged with a given skill

$C_{i,j}$ is the likelihood to produce a correct answer to a question. This calculation is derived from the BKT formulas, and it is done, for the student *i*, as follows:

$$C_{i,t} = L_{t-1} * (1 - S) + (1 - L_{t-1}) * G \quad \text{eq. 2}$$

BKT-BF is, however, is very expensive in computational cost, as all brute force algorithms are, and does not help the identifiability [3] problem from BKT; identifiability results in different combinations of parameter values, some of which make no theoretical sense, giving similar RSS values. The other most usual algorithm is EM [4], which is not as computationally demanding but suffers from local minima issues. There are efforts to develop methods [5], [6], [7] and [8] that use different techniques to tackle the issues we mentioned, however, in this paper we will focus our work on BKT-BF.

Given that BKT-BF is an algorithm that gives good practical results, but it is so computationally expensive, the objective of this paper is to make accurate estimates of the minimum RSS value for any skill. At the same time, this might provide a better understanding of the BKT model.

2. DATA AND METHODS

The data used belongs to the 'Psychology MOOC GT - Spring 2013' dataset, accessed via DataShop (pslcdatashop.org) [9]. This course was designed by the Open Learning Initiative (OLI), who are known for their data driven design [10], [11], this fact and their long experience in course design ensure that skills have been properly tagged. The course was taken by 5615 students that issued around 2 million first attempt answers. There were 226 different skills identified in the course. The skills map used can be also found in [9]

In order to obtain the RSS values, we have used the BKT-BF algorithm. Specifically, we have used values from 0.05 to 0.95, with a 0.15 step, for L_0 and T; for G and S, the bounded approach has been taken in order to avoid model degeneracy [5], so we have used values from 0.05 to 0.30, with a 0.05 step. Given all this, 1764 different RSS values were obtained per skill.

To identify each skill, we have defined three variables:

- *dim*: number of different questions that are tagged with a given skill
- *n*: total number of responses on questions tagged with a given skill. It's the product of *students* and *dim* from eq.1
- *percent_correct (pc)*: Percentage of correct answers to questions tagged with a given skill

These variables have been chosen as they are pieces of information that one may have easy access to before computing BKT-BF.

In order to achieve the aforementioned objective, we will train a linear model using the three variables we defined for each skill. This model will allow us to make predictions of which will be any skill's minimum RSS, if we were to train it using BKT-BF. To train the model, we have extracted the minimum RSS value, resulting from the BKT-BF calculations, for each one of the skills, and used it as the RSS value for that skill. An example of the data we have worked with is shown in table 1.

It has to be noted, that skills that were tagged in less than 4 different questions (*dim*<4) have been discarded. That results in a sample of 103 different skills for training and evaluating the model.

Table 1. Data structure for skills

dim	n	pc	Skill	Grid BKT-BF	RSS min
8	4923	79,8%	1	1764 data	795.6
16	11062	89,5%	2	1764 data	1024.5
…	…	…	…	…	…

The resulting distribution of RSS values is far from being normal, as it could be expected. However, if instead of using the RSS value, we compute the Root Mean Squared Error (RMSE) for each skill, by taking the square root of the RSS divided by n, the resulting distribution is acceptably normal as we can see in the histogram shown in Figure 1 and in the Q-Q plot in Figure 2. This latter plot assesses normality by displaying the normal theoretical quantiles (x axis) and the normal data quantiles (y axis). If the distribution is perfectly normal, data would perfectly fit the dotted line.

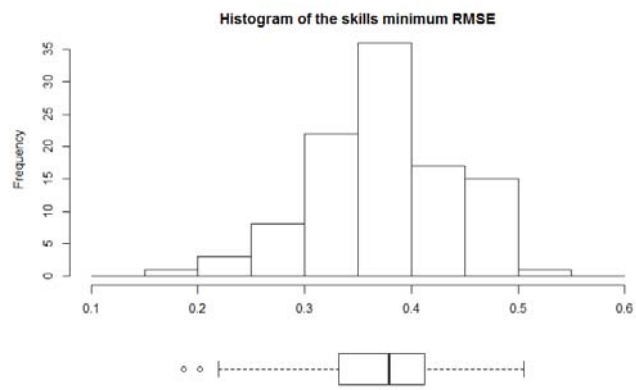


Figure 1. Histogram and boxplot of the RMSE distribution

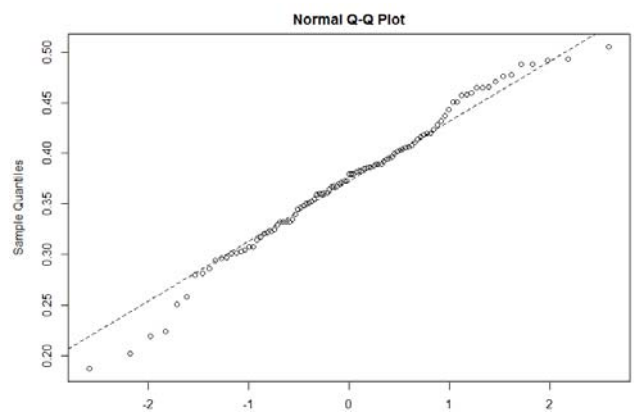


Figure 2. Q-Q plot of the RMSE

3. RESULTS

Firstly, a brief summary for the data we have worked with is shown in the table 2.

Table 2. Summary of the data for training the model.

	n	dim	pc	RMSE	RSS
min	1738	4.00	0.458	0.187	152.0
Median	5899	8.00	0.821	0.379	811.8
Mean	8556	9.65	0.807	0.373	1235.1
Max	47215	23.00	0.964	0.505	8483.4

A linear regression has been performed on the RMSE, using *n*, *dim*, *pc*, some usual transformations, such as using the logarithms and the squares of the variables, and the variable interactions as predictors. A best subset selection (using the leaps package in R, [12]) approach has been taken, resulting the best model the one using a second degree polynomial with *pc*. The results for the linear regression estimates are shown in the table 3.

Table 3. Linear regression results and error metrics

Variable	Estimate	Std. Error	t value	P(>|t|)
Intercept	0.3725	0.0009	415.9	<2e-16
pc	-0.6096	0.0091	-67.1	<2e-16
pc^2	-0.2545	0.0091	-28.0	<2e-16
Adjusted R^2		**Residual standard Error**		
0.981		0.009089		

Finally, using a random validation set (75 skills to train the model and 28 to test it), we have obtained an adjusted R^2 of 0.978, that shows a very good predictive ability for the adjusted model.

In an attempt to have a better knowledge on the relationship between the *RMSE* values and the BKT parameters, we have run a preliminary Principal Components Analysis (PCA). The resulting biplot of the PCA is shown in figure 3. For the sake of a proper understanding of the relationship between the different variables, we have eliminated the data labels from the chart. The variance explained by the first two Components of PCA is 71.4%.

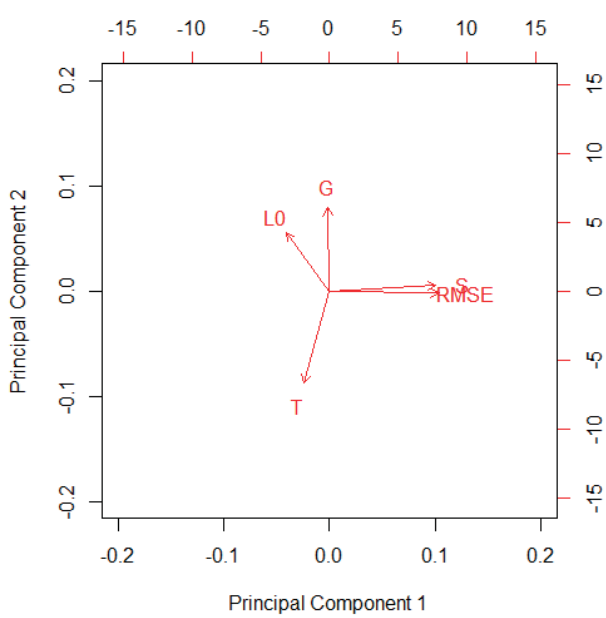


Figure 3. Plot of the PCA loadings of RMSE, L_0, T, G and S

In the chart, we can see how the RMSE is highly correlated with the slip parameter. At the same time, the parameters *G* and *T* seem to be highly inversely correlated, which is something that one can expect as the more likely it is to learn a skill, the less likely it is that you might be guessing the outcome. However, the most noticeable aspect is the orthogonality between *T*, *G* and *RMSE*. In the PCA context, orthogonality is related to poorly correlated variables. If that was to be true, it could imply that T and G have little or no effect in terms of RMSE variation. We have also calculated and drawn the biplots for each skills' RMSEs, using all BKT-BF data points, not just the minimums, and their results lead us to similar conclusions than the ones obtained from figure 3.

4. DISCUSSION AND CONCLUSIONS

We have been able to find a linear model that allows us to estimate the minimum RSS value for the training of the BKT parameters. Using this, we might be able to find a quicker convergence using a modified version of BKT-BF, so that the computational cost will be reduced. Even though that the model has been developed using the RMSE instead of the RSS, the model will also be useful for predicting the latter as the only difference is a transformation involving *dim* and *n*.

We are aware that, in the BKT-BF calculations, we are using a step much larger than the one recommended by the algorithm. This shouldn't be a problem with the conclusions we reached because we are not using BKT-BF for estimating the BKT parameters, but to generate data with which we train a model for estimating the minimum RSS for any skill.

The very high performance of the model, in terms of adjusted R^2, may be indicating that BKT works better when the percentage of correct answers is very high, as the RSS decreases. This has some implications in the BKT model because if the percentage of correct answers is very high, there might not be much room for T and G in the model. We would only be trying to adjust the probability of already knowing the skills before doing the course and the probability of slipping.

To be more certain about the conclusions stated here, the following steps have to include using, at least, a different dataset to shed some light around the suspicions that arise on the influence of T and G in the BKT model. A deeper analysis beyond an exploratory PCA is also required.

5. AKNOWLEDGEMENTS

We acknowledge the help received from colleagues at OLI at Stanford University.

6. REFERENCES

[1] Corbett, A. T. and Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253-278.

[2] Baker, Corbett, Gowda, Wagner, MacLaren, Kauffman, Mitchell, & Giguere, 2010. Bayesian Knowledge Tracing Brute Force model fitting code. http://users.wpi.edu/~rsbaker/edmtools.html

[3] Beck, J. E., Chang, K. M. 2007 Identifiability: A fundamental problem of student modeling. In: Conati, C., McCoy, K., Paliouras, G. (Eds.) UM 2007. LNCS, vol. 4511/2007, pp. 137- 146.

[4] Moon, T. K. 1996. The expectation–maximization algorithm. IEEE Signal Process. Mag., 13, 47–60

[5] Baker, R.S.J.d., Corbett, A. T., Aleven, V.: More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Woolf, B., Aimeur, E., Nkambou, R., Lajoie, S. (Eds.) ITS 2008. LNCS, vol. 5091/2008, pp. 406-415. Springer, Berlin Heidelberg (2008)

[6] Hawkins, W., Heffernan, N.T., Baker, R.S.J.d. (2014) Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. Lecture Notes in Computer Science Volume 8474, 2014, pp 150-155.

[7] Pardos, Z. A., Heffernan, N. T.: Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. In: Baker, R.S.J.d., Merceron, A., Pavlik, P.I. (Eds.) Proceedings of the 3rd International Conference on Educational Data Mining, pp. 161-170 (2010)

[8] Rai, D., Gong, Y., Beck, J.: Using Dirichlet priors to improve model parameter plausibility. In: Barnes, T., Desmarais, M., Romero, C., Ventura, S. (Eds.) Proceedings of the 2nd International Conference on Educational Data Mining, pp. 141-150 (2009)

[9] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. 2010. A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

[10] Strader, R., Thille, C.: The Open Learning Initiative: Enacting Instruction Online. In Oblinger D. G. (Ed.), Game Changers. Education and Information Technologies, pp. 201-213 (2012)

[11] Thille, C. (2012). Changing the Production Function in Higher Education. Making Productivity Real. American Council on Education 2012.

[12] Thomas Lumley using Fortran code by Alan Miller (2009). leaps: regression subset selection. R package version 2.9. http://CRAN.R-project.org/package=leaps

Goodness of fit of skills assessment approaches: Insights from patterns of real vs. synthetic data sets

Behzad Beheshti
Polytechnique Montreal
behzad.beheshti@polymtl.ca

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

ABSTRACT

This study investigates the issue of the goodness of fit of different skills assessment models using both synthetic and real data. Synthetic data is generated from the different skills assessment models. The results show wide differences of performances between the skills assessment models over synthetic data sets. The set of relative performances for the different models create a kind of "signature" for each specific data. We conjecture that if this signature is unique, it is a good indicator that the corresponding model is a good fit to the data.

1. INTRODUCTION

There exists a large array of models to represent and assess student skills. Item Response Theory (IRT) is probably the most established method. It dates back to the 1960's and is still one of the prevailing approaches (see [1]). But many other methods have been introduced in recent years. Among them is the family of models that rely on slip and guess factors [12, 11], such as the DINA (Deterministic Input Noisy And-Gate), DINO (Deterministic Input Noisy Or-Gate), and other variants (see [7]). Other approaches are based on the Knowledge Space theory of Doignon and Falmagne [10, 8], which does not directly attempt to model underlying skills but instead rely on observable items only. Finally, recent methods based on matrix factorization have also emerged in the last decade [16, 15, 5, 2]. They factorize the student per item results matrix into the linear product of the so called Q-matrix (skills required per item) and the skills mastery matrix.

We undertook the effort of comparing prevailing and widely different methods to assess skills. The comparison is based on each method's ability to predict item/task outcome. However, in addition to providing a comprehensive comparison of skills assessment approaches, this research also aims to develop a method that uses synthetic data to characterize item outcome data and yield insights about this data's ground truth structure. Beyond the obvious expectation that the model behind the generation of synthetic data will outperform all others on this data set, we conjecture that the relative performance of all other methods will be unique and can represent a kind of "performance signature" that characterizes this type of data. Therefore, if a data set from a real setting reflects that signature, it would constitute a good indicator that the corresponding model is a good fit.

This work is an extension of [3], and is similar in its general principles to the approach of Rosenberg-Kima and Pardos [13], who take the likelihood of a model's parameter space as a signature instead of the performance of different techniques as we do here. Their idea is that the likelihood function of two parameters of Bayesian Knowledge tracing is a unique characterization of a data set. If the likelihood function of synthetic data generated with estimates of these parameters from real data has the same "signature" as the likelihood function of that real data, then the model is a good fit.

2. SKILLS ASSESSMENT METHODS

We compare a total of seven different skills assessment methods. We briefly describe them here and refer the reader to [7] and [6] for details. They can be grouped into four categories:

(1) The single skill Item Response Theory (IRT) approach. IRT is a well known framework based on logistic regression and represents student proficiency by a single skill (although we also find multiple skills version of IRT, MIRT).

(2) The POKS (Partial Order Knowledge Structrures) represents the order in which items are learned and uses a Naive Bayes framework to make inferences based on this order. It does not represent latent skills, but a Q-matrix can be used aposteriori on the estimated item outcome to assess skills.

(3) The matrix factorization approach decomposes the matrix of m students by n items into the product of m students by k skills representing the latent skills assessment, and an k by n Q-matrix.

(4) The multi-skills family of DINA/DINO approaches are equivalent to a binary matrix factorization framework, where the skill outcome is a boolean product of binary vectors, but they also contain *guess* and *slip* parameters. In the DINA version, the boolean product is based on the AND operator, whereas DINO is based on the OR operator.

Finally, as a baseline for comparison we also consider the *Expected value* as the simplest model. It takes into account the mean item difficulty and student ability to compute the expected score of the corresponding item. The mean difficulty is the average success rate of an item obtained from the training data, while the student ability is the mean success rate obtained from the observed data. The Expected value is the geometric mean of the product of these two means.

3. METHODOLOGY

The performance of each method is assessed on the basis of 10-folds cross-validation, and on observing all items from a student except the one that is to be predicted. For each fold, each item in the set is taken as a target prediction once.

For the IRT and POKS models, the parameters of each models are trained and the testing is based on feeding the models with all but one question. A probability of mastery is obtained and rounded, resulting in a 0/1 error loss function. We report the mean accuracy as the performance measure. The R package `ltm` is used for parameter and skills estimation.

For the other models, they rely on a Q-matrix to estimate the remaining item outcome. For the linear conjunctive and compensatory models, the Q-matrix needs to be normalized such that if all skills for an item are mastered, the inner product of the skills mastered vector and the skills required will be 1. Here too, results are rounded for obtaining a 0/1 loss function. Normalization of the Q-matrix is not necessary for the DINA and DINO models.

4. DATA SETS AND SYNTHETIC DATA GENERATION

The performance of the methods is assessed over a total of 14 data sets, 7 of which are synthetic, and 7 are real data. They are listed in table 1), along with the number of skills of their Q-matrix, their number of items, the number of the student respondents, and the average score. Table 1 also reports the Q-matrix used. To make these data sets more comparable to their real counter part we used Q-matrices and other parameters from real data sets to generate synthetic datasets.

Of the 7 real data sets, only three are independent. The other 4 are variations of a well known data set in fraction Algebra from Tatsuoka's work [14]. The real data sets were obtained from different sources and are freely available from the CDM and NPCD R packages. The Q-matrices of the real data sets were made by experts.

The synthetic data sets are generated from their underlying respective skills assessment model.

For POKS, the structure was obtained from the Fraction data set and the conditional probabilities were generated stochastically, but in accordance with the semantic constraints of these structures and to obtain an average success rate of 0.5.

For IRT, the student ability distributions was obtained from the Fraction data set, and the item difficulty was set to reasonable values: averaging to 1 and following a Poisson distribution that kept most values between 0.5 and 2[1].

Data set	Number of			Mean Score	Q-matrix
	Skills	Items	Students		
Synthetic					
1. Random	7	30	700	0.75	$\mathbf{Q}_{01}$
2. POKS	7	20	500	0.50	$\mathbf{Q}_{02}$
3. IRT-Rasch	5	20	600	0.44	$\mathbf{Q}_{04}$
4. DINA	7	28	500	0.31	$\mathbf{Q}_{5}$
5. DINO	7	28	500	0.69	$\mathbf{Q}_{6}$
6. Linear Conj.	8	20	500	0.24	$\mathbf{Q}_{1}$
7. Linear Comp.	8	20	500	0.57	$\mathbf{Q}_{1}$
Real					
8. Fraction	8	20	536	0.53	$\mathbf{Q}_{1}$
9. Vomlel	6	20	149	0.61	$\mathbf{Q}_{4}$
10. ECPE	3	28	2922	0.71	$\mathbf{Q}_{3}$
Fraction subsets and variants of $\mathbf{Q}_1$					
11. 1	5	15	536	0.53	$\mathbf{Q}_{10}$
12. 2/1	3	11	536	0.51	$\mathbf{Q}_{11}$
13. 2/2	5	11	536	0.51	$\mathbf{Q}_{12}$
14. 2/3	3	11	536	0.51	$\mathbf{Q}_{13}$

Table 1: Datasets

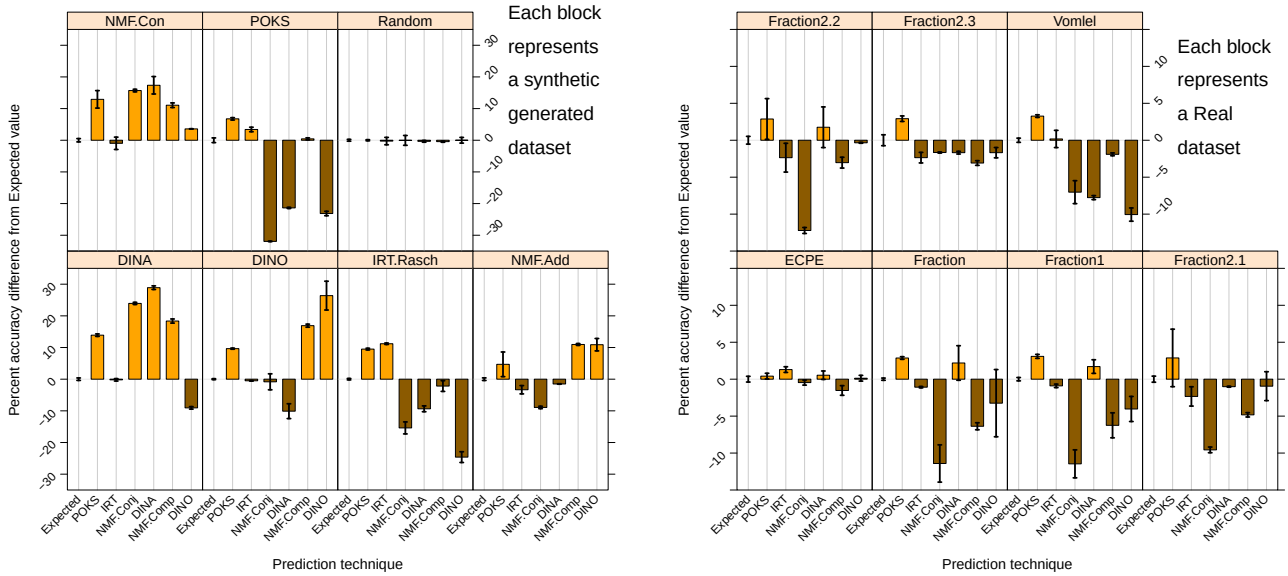
The matrix factorization synthetic data sets of DINO and DINA were generated by taking a Q-matrix of 7 skills that contains all possible combinations of 1 and 2 skills, which gives a total of 28 combinations and therefore the same number of items. Random binary skills matrix were generated and the same process was used for both the DINO and DINA data sets. Item outcome is then generated with a slip and guess factor of 0.1.

A similar process was followed to generate the Q-matrices and the skills matrices $\mathbf{S}$ of the linear matrix factorization data sets

Note that the first 3 models do not rely on any Q-matrix for the data generation process, but the DINO/DINA and matrix factorization assessment methods still require one. To define these Q-matrices (denoted $\mathbf{Q}_{0x}$ in table 1, a wrapper method was used to first determine the number of skills according to [4], then a Q-matrix was derived with the ALS method (see [9]).

All data sets are considered *static* in the sense that they represent a snapshot of student test performance data. This corresponds to the assumption that the student has not mastered new skills during the process of assessment, as we would expect from data from learning environments. This assumption is common to all models considered for this study.

[1]Done by generating random numbers from a Poisson distribution with lambda parameter set to 10 and dividing by 10.



(a) Synthetic datasets

(b) Real dataset

Figure 1: Item outcome prediction accuracy results. Each plot reports the prediction accuracy of the different techniques, whereas each bar shows the percentage difference in accuracy from the Expected value baseline (square root of item × student average success rates).

5. RESULTS AND DISCUSSION

Figure 1 shows the difference between the performance of each technique and the Expected value accuracy as computed by the geometric mean: square root of item × student average success rates. An error bar of 1 standard deviation is reported and computed over the 10 random sampling simulation runs and provides an idea of the variability of the results. Also reported is the performance of random data with a 0.75 average success rate.

As expected, when the generative model behind the synthetic data set is the same as the skills assessment technique, the corresponding technique's performance is generally the best. Exceptions are found for the linear conjunctive case, where the corresponding technique performance comes second. For real data, the performance of many techniques is often lower than the Expected value baseline. This is likely due to the fact that all but one item is observed, the target, and therefore the Expected value is a reliable predictor.

The most consistent performance across the synthetic data sets are those of POKS and IRT, with POKS showing a greater accuracy on average. This consistency also transfers to the real data sets, although the differences are smaller and the Expected value method performance is sometimes better than the IRT one. But as mentioned the good performance of the Expected value may well depend on the relatively high number of observations for each data sets (1 less than the total number of questions per data set).

Also worth noticing is that the random data set has a flat performance across techniques which corresponds to the dominant class prediction. This is not necessarily surprising, but it is reassuring in a sense to know that they all perform the same in the face of random data and this performance is indeed the best that could be obtained.

For the independent real data sets, the differences between techniques are less divergent and closer to the Expected value technique, although the best performers are still significantly better than the Expected value for the Fraction (POKS and DINA) and Vomlel (POKS) data sets. However, for the ECPE data set, the pattern corresponds closely to that of random data: The Expected value performance is close to the dominant class performance, and all techniques are aligned towards this performance. One possibility is that all student perform more or less the same and therefore no technique is good at discriminating high/low performers.

The results from the subsets of the Fraction data shows that the pattern of the Fraction performance data set repeats over Fraction-1, Fraction-2/1 and Fraction-2/2, in spite of the different number of skills and different subsets of questions. However, it differs substantially from Fraction-2/3 for the NMF conjunctive performance which reaches that of the NMF compensatory one. This is readily explained by the fact that the Q-matrix of this data set has the property of assigning a single skill to each item, in which case the two matrix factorization techniques become equivalent.

As mentioned, the performance of the Expected value technique is high for real data, and systematically close to the best performers, POKS and DINA, which only have 2–4% better performance than the Expected value. Note that this is still substantial because we have to look at this difference relative to the remaining error (about 20%), but it is far less than for the synthetic data sets, especially on a relative difference basis.

6. CONCLUSION

This study relies on the assumption that better skills models result in better item outcome prediction. The results do show wide differences in the performance of the techniques for different synthetic data sets. For real data sets, the differences are smaller, though still significant, especially in terms of relative residual errors. Based on the results, we could conclude that POKS and DINA would provide more accurate estimates of skills.

Let us return to the comparison of real vs. synthetic data and to the conjecture that this comparison can help determine whether a specific skill model corresponds to the ground truth of some data set. This is a complex question but some clear hints are given in the results. There is a clear evidence in the DINA vs. DINO performance of figure 1 data that, if a Q-matrix is conjunctive vs. disjunctive, the results show a much better the fit to the corresponding model. Evidence is also some evidence to the claim that unidimensional data sets, i.e. a domain for which a single skill best characterizes the performance data, are best modelled by the IRT single skill IRT or the skill-less POKS models, and the multi-skills NMF conjunctive and DINA approaches do rather poorly. Conversely, multiple skills data sets of the DINO/DINA and linear family of models are better characterized by multi-skills approaches, and the IRT single skill performance is much lower in relative terms.

Another interesting finding is that random data does have a signature of its own: all methods converge towards the score of the majority class. Now, this result could stem from a set of highly similar response patterns from students, but it is clearly different from, for example, the Fraction-2/3 data set, for which all methods have relatively similar performance but they are all well above the majority class condition (AVG Success rate).

Therefore, we do conclude that there is evidence to support the claim that the relative performance of the different skills modelling approaches do create signatures over data sets and can yield some evidence about the ground truth. And if we accept this perspective, then we can also conclude that the real data sets we studied do not correspond to any of the prototypical synthetic data sets. The ground truth may involve correlations between skills, which we did not take into account. Or, the Q-matrices we have studied are not faithful to the reality and, for example, may involve combinations of conjunctive and disjunctive skills. In fact, many explanations can be evoked, but the hope is that by looking at the relative performances of each method we can gain some insights of the best explanations.

References

[1] F. B. Baker and S.-H. Kim. *Item Response Theory, Parameter Estimation Techniques (2nd ed.)*. Marcel Dekker Inc., New York, NY, 2004.

[2] T. Barnes. Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on Educational Data Mining*, 2010.

[3] B. Beheshti and M. C. Desmarais. Predictive performance of prevailing approaches to skills assessment techniques: Insights from real vs. synthetic data sets. In *5th International conference on Educational Data Mining, EDM 2014*, pages 409–410, London, UK 2014.

[4] B. Beheshti, M. C. Desmarais, and R. Naceur. Methods to find the number of latent skills. In *5th International conference on Educational Data Mining, EDM 2012, Chania, Greece, 19–21 June 2012*, pages 81–86. Springer, 2012.

[5] M. Desmarais. Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization. In *4th International Conference on Educational Data Mining, EDM*, pages 41–50, 2011.

[6] M. C. Desmarais, B. Beheshti, and R. Naceur. Item to skills mapping: Deriving a conjunctive Q-matrix from data. In *11th Conference on Intelligent Tutoring Systems, ITS 2012*, pages 454–463, Chania, Greece, 14–18 June 2012 2012.

[7] M. C. Desmarais and R. S. 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.

[8] M. C. Desmarais, P. Meshkinfam, and M. Gagnon. Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction*, 16(5):403–434, 2006.

[9] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-Matrices. In *6th International Conference, AIED 2013, Memphis, TN, USA*, pages 441–450, 2013.

[10] J.-P. Doignon and J.-C. Falmagne. *Knowledge Spaces*. Springer-Verlag, Berlin, 1999.

[11] B. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric IRT, Dec. 27 2000.

[12] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.

[13] R. B. Rosenberg-Kima and Z. Pardos. Is this data for real? In *Twenty Years of Knowledge Tracing Workshop (colocated with EDM 2014*, pages 141–145, London, UK.

[14] K. K. Tatsuoka. *Analysis of errors in fraction addition and subtraction problems*. Computer-based Education Research Laboratory, University of Illinois, 1984.

[15] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. Matrix and tensor factorization for predicting student performance. In A. Verbraeck, M. Helfert, J. Cordeiro, and B. Shishkov, editors, *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011*, pages 69–78. SciTePress, 2011.

[16] T. Winters, C. Shelton, T. Payne, and G. Mei. Topic extraction from item level grades. In *American Association for Artificial Intelligence 2005 Workshop on Educational Datamining*, 2005.

A Transfer Learning approach for applying Matrix Factorization to small ITS datasets

Lydia Voß
Information Systems and Machine Learning Lab
Universitätsplatz 1, 31141
Hildesheim, Germany
lvoss@ismll.uni-hildesheim.de

Carlotta Schatten
Information Systems and Machine Learning Lab
Universitätsplatz 1, 31141
Hildesheim, Germany
schatten@ismll.uni-hildesheim.de

Claudia Mazziotti
Institute of Educational Research, Ruhr-University Bochum
Universitätsstraße 150, 44780
Bochum, Germany
claudia.mazziotti@rub.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
Universitätsplatz 1, 31141
Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

Machine Learning methods for Performance Prediction in Intelligent Tutoring Systems (ITS) have proven their efficacy; specific methods, e.g. Matrix Factorization (MF), however suffer from the lack of available information about new tasks or new students. In this paper we show how this problem could be solved by applying Transfer Learning (TL), i.e. combining similar but not equal datasets to train Machine Learning models. In our case we obtain promising results by combining data collected of German fractions' tasks (517 interactions, 88 students, 20 tasks) with their non-exact translation of a previously American US version (140 interactions, 14 students, 16 tasks). In order to do so we also analyze the performance of MF based predictors on smaller ITS' samples evaluating their usefulness.

Keywords

Transfer Learning, Intelligent Tutoring Systems, Matrix Factorization, Vygotsky Policy Sequencer

1. INTRODUCTION

One of the main uses of Educational Data Mining in Intelligent Tutoring Systems (ITS) is Performance Prediction, which aims to ameliorate the student's model by understanding whether a student mastered a specific set of skills or not. Specific methods, e.g. Matrix Factorization (MF), suffer from the lack of available information about new ITS tasks or new students imposing challenging requirements on organizing trials. This happens because the algorithm is personalized, i.e. there is one model for each student interacting with the system and one for each task one can practice with. If no data are available for one task or for one student no prediction can be computed, this problem is called the cold-start problem. Moreover, first data for new tasks in ITS applications are obligatorily collected in a specific sequence, which is generally fixed or rule-based. As a consequence more interaction data are available for the first tasks in the sequence whereas just a few are available for the last ones making the prediction for specific tasks more challenging. In the FP7 iTalk2Learn project[1] we developed a domain independent sequencer [9] for one of our use cases based on MF Performance Prediction. One of this use cases is a German translation of Fraction Tutor (FT) a web-based Cognitive Tutor for fractions developed by Carnegie Mellon University[2]. Our data collection for the German version (88 students, 20 tasks, 517 interactions) represents, to the best of our knowledge, one of the smallest dataset used to train a MF based recommender for Performance Prediction in ITS. We also possess the data collected with the original US American version (16 tasks, 14 students and 140 interactions), which, according to common practice, should be discarded. In this paper we want to:

- Show, that we can use two different but comparable datasets (the German and English ones) to ameliorate Performance Prediction.
- Analyze in detail the effects of a small dataset on the performances of MF used as performance predictor.
- Propose a practical solution to the data collection to reduce data sparsity.

The paper is structured as follows. the second and third section describe the state of the art and the theory behind the performance predictors we used. In Sec. 4 the data collection, translation and preprocessing is described. In the Experiment Section we discuss the usefulness and measure

[1] www.iTalk2Learn.eu
[2] https://mathtutor.web.cmu.edu/

the performances of MF based predictors. Then we conclude the Section combining the English and German datasets to evaluate the feasibility of Transfer Learning approaches to exploit generally discarded data in ITS.

2. RELATED WORK

As we did not have access to the required skills information in [7, 8], MF and the VPS sequencer presented in [9] are used for Performance Prediction. MF has many applications, its most common use is for Recommender Systems and recently this concept was extended to Performance Prediction and to sequencing problems in ITS [10, 9], but all experiments were done with simulated students' interactions or offline experiments. In [7], we showed how the VPS sequencer could be integrated and worked in a large commercial ITS. A similar analysis on MF was done in [5] where Performance Prediction was tested on a small dense dataset (each student saw each task). The performance predictors were standard Collaborative Filtering techniques, where the best one performing resulted to be Biased Matrix Factorization (see Section 3.1 for more details). In this paper, we possess even less interactions. Not only the students did not interact with all available tasks, but sometimes they also solved less than three tasks. We try to solve this problem with Transfer Learning (TL)[3]. In contrast to classical Machine Learning methods, TL methods exploit the knowledge accumulated from auxiliary data to facilitate predictive modeling consisting of different but similar patterns in the current data [2]. Auxiliary data could mean additional information describing the state of the system and/or data collected with a second slightly modified version of the same system (e.g. using equal movies from different movie rating datasets and transfer the knowledge [4]). In this case correctly done transfer of knowledge, i.e. using similar but not equal datasets, is required and could improve the performance of predictors in classification and regression tasks ([4]) by considering previously unused data. This approach becomes particularly helpful when recollection is expensive or impossible. However TL was never applied to ITS data. Consequently, in Sec. 5.3 we evaluate the feasibility of applying TL to our use case to get a better Performance Prediction.

3. MATRIX FACTORIZATION BASED PREDICTORS

We use MF to predict the students performance. The matrix $Y \in \mathbb{R}^{S \times T}$ can be seen as an incomplete table of T tasks and S students. This matrix is used to train the system. MF is the approximation of this incomplete matrix by decomposing it in two smaller matrices $W \in \mathbb{R}^{S \times K}$ and $H \in \mathbb{R}^{T \times K}$. The elements of the two matrices are called *latent features* and are learned with gradient descend.

Using the available entries (e.g. the score recorded from previous tasks) the missing entries can be computed by means of very fast optimization algorithms. In our experiments we use MF and a simple variation of MF, the Biased Matrix Factorization (BMF) which uses three additional variables: the global average performance μ, the student (user) bias b_s and the task (item) bias b_t. For predicting students performance the following equation is used (for MF without the bold variables):

$$p_{t,s} = \boldsymbol{\mu} + \boldsymbol{b_s} + \boldsymbol{b_t} + \sum_{k=1}^{K} w_{sk} h_{tk}, \tag{1}$$

t represents a task, s a student, k the latent features and K represents the total number of latent features. The optimization function is represented by:

$$\min_{w_s, h_t, \boldsymbol{b_t}, \boldsymbol{b_s}} \sum_{s,t \in \mathcal{D}} (y_{ts} - \hat{y}_{ts})^2 + \lambda \left(\|W\|^2 + \|H\|^2 + \|\boldsymbol{b_t}\|^2 + \|\boldsymbol{b_s}\|^2 \right) \tag{2}$$

with $\mathcal{D}$ the set of collected task student interactions. The final goal of the algorithm is to minimize the Root Mean Squared Error (RMSE) on the set of known scores.

In order to evaluate the performances of BMF and MF generally simple models like Global Average (GA, using the Global Average Score (GAS) of the students as prediction value) are used. To check which is the contribution of the Biases of the BMF to the performance of the MF we use the model called Biases, which has Eq. 2 as optimization function and Eq. 1 as prediction function, but with $K = 0$.

4. DATA COLLECTION AND ITS CHARACTERISTICS

In this section we describe the ITS we used, the data collection and what was done to connect Fraction Tutor and MF approaches.

4.1 Data collection and sequencing

We have carefully translated the English/US American FT tasks into child-friendly German and iteratively adapted to German students' needs. As a result of the translation and adaption process the US American and the German tasks are not 100% identical and we are using TL according to the definition in Sec. 2 and exploiting the knowledge from the auxiliary Englis dataset to ameliorate the German Performance Prediction.

We used three different sequences to have an equal number of interactions for each task, each sequence using a different order of task categories (6 categories). The interleaved sequence starts with one task of each category (hierarchically) and repeats this process. The second sequence refers to the so called blocked practice sequence where first all tasks of category I need to be solved, then category II and so on. Last is the mixed sequence that has a coincidental order.

In order to collect log data and train the MF for the FT we conducted a study with students (i.e. fifth graders) in classrooms (i.e. 21-28 students per class) in Germany. Students of three classes (88 students) of a German Gymnasium could interact with FT which was integrated in the iTalk2Learn platform [4].

The US American data were collected when students (14 of one class) interacted with the US American version of FT [3]. To these students tasks were proposed in a single sequence. All of them completed at least half of the sequence.

4.2 Dataset characteristics

[3]From now on we will refer to Machine Learning's Transfer Learning as TL in order not to mix it with the students' transfer learning

[4]The iTalk2Learn platform is a Plug-In platform used to integrate different components. In our case: FT tasks, database, and simple fixed sequencer.

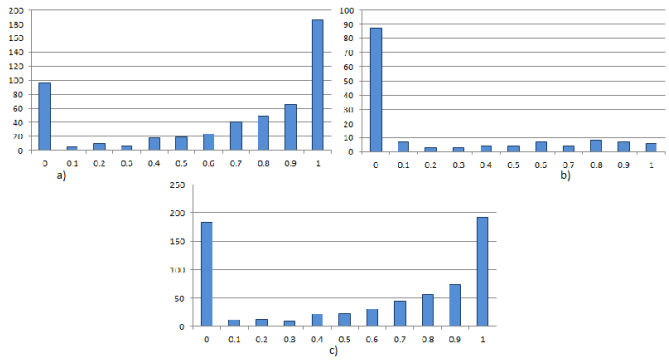


Figure 1: a) German scores b) English scores, c) combined German and English scores

For exploring the task cold-start problem for the German and English datasets (described in Sec. 1) we assigned to each task IDs from 0 to 23, where German and English tasks' (0-15) translations have the same ID. As a result we have: 14 interactions for IDs $0-6$, 11 for ID 7 ((7; 11)), (8; 10), (9; 8), (10; 6), (11; 2), (12; 2), (13; 1), (14; 1), (15; 1). For the German data the interactions are more spread out because of the three different sequences which were used: (0; 38),(1; 59), (2; 36), (3;0), (4;73), (5;47), (6;5), (7;0), (8;22), (9;29), (10;3), (11;0), (12;22), (13;32), (14;0),(15;0), (16;24), (17;32), (18;12), (19;26), (20;29), (21;28), (22;0), (23, 2). There are IDs only used in the English data: (3, 7, 11, 15). The tasks (11, 14, 15, 22, 23) have less than 2 interactions for the German and English datasets and are removed in the preprocessing. Thanks to the different sequences we have a sufficient number ([6]) of interactions for most tasks. For the English experiments we removed the last tasks, since there were too few intercations.

For the students' cold-start problem the dataset can be considered as sparse. The English dataset should be less influenced by the students' cold-start problem, because each student interacted at least with 7 tasks.

In order to have a continuous score measure as we had in [9] we used following equation to compute the score:

$$\text{score} = 1 - \left(\frac{\#hints}{\#totalnumhints} + (\#incorrect * 0.1) \right) \quad (3)$$

If the score is less than zero we set the score to 0 avoiding negative scores. For the German (a)), English (b)) and German+English (c)) data we computed the score Histogram to measure how much the data is unbalanced (See Fig. 1). Both datasets are very unbalanced but by combining the two datasets we can achieve a more balanced distribution. We will explain in the Experiment Section how this is influencing the models' performances.

5. EXPERIMENTS

To split the data in test and train set we used Leave One Out (LOO) for each student; which is a common approach to split for small datasets (here we used the last task seen by the student). To evaluate the error we measure the RMSE averaged over five experiments to avoid the influence of the random initialization of the model parameters on the model performances. The standard deviation of the error for the models prediction lies around 10^{-3}, which is normal for movie recommender datasets and small datasets. For each experiment we used the models described in sec. 3 (GA, MF, BMF). For finding the best hyperparameters we used Grid Search (learning rate: $[0.01, 0.09]$ stepsize 0.01; regularization: $[0.001, 0.009]$ stepsize 0.001, $[0.01, 0.09]$ stepsize 0.01, $[0.1, 0.9]$ stepsize 0.1; num. iterations: $100-300$ stepsize 20; num. latent features: $2-100$ stepsize 10). Moreover for each experiment we computed the performance Global Average Score (GAS) and report the number of students whose data are used.

HL	GA	Biases	MF	BMF
≥ 3	0.337	0.390	0.405	0.386
≥ 4	0.336	0.371	0.385	0.370
≥ 5	0.325	0.325	0.337	0.334
≥ 6	0.319	0.321	0.328	0.322
≥ 7	0.333	0.358	0.355	0.355
≥ 8	0.345	0.298	0.296	0.292

a)

HL	GA	Biases	MF	BMF
≥ 7	0.285	0.240	0.235	0.235
≥ 8	0.295	0.241	0.229	0.218

b)

Figure 2: a) RMSE German, b) RMSE English

HL	≥ 3	≥ 4	≥ 5	≥ 6	≥ 7	≥ 8
GAS	0.673	0.673	0.673	0.673	0.673	0.673
# test students	88	72	53	38	26	22
# interactions	517	501	482	467	455	451

Table 1: GA and test size German data

5.1 Cold-start problem, MF Utility and Intra-Student Variance

For our experiments we studied different History Lengths (HL), i.e. the number of interactions the student had with the ITS, and we deleted the students with a HL less than 2. Starting with HL ≥ 3 we continued removing the students with $HL \leq 4$, $HL \leq 5$, etc. until $HL \leq 8$. We kept the same train data and just removed the test data, so the test set shrinks while increasing the HL requirements. GAS and number of test students are reported in Tab. 1. Table a) in Fig. 2 lists the RMSE for the German dataset.

The performances as well as the behavior of Biases, BMF and MF are coherent with the one reported in [10]. For $HL \leq 5$ Biases, MF and BMF have not sufficient informationto predict the performances (see a) in fig. 2). Keeping students with $HL \leq 5$ in the train influenced BMF negatively. The small gain between BMF and Biases can be explained with the performances of MF which are almost always worse than GA ones. This is coherent with MF and BMF behaviors where generally Biases give a strong contribution to the model performances. We can say that the Performance Prediction of GA was positively influenced by having all data in the train set, since it can be computed on a more robust statistic. BMF and MF are in general influenced by data of students with short history negatively at the beginning, although, for students with a longer history, these data can be used to ameliorate performances.

Next we evaluate the performances of Biases/MF/BMF on an even smaller dataset: the English one. The performances also of GA are quite good, although Biases, MF, and BMF clearly outperform it (see b) in Fig. 2). GA prediction ability is due to the fact that the dataset is highly unbalanced; with a majority of samples with 0 score the probability that a sample of this dataset is similar to the GAS is higher. Fig. 2 shows that BMF outperforms the Biases and the re-

HL	GA	Biases	MF	BMF
≥ 3	0.506	0.391	0.407	0.389
≥ 4	0.500	0.375	0.389	0.375
≥ 5	0.522	0.333	0.342	0.331
≥ 6	0.516	0.322	0.336	0.321
≥ 7	0.523	0.346	0.362	0.344
≥ 8	0.514	0.293	0.285	0.288

a)

HL	GA	Biases	MF	BMF
≥ 7	0.564	0.277	0.288	0.273
≥ 8	0.564	0.283	0.310	0.275

b)

Figure 3: a) RMSE GerEng, b) RMSE EngGer

sults are better than the German ones. According to our previous experience, we think that the difference in the performances (comparing experiments with same HL to avoid the cold-start problem contribution) is due to the variance between the different elements of the students' population under study. In our previous work [1] we showed the negative impact of intra-class variance in the performance of classifiers with small data samples. This applies in our opinion to the case because the intra-student variance of the German data, collected in three classes from different schools, should be higher than the intra-student variance of the English dataset that was collected in one class only.

5.2 Transfer Learning

To test the possibility to use English data to ameliorate the German prediction performances, we combined the English and German datasets as follows. In this experiment the data from an English task and its translation are considered by the MF as the same task. When combining the German and English datasets (See Table a) in Fig. 3), the performances of GA drop to approximately 0.5 because the most samples are almost equally distributed between 0 and 1 with a GAS around 0.56. To prove feasibility of TL we ran more experiments starting with the best results of the previous Sections. We added the English data to the German train set Table a) in Fig. 3), where the addition of the English data in training is always taking to a contribution for $HL \geq 6$.
The same amelioration cannot be seen when adding the German data to the English train, since adding the German data increases the intra-student variance worsening the English model performances (Table b) in Fig. 3, and Tab. 2).

BMF + HL	≥ 3	≥ 4	≥ 5	≥ 6	≥ 7	≥ 8
German	0.386	0.370	0.334	0.322	0.355	0.292
GerEng	0.389	0.375	0.331	0.321	0.344	0.288
English	/	/	/	/	0.235	0.218
EngGer	/	/	/	/	0.273	0.275

Table 2: Comparison of BMFs perfromances for all experiments.

6. CONCLUSIONS

In this paper we proposed a practical solution to the data collection to reduce data sparsity, by proposing tasks with different sequences. Moreover, we analyzed in detail the effects of a small dataset on the performances of MF used as performance predictor. Thanks to these analyses it was also possible to determine the utility of MF based performance predictors and sequencing in new ITS' tasks. Considering the Utility of BMF in comparison to GA, before having at least 7 interactions for a student it would be better to use GA as performance predictor. With using TL we already get better results for BMF with HL ≥ 5. This should hold theoretically also for the use of the VPS, although an experiment with online model update is required for a full evaluation. Finally, we proposed to exploit generally discarded data exploiting the concept of TL. As future work we will investigate more advanced methods to perform TL on small datasets and try to ameliorate performances of the first BMF predictions ($HL \leq 5$).

7. ACKNOWLEDGMENTS

The research leading to the results reported here has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 318051 - iTalk2Learn project (www.iTalk2Learn.eu). Special thanks goes to Jenny Olsen and Martina Rau who agreed to give us access to their versions of Fractions Tutor. Special thanks further goes to Brett Leber and Jonathan Sewall who helped us by adapting the Fractions Tutor tasks into German. Last but not least, we want to thank Hamza Sati, Sebastian Strauss, Jörg Striewski, and Richard Hesse for supporting us when conducting the classroom study in Germany.

8. REFERENCES

[1] R. Janning, C. Schatten, and L. Schmidt-Thieme. Hnnp-a hybrid neural network plait for improving image classification with additional side information. In *ICTAI*. IEEE, 2013.

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang. Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 2015.

[3] J. Olsen, D. Belenky, V. Aleven, and N. Rummel. Using an intelligent tutoring system to support collaborative as well as individual learning. In *ITS*, volume 8474 of *LNCS*, pages 134–143. Springer, 2014.

[4] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, Oct 2010.

[5] Š. Pero and T. Horváth. Comparison of collaborative-filtering techniques for small-scale student performance prediction task. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*. Springer, 2015.

[6] I. Pilászy and D. Tikk. Recommending new movies: Even a few ratings are more valuable than metadata. RecSys, 2009.

[7] C. Schatten, R. Janning, and L. Schmidt-Thieme. Integration and evaluation of a machine learning sequencer in large commercial its. In *AAAI*. Springer, 2015.

[8] C. Schatten, M. Mavrikis, R. Janning, and L. Schmidt-Thieme. Matrix factorization feasibility for sequencing and adaptive support in its. In *EDM*, 2014.

[9] C. Schatten and L. Schmidt-Thieme. Adaptive content sequencing without domain information. In *CSEDU*, 2014.

[10] N. Thai-Nghe, L. Drumond, T. Horvath, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme. Factorization techniques for predicting student performance. *IGI Global*, 2011.

Towards Understanding How to Leverage Sense-making, Induction and Refinement, and Fluency to Improve Robust Learning

Shayan Doroudi[1], Kenneth Holstein[2], Vincent Aleven[2], Emma Brunskill[1]
[1]Computer Science Department, [2]Human-Computer Interaction Institute
Carnegie Mellon University
{shayand, aleven, ebrun}@cs.cmu.edu, kenneth.holstein@gmail.com

ABSTRACT

The field of EDM has focused more on modeling student knowledge than on investigating what sequences of different activity types achieve good learning outcomes. In this paper we consider three activity types, targeting sense-making, induction and refinement, and fluency building. We investigate what mix of the three types might be most effective in supporting robust student learning. To do so, we collected data from students in grades 4 and 5 who completed sequences of activities in largely random order. Students significantly improved from pretest to posttest, suggesting that incorporating all three types can support learning gains. Using hierarchical linear modeling, we found that students who get relatively more fluency problems achieve higher posttest scores. This finding suggests that fluency-building activities are most effective in helping students learn, although our data do not allow us to conclude that fluency alone is sufficient. This work represents a step towards better understanding what combination of different learning mechanisms may best support robust learning.

1. INTRODUCTION

Intelligent tutoring systems (ITSs) have been very effective at enhancing student learning [12, 6]. They typically provide step-level support for complex problem solving such as correctness feedback, next-step hints, and error-specific feedback. ITSs also provide individualized problem selection [11, 3]. It is interesting to consider ITS effectiveness from the perspective of the Knowledge-Learning-Instruction (KLI) framework [5]. KLI posits that three mechanisms of learning—sense-making (SM), induction and refinement (IR), and fluency-building processes—may all be important for robust learning (persistent learning that supports future learning) in any complex domain. However, existing ITSs typically focus only on the IR mechanism through the provision of scaffolded, tutored problem solving. It is possible that providing support for all three learning mechanisms will lead to more robust learning. Supporting the three learning mechanisms would however require a wider range of activity types than typical ITSs offer, to add or enhance support for SM and fluency. Further, it would require that we answer key questions of how and when to provide the different activity types to different learners in an individualized manner, which may itself depend on the student's learning process so far.

In this paper we take a preliminary step towards answering these questions. Fractions Tutor [8] is a web-based intelligent tutoring system for fourth and fifth grade fractions learning. We significantly extended the Fractions Tutor to support all three learning mechanisms. We then collected data from over 600 students with constrained random problem sequences. This allowed us to do a preliminary analysis to understand the contributions of activities targeting the three different learning mechanisms. We did this by fitting a hierarchical linear model (HLM) to our data to see how posttest scores are influenced by the proportion of each activity type in problem sequences as well as looking at the correlation of each activity type with posttest scores. A challenge in drawing conclusions from our data is that the mix of activity types each student was presented with was correlated with the number of problems each student did, but despite this challenge, we show that fluency-building activities are more effective for robust learning.

There has been related work on how to combine two different types of activities, such as worked examples and problem-solving practice [10]. More recent work on MOOCs has analyzed the effectiveness of different activity types chosen by the student (instead of the tutor) [4, 2]. More relevant to the current work is prior work on SM and fluency processes in the Fractions Tutor [8, 9]. While that work also uses hierarchical linear modeling [9], their model includes predictors corresponding to experimental conditions, whereas we have random trajectories with no experimental conditions. Using random sequences gives us the potential to compare a wider variety of relative compositions and sequences of activity types than a standard experimental study.

Finally, prior EDM work has looked at the related problem of how to measure the relative efficacy of different activities [1, 7]. While these works deal with a very similar problem to ours, they differ in at least two main respects from the present work. First, their models consider the efficacy of different activities in performance while being tutored, whereas

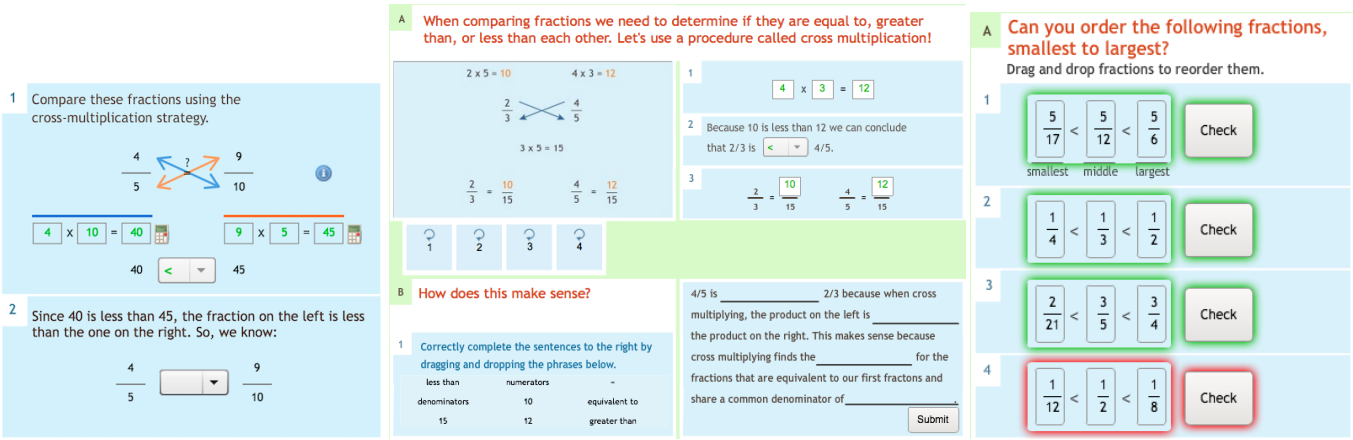


Figure 1: Sample IR (left), SM (center), and fluency (right) activities.

we are interested in robust learning (i.e. performance on a posttest). Second, they only consider which individual activity is best rather than what mix of activities is best. Our modeling approach could in theory suggest optimal mixes of activity types, although we find that in this case the best fitting model reduces to one that can only suggest the relative efficacy of each activity type. It would be worthwhile to compare our findings with the results we can obtain from these models as next steps of our work.

2. METHODS

2.1 Fractions Tutor

For this work, our Fractions Tutor covered topics emphasized in the Common Core[1]: making and naming fractions, fraction equivalence and comparison, and fraction addition. For each topic, we designed three activity types designed to promote each of the KLI learning mechanisms. KLI does not provide strict design guidelines and so we now describe how our designed activities targeting each learning mechanism are in line with KLI's definitions.

Under KLI, IR processes are non-verbal learning processes that improve the accuracy of knowledge [5]. Activities to promote IR processes emphasized procedural learning and practice via fine-grained task decomposition and step-level guidance and feedback, as is typical of ITSs [11]. An IR activity for a procedure for the comparison of two fractions is shown in Figure 1, on the left.

In KLI, SM processes are "explicit, verbally mediated learning in which students attempt to understand or reason" [5]. Our SM activities included instructional videos designed to promote conceptual understanding of targeted fractions topics. The videos were divided into small segments and interspersed with brief supporting problem-solving exercises. Each SM activity concluded with a drag-and-drop fill-in-the-blank question designed to help students self-explain the underlying concepts. An example SM activity for the cross-multiplication procedure is shown in Figure 1 (center). Unlike the IR activities that teach the application of this procedure, the SM activities were designed to help students understand why a certain procedure (e.g., cross-multiplication to compare and order fractions) is effective.

Finally, under KLI, fluency-building processes are non-verbal processes that strengthen memory and enable students to apply their procedural knowledge faster and more fluently [5]. Thus the fluency activities were designed to promote the development of rapid reasoning about fractions and fluent performance on minimally-decomposed problem-solving exercises. Whereas students received support from the tutor via step-level hints in IR activities and video-replays in SM activities, neither were available in fluency activities. See a sample fluency activity in Figure 1, on the right.

2.2 Activity Selection

Since we wish to be able to understand a broader range of activity orderings and mixes rather than a small fixed set, we presented activities to students in a semi-randomized order. A semi-randomized order was chosen as a compromise between two potentially competing objectives. The first is to enhance student learning broadly and for the students that participated in this initial data collection. This objective would push us towards selecting an activity order that draws upon existing research on effective sequencing and satisfies commonly assumed topic orderings. Our second objective is to be able to find effective (potentially adaptive) orderings that may fall outside of the reach of standard procedures. To balance these two competing objectives, we chose to provide students with activity sequences that initially satisfy a prerequisite structure over activity types and topics (designed by the authors). Students could be presented with any activity whose prerequisites had already been presented. This ensured some semantic ordering, e.g. students would not be presented with addition problems before being introduced to the concept of a fraction! However, only a fixed set of 26 problems have prerequisites; once a student finishes the first 26 problems, the student is randomly presented with problems from a large pool of remaining problems.

[1]The Common Core State Standards determine the math curriculum for students from kindergarten through high school in most US states: http://www.corestandards.org/.

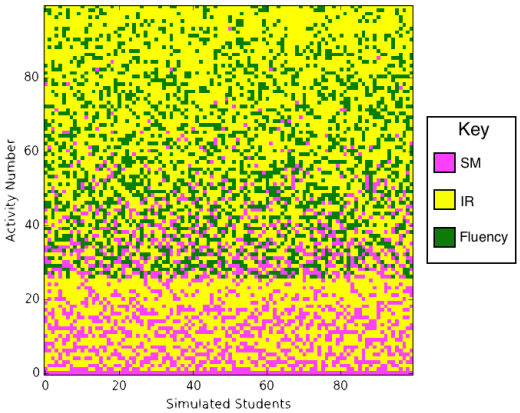


Figure 2: Simulation of potential activity type orderings. Each column represents a sequence of activity types for a student who was given 100 problems.

2.3 Data Collection

We collected data from students using the tutor in eight schools spanning two school districts. Students took a pretest, used the tutor for several sessions, and then took a posttest. The pretest and posttest consisted of 16 items covering conceptual and procedural understanding over skills involved with the three topics. Items were developed by building off of Common Core standards and prior assessment items developed for the Fractions Tutor. For our data analysis we used data from students who started each of the pretest and posttest (639 students).

3. ANALYSIS AND RESULTS

Our ultimate objective for this initial analysis was (1) to evaluate if the new tutor helped students improve their understanding of the material, and (2) to determine what static mix of activity types (SM, IR, and fluency) has the most effective learning outcomes.

3.1 Learning Gains

The mean pretest score is 5.82 ± 3.19 and the mean post test score is 8.23 ± 2.78 (both out of 16). Students significantly improved from pretest to posttest (paired t-test, $t(638) = 27.67$, $p < 10^{-110}$). The effect size was $d = 1.09$, which is considered a large effect size. These results demonstrate that our assortment of activity types can support learning gains, even when those activities are largely randomized.

3.2 Correlation of Variables

Exploratory data analysis revealed a substantial variation in both the number of activities done (mean: 49.6 ± 30.9) and the amount of time students had with the tutor (mean: 183.2 ± 82.3 minutes). Due to the prerequisite structure and semi-randomized ordering used, the number of activities and amount of time spent on the tutor influenced the relative proportion of each activity type that the students completed. To see this we can look at a set of possible simulated sequences that could have been given to students: Figure 2 shows 100 such sequences of 100 problems each. We can observe that students completing 26 problems or less would only receive SM and IR problems. In addition, because the total number of SM activities was fewer than the other two types of activities, if a student did a very large number of activities, the fraction of activities he/she completed would eventually be dominated by IR and fluency.

Predictor	Pearson's r	Partial Pearson's r	p-value
SM	-0.48	-0.15	$5.8 * 10^{-4}$
IR	0.26	-0.033	1
Fluency	0.44	0.18	$5.0 * 10^{-6}$

Table 1: Pearson's r between proportion of problem types and posttest scores, along with partial correlation coefficients when controlling for the number of problems done and amount of time spent on the tutor and Bonferroni corrected p-values for the partial correlations. Predictor variables represent the proportion of problems done by the student that were SM, IR, or F.

To help tease apart the strong correlation between the number of problems and the distribution of activity types completed, we computed the partial correlation between the proportion of problems belonging to each activity type and the posttest score, controlling for both the total number of problems done as well as the amount of time spent by the student. The results are shown in Table 1.

The decrease in magnitude between the raw correlation and partial correlation for each activity type tells us that the number of problems done and total time spent on the tutor accounts for some of the correlation with post test, as expected. More interestingly, the proportion of fluency problems is significantly positively correlated with the posttest scores even after considering the number of problems done and time spent. This suggests that having relatively more fluency problems is beneficial for students, beyond the fact that the students who did more fluency problems tend to have completed more problems; we will verify this with our hierarchical linear modeling. On the other hand, the proportion of SM problems is significantly negatively correlated with the posttest score even after accounting for time and number of problems.

To limit the extent to which students who got more time tended towards a certain mix of activity types, we restricted our subsequent analysis to only those students from one school district who had 150-200 minutes of tutor time in between pretest and posttest (resulting in 268 students).

3.3 Impact of Activity Proportions

The second key issue we wished to investigate was how student learning may be influenced by the mix of different activity types that they complete. To address this issue, we used hierarchical linear modeling to predict posttest scores as a function of the mix of SM, IR and fluency problems that a student completed. In the analysis below, we consider two-level HLMs that treat the class the student is from as a level-2 variable. Using a two-level model resulted in a better fit than just using linear regression. (We tried adding school as a level-3 variable, but this did not improve the

Predictor	Coefficient	p-value
Intercept	12.97	$5.4 * 10^{-9}$
Pretest Score	0.59	$< 1.0 * 10^{-15}$
Proportion SM	-11.20	$9.0 * 10^{-8}$
Proportion IR	-7.67	.021

Table 2: The coefficients of the HLM and their significance with a Bonferroni correction for doing four t-tests. (Satterthwaite approximations were used to compute the degrees of freedom.)

fit.) After trying a variety of models, we found that the best fitting model (in terms of cross-validated RMSE) was one of the simplest. The best model used only three predictor variables: pretest score, proportion of SM problems, and proportion IR problems. (Note that proportion of fluency problems is not a necessary predictor since the three proportions sum to one.) The coefficients for the level-1 variables of the HLM and their p-values are given in Table 2. We see the coefficient for the proportion of SM and the coefficient for the proportion of IR were significant and negative. Thus our model suggests fluency is the most effective activity type (since minimizing the proportion of IR and SM maximizes the posttest score) followed by IR, which agrees with our partial correlation analysis. The apparent lack of efficacy of SM problems may be because these items were substantially more time consuming for students to complete than the two other activity types. Thus even if SM problems are useful, their relative effectiveness per time spent may be lower than more active problems. This is also supported by recent results on the benefit of learning by doing [4].

If our model generalized to all possible sequences, it would suggest that students should do as many fluency problems as possible and not do any SM or IR problems. To allow for non-trivial mixes of activity types, the model would need to include interaction terms between the proportions of different activity types. Such models had statistically insignificant coefficients and worse fits than the model presented.

Nonetheless, it is important to note that any student who did fluency problems in our study necessarily also did SM and IR problems due to the prerequisite structure. Therefore we cannot reliably evaluate the value of a sequence consisting of only a single activity type using our model; such a sequence is *very* different from sequences the students actually received. Rather, the conclusion we can draw from our model is that if we were able to provide additional tutoring to students who already did many problems using our tutor, we should probably just give them more fluency problems.

Notice that our model includes no term for the total number of problems a student did (which we know correlates well with the posttest score). When adding such a term to our model, the fit was worse and the coefficient for that term was both small and statistically insignificant. This implies that the proportion of fluency problems is a better predictor than the number of problems a student did!

4. CONCLUSION

We have extended an existing ITS to include activity types that support all three learning mechanisms posited by the Knowledge-Learning-Instruction framework. In a large-scale classroom study, our ITS had learning gains with a large effect size. A preliminary analysis indicates that students who have a high percentage of fluency problems have the largest posttest scores, suggesting that fluency-building activities are most effective in helping students learn. However, many open questions remain. To what extent are SM and IR problems necessary? Does the appropriate mix of activity types differ for different topics (e.g. making fractions vs. fractions addition)? We hope to address these questions as we work towards our goal of learning personalized policies that best support robust student learning.

5. ACKNOWLEDGEMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A130215 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept. of Education.

6. REFERENCES

[1] J. E. Beck and J. Mostow. How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students. In *Proc. of ITS*, pages 353–362, 2008.

[2] 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 *Proc. of L@S*, pages 11–20, 2014.

[3] A. Corbett, M. McLaughlin, and K. C. Scarpinatto. modeling student knowledge: cognitive tutors in high school and college. *UMUI*, 10:81–108, 2000.

[4] K. Koedinger. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proc. of L@S*, 2015.

[5] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 4 2012.

[6] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of Cognitive Tutor Algebra I at scale. *Educational Eval. & Policy Analysis*, 2013.

[7] Z. A. Pardos and N. T. Heffernan. Detecting the learning value of items in a randomized problem set. In *Proc. of AIED*, 2009.

[8] M. A. Rau, V. Aleven, and N. Rummel. Complementary effects of sense-making and fluency-building support for connection making: A matter of sequence? In *AIED*, 2013.

[9] M. A. Rau, V. Aleven, N. Rummel, and S. Rohrbach. Sense making alone doesnt́ do it: Fluency matters too! its support for robust learning with multiple representations. In *Proc. of ITS*, pages 174–184, 2012.

[10] R. J. Salden, K. R. Koedinger, A. Renkl, V. Aleven, and B. M. McLaren. Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review*, 22(4):379–392, 2010.

[11] K. VanLehn. The behavior of tutoring systems. *IJAIED*, 16(3):227–265, 2006.

[12] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

Learning Behavior Characterization with Multi-Feature, Hierarchical Activity Sequences

Cheng Ye
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
cheng.ye@vanderbilt.edu

John S. Kinnebrew
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
john.s.kinnebrew@vanderbilt.edu

James R. Segedy
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
james.segedy@vanderbilt.edu

Gautam Biswas
Department of EECS and ISIS
Vanderbilt University
1025 16th Ave S, Ste 102
Nashville, TN 37212
gautam.biswas@vanderbilt.edu

ABSTRACT

This paper discusses Multi-Feature Hierarchical Sequential PAttern Mining, MFH-SPAM, a novel algorithm that efficiently extracts patterns from students' learning activity sequences. This algorithm extends an existing sequential pattern mining algorithm by dynamically selecting the level of specificity for hierarchically-defined features individually for each pattern. Consequently, MFH-SPAM operates on a larger space of patterns in the activity sequences. In this paper, we employ a differential version of MFH-SPAM to extract a small set of patterns that best differentiate students with different learning behavior profiles in the Betty's Brain system. Our results illustrate that: (1) MFH-SPAM identifies important patterns missed by traditional sequence mining approaches; and (2) the differential patterns provide additional information for characterizing learning behaviors. This has implications for developing targeted and adaptive scaffolding in open-ended learning environments.

1. INTRODUCTION

Open-Ended Learning Environments (OELEs [4, 7]) present students with a challenging problem-solving task, along with resources and tools for solving the task. Students have the choice to explore, and, therefore, can evolve their solutions in a variety of ways. In previous work, we proposed a theory-based approach called *coherence analysis* (CA) [7] for analyzing student behavior in OELEs. Experimental results showed that grouping students using the CA metrics produced distinct behavior profiles that are discussed in greater detail in Sections 3 and 4. To date we have established the stability and usefulness of our CA measures across extended periods of student work, which does not make this approach directly applicable to adaptive scaffolding as students work in the OELE. To address this problem, our goal has been to use sequence mining methods to find students' activity patterns that are indicators of their behavior profiles. In this paper, we present a case study illustrating that action patterns derived using a novel hierarchical sequence mining approach followed by differential analysis enable classification performance on a par with the groupings derived using CA. Occurrence of individual action patterns can be easily detected online, and future work will assess their utility for early identification of behavior profiles and contextualized scaffolding in OELEs.

In the Betty's Brain OELE [5] each action performed by a student has a number of accompanying features that capture context and consequences of the action. In past work, we used pre-processing methods to select specific features and the level of granularity for each feature to generate 'flat' sequences for pattern mining [2]. This largely ad hoc process resulted in our running many different mining analyses, but often missing potentially important patterns. Other work, such as Plantevit et al. [6], has addressed some aspects of the search in large feature spaces. They define a two-phase technique that first determines frequent combinations of features and levels of specificity in hierarchical representations to pre-processes multi-feature (hierarchical) sequences into a 'flattened' representation. While this approach provides clear advantages over numerous mining analyses with ad hoc feature and granularity choices, many frequent patterns can still be missed due to the initial flattening phase.To address this issue, we have developed a novel **M**ulti-**F**eature, **H**ierarchical **S**equential **PA**ttern **M**ining algorithm (MFH-SPAM).

MFH-SPAM extends the sequence mining algorithm SPAM [1] to simultaneously operate on the entire feature space of action sequences for pattern mining. In this work, we start with MFH-SPAM, and then apply a classifier wrapper method [3] to discover a small subset of mined patterns that are useful for differentiating students across the CA-derived learn-

ing behavior profiles. We have evaluated MFH-SPAM and other traditional sequence mining approaches in this behavior profile classification task using data from a recent study with the *Betty's Brain* OELE. Results show that MFH-SPAM consistently outperforms traditional sequence mining approaches on this task. Further, the differential patterns provide additional information for characterizing student learning behaviors, which has implications for developing targeted and adaptive scaffolding in OELEs.

2. MFH-SPAM APPROACH

Our approach to efficient mining of Multi-Feature, Hierarchical (MFH) sequences extends the SPAM algorithm [1] by directly working with the MFH representation of actions during the mining process. To illustrate this representation, we consider a generic set of possible items/actions to make up sequences (A, B, or C) with an additional feature (*e.g.*, a measure of the action's outcome) that can take on values of $+$ or $-$ at the most general level. In this example, $+$ values for the outcome feature can be further specified as either $+Big$ or $+Small$ at the next level of the hierarchy. Therefore, an individual action might be represented as $B^{+\text{Big}}$, and both $B^{+\text{Big}}$ and $B^{+\text{Small}}$ actions could be more generally represented as B^{+} by abstracting the outcome feature to the more general $+$ level. Further, $B^{+\text{Big}}$, $B^{+\text{Small}}$, and B^{-} actions could all be represented as simply a B action by ignoring the outcome feature entirely. We represent one action followed by another in a sequential pattern using the $\rightarrow$ symbol, such as $A \rightarrow B$ to indicate A followed by B. Itemsets (*i.e.*, co-occurring items in the sequence) are surrounded with parentheses, such as (A, B) to indicate both A and B occurring at the same position in a sequence (*i.e.*, simultaneously).

The core SPAM [1] algorithm searches the space of possible sequential patterns by incrementally extending the current pattern (starting with an empty pattern) in a depth-first manner. For each pattern in the search, SPAM generates the potential "child" patterns by applying one of two types of extensions to the current pattern: 1) a Sequence-extension step (S-step), which appends an item to the end of the sequence (occurring after the last item/itemset), or 2) an Itemset-extension step (I-step), which adds an additional item to the last itemset in the current pattern. For each pattern considered, SPAM calculates the number of sequences in which the pattern occurs using a vertical bitmap representation, explained in more detail later. If the number of sequences in which the new pattern is contained is less than the specified support threshold, SPAM rejects the pattern and does not consider any subsequent extensions to it.

MFH-SPAM augments SPAM with two new pattern extension steps in the pattern search: Feature extensions (F-steps) and Hierarchical extensions (H-steps). During an F-step, MFH-SPAM adds an additional feature to the last item of the current sequence using one of the most general values in the feature hierarchy. For example, the possible extensions to the pattern $A \rightarrow B$ with an F-step would result in $A \rightarrow B^{+}$ or $A \rightarrow B^{-}$. During an H-step, MFH-SPAM selects the last feature of the last item of the current sequence and specifies its value at one level deeper in the feature hierarchy. For example, the possible extensions to the pattern $A \rightarrow B^{+}$ with an H-step would result in $A \rightarrow B^{+\text{Big}}$ or $A \rightarrow B^{+\text{Small}}$.

In addition to these two new extension steps in MFH-SPAM, we define a corresponding extension to the vertical bitmap approach employed in SPAM to efficiently calculate the support for a new pattern[1]. For each data sequence, SPAM initially defines a bitmap for each possible item (*e.g.*, A, B, and C) that represents the locations of that item in the sequence with a value of 1 (all other locations have a value of 0). For example, the sequence $A \rightarrow B \rightarrow B$ would be represented with an A bitmap of [1 0 0], a B bitmap of [0 1 1], and a C bitmap of [0 0 0]. As SPAM generates patterns, it combines item bitmaps to produce *pattern bitmaps* in which 1's represent the endpoints of the corresponding pattern in the sequences. Consequently, for a trivial, single-item pattern like A, the pattern bitmap is exactly the same as the initial item bitmap.

For an S-step extension of a pattern (*e.g.*, extending A to $A \rightarrow B$), SPAM first transforms the current pattern bitmap ([1 0 0]) to indicate where the extension to the current pattern could occur. This is performed by shifting the bitmap to make each location following the occurrence of a 1 in the pattern bitmap a 1 (indicating a *candidate location* for the additional item being added in the S-step) and making all other locations 0 (e.g., resulting in the bitmap [0 1 0]). In other words, $A \rightarrow B$ exists in the sequence if B exists in the candidate location of the second position in the sequence. To complete the S-step (*e.g.*, for A to $A \rightarrow B$) SPAM performs a bitwise AND operation on the transformed pattern bitmap and the item (B) bitmap, resulting in the new pattern bitmap of [0 1 0] indicating that the pattern $A \rightarrow B$ exists and ends at the second position in the sequence.

We extend the SPAM bitmap procedure in F- and H-steps by first creating bitmaps for each possible feature value (at every level of the hierarchy) in the sequence, just as SPAM does with each possible item. Thus, if the original sequence were $A^{-} \rightarrow B^{+\text{Big}} \rightarrow B^{+\text{Small}}$, we would have a $-$ bitmap of [1 0 0], a $+$ bitmap of [0 1 1], a $+Big$ bitmap of [0 1 0], and a $+Small$ bitmap of [0 0 1]. The bitmap operations for F- and H-steps are then analogous to those for S-steps except without the bitmap shift[2] and using the feature value bitmap corresponding to the chosen extension. For example, applying an F-step to add the outcome feature with a value of $+$ to the pattern $A \rightarrow B$, producing $A \rightarrow B^{+}$, would correspond to [0 1 0] (the pattern bitmap) AND [0 1 1] (the feature value bitmap), giving the new pattern bitmap [0 1 0], indicating that this pattern does occur in the example sequence and ends at the second position in the sequence. With the additional F- and H-steps, as well as corresponding bitmap operations for calculating support, MFH-SPAM extends SPAM to efficiently search the space of possible patterns in MFH sequences. Finally, to choose a small subset of the frequent patterns identified by MFH-SPAM (or by SPAM for the experimental comparison) that differentiate the pre-defined learning profiles, we apply a classifier wrap-

[1]In the algorithm description, we describe only the case in which no gaps are allowed between items in the pattern, however, implementing more general gap constraints works in the same manner as with extensions to the original SPAM algorithm

[2]No shift is necessary because the candidate location is for adding further detail to the last item in the current pattern rather than adding an item after it.

per method [3]. Using a greedy approach, the classifier wrapper iteratively identifies the best pattern to include next[3].

3. DATA AND EVALUATION METHODS

The data presented in this paper comes from a study of 98 students from four middle school science classrooms using *Betty's Brain* for six weeks [7]. Six coherence measures were employed to describe the quality and quantity of various problem-solving activities for each student, and hierarchical clustering with these measures identified three primary clusters of students characterized by different behavior profiles [7]. In total, 87 of the students fell into one of these three clusters, and the other 11 students exhibited behavior profiles indicative of either extreme confusion or disengagement. The primary clusters were defined as:(1) *Frequent researchers and careful editors*, who spent large proportions of their time viewing sources of information and did not edit their maps very often; (2) *Strategic experimenters*, who spent a fair proportion of their time viewing sources of information, but often did not take advantage of this information; and (3) *Engaged & efficient students*, who edited their maps very frequently, and usually supported by information from previous activities.

To generate MFH activity sequences for mining, we categorized learning actions into seven primary categories, defined hierarchically (these categories are discussed in more detail in [2]): *Read*ing resource pages; *Search*ing the resources for keywords; causal *Map Edit*ing; *Query*ing the teachable agent, Betty; having Betty take a *Quiz*; asking Betty to *Explain* her answer; or taking *Note*s or causal link annotations (*LinkEval*) indicating whether a link is believed to be correct. To capture the context associated with these actions, we use additional features: (1) the "Length" dimension (applied to Read actions) indicates whether the student spent enough time on the page to have read a significant amount of the material (Full) or only spent a brief period of time on the page (Short) [2]; (2) the "Previous (Full) Read" dimension indicates whether the student has previously done an in-depth ("Full") read of the page or not; (3) the "Supported" dimension indicates whether or not an EditLink action was based on either recently viewed reading materials or quiz results [7], with supported actions denoted by *Sup* and unsupported actions denoted by *NoSup*; and (4) the "Map Score Change" dimension indicates what effect an EditLink action had on the quality of the student's map - whether the quality improved (denoted by +), worsened (denoted by −), or did not change (denoted by =).

We evaluate our MFH-SPAM approach with comparison to four alternative approaches: *Flattened Features (SPAM)* first flattened all activity sequences using all features and the greatest level of action specificity and then used SPAM to generate candidate patterns (e.g., this approach would consider the pattern $\text{LinkRem}^{+}{}_{\text{Sup}} \rightarrow \text{LinkAdd}^{-}{}_{\text{NoSup}}$, but it would not consider the more general pattern LinkEdit → LinkEdit); *Actions-only (SPAM)* considered only the frequent patterns at the most general level of specificity and did not consider any additional features; *MFH-SPAM Baseline by Frequency* used our MFH-SPAM algorithm to generate candidate patterns and simply selected the 10 most frequent patterns; and *Coherence Metrics* classified students using the coherence measures. The performance of each approach was evaluated as the average F1 score of the resulting classifier using 10-fold cross-validation. We chose decision trees as the classifiers and performed this analysis at mining support thresholds ranging from 1.0 to 0.5 in increments of 0.02.

[3]A limit of 10 patterns and an increase of at least 0.1% in performance over the previous pattern set was used in our implementation of the wrapper. A stratified 5-fold cross-validation approach was used for building the classifier in the wrapper with F1 score for evaluation.

4. RESULTS

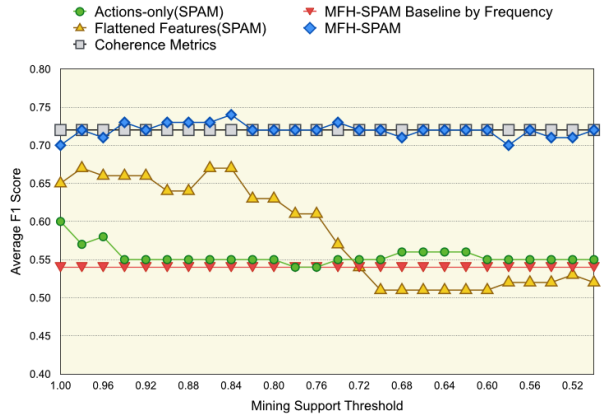


Figure 1: Classification performances of MFH-SPAM and alternative approaches

Figure 1 illustrates the performances of the classifiers built using the candidate feature sets mined in each approach. At each level of support, MFH-SPAM achieved an average F1 score that was much higher than the scores produced by the other sequence mining methods. When using a particularly high mining support threshold, the Flattened Features approach achieves performance close to that of MFH-SPAM, but its performance decreases dramatically as the support threshold is reduced (and the search space is increased). One striking result from this analysis is that MFH-SPAM's performance is on par with the performance of the classifier trained with the features used to perform the original clustering that defined these behavior profile classes. Further, Table 1 presents the five patterns chosen most frequently across the 10 cross-validation folds at a support threshold of 0.9. Considering these top patterns, it is clear that the first three patterns could not have been identified without MFH-SPAM, as they involve multiple levels of hierarchies and feature specificity.

Interestingly, the top MFH-SPAM patterns all involve various forms of causal link edits. This suggests that the way a student went about building their map, as opposed to the way they navigated the resources and investigated Betty's quiz results, was the most useful in predicting their overall learning behavior profile. However, the edit actions, through the support feature, can also incorporate the action's relationship to reading and quiz actions. In other words, what was most helpful in predicting a student's cluster was not the way they acquired information (either from the resources or quiz results), but how they applied previously acquired information to editing their maps. When comparing frequency of use across the three groups, their relative magnitudes are

Table 1: Pattern Frequency Mean (Std Dev) by Cluster for MFH Wrapper with Support 0.9

Pattern	Researchers	Experimenters	Efficient
$\text{LinkRem}^{+}{}_{\text{Sup}} \rightarrow \text{LinkEdit}$	2.6 (2.5)	3.6 (3.0)	14.3 (8.4)
$\text{LinkEdit}^{+}{}_{\text{Sup}} \rightarrow \text{LinkAdd}$	2.3 (1.9)	2.5 (2.6)	12.0 (6.5)
$\text{LinkEdit}^{+}{}_{\text{NoSup}} \rightarrow \text{LinkEdit}$	3.3 (2.9)	16.4 (16.9)	15.6 (12.3)
$\text{LinkEdit}^{-} \rightarrow \text{LinkEdit}^{-}$	3.7 (3.1)	17.5 (16.0)	18.3 (16.2)
LinkAdd^{-}	15.3 (7.2)	28.6 (12.1)	43.5 (21.6)

compatible with the behavior descriptions; *e.g.*, researchers and careful editors make the least number of these edits; engaged & efficient students have the most; and strategic experimenters fall in between. This confirms that the engaged & efficient students, who exhibited the best learning behaviors and the largest learning gains [7], are broadly distinguished from the other groups by more map editing overall: ineffective and effective; supported and unsupported. The usage distributions for these patterns also revealed interesting characteristics about strategic experimenters. These students performed patterns with supported edits far less frequently than engaged & efficient students. Conversely, they performed patterns with unsupported edits far more frequently than researchers and careful editors. Thus, even though the engaged & efficient students made several unsupported and ineffective edits, it would seem that their overall edit distribution is far more favorable to achieving better map scores (and in their case, better pre-post gains on domain knowledge) than that of the strategic experimenters.

To better characterize these three groups, we followed up on previous experimental results [2] and further analyzed the top behavior pattern: (1) $LinkRem^{+}{}_{\text{Sup}} \rightarrow LinkEdit$ that indicates an effective map correction behavior (removing an incorrect link with supporting evidence) followed by further editing. Overall, an average of 19% (s.d. 9%) of the engaged/efficient students' total number of link edits involved this pattern versus 9% (s.d. 8%) for researchers/careful-editors and 9% (s.d. 7%) for strategic experiments. This behavior of incorporating effective map correction in periods of extended map editing appears to be a key characteristic of the engaged/efficient students. Further analysis also suggested that engaged/efficient students were relatively more likely to follow this pattern with a quiz to evaluate their revised map than the researchers/careful-editors and strategic experimenters. This may indicate a greater propensity for the engaged/efficient students to effectively combine evaluation of the causal map with map construction and correction. In summary, going back to OELE characteristics, the engaged and efficient students seem to be better at exploring the problem-solving space, and in distinguishing correct and incorrect approaches to solving complex problems.

5. DISCUSSION AND CONCLUSIONS

MFH-SPAM provides a comprehensive approach to mining OELE activity sequences by efficiently covering the entire MFH action-feature space to generate patterns. Results showed that MFH-SPAM consistently outperforms traditional sequence mining approaches on a behavior profile classification task. Further, analysis of the MFH-SPAM patterns illustrated that a nice, compact way for differentiating these student groups, while retaining high accuracy, was in their approach to map construction and refinement using various forms of editing actions. Overall, these results showed the importance of behavior patterns identified by MFH-SPAM and illustrated the potential to use these patterns to better characterize and ultimately scaffold student learning. In general, effective virtual agents for adaptive scaffolding in OELEs like Betty's Brain may do well to focus on behavior patterns to gain an understanding of how students' *apply their acquired knowledge* (*e.g.*, from reading the resources and studying quiz results) to build and refine models. Detection of specific suboptimal (not using acquired information well) or erroneous behaviors in this context may provide the needed cue for effective scaffolding.

6. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A120186 to Vanderbilt University. The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

7. REFERENCES

[1] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 429–435. ACM, 2002.

[2] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1):190–219, 2013.

[3] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[4] S. Land. Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3):61–78, 2000.

[5] K. Leelawong and G. Biswas. Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.

[6] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, and Y. W. Choong. Mining multidimensional and multilevel sequential patterns. *ACM Transactions on Knowledge Discovery from Data*, 4(1):4:1–4:37, Jan. 2010.

[7] J. R. Segedy, J. S. Kinnebrew, and G. Biswas. Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*, To Appear.

Discrimination-Aware Classifiers for Student Performance Prediction

Ling Luo
School of Information Technologies, The University of Sydney, Australia
National ICT Australia
ling.luo@sydney.edu.au

Irena Koprinska
School of Information Technologies, The University of Sydney, Australia
irena.koprinska@sydney.edu.au

Wei Liu
Faculty of Engineering & IT, University of Technology Sydney, Australia
National ICT Australia
wei.liu@uts.edu.au

ABSTRACT

In this paper we consider discrimination-aware classification of educational data. Mining and using rules that distinguish groups of students based on sensitive attributes such as gender and nationality may lead to discrimination. It is desirable to keep the sensitive attributes during the training of a classifier to avoid information loss but decrease the undesirable correlation between the sensitive attributes and the class attribute when building the classifier. We illustrate, motivate, and solve the problem, and present a case study for predicting student exam performance based on enrolment information and assessment results during the semester. We evaluate the performance of two discrimination-aware classifiers and compare them with their non-discrimination-aware counterparts. The results show that the discrimination-aware classifiers are able to reduce discrimination with trivial loss in accuracy. The proposed method can help teachers to predict student performance accurately without discrimination.

Keywords

Predicting student performance; association rule mining; decision tree; discrimination-aware classification

1. INTRODUCTION

Educational data often contains sensitive attributes such as age, gender and nationality. Mining such data may generate discriminating rules. For example, if our goal is to predict the exam mark of current students, and in the historic dataset used for training of the prediction algorithm, males have achieved significantly higher exam marks than females, a prediction rule using the attribute gender may be generated. It may produce high accuracy but we cannot use it for providing feedback to students or other decision making, as it can be seen as discriminating based on gender, which is unethical and also against the law. Sensitive attributes such as gender should be used as an information carrier and not as distinguishing factors [1]. In this paper we consider building discrimination-aware classification models for predicting student performance.

The task of discrimination-aware classification can be defined as follows [2; 3]: given a labelled dataset and an attribute *S*, find a classifier with high accuracy that does not discriminate on the basis of *S*. There are two approaches to deal with this problem: 1) not using the sensitive attribute to build the classifier and 2) modifying the classification algorithm by integrating a discrimination-aware mechanism to reduce discrimination. The first approach, simply removing the sensitive attribute from the training data, results in information loss and also typically doesn't solve the problem as other attributes are correlated with the sensitive attribute, and will discriminate indirectly. In this paper, we develop and apply methods from the second group which incorporate discrimination awareness during the building of the classifier and use information from the sensitive attribute without causing discrimination.

There are two important aspects that need to be considered when applying discrimination-aware classifiers in educational settings. Firstly, adjusting the classifier to reduce discrimination typically leads to lower predictive accuracy. Given this trade-off between accuracy and discrimination, our aim is to build a classifier with lower discrimination without significant loss in accuracy. Secondly, the output of the classifier should be easy to understand and use by teachers and students. Therefore, we consider classifiers based on decision tree and association rules, which generate sets of rules to guide prediction and decision making.

Our contribution can be summarized as follows:

- We illustrate and motivate the problem of discrimination-aware classification for mining educational data, and show its importance and challenges in educational data mining. Discrimination-aware classification has not been studied for educational data mining and our main goal is to raise the awareness of the community to this problem.
- We introduce our recently proposed classification method Discrimination-aware Association Rule classifier (DAAR) [4]. DAAR uses the novel Discrimination Correlation Indicator (DCI) to measure the discrimination severity of an association rule and select non-discriminatory rules.
- We consider the task of predicting the student exam performance in a first year computer programming course. We apply two discrimination-aware classifiers: our method DAAR and the state-of-the-art Discrimination-Aware Decision Tree (DADT) [3], and compare their performance with standard non-discrimination-aware association rules and decision tree. We show that both DAAR and DADT are able to produce non-discriminatory rules with minimum loss in accuracy.

2. RELATED WORK

Mining educational data to predict student performance has gained increasing popularity. Romero et al. [5] predicted the final student mark based on the Moodle usage data such as the number of passed and failed quizzes, number of completed assignments, number of sent and read messages on the discussion board and the time spent on the assignments, quizzes and discussion board. In their subsequent work [6], the same group studied predicting the

student grade (pass or fail) based on the student participation in a discussion forum, using a number of machine learning algorithms, in the middle and at the end of the semester. Kotsiantis et al. [7] applied an ensemble of classifiers to predict the exam grade (pass or fail) from assessment data during the semester in an online informatics course. Lykourentzou et al. [8] predicted dropouts and completers in e-learning courses on computer networks and web design, using demographic and assessment data.

The discrimination-aware classification problem was introduced in by Pedreshi et al. [2] and Kamiran and Calders [9]. Discrimination-aware naïve Bayes approaches were proposed in [1] and discrimination-aware decision trees were developed in [3].

In this paper, we investigate discrimination-aware classifiers for mining of educational data. We apply our recently proposed discrimination-aware classifier based on association rules and also a discrimination-aware decision tree. We show how these algorithms can be applied for predicting student performance in a first year programming course, discuss the results, and raise the awareness of the Educational Data Mining community to the importance of discrimination-free classification.

3. METHODOLOGY

In this section we describe the main principles of the two discrimination-aware classifiers: our method DAAR and the state-of-the-art DADT. Both classifiers are designed to decrease the discrimination of the predictive model with minimal impact on the accuracy. They are based on the popular and successful association rule classifiers and decision trees, which produce rules that can be easily understood and directly applied by teachers and students.

3.1 Association Rule Classifiers and DAAR

Association analysis discovers relationships among items in a dataset. An association rule takes the form $X \rightarrow Y$, where X and Y are disjoint item sets [10]. Two measures, *support* and *confidence*, are used to evaluate the quality of an association rule. Given a dataset containing N instances and an association rule $X \rightarrow Y$, the support and confidence of this rule are defined as:

$$\text{Support } (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}, \qquad \text{Confidence } (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

where $\sigma(\cdot)$ is the frequency of an item set $(\cdot)$. High-quality rules have high support and confidence.

Classification Based on Association (CBA) [10] uses association rules to solve classification problems. In a standard association rule, any attribute which is not included in X, can appear in Y while in CBA only class attributes can appear in Y.

3.1.1 DCI Measure

To measure the degree of discrimination for an association rule, we propose a new measure called DCI. Given a rule $X \rightarrow y$ and a sensitive attribute S, DCI is defined as:

$$\text{DCI} = \begin{cases} \dfrac{|P(C = y|\, S = S_{\text{rule}}) - P(C = y|\, S = S_{\text{others}})|}{\left(P(C = y|\, S = S_{\text{rule}}) + P(C = y|\, S = S_{\text{others}})\right)} \\ 0 \qquad \text{if either of the above } P(\cdot) \text{ is } 0 \end{cases}$$

where $P(C = y|S = S_{\text{rule}})$ is the probability of the class to be y given the value of the sensitive attribute S is S_{rule}.

When S is a binary or multi-valued attribute, the specific S value in the rule is considered as S_{rule}, and the S_{others} includes the set of all attribute values except the one which appears in the rule. For example, if the rule is "*gender = female, degree = CS → assessment = low*", where *gender* is the sensitive attribute, then S_{rule} refers to *female*, and S_{others} refers to *male*. The DCI for this rule will be:

$$\frac{|P(C = \text{low}|\text{gender} = \text{female}) - P(C = \text{low}|\text{gender} = \text{male})|}{P(C = \text{low}|\text{gender} = \text{female}) + P(C = \text{low}|\text{gender} = \text{male})}$$

When the sensitive attribute does not appear in that rule, we define DCI to be 0.

Therefore, DCI has a range of [0, 1) and its interpretation is the following:

- If DCI is 0, the rule is free of discrimination. DCI is 0 when the probability of the class value to be y is the same for different values of the sensitive attribute S.
- If DCI is not 0, the higher the value, the more discriminatory the rule is with respect to the sensitive attribute S. Thus, the DCI value is monotonically increasing with the discriminatory severity of a rule.

3.1.2 DAAR

DAAR uses DCI together with minimum confidence and support to efficiently select non-discriminatory rules. DAAR's algorithm is shown in Figure 1.

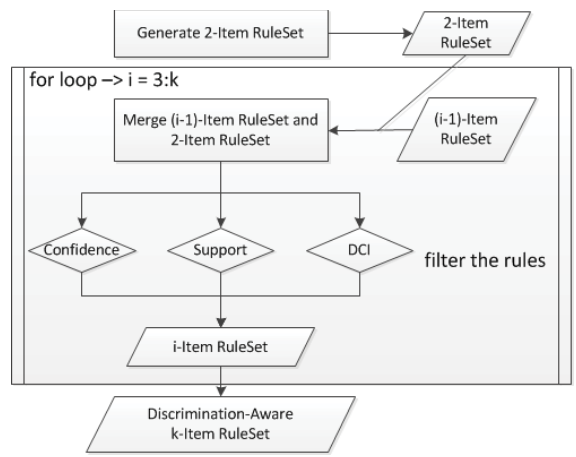


Figure 1. DAAR's Algorithm

DAAR starts from the set of 2-item rules (i.e. the rules with one attribute value and the class attribute), which is the base case, and merges with other 2-item rules iteratively until it gets the k-item rules, where k is the upper bound for the number of items in the rule. In each iteration, the rules are filtered by confidence, support and DCI. To classify new instances, DAAR uses majority voting based on the number of rules that predict the same class. If the vote is tied, the DCI sum for all rules for each class is compared and the class with lower sum (i.e. less discrimination) is selected.

3.2 Decision Tree and DADT

Decision Trees (DTs) are one of the most popular machine learning algorithms. The standard DT algorithm uses information gain to select the best attribute at each step as a root of the tree/subtree, until all examples in the subset belong to the same class, in which case it creates a leaf node labelled with this class. DTs can be seen as generating a set of mutually exclusive rules – each path from the root of the tree to a leaf node is one rule, and each rule is a conjunction of attribute tests. DADT is a discrimination-aware version of DT introduced by Faisal et al. in [3]. The tree is constructed in two phases. In the first phase, it generates a tree by using a new splitting criterion: IGC-IGS. IGC is the standard information gain (Information Gain regarding the

Class label) and IGS is Information Gain regarding the Sensitive attribute, defined as:

$$IGS = H_S(D) - \sum_{j=1}^{k} \frac{|D_j|}{|D|} H_S(D_j)$$

where S is the sensitive attribute, $H_S(D) = -\sum_{i=1}^{n} P_i * \log_2 P_i$ is the entropy of set D with respect to S and P_i is the proportion of items with the ith value of the sensitive attribute.

As the aim is to have higher IGC but lower IGS, the difference IGC-IGS is an appropriate criterion. In the second phase, the leaves are relabeled to decrease the discrimination severity to less than ε (where ε is a non-discriminatory constraint), while sacrificing as little accuracy as possible. Experiments on census income datasets showed that DADT can produce a tree with a lower discrimination while maintaining accuracy [3].

4. EXPERIMENTS AND RESULTS

We consider the task of predicting exam performance in a fist year programming course. We compare the performance of the discrimination-aware classifiers DAAR and DADT with their standard non-discrimination-aware counterparts CBA (standard AR) and C4.5 (standard DT).

4.1 Dataset and Experimental Setup

Learning computer programming is difficult as it requires a lot of practice with feedback, and a very precise way of thinking. It is easy for students to fall behind, especially since introductory computer programming courses have a large number of students. Predicting students at risk of failing or not performing well is highly desirable.

Our evaluation is conducted using data from a first year computer programming course at an Australian University with 220 students. Our goal is to predict the exam performance, *high* or *low*, based on the student grades on the assessments during the semester and some enrolment attributes such as country of residence, degree name and if the student is local or international. A description of the attributes and their values is given in Table 1.

Table 1. Description of Attributes

Attribute	Description	Number of Attribute Values
Country	Country of permanent residence: {Australia, Brazil, China, …}	26
Degree	Name of the degree the student is enrolled into: {Bachelor of Science, Bachelor of Engineering,…}	27
Local	Indicates if the student is Australian or not: {Local, International}	2
a1_grade	The grade of assessment 1 during semester: {HD, D, CR, P, F}	5
a2_grade	The grade of assessment 2 during semester: {HD, D, CR, P, F}	5
a3_grade	The grade of assessment 3 during semester: {HD, D, CR, P, F}	5
a4_grade	The grade of assessment 4 during semester: {HD, D, CR, P, F}	5
a5_grade	The grade of assessment 5 during semester: {HD, D, CR, P, F}	5
Exam	Exam performance during examination period: {high, low}	2

The grades for the 5 assessments during the semester are the standard grades used at the university defined as follows: *HD* (High Distinction, mark of [85, 100]), *D* (Distinction, mark of [75, 84]), *CR* (Credit, mark of [65, 74]), *P* (Pass, mark of [50, 64]) and *F* (Fail, mark below 50). The exam performance is defined as *high* if the exam mark is 65 or higher (i.e. HD, D or CR), and *low* if it is below 65 (i.e. P or F). There were 105 students in the *high* group and 115 in the *low* group.

We selected the exam grade as a variable to predict rather than the final grade in the course, as the exam is the major assessment component (worth 50% and covering all topics) and it is also independent of the assessment components during the semester, while these components contribute to calculating the final grade for the course.

Among the 8 predictors, we consider *country* as the sensitive attribute, which means that we would like to avoid discrimination based on the student nationality. Originally, this attribute had 26 different values, with 5 or less number of students for most of the countries, so we aggregated these values into three groups: *Australia*, *China* and *Others*. The number of students in each group was 127, 54 and 39, respectively.

4.2 Results and Discussion

To evaluate the performance of the classification methods, we use 10-fold cross validation in all experiments. We report both the average value and the standard deviation for the 10 folds. As predictive accuracy measures, we use both classification accuracy and F-measure.

To assess the discrimination severity of the classifier, we calculate a discrimination score. In [1] a discrimination score for a binary sensitive attribute S with values S_1 and S_2, and class values C_+ and C_- is defined as:

$$\text{Score} = |P(C = C_+ | S = S_1) - P(C = C_+ | S = S_2)|$$

As our sensitive attribute has three values, we extend this definition to multi-valued attribute with m ($m>2$) values. We compute the score for each value S_i and then average the m scores:

$$\text{Score} = \frac{1}{m} * \left(\sum_{i=1}^{m} |P(C = C_+ | S = S_i) - P(C = C_+ | S = S_{others})| \right)$$

where S_{others} represents all the attribute values other than S_i.

If the score is 0, there is no discrimination. Otherwise, a higher score corresponds to a higher discrimination severity.

4.2.1 DAAR

Table 2 presents the accuracy results and discrimination score for the standard AR and DAAR. We can see that DAAR was able to decrease the discrimination score of AR from 0.2831 to 0.2653. The trade-off was a slightly lower accuracy - DAAR achieved 73.92% accuracy, which is 4.72% lower than AR's accuracy.

Table 2. Results for Standard AR and DAAR

	Standard AR		DAAR	
	Mean	**Std.**	**Mean**	**Std.**
Accuracy	78.64%	0.0037	73.92%	0.0128
F-measure	0.7863	0.0037	0.7389	0.0131
Disc. score	0.2831	0.0109	0.2653	0.0163

Table 3 shows some representative and interesting rules produced by DAAR with their confidence, support and DCI. These rules are very compact, easy to understand and apply by teachers.

Table 4 shows the rules with high confidence and support that were filtered out by DAAR, as they were discriminatory with respect to the sensitive attribute *country*.

Table 3. Sample Rules Produced by DAAR

Rules	Conf.	Sup.	DCI
a1_grade=CR → exam = low	1.0	0.01	0
degree= Bachelor of Commerce, a4_grade=HD → exam = high	1.0	0.01	0
a4_grade=F → exam = low	1.0	0.19	0
degree= Bachelor of Engineering & Bachelor of Science, a5_grade=HD → exam = high	0.84	0.08	0

Table 4. Discriminatory Rules Removed by DAAR

Rules	Conf.	Sup.	DCI
country=Other → exam = high	0.62	0.12	0.17
country=CH → exam = low	0.77	0.18	0.26
country=Others, a5_grade=HD, a4_grade=HD → exam = high	0.83	0.08	0.17

4.2.2 DADT

The trees produced by the standard DT and DADT are shown in Figure 2 and Figure 3, respectively. The standard DT achieved accuracy of 83.46% but it used the sensitive attribute *country* and its discrimination score was 0.2298. DADT achieved a slightly lower accuracy of 82.73% without using the sensitive attribute. Thus, DADT is able to avoid discrimination with a minimum loss in accuracy. Both DTs included the attribute *a4_grade* as a root of the tree, which shows the importance of this attribute for predicting exam performance.

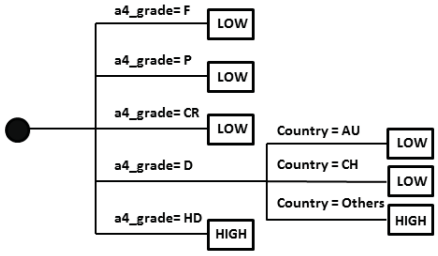


Figure 2. Tree Produced by the Standard DT

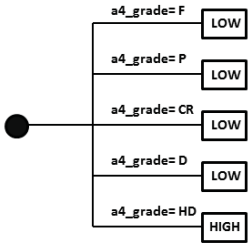


Figure 3. Tree Produced by DADT

4.2.3 Discussion

In terms of overall performance, all four methods had reasonable accuracy, from 73.92% to 83.46%, with the DT-based classifiers outperforming the AR-based classifiers. All classifiers generated a small set of rules that are easy to understand and use by teachers. The AR classifiers used more attributes in the rules which, for our case study, provided additional insights about the important attributes in predicting student performance and providing feedback to students.

In terms of discrimination, we can see that both DAAR and DADT decreased the severity of the discrimination compared to their standard counterparts, with trivial loss in accuracy. Specifically, DAAR removed the rules with higher DCI values and reduced the discrimination score, and DADT using IGC-IGS as an attribute selection criterion, built a DT without using the sensitive attribute *country*.

5. CONCLUSIONS

Educational data often contains sensitive attributes, which should only be used as information carriers rather than factors to distinguish students and potentially discriminate them. We investigated discrimination-aware classification for mining of educational data, with a case study in predicting student exam performance based on enrolment information and assessment marks during the semester, in the context of a computer programming course. We applied our discrimination-aware method DAAR, which is based on association rules, and also DADT, a discrimination-aware decision tree method, and compared DAAR and DADT with their non-discrimination-aware alternatives. The experiment results showed that both DAAR and DADT decreased the discimination with minor impact on the predictive accuracy. Both classifiers generated a small set of rules that are easy to understand and use by teachers and students. The discrimination-aware classifiers can be used for any classification tasks in educational settings, such as identifying students at risk, to provide timely feedback and intervention.

6. REFERENCES

[1] Calders, T. and Verwer, S., 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery 21*, 2, 277-292.

[2] Pedreshi, D., Ruggieri, S., and Turini, F., 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* ACM, 560-568.

[3] Kamiran, F., Calders, T., and Pechenizkiy, M., 2010. Discrimination aware decision tree learning. In *Proceedings of the 10th IEEE International Conference on Data Mining* IEEE, 869-874.

[4] Luo, L., Liu, W., Koprinska, I., and Chen, F., 2015. Discrimination-Aware Association Rule Mining for Unbiased Data Analytics. *TR700*, School of Information Technologies, The University of Sydney.

[5] Romero, C., Ventura, S., Espejo, P.G., and Hervás, C., 2008. Data Mining Algorithms to Classify Students. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, 8-17.

[6] Romero, C., López, M.-I., Luna, J.-M., and Ventura, S., 2013. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education 68*, 458-472.

[7] Kotsiantis, S., Patriarcheas, K., and Xenos, M., 2010. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems 23*, 6, 529-535.

[8] Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., and Loumos, V., 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education 53*, 3, 950-965.

[9] Kamiran, F. and Calders, T., 2009. Classifying without discriminating. In *International Conference on Computer, Control and Communication* IEEE, 1-6.

[10] Ma, Y., Liu, B., and Yiming, W.H., 1998. Integrating classification and association rule mining. In *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 80-86.

Language to Completion: Success in an Educational Data Mining Massive Open Online Class

Scott Crossley
Georgia State U.
Atlanta, GA 30303
scrossley@gsu.edu

Danielle S. McNamara
Arizona State Univ.
Tempe, AZ, 85287
dsmcnama@asu.edu

Ryan Baker, Yuan Wang, Luc Paquette
Teachers College
Columbia University
New York, NY 10027
ryanshaunbaker@gmail.com

Tiffany Barnes
NC State Univ.
Raleigh, NC 27606
tmbarnes@ncsu.edu

Yoav Bergner
Educational Testing Service
Princeton, NJ 08541
ybergner@ets.org

ABSTRACT

Completion rates for massive open online classes (MOOCs) are notoriously low, but learner intent is an important factor. By studying students who drop out despite their intent to complete the MOOC, it may be possible to develop interventions to improve retention and learning outcomes. Previous research into predicting MOOC completion has focused on click-streams, demographics, and sentiment analysis. This study uses natural language processing (NLP) to examine if the language in the discussion forum of an educational data mining MOOC is predictive of successful class completion. The analysis is applied to a subsample of 320 students who completed at least one graded assignment and produced at least 50 words in discussion forums. The findings indicate that the language produced by students can predict with substantial accuracy (67.8 %) whether students complete the MOOC. This predictive power suggests that NLP can help us both to understand student retention in MOOCs and to develop automated signals of student success.

Keywords

Natural language processing, MOOCs, student success

1. INTRODUCTION

The sheer size of student populations in massive open online classes (MOOCs) requires educators to rethink traditional approaches to instructor intervention and the assessment of student motivation, engagement, and success [11]. As a result, a good deal of MOOC research has focused on predicting or explaining attrition and overall student success. Most research assessing student success in MOOCs has involved the examination of click-stream data. Such data provides researchers with evidence of engagement within the course and activities associated with individual course goals [6]. Additional approaches to assessing student success include the use of sentiment analysis tools to gauge students' affective states [15, 16] and individual difference measures such as student backgrounds and other demographic variables [5].

In this paper, we explore the potential for natural language processing (NLP) tools that include but also go beyond sentiment analysis to predict success in an educational data mining MOOC. Our goal is to develop an automated model of MOOC success based on NLP variables such as text length, text cohesion, syntactic complexity, lexical sophistication, and writing quality that can be used to predict class completion. Thus, in line with Koller et al. [7], we hope to better understand the language produced by MOOC students, especially differences in the language between those students that complete a course and those that do not. Using NLP variables affords the opportunity to go beyond click-stream data to examine student success and allows the personalization of predictive variables based solely on the language differences exhibited by students. Such fine-grained content analyses may allow teachers to monitor and detect evidence of student engagement, emotional states, and linguistic ability to predict success and intervene to prevent attrition.

1.1 NLP and MOOC Success

Researchers and teachers have embraced MOOCs for their potential to increase accessibility to distance and lifelong learners [7]. From a research perspective, MOOCs provide a tremendous amount of data via click-stream logs within the MOOC platform. These data can be mined to investigate student learning, student completion, and student attitudes. Typical measures include frequency of access to various learning resources, time-on-task, or attempt rates on graded assignments [14]. Less frequently mined, however, are data related to language use [15, 16].

NLP refers to the examination of texts' linguistic properties using a computational approach. NLP centers on how computers can be used to understand and manipulate natural language texts (e.g., student posts in a MOOC discussion forum) to do useful things (e.g., predict success in a MOOC). The principal aim of NLP is to gather information about human language understanding and production through the development of computer programs intended to process and understand language in a manner similar to humans [3]. Traditional NLP tools focus on a text's syntactic and lexical properties, usually by counting the length of sentences or words or using databases to compare the contents of a single text to that of a larger, more representative corpus of texts. More advanced tools provide measurements of text cohesion, the use of rhetorical devices, syntactic similarity, and more sophisticated indices of word use.

In MOOCs, the most common NLP approach to analyzing student language production has been through the use of sentiment analysis tools. Such tools examine language for positive or negative emotion words or words related to motivation, agreement, cognitive mechanisms, or engagement. For instance, Wen et al. [16] examined the sentiment of forum posts in a MOOC to examine trends in students' opinions toward the course and course tools. Using four variables related to text sentiment (words related to application, cognitive words, first person pronouns, and positive words), Wen et al. reported that students' use of words related to motivation had a lower risk of dropping out of the course. In addition, the more students used personal pronouns in forum posts, the less likely they were to drop out of the course. In a similar study, Wen et al [15] reported a significant correlation between sentiment variables and the number of students who dropped from a MOOC on a daily basis. However, Wen et al. did not report a consistent relation between students' sentiment across individual courses and dropout rates (e.g., in some courses negative words such as "challenging" or "frustrating" were a sign of engagement), indicating a need for caution in the interpretation of sentiment analysis tools.

2. METHOD

The goal of this study is to examine the potential for NLP tools to predict success in an EDM MOOC. Specifically, we examine the language used by MOOC students in discussion forums and use this language to predict student completion rates.

2.1 The MOOC: Big Data in Education

The MOOC of interest for this study is the Big Data in Education MOOC hosted on the Coursera platform as one of the inaugural courses offered by Columbia University. It was created in response to the increasing interest in the learning sciences and educational technology communities in learning to use EDM methods with fine-grained log data. The overall goal of this course was to enable students to apply each method to answer education research questions and to drive intervention and improvement in educational software and systems. The course covered roughly the same material as a graduate-level course, Core Methods in Educational Data Mining, at Teachers College Columbia University. The MOOC spanned from October 24, 2013 to December 26, 2013. The weekly course comprised lecture videos and 8 weekly assignments. Most of the videos contained in-video quizzes (that did not count toward the final grade).

All the weekly assignments were automatically graded, numeric input or multiple-choice questions. In each assignment, students were asked to conduct an analysis on a data set provided to them and answer questions about it. In order to receive a grade, students had to complete this assignment within two weeks of its release with up to three attempts for each assignment, and the best score out of the three attempts was counted. The course had a total enrollment of over 48,000, but a much smaller number actively participated; 13,314 students watched at least one video; 1,242 students watched all the videos; 1,380 students completed at least one assignment; and 710 made a post in the weekly discussion sections. Of those with posts, 426 completed at least one class assignment; 638 students completed the online course and received a certificate (meaning that some students could earn a certificate without participating in the discussion forums at all).

2.2 Student Completion Rates

We selected completion rate as our variable of success because it is one of the most common metrics used in MOOC research [17]. However, as pointed out by several researchers, learner intent is a critical issue [5, 6, 7]. Many MOOC students enroll based on curiosity, with no intention of completing the course. The increased use of entry surveys is no doubt related to this inference problem. In the present analysis, however, we do not have access to this information. Therefore, we compute completion rates based on a smaller sample of forum posters as described below. "Completion" was pre-defined as earning an overall grade average of 70% or above. The overall grade was calculated by averaging the 6 highest grades extracted out of the total of 8 assignments.

2.3 Discussion Posts

We selected discussion posts because they are one of the few instances in MOOCs that provide students with the opportunity to engage in social learning [11, 16]. Discussion forums provide students with a platform to exchange ideas, discuss lectures, ask questions about the course, and seek technical help, all of which lead to the production of language in a natural setting. Such natural language can provide researchers with a window into individual student motivation, linguistics skills, writing strategies, and affective states. This information can in turn be used to develop models to improve student learning experiences [11]. In the EDM MOOC, students and teaching staff participated in weekly forum discussions. Each week, new discussion threads were created for each week's content including both videos and assignments under sub-forums. Forum participation did not count toward student's final grades. For this study, we focused on the forum participation in the weekly course discussions.

For the 426 students who both made a forum post and completed an assignment, we aggregated each of their posts such that each post became a paragraph in a text file. We selected only those students that produced at least 50 words in their aggregated posts (n = 320). We selected a cut off of 50 words in order to have sufficient linguistic information to reliably assess the student's language using NLP tools. Of these 320 students, 132 did not successfully complete the course while the remaining 188 students completed the course.

2.4 Natural Language Processing Tools

We used several NLP tools to assess the linguistic features in the aggregated posts of sufficient length. These included the Writing Assessment Tool (WAT [9]), the Tool for the Automatic Analysis of Lexical Sophistication (TAALES [8]), and the Tool for the Automatic Assessment of Sentiment (TAAS). We provide a brief description of the indices reported by these tools below.

2.4.1 WAT

WAT was developed specifically to assess writing quality. As such, it includes a number of writing specific indices related to text structure (text length, sentence length, paragraph length), cohesion (e.g., local, global, and situational cohesion), lexical sophistication (e.g., word frequency, age of acquisition, word hypernymy, word meaningfulness), key word use, part of speech tags (adjectives, adverbs, cardinal numbers), syntactic complexity, and rhetorical features. It also reports on a number of writing quality algorithms such as introduction, body, and conclusion paragraph quality and the overall quality of an essay.

2.4.2 TAALES

TAALES incorporates about 150 indices related to basic lexical information (e.g., the number of tokens and types), lexical frequency, lexical range, psycholinguistic word information (e.g., concreteness, meaningfulness), and academic language for both single words and multi-word units (e.g., bigrams and trigrams).

2.4.3 TAAS

TAAS was developed specifically for this study. The tool incorporates a number of language-based sentiment analysis databases including the Linguistic Inquiry and Word Count database (LIWC [10]), Affective Norms for English Words (ANEW [1]), Geneva Affect Label Coder (GALC [13]), the National Research Council (NRC) Word-Emotion Association Lexicon [12], and the Senticnet database [2]. Using these databases, TAAS computes affective variables related to a number of emotions such as anger, amusement, fear, sadness, surprise, trust, pleasantness, attention, and sensitivity.

2.5 Statistical Analysis

The indices reported by WAT, TAALES, and TAAS that yielded non-normal distributions were removed. A multivariate analysis of variance (MANOVA) was conducted to examine which indices reported differences between the postings written by students who successfully completed the course and those who did not. The MANOVA was followed by stepwise discriminant function analysis (DFA) using the selected NLP indices that demonstrated significant differences between those students who completed the course and those who did not, and did not exhibit multicollinearity ($r > .90$) with other indices in the set. In the case of multicollinearity, 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 postings. This model was then used to predict group membership of the postings 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 WAT, TAALES, and TAAS as the dependent variables and the postings by students who completed the course and those who did not as the independent variables. A number of indices related to positing length, number of posts, use of numbers, writing quality, lexical sophistication, n-gram use, and cohesion demonstrated significant differences (see Table 1 for the MANOVA results). These indices were used in the subsequent DFA.

The results indicate that those who completed the course, even though course completion depended solely on success on technical assignments, tended to be better writers (i.e., received higher scores based on the essay score algorithm in WAT), to use a greater variety of words, to write more often with more words, and with greater cohesion. They also used more words relevant to the domain of the course, more concrete words, more sophisticated words, words with more associations to other words, and more common bigrams and trigrams.

3.2 Discriminant Function Analysis

A stepwise DFA using the indices selected through the MANOVA retained seven variables related to post length, lexical sophistication, the use of numbers, cohesion, and writing quality as significant predictors of whether a student received a certificate or not. These indices were *Average post lengths, Word age of acquisition, Cardinal numbers, Hypernymy standard deviation, Situational cohesion, Trigram frequency,* and *Essay score algorithm*. The remaining variables were removed as non-significant predictors.

Table 1. MANOVA Results Predicting Whether Students Completed the MOOC

Index	*F*	*η2*
Essay score algorithm	13.071**	0.039
Type token ratio	12.074**	0.037
Number of word types	11.371**	0.035
Number of posts	10.919*	0.033
Average post length	10.596*	0.032
Concreteness	10.017*	0.031
Cardinal numbers	10.081*	0.031
Trigram frequency	9.445*	0.029
Bigram frequency	8.903*	0.027
Number of sentences	8.451*	0.026
Frequency content words	8.219*	0.025
Situational cohesion	8.041*	0.025
Hypernymy standard deviation	7.643*	0.023
Word meaningfulness	7.378*	0.023
Lexical diversity	6.180*	0.019
Average word length	5.150*	0.016
Essay body quality algorithm	4.409*	0.014
Logical connectors	3.915*	0.012
Word age of acquisition	3.854*	0.012

** $p < .001$, * $p < .050$

The results demonstrate that the DFA using these seven indices correctly allocated 222 of the 320 posts in the total set, $\chi 2$ (df=1) = 46.529 $p < .001$, for an accuracy of 69.4%. For the leave-one-out cross-validation (LOOCV), the discriminant analysis allocated 217 of the 320 texts for an accuracy of 67.8% (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 0.379, demonstrating fair agreement.

Table 2. Confusion matrix for DFA classifying postings

		predicted		
	actual	- Cert	+Cert	F_1 score
Whole set	- Certificate	**91**	41	0.650
	+Certificate	57	**131**	0.728
LOOCV	- Certificate	**87**	45	0.628
	+Certificate	58	**130**	0.716

4. DISCUSSION AND CONCLUSION

Previous MOOC studies have investigated completion rates though click-stream data and sentiment analysis tools. The current study adds another tool for examining successful completion of a MOOC: natural language processing. The tools assessed in this study show that language related to forum post length, lexical sophistication, situational cohesion, cardinal numbers, trigram production, and writing quality can significantly predict whether a MOOC student completed an EDM course. Such a finding has important implications for how students' individual differences (in this case, language skills) that go beyond observed behaviors (i.e., click-stream data) can be used to predict success.

Overall, the results support the basic notion that students that demonstrate more advanced linguistic skills, produce more coherent text, and produce more content specific posts are more likely to complete the EDM MOOC. For instance, students were more likely to complete the course if their posts were shorter (i.e., more efficient), used words that are less frequent or familiar (i.e., higher age of acquisition scores), used more cardinal numbers (i.e., content specific), used words that were more consistent in

terms of specificity (i.e., less variance in terms of specificity), produced posts that were more cohesive (i.e., greater overlap of ideas), used more frequent trigrams (i.e., followed expected combinations of words), and produced writing samples of higher quality (i.e., samples scored as higher quality by a automatic essay scoring algorithm). Interestingly, none of our affective variables distinguished between students who completed or did not complete the EDM MOOC. This may be the result of the specific MOOC under investigation, a weakness of the affective variables examined, or a weakness of affective variables in general.

The findings have important practical implications as well. The linguistic model developed in this paper through the DFA could be used as a prototype to monitor MOOC students and potentially identify those students who are less likely to complete the course. Such students could then be target for interventions (e.g., sending e-mails, suggesting assignments or tutoring) to improve immediate engagement in the MOOC and promote long-term completion.

The results reported in this study are both significant and extendible to similar datasets (as reported in the LOOCV results). They also open up additional research avenues. For instance, to improve detection of students who might be unlikely to complete the MOOC, follow-up models that include click-stream data could be developed and tested. Such models would likely provide additive power to detection accuracy. One concern with the current model is that it requires language samples for analysis. This suggests that NLP approaches like this one may be even more useful in classes that have activities such as collaborative chat, a feature now emerging in some MOOCs.

5. ACKNOWLEDGMENTS

This research was supported in part by the Institute for Education Sciences and National Science Foundation (IES R305A080589, IES R305G20018-02, 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.

6. REFERENCES

[1] Bradley, M. M., and Lang, P. J. 1999. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. *Technical report. The Center for Research in Psychophysiology, University of Florida*.

[2] Cambria, E. and Hussain, A. 2015. *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Cham, Switzerland: Springer.

[3] Crossley, S. A. 2013. Advancing research in second language writing through computational tools and machine learning techniques: A research agenda. *Language Teaching, 46* (2), 256-271.

[4] DeBoer, J., Stump, G. S., Seaton, D., Ho, A., Pritchard, D. E., and Breslow, L. 2013. Bringing student backgrounds online: MOOC user demographics, site usage, and online learning. *In the Proceedings of the 6th International Conference on Educational Data Mining*, 312-313.

[5] DeBoer, J., Ho, A. D., Stump, G. S., & Breslow, L. 2014. Changing "Course": Reconceptualizing Educational Variables for Massive Open Online Courses. *Educational Researcher*, *March*, 74–84.

[6] Kizilcec, R. F., Piech, C., and Schneider, E. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. *In the Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 170-179.

[7] Koller, D., Ng, A., Do, C., and Chen, Z. 2013. Retention and Intention in Massive Open Online Courses. *Educause*.

[8] Kyle, K., and Crossley, S. A. in press. Automatically Assessing Lexical Sophistication: Indices, Tools, Findings, and Application. *TESOL Quarterly.*

[9] McNamara, D. S., Crossley, S. A., & Roscoe, R. 2013. Natural Language Processing in an Intelligent Writing Strategy Tutoring System. *Behavior Research Methods, 45* (2), 499-515.

[10] Pennebaker, J. W., Booth, R. J., and Francis, M. E. 2007. *LIWC2007: Linguistic inquiry and word count.* Austin, Texas.

[11] Ramesh, A., Goldwasser, D., Huang, B., Daume, H., and Getoor, L. 2014. Understanding MOOC Discussion Forums using Seeded LDA. *ACL Workshop on Innovative Use of NLP for Building Educational Applications,* 22-27.

[12] Saif, M., and Turney, P. 2013. Crowdsourcing a Word-Emotion Association Lexicon**,** *Computational Intelligence,* 29 (3), 436-465.

[13] Scherer, K. R. 2005. What are emotions? And how should they be measured? *Social Science Information*, 44 (4), 695-729.

[14] Seaton, D. T., Bergner, Y., Chuang, I., Mitros, P., & Pritchard, D. E. (2014). Who does what in a massive open online course? *Communications of the ACM*, *57*(4), 58–65.

[15] Wen, M., Yang, D. and Rose, C. P. 2014. Sentiment Analysis in MOOC Discussion Forums: What does it tell us? *In the Proceedings of the 7th International Conference on Educational Data Mining, 130-137*.

[16] Wen, M., Yang, D. and Rose, C. P. 2014. Linguistic Reflections of Student Engagement in Massive Open Online Courses. *In the Proceedings of the International Conference on Weblogs and Social Media*.

[17] Wang, Y. 2014. MOOC Leaner Motivation and Learning Pattern Discovery. *In the Proceedings of the 7th International Conference on Educational Data Mining*, 452-454.

A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance

Pedro Strecht
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
pstrecht@fe.up.pt

Luís Cruz
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
luiscruz@fe.up.pt

Carlos Soares
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
csoares@fe.up.pt

João Mendes-Moreira
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
jmoreira@fe.up.pt

Rui Abreu
INESC TEC
Rua Dr. Roberto Frias
4200-465 Porto, Portugal
rma@fe.up.pt

ABSTRACT

Predicting the success or failure of a student in a course or program is a problem that has recently been addressed using data mining techniques. In this paper we evaluate some of the most popular classification and regression algorithms on this problem. We address two problems: prediction of approval/failure and prediction of grade. The former is tackled as a classification task while the latter as a regression task. Separate models are trained for each course. The experiments were carried out using administrate data from the University of Porto, concerning approximately 700 courses. The algorithms with best results overall in classification were decision trees and SVM while in regression they were SVM, Random Forest, and AdaBoost.R2. However, in the classification setting, the algorithms are finding useful patterns, while, in regression, the models obtained are not able to beat a simple baseline.

Keywords

Regression, Classification, Academic Performance

1. INTRODUCTION

Recently, the University of Porto (UPorto) identified modelling of the success/failure of students in each course as one of its priorities. The goal is to use the models for two tasks: make predictions for the individual performance of students in courses and understand the factors associated with success and failure. These models are relevant to five levels of decision, namely: Course teacher, Program Director, Department Director, Faculty Director and University Rector. Course teachers and program directors can use the models to identify students at risk and devise strategies that can reduce the risk of failure. Also, program directors as well as department directors can find them useful in designing program syllabus. Finally, the top levels of university management can use these models to understand general trends and behaviours in student performance, which can lead to new or adapted pedagogical strategies.

The fact that models are needed for different levels of decision requires that these models have different granularities. In other words, course teachers and program directors are able to work with a few or a few dozen models, respectively. However, the other levels of management would have to deal with hundreds, maybe even thousands of models, which is not feasible. On the other hand, each course presents different particularities which makes the creation of a unique model to predict academic success for all the courses, an extremely hard task. Such a model would have to aggregate the different factors that influence success in very different courses. Therefore, we train a model separately for each course.

So far, the results obtained and the domain-specific constraints provide a satisfactory justification for the choice of decision trees. However, there is a need to understand the impact of this choice in the predictive accuracy of the algorithms, namely when compared with others. Additionally, although the problem of predicting if a student will pass or fail (classification task) is relevant for all levels of management of the university, the related problem of predicting the actual grade (regression task) may provide additional useful information. Therefore, this study also considers a comparative analysis of different regression algorithms. This comparison will also address the question of whether the features that are useful for classification are equally useful for regression.

The main contributions of this paper are: 1) to compare the predictive accuracy of different algorithms on the problems of predicting the performance of students in both classification (predicting success/failure) and regression (predicting the grade) tasks, particularly when comparing with decision trees, which have some other properties that deem

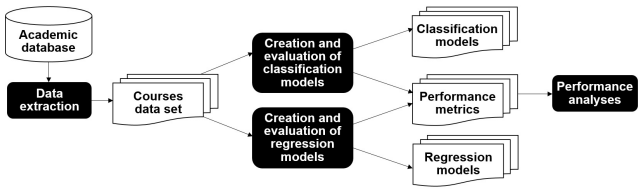


Figure 1: Experimental Setup

them suitable for this problem; 2) to assess whether the features which have obtained positive results in the classification task, and that represent essentially administrative information, are also useful to predict the grades.

The remainder of this paper is structured as follows. Section 2 presents related work. Section 3 describes the experimental set-up and methodology for both classification and regression models. Section 4 presents the results followed by section 5 with the conclusions and future work.

2. RELATED WORK

Predicting students' performance has been an issue studied previously in educational data mining research in the context of student attrition [24, 23]. Minaei-Bidgoli [13] used a combination of multiple classifiers to predict their final grade based on features extracted from logged data in an education webbased system.

Pittman [15] performed a study to explore the effectiveness of data mining methods to identify students who are at risk of leaving a particular institution. Romero et al. [16] focused on comparing different data mining methods and techniques for classifying students based on their Moodle (e-learning system) usage data and the final marks obtained in their respective programmes. The conclusion was that the most appropriate algorithm was decision trees for being accurate and comprehensible for instructors. Kabakchieva [10] also developed models for predicting student performance, based on their personal, pre-university and university performance characteristics. The highest accuracy is achieved with the neural network model, followed by the decision tree model and the kNN model.

Strecht, Mendes-Moreira and Soares [20] work predicted the failure of students in university courses using an approach to group and merge interpretable models in order to replace them with more general ones. The results show that merging models grouped by scientific areas yields an improvement in prediction quality.

3. METHODOLOGY

To carry out the experiments, a system with four processes was developed following the architecture presented in Figure 1. The first process creates the data sets (one for each course in the university) from the academic database, containing enrolment data. The courses data set were then used by two processes to create classification and regression models for each course using various algorithms. These models were evaluated using suitable performance metrics (different for classification and regression) that are collected to allow analyses and comparison by the final process.

3.1 Data Extraction

This process extracts data sets from the academic database of the university information system. The analysis done focuses on the academic year 2012/2013. A total of 5779 course data sets were extracted (from 391 programmes). The variables used were: age, sex, marital status, nationality, displaced (whether the student lived outside the Porto district), scholarship, special needs, type of admission, type of student (regular, mobility, extraordinary), status of student (ordinary, employed, athlete, ...), years of enrolment, delayed courses, type of dedication (full-time, part-time), and debt situation. The target variables are approval for classification and final grade for regression.

The final grade in these data sets is stored as a numerical value between 0 and 20. However, there are some special cases in which the grade is given as an acronym (e.g, RA means fail because of dropout), which is not feasible for regression. In such cases, in which a student failed, we converted the grade to 0.

3.2 Creation and evaluation of models

Two processes trained a set of models for classification and regression respectively for each course using different algorithms. For classification we have used k-Nearest Neighbors (kNN) [9], Random Forest (RF) [2], AdaBoost (AB) [7], Classification and Regression Trees (CART) [3], Support Vector Machines [21], Naïve Bayes (NB) [12] and for regression we used Ordinary Least Squares (OLS) [18], SVM, CART, kNN, Random Forest, and AdaBoost.R2 (AB.R2) [8].

This selection of algorithms was based on the most used algorithms for general data mining problems [22]. In this set of experiments a standard values of parameters was used. As baseline in classification we defined a model which always predicts failure. For regression, the baseline model predicts the average grade of the training set of a given course.

Models were evaluated using the k-fold cross-validation method [19] with stratified sampling [11]. The distribution of positive and negative instances is not balanced, thus it is necessary to ensure that the distribution of students in each fold respect these proportions. Failure is the positive class in this problem and we used F1 score for evaluation [5]. All regression models used 10-fold cross validation and the Root Mean Squared Error (RMSE) as evaluation measure [4].

Training and evaluation of models was replicated for each course. Courses with less then 100 students were skipped. This resulted in around 700 models for each algorithm in both classification and regression.

3.3 Performance Analyses

In both classification and regression, the algorithms were compared by placing box plots side by side relating to F1 and RMSE respectively. To get a better perspective of the distribution of results, violin plots are presented together with the box plots. The longest horizontal lines inside the boxes refer to the median while the shortest refer to the average. A few descriptive statistics were also collected and presented in tables.

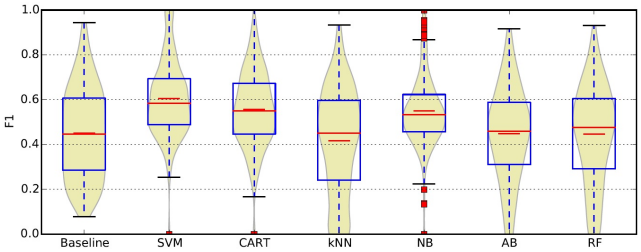


Figure 2: F1 score for each classification algorithm

In order to statistically validate the results obtained in the experiments we have used the Friedman test as suggested by Demšar to compare multiple classifiers [6]. We have used the typical value of 12 groups of models often referred as data sets in this context.

4. RESULTS

This section presents the results obtained by running experiments to train models for both classification and regression.

4.1 Classification

Figure 2 presents the F1 score distribution of models across algorithms. Table 1 presents some basic statistics about the results. Algorithms are ranked by descending order of values of the average and standard deviation of F1 scores.

The first fact that stands out from Figure 2 is that none of the algorithms present exceptional results. Albeit this, some of them seem to systematically outperform the baseline, namely SVM, CART and NB.

Table 1 confirms that SVM is the algorithm with the best performance, clearly outperforming the baseline. Not only it provides the highest average F1 score, 0.60±0.17, but sometimes it also achieves a maximum F1 score of 1.0, while the maximum score of the baseline is 0.94. Finally, although the minimum score is lower than the baseline's (0 vs. 0.08), the standard deviation is lower (0.17 vs. 0.20) which indicates that overall, it obtains more robust results.

Similar observations can be made for CART and NB. The performance of RF and AB is very similar to that of the baseline, while kNN is worse. The results of Random Forest, in particular, are surprising as this algorithm usually exhibits a very competitive performance [17].

In spite of the showing some systematic differences, the results are, overall, not very different. This is confirmed by the results of the Friedman test, $\chi^2(6) = 2.6071, p = 0.8563$, as the p-value is very high.

4.2 Regression

Figure 3 presents the distribution of RMSE values of models obtained by the algorithms. Table 2 presents some basic statistics about the results. The algorithms are ranked by ascending order of RMSE values.

As in classification, it is also quite straightforward that none of the algorithms present exceptional results. Also in this case, there is one algorithm which performs clearly worse than the baseline, CART (Table 2). Unlike classification, all violin plots show exactly the same shape, i.e., equally sized upper and lower tails. Therefore, differences are more related to overall performance (i.e. location). This shows that to compare models it is enough to consider the average and standard deviation.

The differences in performance are even smaller than in classification. However, Table 2 suggests that SVM was the best algorithm with an average of 4.65±1.19, but the standard deviation is quite large (1.19) taking into account the RMSE of the baseline (4.92). These observations are confirmed by the Friedman test ($\chi^2(6) = 3.3697, p = 0.7612$). In the case of regression, the value of the RMSE is interpretable, as it is in the same scale as the target variable. All algorithms obtain an error around 5, which is very high according to the scale (0 to 20).

In light of the results obtained in the classification setting, this is somewhat surprising, since the independent variables are the same and many of the algorithms used are based on the same principles.[1] Further analysis of the results is necessary to understand them and to identify possibilities to improve the results.

[1] Although this must be interpreted carefully as it is arguable to say that, for instance, SVM for classification and regression are the same algorithm.

Table 1: Classification models results (F1)

Rank	Algorithm	Avg	Std Dev	Max	Min
1	SVM	0.60	0.17	1.00	0.00
2	CART	0.56	0.17	1.00	0.00
3	NB	0.55	0.16	1.00	0.00
4	RF	0.45	0.22	0.93	0.00
5	AB	0.45	0.21	0.92	0.00
6	Baseline	0.45	0.20	0.94	0.08
7	kNN	0.42	0.24	0.93	0.00

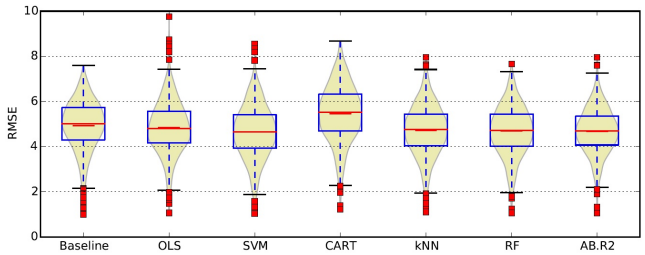


Figure 3: RMSE for each regression algorithm

Table 2: Regression models results (RMSE)

Rank	Algorithm	Avg	Std Dev	Max	Min
1	SVM	4.65	1.19	8.54	1.03
2	RF	4.69	1.10	7.66	1.06
3	AB.R2	4.69	1.02	7.96	1.07
4	kNN	4.72	1.12	7.96	1.10
5	Baseline	4.92	1.11	7.59	1.00
6	OLS	4.84	1.19	9.75	1.06
7	CART	5.46	1.26	8.68	1.22

5. CONCLUSIONS

Positive results were obtained on the classification approach where the goal is to predict whether a student will pass or fail a course. Surprisingly, however, the results on the regression approach, where the goal is to predict the grade of the student in a course, were bad. Additionally, we found no statistical evidence that the differences in performance between the algorithms are significant, although some trends are observed. Further analysis is necessary to better understand these results, which could lead to ideas for improvement. As a complement of the problems studied in this work, it should be interesting to predict an interval for a grade [1].

Some algorithms are more sensitive to parameter tuning than others. Thus it is not guaranteed that they ran with the best configuration. As future work, some optimisation could be made using an automate tuning methodology. In addition, feature selection and feature weighting can be carried out which has proven to yield good results in educational data [14].

Although the feature set used in the experiments provided some interesting results in classification, the same did not happen in regression. Thus, new features could be added. Features related to academic goals, personal interests, time management skills, sports activities, sleep habits, etc. are worthwhile investigating.

6. ACKNOWLEDGMENTS

This work is funded by projects "NORTE-07-0124-FEDER-000059" and "NORTE-07-0124-FEDER-000057", financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

7. REFERENCES

[1] R. Asif, A. Merceron, and M. Pathan. Predicting student academic performance at degree level: A case study. 2015.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[4] T. Chai and R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? *Geoscientific Model Development Discussions*, 7:1525–1534, 2014.

[5] N. Chinchor. MUC-4 Evaluation Metrics. In *Proceedings of the 4th Message Understanding Conference (MUC4 '92)*, pages 22–29. Association for Computational Linguistics, 1992.

[6] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[7] T. G. Dietterich. Machine-learning research: four current directions. *AI magazine*, 18(4):97, 1997.

[8] H. Drucker. Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115, 1997.

[9] E. Fix and J. L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.

[10] D. Kabakchieva. Predicting Student Performance by Using Data Mining Methods for Classification. *Cybernetics and Information Technologies*, 13(1):61–72, 2013.

[11] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Conference on AI (IJCAI)*, pages 1137–1145, San Mateo, CA, 1995. Morgan Kaufmann.

[12] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, volume 33, pages 81–93, 1994.

[13] B. Minaei-Bidgoli. Predicting student performance: an application of data mining methods with an educational web-based system. In *33rd ASEE/IEEE Frontiers in Education Conference*, pages 1–6, 2003.

[14] B. Minaei-Bidgoli and W. F. Punch. Using genetic algorithms for data mining optimization in an educational web-based system. In *Genetic and Evolutionary Computation—GECCO 2003*, pages 2252–2263. Springer, 2003.

[15] K. Pittman. *Comparison of data mining techniques used to predict student retention*. PhD thesis, Nova Southeastern University, 2008.

[16] C. Romero. Data mining algorithms to classify students. In *1st International Educational Data Mining Conference (EDM08)*, 2008.

[17] M. R. Segal. Machine learning benchmarks and random forest regression. *Center for Bioinformatics & Molecular Biostatistics*, 2004.

[18] S. M. Stigler. Gauss and the invention of least squares. *The Annals of Statistics*, pages 465–474, 1981.

[19] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B*, 36(2):111–147, 1974.

[20] P. Strecht, J. Mendes-Moreira, and C. Soares. Merging Decision Trees: A Case Study in Predicting Student Performance. In X. Luo, J. Yu, and Z. Li, editors, *Advanced Data Mining and Applications*, Lecture Notes in Computer Science, pages 535–548. Springer International Publishing, 2014.

[21] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

[22] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[23] A. Zafra and S. Ventura. Predicting student grades in learning management systems with multiple instance genetic programming. *International Working Group on Educational Data Mining*, 2009.

[24] J. Zimmermann, K. H. Brodersen, J.-P. Pellet, E. August, and J. M. Buhmann. Predicting graduate-level performance from undergraduate achievements. In *4th International Educational Data Mining Conference (EDM11)*, pages 357–358, 2011.

Predicting Student Grade based on Free-style Comments using Word2Vec and ANN by Considering Prediction Results Obtained in Consecutive Lessons

Jingyi Luo
Graduate School of Information Science and Electrical Engineering,Kyushu University
Fukuoka,Japan

Shaymaa E.Sorour
Graduate School of Information Science and Electrical Engineering,Kyushu University
Fukuoka,Japan

Kazumasa Goda
Kyushu Institute of Information Science
Dazaifu,Japan

Tsunenori Mine
Faculty of Information Science and Electrical Engineering,Kyushu University
Fukuoka,Japan

ABSTRACT

Continuously tracking students during a whole semester plays a vital role to enable a teacher to grasp their learning situation, attitude and motivation. It also helps to give correct assessment and useful feedback to them. To this end, we ask students to write their comments just after each lesson, because student comments reflect their learning attitude towards the lesson, understanding of course contents, and difficulties of learning. In this paper, we propose a new method to predict final student grades. The method employs Word2Vec and Artificial Neural Network (ANN) to predict student grade in each lesson based on their comments freely written just after the lesson. In addition, we apply a window function to the predicted results obtained in consecutive lessons to keep track of each student's learning situation. The experiment results show that the prediction correct rate reached 80% by considering the predicted student grades from six consecutive lessons, and a final rate became 94% from all 15 lessons. The results illustrate that our proposed method continuously tracked student learning situation and improved prediction performance of final student grades as the lessons go by.

Keywords

PCN Method, Word2Vec, ANN, Comment Mining, Grade Prediction

1. INTRODUCTION

Learner performance assessment is a continuous and an integral part of the learning process [4]. During studying, exams are used to help teachers know how good students are learning, as well as to help them find out the difficulties with the course. However preparing a good exam is a laborious and resource demanding work, so it's still hard to obtain assessment by exams over all periods of a semester.

Thus, in the past four decades, researchers have been working on predicting individual or group performance in courses for getting assessments. By accurate predictions, we can detect students who have difficulties with the courses early, and help them improve [1].

To control students' learning behavior and situations, previous studies have used various regular assessment methods, such as e-learning logs, test marks and questionnaires. The current study proposes a new method to predict student grades. Our method is based on students' free-style comments collected after each lesson.

K.Goda, S.Hirokawa, and T.Mine [3] [2]proposed the PCN method to estimate student learning situations from free-style comments written by the students. The PCN method categorizes the comments into three items: P (Previous activity), C (Current activity), and N (Next activity).

In this paper, we apply the Word2Vec method to the comments data to get a vector representation of each comment. Then we use an artificial neural network (ANN) model to predict student grades based on the vectors. The experiments were conducted to validate the proposed methods by calculating the F-measure and accuracy for each lesson. After acquiring a prediction result for each lesson, we applied a window function and a majority vote method to get a final prediction result based on multiple lessons. The experiment results illustrate that the prediction correct rate reached 80% by considering the predicted student grades obtained from six lessons, and the final rate became 94% from all 15 lessons.

Contributions of this paper are threefold. First, we propose a new method to predict final student grades by using Word2Vec and ANN. Second, we improve the prediction performance by considering the results obtained in consecutive lessons. We show as the size of the lessons increases, the prediction performance becomes better. Third, we conduct experiments to illustrate the effectiveness of the proposed methods. The experiment results show the validity of the proposed methods.

2. RELATED WORK

Extensive literature reviews of the Educational Data Mining (EDM) research field are mainly focused on retention of students, improving institutional effectiveness, enrollment management and alumni management. In the past four decades, a considerable amount of research has gone into predicting individual or group success in exams and courses.

Schoor and Bannert [7] studied sequences of social regulatory processes (i.e. individual and collaborative activities of analyzing, planning...aspects) during collaborative sessions

and their relationship to group performance. They used process mining to identify process patterns for high versus low group performance dyads. The result models showed that there were clear parallels between high and low achieving dyads in a double loop of working on the task, monitoring, and coordinating.

Liu and Xing [5] aimed to develop a predictive model of student behavior by an ensemble approach composed of creation of sampled sets, generation of base models, and selection of base models to be aggregated for obtaining the final ensemble model. The solution required less computation resource, had satisfying prediction performance and produced prediction models with good capability of generalization.

Different from the above studies, Goda et al. [3] proposed the PCN method to estimate students' learning situations with their free-style comments written just after a lesson. They applied Support Vector Machine (SVM) to the comments for predicting final student results in 5 grades. The experiment results illustrate that as student comments get higher PCN scores, prediction performance of student grades becomes better. Sorour et al.[8] applied machine learning technique: artificial neural network (ANN) and made it learn the relationships between comments data analyzed by Latent semantic analysis(LSA) and the final student grades. They constructed a network model to each lesson. The average prediction accuracy of student final grades was 82.6%. In this study, as an extension of Sorour et al. [8], we focused on using different text mining method Word2Vec combined with the ANN model to get prediction on each lesson, and obtain prediction results based on consecutive multiple lessons. Our method outperformed the method of Sorour et al.[8].

3. METHODOLOGY

3.1 Collecting Comments

In this research, we used the same comment data as Sorour et al.[8]. The comments were collected after each lesson in a course including 15 lessons. 123 students attended this course. They were asked to fill in three simple questionnaire items about their learning status. Goda et al. [3] called the three items, P (Previous), C (Current) and N (Next) items. In this paper, we mainly focus on the C (Current) comments. Table 1 displays the real number of comments in each lesson that we analyzed. On average there is 111.13 comments in each lesson.

Table 1: Number of comments for each lesson

Lesson	Num	Lesson	Num	Lesson	Num
1	100	6	116	11	107
2	121	7	104	12	109
3	118	8	103	13	107
4	115	9	107	14	111
5	123	10	111	15	121

3.2 Comments Data Preparation

3.2.1 Comments Data Preprocessing

This step covers all the preparations required for constructing the final dataset from the initial data. Our method used a Japanese morphological analyzer Mecab[1] to analyze C comments, extract words and part of speech. In this experiment, we only used noun, verb, adjective and adverb. The number of words appeared in the comments is about 1400 in each lesson, and the number of words in all the comments without duplication is over 430 in each lesson.

3.2.2 Word2Vec

Word2vec is a popular neural network based approach to learning distributed vector representations for words released by Google in 2013. This tool adopts two main model architectures, Continuous Bag-of-Words (CBOW) and Skip-Gram[6].

3.3 Training Phase

After the previous step and before we applied ANN to train the data, we have some pretreatments for preparing training data for ANN.

We have got a list of vocabularies and their corresponding vectors after the previous step. Now we need to find out all the words one student have used in his/ her comment which existed in the vocabulary list, and add the vectors indicating these words up to get a final vector for that student.

After obtaining a list of vectors for each student, we need to proceed the training phase with the list. In this research, we used a three-layered Artificial Neural Network to estimate student grades. In our work, we used FANN Libraries[2] to build our network model. We took the results from the former step and put them into the input layer of ANN. For all the lessons, we applied the same model with 0.1 learning rate and 0.3 momentum.

3.4 Test Phase

To predict student grades, we used 5 grade categories instead of real marks to classify final student marks.

Table 2: 5 Grades Categories

Real Marks	Grades	Num of Students
≥ 90	S	21
80-89	A	41
70-79	B	23
60-69	C	17
≤ 60	D	21

Since in each lesson, there exist some students who did not fill in questionnaires, we can't predict their grade. In these cases, we treat them as grade D instead.

After training the ANN model, we proceed the test phase to get prediction results of final student grades in each lesson. In the test phase, we evaluated prediction performance (Accuracy, F-measure) by 10-fold cross validation. We separated comments data by using 90% as training data and the rest 10% as test data. The procedure was repeated 10 times and the results were averaged. Afterwards, we apply an window function and the majority vote method to obtain a continuous prediction. The details of the window function and the majority vote method will be described in Section 4.1.

[1] http://sourceforge.net/projects/mecab/

[2] http://leenissen.dk/fann/wp/

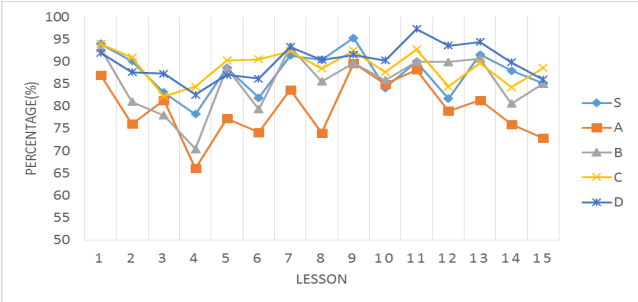


Figure 1: Accuracy for different grades

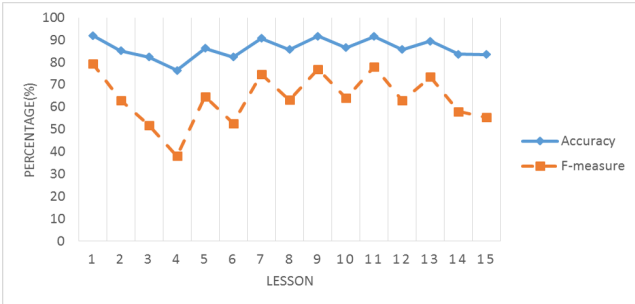


Figure 2: Average accuracy and F-measure of all the grades in each lesson

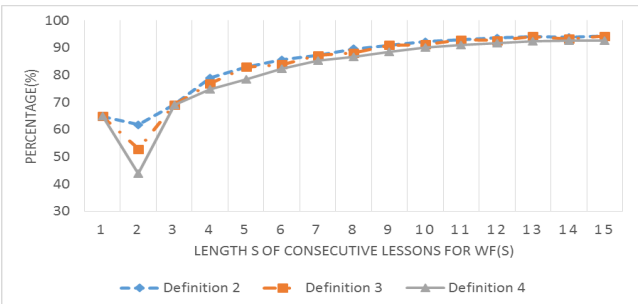


Figure 3: Average TP rate based on different definitions

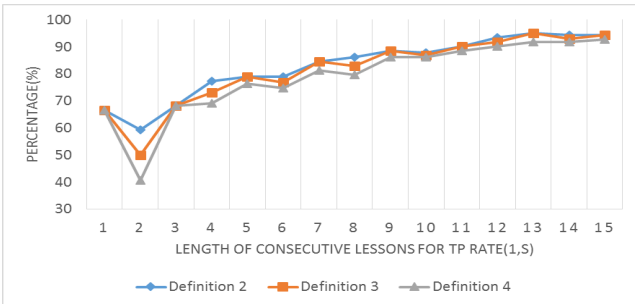


Figure 4: TP rate for different length of consecutive lessons from lesson 1

4. PREDICTION PERFORMANCE

4.1 Measure of Prediction Performance

We define the majority vote method and the window function as follows:

Let G be a set of grades $\{g_0, g_1, g_2, g_3, g_4\}$; each element of G corresponds to each grade, i.e., g_0, g_1, g_2, g_3, and g_4 correspond to S, A, B, C, and D, respectively. Let $MV_k(m,n)$ be the function of Majority Vote of student k from lessons m to n . $MV_k(m,n)$ returns a set of predicted student k's grades whose occurrence frequency from lessons m to n became the greatest. We define $MV_k(m,n)$ in Definition 1.

Definition 1. $MV_k(m,n)$

$$MV_k(m,n) = \underset{g_i \in G}{\operatorname{argmax}}\ f(k,g_i)(m,n)$$

where $f(k,g_i)(m,n)$ returns the occurrence frequency of predicted grade g_i of student k from lessons m to n.

For example, if the predicted grades of student 1 from lessons 1 to 3 are respectively S ($=g_0$), A ($=g_1$), and S ($=g_0$), then $f(1,g_0)(1,3) = 2$ and $f(1,g_1)(1,3) = 1$. So, $MV_1(1,3)$ returns $\{g_0\}$. If the predicted grades of student 1 from lessons 1 to 3 are respectively S($=g_0$), A($=g_1$), B($=g_2$), then $f(1,g_0)(1,3) = 1$, $f(1,g_1)(1,3) = 1$, and $f(1,g_2)(1,3) = 1$. So, $MV_1(1,3)$ returns $\{g_0, g_1, g_2\}$.

Function δ returns a score according to the results returned by a Majority Vote function $MV(m,n)$ defined in Definition 1. Three δ functions: δ_1, δ_2, and δ_3, are defined in Definitions 2, 3, and 4. Here we use the notation $|.|$ that denotes the cardinality of a set. For example, if $MV_1(1,3)$ returns $\{g_0, g_1, g_2\}$, then $|MV_1(1,3)| = 3$.

Definition 2. δ_1

$\delta_1(MV_k(m,n))$ returns 1 if g_k is the actual grade of student k, $g_k \in MV_k(m,n)$ and $g_l \notin MV_k(m,n)$ such that $|l-k| > 1$, 0 otherwise.

For example, we assume that the actual grade of student k is g_0, if $MV_k(m,n) = \{g_0, g_1\}$, then $\delta_1(MV_k(m,n)) = 1$. If $MV_k(m,n) = \{g_0, g_2\}$ then $\delta_1(MV_k(m,n)) = 0$, because $|2-0| > 1$.

Definition 3. δ_2

$\delta_2(MV_k(m,n))$ returns $\frac{1}{|MV_k(m,n)|}$ if $g_k \in MV_k(m,n)$ where g_k is the actual grade of student k, 0 otherwise.

Definition 4. δ_3

$\delta_3(MV_k(m,n))$ returns 1 if $g_k \in MV_k(m,n)$ and $|MV_k(m,n)| = 1$, 0 otherwise.

Next, we define $TP(m,n)$ that returns True Positive (TP) rate from lessons m to n in Defition 5.

Definition 5. $TP(m,n)$

$$TP(m,n) = \frac{\sum_{k=1}^{N_s} \delta(MV_k(m,n))}{N_s}$$

where N_s is the number of students.

Now we define function $WF(s)$, which returns the average TP rate in s consecutive lessons, in Defition 6. Here s denotes the length of consecutive lessons, i.e. the number of lessons.

Definition 6. $WF(s)$

$$WF(s) = \frac{\sum_{k=1}^{N-s+1} TP(k,k+s-1)}{N-s+1}$$

where N is the number of all lessons in a course, 15 in this research.

For example, when $N = 15$, $WF(1)$ to $WF(15)$ are computed as follows:

$$WF(1) = \frac{TP(1,1) + TP(2,2) + .. + TP(15,15)}{15}$$
$$WF(2) = \frac{TP(1,2) + TP(2,3) + .. + TP(14,15)}{14}$$
$$WF(3) = \frac{TP(1,3) + TP(2,4) + .. + TP(13,15)}{13}$$
$$\cdots$$
$$WF(14) = \frac{TP(1,14) + TP(2,15)}{2}$$
$$WF(15) = \frac{TP(1,15)}{1} = TP(1,15)$$

4.2 Results in Each Lesson

We examined the same model on all the students with different final grades. Results are shown in Figures 1 and 2. Figure 1 displays the plot of accuracy results of students with different grades in each lesson. Table 3 shows the average overall prediction accuracy and F-measure for the different grades. As for accuracy, the result of grade D is the highest, which scores 89.5%, and the lowest average is grade A, which scores 79.1%. Also, according to Figure 2, lesson 1 has the highest accuracy and F-measure, while lesson 4 has the lowest results.

Table 3: Average accuracy and F-measure for different grades

Grades	Accuracy	F-measure
S	87.3	65.6
A	79.1	**71.3**
B	85.0	62.6
C	88.5	57.2
D	**89.5**	62.3
Average of all grades	85.9	63.8

4.3 Results after Using Window Function and Majority Vote

Before we apply the window function to all the consecutive lessons, we first treat all the students who did not describe comments as Grade D. After this step, it also ensures that for each lesson, every student has one predicted grade. After we get the prediction result in each lesson, we apply the window function and the majority vote method to get a continuous track of student performance.

Here, we only consider TP rates. First we investigated the effect of size s of $WF(s)$ by varying the value of s from 1 to 15. As we can see, in Figure 3, the TP rate was increased as the value of s increased. As an example of the results, even though the strictest way of counting the correct case by Definition 4, the correct rate still raised over 80% after considering more than six lessons. In addition, with all the lessons, the correct rates all reached over 90%. And with Definition 2 and 3, they both reached 94%. The results by Definition 4 reached 92.7%.

Figure 4 shows the result of TP rate from TP(1, 1) ,TP(1, 2),TP(1, 3) to TP(1, 15) with three different definitions. With the growing of window function size, the TP rate raised over 80% with more than 7 lessons, which is slightly lower than the average.

Considering the results of Figures 3 and 4, we can say the both results took similar tendency that the TP rates became greater as the size of lessons increased.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the prediction method of student grade based on the C comments data from Goda et al. [3]. We applied the Word2Vec and ANN methods to the comments to obtain prediction of their grades in each lesson. Then we used the window function and the majority vote method to improve the prediction results based on consecutive multiple lessons. The experiment results illustrate the validity of the proposed method.

This study expressed the correlation between self-evaluation descriptive sentences written by students and their academic performance by predicting their grade. Especially when using prediction results obtained in consecutive lessons, the prediction result has quite high credibility. This could help giving feedback to students during the semester to help students achieve higher motivation and know their learning conditions better.

However, there still remain some room for improving prediction results in each lesson. In the future, we will try to apply better models to achieve higher accuracy in predicting student grades.

6. ACKNOWLEDGMENTS

This work was partially supported by JSPS KAKENHI Grant Numbers 25350311 and 26540183.

7. REFERENCES

[1] M. Fire, G. Katz, Y. Elovici, B. Shapira, and L. Rokach. Predicting student exam's scores by analyzing social network data. Lecture Notes in Computer Science Volume 7669, pp 584-595, ,2012.

[2] K.Goda, S.Hirokawa, and T.Mine. Correlation of grade prediction performance and validity of self evaluation comments. In Proc. of the 14th annual ACM SIGITE conference on Information technology education, Florida, USA, 2013.

[3] K.Goda and T.Mine. Analysis of student' learning activities through qualifying time-series comments. Proc. of the KES 2011, Part 2, LNAI 6882, Springer-Verlag Berlin Heidelberg, pp.154-164, 2011.

[4] L.Earl. Assessment of Learning, for Learning, and as Learning. Thouand Oaks,CA,Corwin Press, 2003.

[5] K. Liu and Y. Xing. A lightweight solution to the educational data mining challenge. In Proceedings of the KDD 2010 cup 2010 workshop knowledge discovery in educational data(pp. 76-82), 2010.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. Advances in Neural Information Processing Systems (NIPS), 2013.

[7] C. Schoor and M. Bannert. Exploring regulatory processes during a computer-supported collaborative learning task using process mining. Computers in Human Behavior, 2012.

[8] S. E. Sorour, T. Mine, K. G., and S. Hirokawa. Predictive model to evaluate student performance. In Journal of Information Processing (JIP), Vol. 23, 2015.

Learning the Creative Potential of Students by Mining a Word Association Task

Cristian Olivares-Rodríguez
Engineering Faculty, Universidad Andres Bello
Avenida República 237
Santiago 8370146, Chile
colivares@unab.cl

Mariluz Guenaga
DeustoTech - Deusto Institute of Technology -
University of Deusto
Avenida Universidades 24
Bilbao 48007, Spain
mlguenaga@deusto.es

ABSTRACT

Creativity is a relevant skill for human beings in order to overcome complex problems and reach novel solutions based on unexpected associations of concepts. Thus, the education of creativity becomes relevant, but there are not tools to automatically track the creative potential of learners over time. This work provides a novel set of behavioural features about creativity based on associative skills. These associations are processed to define two models that depict students' creative potential. This way, we have reached an acceptable accuracy rate in the classification of creative potential, hence we have found concrete evidence regarding the ability to automatically predict the creative potential of students based on their association capabitilies.

1. INTRODUCTION

Creativity generally emerges when people face a problematic or new situation, where constraints and concepts are probably unknown. An intensive search of novel solutions is required to solve real problems. This search can be done by exploration, transformation or combination of concepts. Therefore, there is a need of new associations of concepts to reach unknown solutions [7].

Nowadays, students are the centre of their learning when solving authentic problems, and creative skills provide students adaptation abilities to overcome heuristic environment, where nor the path to the solution nor the solution are known and therefore, you need to establish strategies to achieve the goal. In this context, the intensive use of technology by students to produce web searches and social data is a rich source of information to learn how students behave [2], thus a monitoring framework of creativity, based on associative features, becomes feasible.

A set of creative challenges have been applied to 64 students of sixth grade in two primary schools in Spain. First, we have applied an "unusual uses" test as a measure of creativity. After that, we have applied two "word association" tasks in order to depict their associative skills over time, similarly to [4]. Based on data captured from these activities we have extracted a set of relevant features regarding to creativity in order to model the user behaviour.

Our hypothesis bases on the fact that the local frequency of words and the time when they came out provide relevant information about originality. Also, we set that time provides a measure about fluency and that part of speech gives information about flexibility.

We have tested the strength of features associated to creativity with a supervised classification approach. We propose a model to track the creative potential of students based on their associative skills, but it still requires a more powerful set of semantic features and a learning algorithm that works properly with sequential data. However, the developed model provides an acceptable accuracy rate (over 81% in the best case) and outperform a Bag-of-Words approach.

(Sec-2) describes the concept of creativity and provides formal definitions of existing models (Sec-3). (Sec-4) defines the experiment carried out to collect data and evaluates the predictability of proposed models. Then, we present the main results using these models (Sec-5) and provide a discussion regarding the monitoring of creativity (Sec-6). Finally, we summarize our contributions and outline future works (Sec-7).

2. BACKGROUND

Creativity is a mental process based on associations in our mind and it has been characterized by: fluency, flexibility, originality and elaboration. The conceptual model of creativity of Amabile [1] defines a general process to solve problems that is grouped in three phases: a *conceptualization* phase to establish several problem definitions; a *search* phase to reach concepts, make new associations and establish new solutions; and a *development* phase to implement a solution and to update the knowledge. The model also introduces relevant skills related to creativity: associative and executive, which have been studied against the creative performance [4].

Mednick [7] proposed an associative theory of creativity where affirms that a creative person is able to find new solutions to real problems making as many associations as possible

(fluency), as diverse as possible (flexibility) and as unexpected as possible (originality). Benedek et al. have found a positive relation between fluency and creativity [4].

There are some procedures to measure the creativity [8] [4]. Generally, it is measured by an *unusual uses* test, where each participant must achieve as many uses as possible for a particular object (e.g. a brick) in a short period of time, and the expert evaluation of creativity is based on a Likert-scale of fluency, flexibility and originality. This methods do not include measures of the creative potential obtained during data from the process to carry out the activity (i.e. search on the Web, social networks, etc.) and, thus there is an opportunity to model the creativity to implicitly depict students' behaviour.

A *word association* task makes visible the association skills by retrieving as many words as possible with respect to a query word, in a short period of time. This task is an heuristic process of word retrieval, where a user defines an association model Q^u in order to provide words w_i, in a certain time t_i and related to a query word q^s (Eq-1).

$$Q^u(q^s) = [(w_1, t_1), (w_2, t_2), \ldots, (w_n, t_n)] \forall w_i \in W^u \quad (1)$$

Moreover, each user defines an heuristic measure h_u^* based on a hidden similarity measure S^u (Eq-2).

$$h_u^*(q^s, w) = S^u(q_j^s, w|t) \mid u \in U \quad (2)$$

Even heuristic is hidden, we can derive an empirical model through a set of association features [4].

2.1 Computational Models of Creativity

In art, a search behaviour analysis based on a visual creative task has been developed to figure out the hidden process [5]. A computerized aesthetic composition task was implemented in order to capture the search flow followed by each participant to design a new image. Thus, the user actions are used to depict the heuristic applied in the search process.

In education of creativity, a personalized creativity learning system (PCLS) based on decision trees has been proposed [6]. Nevertheless, they are not focused in track the creativity, but in enhance the process to teach creativity. The purpose of the PCLS is to adapt the student path based on a set of creativity measures and demographic information (gender, college, etc.).

3. MODEL OF WORD ASSOCIATIONS

We have defined two user models: the Bag-of-words (U_{BoG}) and the Features of Creativity (U_{FoC}).

3.1 Bag-of-Words for Association Tasks

The bag-of-words (BoW) is a basic model to describe the content of documents in the information retrieval domain (Eq-3).

$$BoG(d_j) = (f(w_1), f(w_2), \ldots f(w_n)) \forall w_i \in W \quad (3)$$

Where d_j is a document, $f(w_i)$ is a function that defines the relation of the word w_i with the document d_j and w_i is in the dictionary W. This model provides a measure of the originality of words. A document d_j can be defined as the set of all associated words that users have provided against a query word q^s (Eq-4).

$$d_j(q^s) = \bigcup_{k=1}^{|U|} Q^k(q^s) \quad (4)$$

And, the word frequency wf (Eq-5) is defined as an originality measure of the word regarding each document and a relative time measure [3].

$$wf(w_i) = \sum_{j=1}^{|D|} \left[\frac{f(w_i, d_j)}{\max\{f(w, d_j)|w \in d_j\}} \times \frac{t_i}{|T|}\right] \quad (5)$$

We define a model of the user based on BoW (Eq-6), where $\overline{W}^u$ is the set of all words provided by the user u.

$$U_{BoW} = \bigcup_{i=1}^{|W|} \begin{cases} wf(w_i) & , w \in \overline{W}^u \\ 0 & , otherwise \end{cases} \quad (6)$$

3.2 Features of Creativity

We measure *Fluency* as the time variance of each query word of the user as you can see in the Eq-7, where T^u is the set of timestamp of each answer of the user u to the query word q^s.

$$t_v^u(q^s) = var[(t_1), (t_2), \ldots, (t_n)] \forall t_i \in T^u \quad (7)$$

We measure *Flexibility* as the variance in the Part of Speech (PoS) of associated words (Eq-8), where W^u are the answers of the user u to the query word q^s.

$$PoS_v^u(q^s) = var[PoS(w_1), \ldots, PoS(w_n)] \forall w_i \in W^u \quad (8)$$

We define *Originality* through two features based on the word frequency: 1) the variance of the word frequency (Eq-9), where W^u is the set of answers of the user u to the query word q^s.

$$wf_v^u(q^s) = var[f(w_1, d_j), \ldots, f(w_n, d_j)] \forall w_i \in W^u \quad (9)$$

And 2) the dot product of frequency and time (Eq-10)

$$t.f^u(q^s) = [\frac{t_1}{|T|}, \ldots, \frac{t_n}{|T|}] \cdot [f(w_1, d_j) \ldots f(w_n, d_j)] \quad (10)$$

Hence, we define a feature vector $fv^u(q^s)$ integrating the Equations 7, 8, 9 and 10, as follows:

$$fv^u(q^s) = [t_v^u(q^s), PoS_v^u(q^s), wf_v^u(q^s), t.f^u(q^s)] \quad (11)$$

Finally, we model the user behaviour U_{FoC} (Eq-12) concatenating the feature vector fv^u of each association task (Eq-11), where $|Q|$ is the set of all association tasks driven by the query words q_i^s.

$$U_{FoC} = \bigcup_{i=1}^{|Q|} fv^u(q_i^s) \quad (12)$$

4. EXPERIMENTAL SETUP

This work aims to model the hidden heuristic in association tasks based on the behaviour of creative people. We have designed a practical experiment based on a Web platform to generate a novel dataset that relates associative skills of users and their creative potential. We applied this experiment on sixth grade students from two different schools in Spain, in a relation of 67% from one school and 33% from the other. The whole sample was composed by 47% of male and 53% of female.

The experiment involved two creative challenges developed during a class: *unusual uses* and *word association* tasks. First, the users were asked to: *write down as many unusual uses as possible for the object 'Shoe' during 60 seconds.* With this task we captured data about the divergent thinking potential of each user and, thus, we can compute a measure of their creative potential. The users were also asked to: *write down as many associated words as possible for a 'Book' ('Door') during 60 seconds.*

In order to form the dataset, the platform has registered the query object, the unusual uses listed by students and the timestamp for each use. It also has saved the query words, the associated words provided by students and the timestamp of each word. These data is represented by equation 1. Demographic data collected from each student includes age, gender and country.

In addition, a label about their creativity was provided based on the unusual uses challenge and the intrinsic characteristics of creativity. Two reviewers labelled each user as creative or non-creative using a Linkert-scale (5) of flexibility, fluency and originality. Accordingly, a labelled dataset was defined to perform a supervised learning of the creative behaviour of users. The dataset structure is depicted in the Table 1.

We have modelled the user behaviour (Sec-3) using the modified Bag-of-Words (BoW) and the Feature of Creativity (FoC). We have designed a two-class supervised learning experiment and trained a set of learning algorithms: Naïve Bayes (NB), Decision Tree ($dTree$), Support Vector Machine (three kernels) and Random Forest ($rTree$). In order to evaluate the accuracy of the learning algorithms we performed a cross-validation method. Thus, we have iteratively divided the dataset in k subsets, where the $k-1$ subsets were used to train the algorithms and the last one was used to validate the prediction quality based on its accuracy. Finally, we have performed an analysis of accuracy results against the percentage of the instances used in the cross-validation method.

Table 1: User information in dataset

Attribute	Description
`Gender`	The user gender
`Age`	The user age
`Country`	The user country
`Creative tag`	Creative (+) or No creative (-)
`Unusual Uses`	A set of tuples *(use, time)* per object
`Associations`	A set of tuples *(word, time)* per query

Table 2: Dataset statistics

Avg. Attr. per minute	Creative (53%)		No Creative (47%)	
	M (35%)	F (65%)	M (60%)	F (40%)
#Uses 'Shoe'	3.33	4.20	**4.88**	4.10
#Asoc 'Book'	**6.67**	**6.05**	5.30	5.73
#Asoc 'Door'	**5.92**	**5.77**	4.5	4.18
	Age of Students			
	11	12	11	12
#Uses 'Shoe'	3.94	3.33	3.91	4.00
#Asoc 'Book'	**6.42**	4.67	5.52	5.33
#Asoc 'Door'	**5.93**	4.67	4.72	5.33

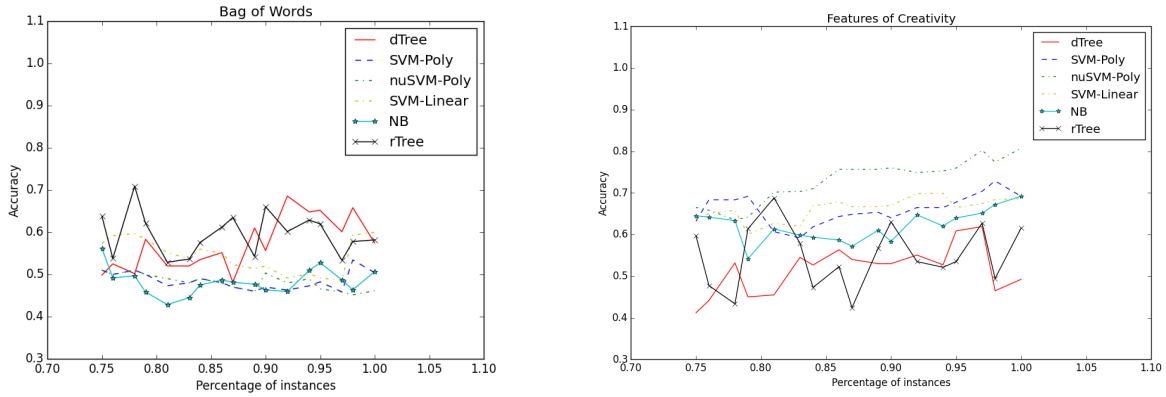
Diccionary Size (# unique words)		
Global	'Libro'	'Puerta'
247	127	129

5. RESULTS

By applying the challenges, a dataset was defined based on Eq-1. We highlight that creative students are more fluent than non-creative ones and younger students provide more associated words per minute. We have also defined a global dictionary with all associated words W provided by users and local dictionaries (W^{q^s}) for each association task. A more detailed information is shown in the Table 2

We have analysed the size of the dataset, because the features are based on statistics. In the figure 1a you can see the accuracy of the U_{BoW} model, which approximately ranges in 10 points at each model. The results of the model are similar for different sizes of the dataset, so this model can be seen independent of the size of the dataset. In the figure 1b we show the accuracy of the U_{FoC} model, which generally increases with respect to the size of the dataset, except in the case of the tree-based method. This model can be seen as dependent of the dataset size and it should improve as the dataset grows.

The most stable algorithms are the kernel-based (SVM) because they fit more precisely with the features of creativity.



(a) (b)

Figure 1: Accuracy performance against different sizes of dataset: The strength of a) U_{BoW} and b) U_{FoC}.

Also, we reach high levels of accuracy in the classification of creative behaviour based on a simple set of features and a moderate number of available samples, which reach up to 81% in the U_{FoC}.

6. DISCUSSION

The work by Jennings et al. is a not context free proposal (art) and, it is too invasive for students [5]. The associative actions of users are mined to figure out the hidden strategy of users through a design task. We define a model based on a ordinary task of word association that is common in problem-solving contexts, web searches and social networks. Therefore, the provided models can be applied in active learning contexts where students make associations. The proposal of Lin et al. [6] is seeking to improve the learning of creativity by recommendation in a personalized tutoring system. We propose a complementary work to identify the creative potential and, thus, it could be possible to provide better learning paths to students based on such prediction.

The bag-of-words approach U_{BoW} has reached an acceptable accuracy level, but it has a high variance. The features of creativity approach U_{FoC} is more stable and it has a growing accuracy along the number of samples. This model is based on a small number of features, which are highly related with the theoretical features of creativity: fluency, frequency and originality.

7. CONCLUSION

We have proposed two user models to identify creative students when they associate words: U_{BoW} and U_{FoC}. These models outline the creativity of students over time by exploiting their word associations (Web search, Social Network, etc). Thus, we depicted that it is possible to learn a classifier based on associative features with an acceptable accuracy. We have developed a dataset that relates the association skills and the creative potential of students.

In the future we will integrate the sequentially of associations, so there is the possibility to use sequential learning algorithms. The flexibility could be described using the sense/meaning of the word as a more informative similarity. Finally, we have depicted that a higher number of instances improves performance, then a more diverse set of samples should be considered.

8. ACKNOWLEDGMENTS

This work is partially founded by the Erasmus Mundus SUDUE program.

9. REFERENCES

[1] T. Amabile. The social psychology of creativity: A componential conceptualization. In *Journal of Personality and Social Psychology*, volume 45, pages 357–376, 1983.

[2] P. André, M. c. Schraefel, J. Teevan, and S. D. T. Discovery is never by chance: designing for (un) serendipity. *C and C '09*, pages 305–314, 2009.

[3] R. E. Beaty and P. J. Silvia. Why do ideas get more creative across time? an executive interpretation of the serial order effect in divergent thinking tasks. *Psychology of Aesthetics, Creativity, and the Arts*, 6(4):309–319.

[4] M. Benedek and A. C. Neubauer. Revisiting mednick's model on creativity-related differences in associative hierarchies. evidence for a common path to uncommon thought. *The Journal of Creative Behavior*, 47(4):273–289, 2013.

[5] K. E. Jennings, D. K. Simonton, and S. E. Palmer. Understanding exploratory creativity in a visual domain. In *Proceedings of the 8th ACM conference on Creativity and cognition*, pages 223–232. ACM, 2011.

[6] C. F. Lin, Y.-C. Yeh, Y. H. Hung, and R. I. Chang. Data mining for providing a personalized learning path in creativity: An application of decision trees. *Comput. Educ.*, 68:199–210, Oct. 2013.

[7] S. Mednick. The associative basis of the creative process. In *Psychological review*, volume 69, pages 220–232, 1962.

[8] E. P. Torrance. The nature of creativity as manifest in its testing. In *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge University Press, New York, 1988.

Optimizing Partial Credit Algorithms to Predict Student Performance

Korinn Ostrow, Christopher Donnelly, Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
{ksostrow, cdonnelly, nth}@wpi.edu

ABSTRACT

As adaptive tutoring systems grow increasingly popular for the completion of classwork and homework, it is crucial to assess the manner in which students are scored within these platforms. The majority of systems, including ASSISTments, return the binary correctness of a student's first attempt at solving each problem. Yet for many teachers, partial credit is a valuable practice when common wrong answers, especially in the presence of effort, deserve acknowledgement. We present a grid search to analyze 441 partial credit models within ASSISTments in an attempt to optimize per unit penalization weights for hints and attempts. For each model, algorithmically determined partial credit scores are used to bin problem performance, using partial credit to predict binary correctness on the next question. An optimal range for penalization is discussed and limitations are considered.

Keywords

Partial Credit, Student Modeling, Next Question Correctness, Adaptive Tutoring Systems, Maximum Likelihood, Grid Search

1. INTRODUCTION

Adaptive tutoring systems provide rich feedback and an interactive learning environment in which students can excel, while teachers maintain data-driven classrooms by using the systems as powerful assessment tools. Simultaneously, these platforms have opened the door for researchers conducting minimally invasive educational research at scale while offering new opportunities for student modeling. Still, they are commonly restricted to measuring performance through binary correctness on each problem. Arguably the most popular form of student modeling within computerized learning environments, Knowledge Tracing, is rooted in the binary correctness of each opportunity or problem a student experiences within a given skill [1]. Knowledge Tracing (KT) drives the mastery-learning component of renowned tutoring systems including the Cognitive Tutor series, allowing for real time predictions of student knowledge, skill mastery, or next problem correctness [4]. Similar modeling methods consider variables that extend beyond correctness but rarely escape the binary nature of the construct, including Item Response Theory [2] and Performance Factors Analysis [9]. By restricting input to a binary metric across questions, these modeling techniques fail to consider a continuous metric that is commonplace for many teachers: partial credit.

Partial credit scoring used within adaptive tutoring systems could provide more individualized prediction and thus establish models with better fit. It is likely that binary correctness has remained the default for learning models due to the inherent difficulty of defining a universal algorithm to generalize partial credit scoring across platforms. Some of the onus may also fall on users' familiarity with current system protocol; students tend to avoid using system feedback regardless of the benefits it may provide because requesting feedback results in score penalization. However, the primary goal of these platforms is generally to promote student learning rather than simply acting as an assessment tool, and thus, binary correctness is flawed.

The present study considers data from ASSISTments, an online adaptive tutoring system that provides assistance and assessment to over 50,000 users around the world as a free service of Worcester Polytechnic Institute. Researchers have previously used ASSISTments data to modify student-modeling techniques in a variety of ways including student level individualization [7], item level individualization [8], and the sequence of student response attempts [3]. Previous work has also shown that naïve algorithms and maximum likelihood tabling methods that consider hints and attempts to predict next problem correctness can be successful in establishing partial credit models meant to supplement KT [10; 11]. More recently, algorithmically derived partial credit scoring resulted in stand-alone tabled models using data from only the most recent question and yet showing goodness of fit measures on par with KT at lower processing costs [6]. However, we hypothesize that some conceptualizations of partial credit may lead to better predictive models than others. Rather than subjectively defining tables or algorithms, a data driven approach should be considered. Thus, considering student performance within the ASSISTments platform, the current study employs a grid search on per unit penalizations of hints and attempts to ask:

1. Based on penalties for hints and attempts dealt per unit, is it possible to algorithmically define partial credit scoring that optimizes the prediction of next problem correctness?
2. Does the optimal model of partial credit differ across different granularities of dataset analysis?

Establishing an optimal partial credit metric within ASSISTments would allow teachers using the tool to more accurately assess student knowledge and learning, while allowing students to alter their approach to system usage by taking advantage of adaptive feedback. The optimization of partial credit scoring would also enhance student modeling techniques and offer a new approach to answering complex questions within the domain of educational data mining.

2. DATA

The ASSISTments dataset used for the present study is comprised solely of assignments known as Skill Builders. This type of assignment requires students to correctly answer three consecutive questions to complete the problem set. Questions are randomly pulled from a large pool of skill content and are typically presented with tutoring feedback, most commonly in the form of hints. The dataset has been de-identified and is available at [5] for further investigation.

The dataset used in the present study is a compilation of Skill Builders from the 2012-2013 school year, containing data for 866,862 solved problems. Recorded data includes students' performance on the problem (i.e., binary correctness, hint count, attempt count), variables that identify the problem itself (i.e., problem type, unique problem identification number) and information pertaining to the assignment housing the problem (i.e., unique identifiers for assignments, skill type, teachers, and schools). The dataset was representative of 120 unique skills and 24,912 unique problems, solved by 20,206 students.

On average, students made 1.53 attempts per problem (SD = 15.08). The minimum number of attempts was 0 (i.e., a student who opened the problem and then left the tutor), while the maximum number of attempts was a daunting 12,246 (i.e., a student who hit 'Enter' repeatedly for a prolonged period of time, likely out of frustration or boredom). Students made a total of 1,324,226 attempts across all problems. The majority of problems (74.9%) had just one logged attempt per student (typically correct answers), while 15.1% of problems carried only two logged attempts.

Hint usage among all students averaged 0.61 hints per problem (SD = 1.29). The minimum number of hints used was 0 (i.e., no feedback requested), while the maximum number of hints used was 10. Interestingly, the maximum number of hints available for any particular problem was 7. Thus, a handful of students who logged more than 7 hints were accessing the tutor in multiple browser windows (i.e., cheating). On average there were 3.22 hints available per problem (SD = 0.89). The majority of problems contained 3 hints (44.6%), 4 hints (28.9%), or 2 hints (18.2%). Although there were 2,768,299 hints available across all problems, students only used 529,394 hints, or approximately 19% of available feedback. Bottom out hints, or those providing the problem's solution, were only used on 146,742 (16.9%) of problems.

Additional analyses were performed on the 261,787 problems that students answered incorrectly out of the original 866,862 problems solved. Within this subset of data, students made an average of 2.75 attempts per problem (SD = 27.40). Students also used an average of 2.02 hints (SD = 1.63). This subset of problems had 860,131 total hints available, of which students used 528,644 hints (61.5%).

Hint usage would likely increase if partial credit scoring was implemented within the ASSISTments platform. In many classrooms, binary first attempt scoring has created an environment in which students are afraid to use hints although they would benefit from feedback, as they know they will receive no credit. Further, the dataset suggests that once students are marked wrong, they are more likely to jump through all available hints and seek out the answer (56% of incorrect first attempts led to bottom out hinting). This reflects another substantial downfall in the system's current protocol: once the risk has passed, so has the drive to learn. The implementation of partial credit scoring has the potential to alleviate this misuse.

3. METHODS

The present study presents an extensive grid search of potential per hint and per attempt penalizations. The full dataset was used to define partial credit scores algorithmically based on per unit penalizations ranging from 0 to 1 in increments of 0.05 for both hints and attempts. Thus, for each solved problem in the dataset, 441 partial credit scores were established based on each possible combination of per unit penalization. For example, in a model in which each attempt earned a penalization of 0.05, and each hint earned a penalization of 0.1, a student who made three attempts and used one hint would receive a penalty of 0.25 ((3x0.05) + (1x0.1)), effectively scoring 0.75 on that problem. This process was used to score each problem in the dataset for each possible penalty combination, with a floored per problem score of 0 (students could not receive negative scores). This method was similar to that presented by Wang & Heffernan in the Assistance Model [10] which established a tabling method to calculate probabilities of next problem correctness based on combinations of hints and attempts that resulted in twelve possible bins or parameters.

For each of the 441 partial credit models, a maximum likelihood tabling method was employed using five fold cross validation. Within each model, a modulo operation was used on each student's unique identification number to assign students to one of five folds. Note that this method resulted in folds that all represented approximately 20% of students in the dataset. Maximum likelihood probabilities for next problem correctness were then calculated for each partial credit score within each model. Table 1 presents an average of test fold probabilities for the model in which each attempt and each hint are penalized 0.1. For instance, a student using two attempts (2 x 0.1) and one hint (1 x 0.1) would be penalized 0.3, thus falling into the score bin of 0.7 (PC Score). Following through with this example, based on 11,174 problems solved that fit this scoring structure, the average of known binary performance on the following problem was 0.599. This value becomes the prediction for next problem correctness for students scoring 0.7 on the current problem.

Using the maximum likelihood probabilities for next problem correctness within each test fold as predicted values, residuals were then calculated by subtracting predictions directly from actual next problem binary correctness (i.e., 1 – 0.725 = 0.275; 0 – 0.571 = -0.571). This approach was used rather than selecting an arbitrary cutoff point to classify a prediction as correct or incorrect in the binary sense (i.e., values greater than or equal to 0.6 serve as predictions of correctness) because it reduced the potential for researcher bias.

Table 1. Probabilities averaged across test folds for the model in which the penalization per hint and per attempt is 0.1

PC Score	n	Max. Likelihood NPC
0	149,504	0.467
0.1	422	0.571
0.2	685	0.581
0.3	1,055	0.578
0.4	1,784	0.574
0.5	3,442	0.583
0.6	6,623	0.585
0.7	11,174	0.599
0.8	18,679	0.662
0.9	49,972	0.725
1.0	476,523	0.802

4. RESULTS

For each model, residuals were used to calculate RMSE, R^2 & AUC at three levels of granularity: problem level, student level, and skill level. Heat maps are only presented here for RMSE, as the other metrics established almost identical maps. Metrics representing greater model fit are depicted using the purple end of the spectrum, while those representing poorer fit are represented using the red end of the spectrum. Further, a series of ANOVAs were conducted to compare each set of models within the same penalization level for attempts and hints. For example, the 21 models in which attempt penalty was set to 0.2 were compared to all other sets of attempt penalty models to investigate significant differences across penalties. This method was used rather than comparing each model with all other models using paired samples t-tests, as the resulting 194,481 analyses (441^2) would greatly inflate the rate of Type I error without unrealistic corrections.

Initial analysis was performed at the problem level; residuals were calculated for each problem that contained next problem correctness metrics and goodness of fit measures were averaged across the dataset. Each metric followed a similar structure in which low attempt penalties appear to result in better fitting models, while hint penalty does not appear to be significant. Thus, partial credit scoring algorithms using lower penalties for attempts were better at predicting next problem performance, as depicted in Figure 1. The ANOVA results depicted in Table 2 suggest that differences in attempt penalty models were significant. Thus, the set of models with per attempt penalties of 0.1 differed significantly from the set of models with per attempt penalties of 0.8. Differences among hint penalty models were not reliably significant. Figure 1 also suggests that the current binary scoring protocol used by ASSISTments results in predictive models that are inadequate. First attempt binary correctness is the equivalent of the model in which per attempt and per hint penalty are both set to 1, or the upper right corner of each heatmap). This model resulted in consistently poor fit metrics, suggesting that modeling techniques such as KT should employ continuous or binned partial credit values as input as they enhance next problem prediction ability. It has not yet been investigated how this alteration would change the prediction of other variables commonly predicted through KT, such as latent student knowledge or skill mastery.

Student level analysis was undertaken using a subset of the original data file. At this granularity, goodness of fit metrics were calculated for each student and averaged across students to obtain final metrics for each of the 441 models. As the ASSISTments system measures completion of a Skill Builder as three consecutive correct answers, a number of high performing students had limited opportunity counts within skills. For students with too few data points, it was not possible to calculate R^2 and AUC. Therefore, student level analysis incorporated 7,429 students from the original dataset, or 651,849 problem logs. Answering our second research question, it appears as though the region of optimal partial credit values observed at the problem level remains consistent at the student level, as shown in Figure 2. ANOVA results depicted in Table 2 show reliably significant differences across attempt penalty models but not across hint penalty models.

Skill level analysis was also undertaken using a subset of the original data file. One skill did not have enough data based on a low number of users and high mastery within those users, and was

Table 2. ANOVA results for groups of attempt and hint penalty models at each level of analysis

			Attempt Penalty			Hint Penalty		
Level	*Min*	*Max*	*F*	*p*	R^2	*F*	*p*	R^2
Problem								
RMSE	.430	.435	302.70	.000	.935	0.95	.519	.043
AUC	.626	.655	295.46	.000	.934	1.14	.304	.052
R^2	.070	.091	304.34	.000	.935	0.95	.525	.043
Student								
RMSE	.424	.429	222.49	.000	.914	1.34	.149	.060
AUC	.578	.593	208.19	.000	.908	1.42	.106	.063
R^2	.096	.110	374.52	.000	.947	0.80	.715	.037
Skill								
RMSE	.423	.429	517.85	.000	.961	0.55	.944	.026
AUC	.624	.647	250.17	.000	.923	0.72	.805	.033
R^2	.073	.090	510.96	.000	.961	0.49	.971	.023

Note. For all models, df = (20, 420).

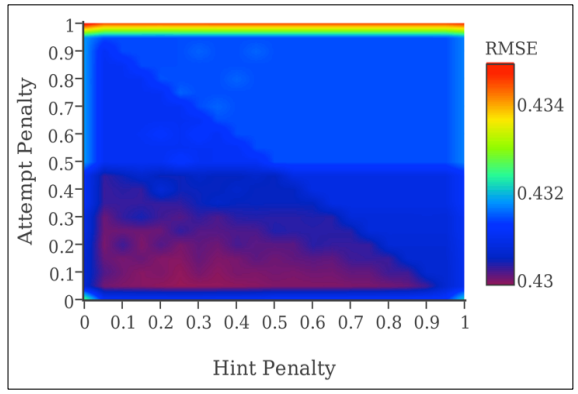


Figure 1. Problem Level RMSE

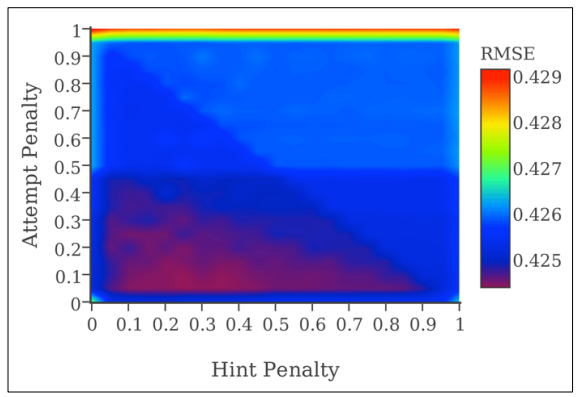


Figure 2. Student Level RMSE

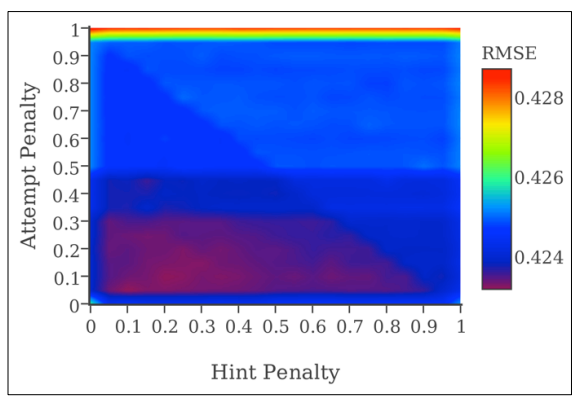


Figure 3. Skill Level RMSE

excluded from skill level analysis, resulting in a file with 119 skills. At this granularity, goodness of fit metrics were calculated for each skill and averaged across all skills to obtain final metrics for each of the 441 models. Results are depicted in Figure 3. The heat map shows that the region of optimal penalization has grown more concise, showing optimal fit among models with low per hint and per attempt penalties (< 0.3). ANOVA results depicted in Table 2 again suggest reliably significant differences in all metrics across attempt penalty models but not across hint penalty models.

Post-hoc analyses were conducted on ANOVA results using multiple comparisons to examine significant differences between attempt penalty and hint penalty model groups when considering problem level AUC. Using a Bonferroni correction to reduce Type I error, this process resulted in a series of significance estimates for penalty group comparisons (i.e., all models where attempt penalty is 0.1 compared to all models where attempt penalty is 0.3 results in a non-significant difference, $p = 0.88$). Results suggested that models close in penalty were less likely to differ significantly than models with greater difference in penalty. For instance, models with an attempt penalty of 0.1 were significantly different than those with an attempt penalty of 0.4, but were not significantly different than those with an attempt penalty of 0.2. This information can be used to help optimize partial credit penalizations, as it may be more motivating and productive for students to receive smaller penalizations. Such information could also allow systems like ASSISTments to define a range of possible penalizations that could then be refined by the teacher, providing all users with a greater sense of control.

5. DISCUSSION & CONTRIBUTION

The initial findings of a grid search on partial credit penalization through per unit hint and attempt docking suggest that the implementation of partial credit within adaptive tutoring systems can be established using a data driven approach that will ultimately produce stronger predictive models of student performance while enhancing the way adaptive tutoring systems are used by students and teachers.

Our first research question was answered with a resounding "Yes," certain algorithmically derived combinations of partial credit penalization are better than others when used to predict next problem performance. Optimal partial credit models were visible in heat maps spanning three levels of data granularity and remained relatively consistent across granularities, thus answering our second research question. ANOVAs revealed that differences in attempt penalty models were consistently significant across dataset granularities, while differences in hint penalty models were not reliable. This finding is likely due to the fact that hint usage is lower and less distributed than attempt count across problems in the dataset, and it is possible that this finding would diminish in a system that more readily promoted the use of tutoring feedback without penalization, or a system already employing partial credit scoring.

The partial credit models that we define here as optimal, based on their ability to predict next problem performance, were models with per hint and per attempt penalties of 0.3 or less. Additional analyses revealed that at the problem level, there should be no reliable difference in predictive ability of a model penalizing 0.3 per attempt from a model penalizing 0.1 per attempt, with variable hint penalization. This finding suggests that less penalization is just as effective, offering an opportunity to consider student motivation and affect when defining a partial credit algorithm. This grid search also revealed that partial credit metrics outperform binary metrics when predicting next problem performance, as previously shown in [6]. Thus, it is possible to improve prediction of student performance within adaptive tutoring systems simply by implementing partial credit scoring. It should also be noted that a leading limitation of the approach presented here is that we have only been predicting next problem correctness, rather than latent variables such as skill mastery or student knowledge. It is possible that optimizing partial credit would also provide benefits for the prediction of latent effects, but further research is necessary in this domain.

6. ACKNOWLEDGMENTS

We acknowledge funding from NSF (1316736, 1252297, 1109483, 1031398, 0742503, 1440753), ONR's "STEM Grand Challenges," and IES (R305A120125, R305C100024). Thanks to S.O. & L.P.B.O.

7. REFERENCES

[1] Corbett, A.T. & Anderson, J.R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4: 253-278.

[2] Drasgow, F. & Hulin, C.L. (1990). Item response theory. In M.D. Dunnette & L.M. Hough (Eds.), Handbook of Industrial and Organizational Psychology, Vol. 1, pp 577-636. Palo Alto, CA: Consulting Psychologists Press.

[3] Duong, H.D., Zhu, L., Wang, Y., & Heffernan, N.T. (2013). A Prediction Model Uses the Sequence of Attempts and Hints to Better Predict Knowledge: Better to Attempt the Problem First, Rather Than Ask for a Hint. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th Int Conf on EDM. pp. 316-317.

[4] Koedinger, K.R. & Corbett, A.T. (2006). Cognitive tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61-78). NY: Cambridge University Press.

[5] Ostrow, K. (2014). Optimizing Partial Credit Data. Accessed 12/8/14. https://tiny.cc/OptimizingPartialCredit

[6] Ostrow, K., Donnelly, C., Adjei, S. & Heffernan, N. (2015). Improving Student Modeling Through Partial Credit and Problem Difficulty. In Russell, Woolf & Kiczales (Eds.), Proceedings of the 2nd ACM Conf on L@S. pp. 11-20.

[7] Pardos, Z.A. & Heffernan, N.T. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Proceedings of the 18th Int Conf on UMAP. pp. 255-266.

[8] Pardos, Z.A., & Heffernan, N.T. (2011). KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Joseph A. Konstan et al. (Eds.): UMAP 2011, LNCS 6787, pp. 243-254.

[9] Pavlik, P.I., Cen, H., Koedinger, K.R. (2009). Performance Factors Analysis - A New Alternative to Knowledge Tracing. In: Proceedings of the 14th Int Conf on AIED, pp. 531-538.

[10] Wang, Y. & Heffernan, N.T. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. The 24th International FLAIRS Conference.

[11] Wang, Y. & Heffernan, N. (2013). Extending Knowledge Tracing to Allow Partial Credit: Using Continuous versus Binary Nodes. In K. Yacef et al. (Eds.) AIED 2013, LNAI 7926, pp 181-188.

Identifying Styles and Paths toward Success in MOOCs

Kshitij Sharma[1], Patrick Jermann[2], Pierre Dillenbourg[1]
1. Computer Human Interaction in Learning and Instruction
2. Center for Digital Education, École Polytechnique Fédérale de Lausanne, Switzerland
<firstname>.<lastname>@epfl.ch

ABSTRACT
Current schemes to categorise MOOC students result from a single view on the population which either contains the engagement of the students or demographics or self reported motivation. We propose a new hierarchical student categorisation, which uses common online activities capturing both engagement and achievement of MOOC students. A first level is based on the online engagement with the course structure, i.e., whether they take part in graded activities or not. Based on this criterion, we divide students into two major categories: *active students* and *viewers*. The second levels are based on the different activities typically performed by the students in these two categories. For the "active students" we categorise them based on their final result. For the "viewers", we further divide the category based on their engagement quotient, i.e., how much of the course content they follow and whether they involve with the non-mandatory exercises in the course or not. Further, in this contribution we analyse the behaviour of the students in different categories to highlight the basic differences among them.

Keywords
Student categorisation, Student achievement, Massive open online courses, Student engagement

1. INTRODUCTION
The global wave of free, large and virtual courses attracts an incredibly diverse student population. With this diversity comes a huge variety of online behaviours. For data scientists it is a challenge to find categories that are suitable for sampling the whole population. It is also important to keep the categorisation scalable and robust.

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 [10] or engagement patterns [6, 7, 9, 4, 3, 5] or demographics [2, 1].

Based on student motivation (their "stated intent") of the students, [10] categorised the students, No-shows, Observers, Casual Learners and Completers. Where No-shows only register, Observers want to know about how a MOOC looks like, Casual Learners want to learn a few things only, and Completers want to earn a finishing certificate.

There are many categorisation schemes depending on engagement patterns. [6] categorised 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, [7] categorised students into Active (students who adapt well to the connectivest 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) [5]. Later he revised his categories and divided the Lurkers into No-shows and Observers [3, 4].

Petty and Farinde [9] used the engagement categories from [8] to categorise students in an online mathematics course. These categories, based on the students' engagement patterns into critical thinking, were Clarification, Assessment, Inference, and Strategies.

The other dimension used to categorise students is to look at the demographics. For an electrical engineering course [2] categorised students based on their country of origin, education qualifications and backgrounds. Looking at the demographics of University of Pennsylvania's Open Learning Initiative [1] also categorised MOOC students based on their country of origin and educational background as [2] did. However, [1] added a few more categories based on gender, age and employment status of the MOOC students.

One common feature about these categorisation schemes is that they all consider only one of the dimensions of student behaviour, for example, engagement with the course content or forums or demographics or motivation. In this contribution, we present a novel categorisation scheme that considers both the engagement and the achievement of MOOC students. We further report on the different patterns shown by

the students from different categories. Moreover, the categories like Completing [6] and Active [4] are more than just engagement patterns; they also represent a mixed population of students with some achievement "flag". Therefore, we propose to further divide this category into subcategories based on the students' achievement.

2. RESEARCH QUESTIONS

In this study, we ask two main research questions:
Question 1: How can we categorise the MOOC students into categories that reflect both their achievement and engagement?
Question 2: What are the basic differences in the online behaviour of the students representing populations from different categories? More specifically, we are interested in finding the different ways to succeed in a MOOC which leads us to the following research questions. **Question 2.1** How does the engagement with the course content relate to the achievement? **Question 2.2** How does the timing of engagement i.e., the engagement with the course structure relate to the achievement? **Question 2.3** How does the effort during graded assignments relate with the achievement?

3. COURSE DETAILS

For this analysis we chose four courses from Coursera. The courses were basic JAVA and C++ both at the fundamental levels and as an introduction to object oriented programming. The courses were in French and were developed at École Polytechnique Fédérale de Lausanne, Switzerland. All the courses were basic level programming courses. All the courses had 7 weeks of lecture material. All the courses had programming assignments to grade the students. Also they had additional non-graded quizzes for practice. All the courses had the last deadline in the 11th week from the beginning of the course. They also had soft deadline for the programming assignments after which the effective submission score reduced to 50 % of the actual score. All the courses were open after the final deadline as well.

4. CATEGORIES

We propose a hierarchical categorisation scheme. The first reason for having a few second levels in the scheme is to be able to include the achievement of MOOC students in the analysis of online behavioural patterns. The existing categorisation schemes lack on this front. They put the completion of the course as the only criterion for having a category, which oversimplifies the different levels of achievement. Having more levels for the students' achievement enables us to identify the different trends to succeed in a MOOC.

We have two first level categories: active students and viewers (based on whether the student participated in the grades assignments or not). Active students are subcategorised based on their achievement levels and viewers are subcategorised based on their further engagement with the course content. The motivation for subcategorising viewers was to have equally distributed categories so that none of the categories have a vast majority of the student population. This improves the generalisation of the categorisation schemes beyond the courses we chose to establish the categories.

We divide the whole student population in two major categories. First, those students who actively participate in the course, i.e., they take part in the assessment processes. We simply call these students "Active students". The active students get an achievement label at the end of the course. Second, those students who just watch the videos from the course (irrespective of the number of videos they watch). We call these students "Viewers". The viewers do not get any achievement label at the end of the course.

We further divide the active students based on their achievement labels that they get at the end of the course. Active students can either be "failed", "normal", or "distinction". The levels of "normal and distinction students may vary from on course to another, but for the courses we chose the criteria is the same for differentiation of these two subcategories of active students. Moreover, all the data for the active students is collected between the start week of the course and the last week of the assignment submission deadline.

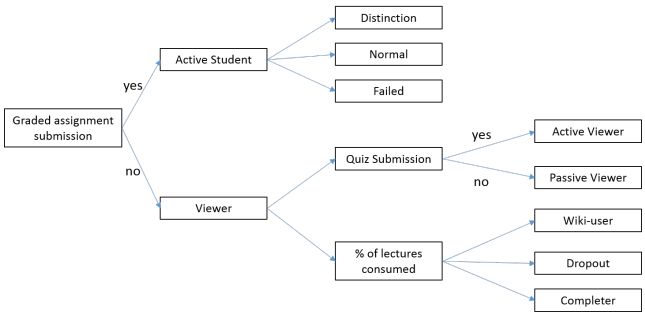


Figure 1: Hierarchy used in the present categorisation scheme.

The viewers, are further divided based on two factors. First, the amount of videos they watch; and second, whether they assess their learning by the means of the non-mandatory quizzes (in-video quizzes or regular non-graded quizzes) or not. Using the first factor, we divide the students into: 1. "wiki viewers" (if a student watches less than 10% of the videos). 2. "dropouts" (if a student watches between 10% and 70% of the videos). 3. "completers" (if a student watches more than 70% of the videos).

Using the second factor, we divide the the student into "Active Viewers" and "Passive Viewers". Since the courses were open even after the last assignment deadline, we consider the data till date of data export from Coursera (20th week) for analysing the behaviour of the viewers.

5. VARIABLES

We used the following variables to analyse the behaviour of the students in different categories:

5.1 Active students

For analysing the differences in the activities among different achievement levels of Active students we defined the **First submission score:** the average score of the first attempt of all the programming assignments, as a proportion of maximum attainable score for each assignment. **First action week:** the first week of any kind of activity after registering for the course, once the course had started. **Activity span:** the difference in weeks between the first activity (as described in the previous item) and the last activity.

Progress within programming assignments: the difference between the two consecutive submissions for the same assignment, as a proportion of maximum attainable score for each assignment. Average number of attempts for each programming assignment. Proportion of videos watched **Delay in watching the lectures:** the time difference in weeks, between the time when the video was released online and the time the students watched it for the first time. Number of forum Views. **Procrastination index:** the ratio of the time difference between the submission time and the hard deadline and time difference between assignment being posted online and the hard deadline.

5.2 Viewers

For analysing the differences across the viewers' subcategories, we use only four of the above mentioned variables: first action week, delay in watching the lectures, activity span and the number of forum views.

6. RESULTS

In this section, we describe the differences between the different levels of active subcategories and viewer subcategories.

6.1 Active students

Concerning the lecture activities, the number of lectures watched by the failed students is significantly lower than the students having normal passing grades or the students with distinction $F[(2, 9914) = 741.95, p < .001]$. The lecture delay (overall and across the 7 weeks of lectures) decreases significantly as we move from distinction to normal to failed students $[F(2, 9914) = 91.43, p < .001]$.

Concerning assignment submissions, we see many differences across the three achievement levels. The first submission score decreases significantly as we move from distinction to normal to failed students $[F(2, 9914) = 210.65, p < .001]$. Number of attempts decreases significantly as we move from failed to distinction to normal students [F(2,2,9914)=222.86, $p < .001$]. The average improvement in two consecutive submissions for the same assignment is significantly higher for the students with distinction than the students with normal and failed levels $[F(2, 9914) = 101.58, p < .001]$. Moreover, the average procrastination index for the students with distinction level is significantly lower than the students from other two subcategories $[F(2, 2, 9914) = 343.83, p < .001]$.

The probability of achieving a higher grade decreases as the first action week approaches the 11th week $[\chi^2(N = 9917) = 201.73, p < .001]$. The activity span for failed students is significantly smaller than passed students (normal and distinction) the course $[F(2, 2, 9914) = 972.68, p < .001]$. If we look at the forum views, the average number of forum views decreases significantly as we move from distinction to normal to failed students $[F(2, 2, 9914) = 135.42, p < .001]$.

6.2 Viewers

The viewer subcategories are based on two factors; first, how much video content they watch and second, whether they participate in non-mandatory quizzes or not. Here we present the results of the different activities for the viewer subcategories. The wiki-users tend to be passive viewers and completers tend to be active users $[\chi^2(N = 35, 193) = 4322.85, p < .001]$.

We observed an interaction effect of the two viewer subcategories on the first action week $[F(2, 35187) = 95.60, p < .001]$. For passive wiki-users and completers the first action week is significantly higher than the active wiki-users and completers. However, we see the opposite trend for the active and passive dropout viewers.

There were two single effects for the two viewer sub-categories on the activity spans. The activity span is more for the active viewers than the passive viewers $[F(1, 35191) = 1484.3, p < .001]$. Also, the activity span increases significantly as we more from wiki-users to dropouts to completers $[F(2, 35190) = 1919.63, p < .001]$.

There was an interaction effect of the two viewer sub-categories on the lecture delays $[F(2, 35187) = 67.50, p < .001]$. For passive wiki-users and completers the first action week is significantly higher than the active wiki-users and completers. However, we see the opposite trend for the active and passive dropout viewers.

7. DISCUSSION

We show that there are clear differences across the subcategories of active students and viewers. Active students are further subdivided into failed, normal and distinction categories. In section 3.1, we can see that the three categories are very different in terms of lecture, assignment, forum activities as well as their timing of these activities. What emerges from the results that the final achievement label that the active students get depends on a number of factors: 1) initial score, 2) engagement with the course content and forums, 3) efforts in assignment submissions and 4) timing of the activities. The variables we chose to differentiate among the achievement subcategories cover all these factors.

The distinction students get higher scores in their first submissions for the graded assignments than the normal and failed students, they improve more than the other two categories within two consecutive submissions for the same assignments and hence they reach the maximum attainable grade in fewer attempts. This reflects the effect of the initial score and efforts on the achievement level (**Question 2.2**). On the other hand, in spite of having similar improvements to the failed students the normal students get a better achievement level because of submitting more number of times. This shows the relationship between efforts and achievement (**Question 2.3**). Moreover, the distinction students have lower procrastination index for all the assignments than the other two categories. This reflects the relation between engagement with the structure (**Question 2.3**) and the achievement level.

The students who pass the course (distinction and normal) watch more videos than the students who fail. This simply reflects the fact that the students who pass the course engage more with the course content than those who fail the course, and establishes a relation between the engagement with the course content and achievement (**Question 2.1**). More interesting fact is that there is almost no difference between the distinction and normal students in terms of en-

gagement with the course content, however, there is a big difference in the delays that the students display in watching the video lectures. The distinction students have a smaller delay, especially in weeks 2 to 6, than the normal students. This shows the that there is a effect of engagement with the course structure (**Question 2.2**) on the achievement level.

Furthermore, the distinction students visit forums more often than the students from other two categories and the passed students (distinction and normal) have longer activity span than the failed students. It also reflects the effect of engagement on the achievement level (**Question 2.1**).

We see some peculiar behavioural patterns for viewers. One clear relation we see is between the engagement level and the activity span of the viewers. The passive users have smaller activity span than the active users. This simply translates to the fact that the people who assess their knowledge in some manner they tend to engage longer with the course content. We observed this fact for all the viewers.

The wiki-users have a very short activity span. This could be explained in two ways: either they started the course very late and realised that they can not pass the course and hence they left; or, they look for very specific content, look at a few videos for the required content and leave the course. The second behaviour is very similar to a Wikipedia user who looks for a very specific piece of information, obtains it and leaves the website. This was the main reason we called this category wiki-users. The passive wiki users start the course very late (only earlier than the passive completers), have an activity span of less than a week, i.e., they visit the course for some very specific content, then leave the course, this behaviour is closer to what we called a wiki-user's behaviour.

The completers display very interesting patterns, viewers in this category watch more than 70% of the video lectures. The difference in the activity spans of passive and active completers is about 4 weeks, this can be explained by the fact that the passive completers are only interested in the content and not in any kind of self assessment, hence they go through the whole content at a very high pace.

There are some overlaps between the categories we propose and the categories proposed by other researchers. For example, the wiki-users are similar to the sampling in [6] and observers in [4, 3]. Similarly, dropouts are a midway (or a mixed population of) category to disengaging in [6] and drop-ins in [4]. The passive viewers are similar to auditing and passive in [6] and [3] respectively. The completing category in similar to active students and completers in viewer population are similar to auditing [6]. However, the main motivation of putting these two in different categories was to capture there different activities which are clearly driven by different motivations, for the active students the main motivation is to get a certificate and for the completers in viewer population just want to watch the videos as a source of knowledge but do not want a completion certificate.

8. CONCLUSIONS

We presented a new MOOC student categorisation scheme. Its basic idea is to have a hierarchy to categorise MOOC students. We used both engagement and achievement to achieve this goal. First, we categorise students into two broad categories active students and viewers. Active students are those who submit graded assignments and viewers do not take part in this process. Further, we divide active students into normal, distinction and failed students, based on their grades; and we divide viewers into active and passive viewers (whether they attempt quizzes or not) and into wiki-users, dropouts and completers (based on how many video lectures they consume).

Throughout our analysis, we highlight the basic activity differences between subcategories of active students and viewers, proposing a few novel variables, like delay in watching lectures and procrastination index. We identify the different paths of success for the active students and different styles for the viewers. One clear difference between the proposed categories and existing categories is that in all the existing categories there is one category that contains a majority of the student population; whereas in the categories we propose, there is no such category.

The present categorisation scheme might have long term implications. First, for initiating a feedback system for those who dropout midway out of a course, we need a benchmark behaviour to compare against. The online behaviour of the students who passed and/or the completers in the viewer categories can be used in such cases. From the differences among different subcategories we report, it is clear that the different behaviour tend to start emerging as early as from the second week. This can be used to proactively help those students who are lagging behind in their engagement with the course content and course structure.

9. REFERENCES

[1] 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? *SSRN 2350964*, 2013.

[2] 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 *Sixth Learning International Networks Consortium Conference*, 2013.

[3] P. Hill. Emerging student patterns in moocs: A graphical view, 2013.

[4] P. Hill. Emerging student patterns in moocs: A (revised) graphical view, 2013.

[5] P. Hill. The four student archetypes emerging in moocs. *E-Literate. March*, 10, 2013.

[6] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Third international conference on learning analytics and knowledge*. ACM, 2013.

[7] C. Milligan, A. Littlejohn, and A. Margaryan. Patterns of engagement in connectivist moocs. *MERLOT Journal of Online Learning and Teaching*, 9(2), 2013.

[8] C. Perkins and E. Murphy. Identifying and measuring individual engagement in critical thinking in online discussions: An exploratory case study. *Educational Technology & Society*, 9(1), 2006.

[9] 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), 2013.

[10] J. Wilkowski, A. Deutsch, and D. M. Russell. Student skill and goal achievement in the mapping with google mooc. In *First ACM conference on Learning@ scale conference*. ACM, 2014.

Analyzing student inquiry data using process discovery and sequence classification

Bruno Emond
National Research Council Canada
bruno.emond@nrc.gc.ca

Scott Buffett
National Research Council Canada
scott.buffett@nrc.gc.ca

ABSTRACT

This paper reports on results of applying process discovery mining and sequence classification mining techniques to a data set of semi-structured learning activities. The main research objective is to advance educational data mining to model and support self-regulated learning in heterogeneous environments of learning content, activities, and social networks. As an example of our current research efforts, we applied temporal data mining analysis techniques to a PSLC DataShop data set [17, 18, 19, 20]. First, we show that process mining techniques allow for discovery of learning processes from student behaviours. Second, sequential pattern mining is used to classify students according to skill. Our results show that considering sequences of activities as opposed to single events improved classification by up to 230%.

1. INTRODUCTION

The Learning Performance Support Systems program (LPSS) at the National Research Council Canada aims at delivering a personal learning environment (LPSS.me), software algorithms, and prototypes to enable Canada's training and development sector to offer learning solutions to industry partners that will address their immediate and long-term skills challenges. The main elements of the personal learning environment include a common platform architecture, a personal learning assistant, a personal cloud, learning resources repository network, personal learning records, and analytics to discover and assess competencies. The program is at an early stage of development.

One of the main thrusts within this research program seeks to advance and apply educational data mining to model and support self-regulated learning in heterogeneous environments of learning content, activities, and social networks. Our initial position points towards a complementary use of latent knowledge estimation and performance prediction methods [3], and temporal data mining methods. A main research trend in educational data mining consists of analyzing students' performance within intelligent tutoring systems, focusing on the correctness of previous questions or the number of hints and attempts students needed in order to predict their future performance [6]. Predictive mathematical models resulting from this analysis characterize, through parameter values, some information contained in the sequence of actions leading to student performances, but do not represent explicitly those sequences. Over the years there has been a growing interest to examine explicitly learning sequences as a complementary approach. Process and sequence mining have been applied for the analysis of content sequencing and curriculum sequencing [5, 15], group behaviour sequences in collaborative software development tasks [16], problem solving behaviours over a shared tabletop [14], as well as self-regulated learning and meta-cognition [7].

The remainder of this paper consists of a short presentation of temporal data mining, followed by process mining and sequence mining analyses of a semi-structured inquiry learning activity data set [17, 18, 19] obtained from the Pittsburgh Centre for Science and Learning DataShop [8]. We show that process mining techniques allow for the discovery of learning processes, and that sequential pattern mining can used to identify the level of skill exhibited by each student.

2. TEMPORAL DATA MINING

Temporal data mining refers to the extraction of information and knowledge from potentially large collections of temporal or sequential data [12]. According to Laxman and Sastry [9], sequential data refers to any type of data where data points are explicitly ordered, either by time stamps or some other sequencing mechanism. This includes data such as moves in a chess game or commands entered by a computer user, but also other forms of data that are not explicitly time-stamped but are still otherwise ordered, such as text or protein sequences.

Temporal data is often divided into two categories: sequences that consist of continuous, real-valued data points taken at regular intervals, which are referred to as *time series data*, and sequences that may be represented by compositions of nominal symbols from a particular alphabet, which are referred to as *temporal sequences* [2]. As the field of time series analysis has a long history with many established techniques, the more recent field of temporal data mining instead focuses on information extraction from temporal sequences.

Given a set of temporal sequences, the general tasks of tem-

poral data mining consist of 1) prediction, 2) classification, 3) clustering, 4) search and retrieval, and 5) pattern discovery. These tasks can be accomplished using a number of established techniques in the area. A few of the more prevalent techniques include: A) *Sequential pattern mining*: The goal of sequential pattern mining [1] is to identify highly frequent sequences that appear within a database of ordered items or events; B) *Sequence classification*: Sequence classification [11] attempts to assign a candidate sequence to one of possibly several classes of existing sequences, typically according to either similarity or common features such as frequent sub-sequences; C) *Episode mining*: Frequent episodes [13] are sets of partially ordered events that are found to occur close together frequently and consistent with the specified partial order; and D) *Process mining*: Process mining refers to the extraction of process-related information from event logs [21]. Process mining algorithms are used to build a model of the business process by representing the different ways cases in the process can be executed. However, there are some key differences between business processes and learn flows [4].

3. TEMPORAL EDM ANALYSIS

To demonstrate the potential of temporal data mining in the analysis of educational data, we conducted a study utilizing process mining and sequential pattern mining to discover learning processes and to identify the level of student skill using a data set [17, 18, 19] taken from the Pittsburgh Science of Learning Center DataShop [8]. This data set contains data on 148 middle school students performing activities logged while working within a micro-world, where students engage in "scientific inquiry" to study liquid phase change. Here, the students form hypotheses and conduct experiments as they investigate whether container size, heat level, substance amount, and cover status affected the boiling/freezing point of water, or the time it took to freeze/boil. All students' fine-grained actions were attributed a time stamp and recorded by the system. These actions included: interactions with the inquiry support widgets, interactions with the simulation including changing simulation variable values and running/pausing/resetting the simulation, and transitioning between inquiry tasks [18].

Given that we are mostly interested in the discovery of self-regulated learning, the fact that students had a moderate degree of freedom to choose their own procedures for conducting experiments, less than in purely exploratory learning environments though [19], was an interesting data set for studying sequences of student behaviours and how they correlate with student success.

3.1 Process Mining and Discovery

Process mining offers a set of techniques and tools to discover sequential patterns represented as workflows. The analysis in this section was performed using the *Inductive visual miner* [10]. We were interested to discover, from the log of students inquiry activities, similar process models to the one depicted in Figure 1. For this discovery analysis, we limited ourselves to the whole data set, and we did not try to distinguish between groups of students. The purpose was to explore and compare the actual processes that students followed to the expected process from the author of the learning environment given in Figure 1, rather than suggest alternative learning processes. The log file contained 29679 events for 147 students. The overall distribution of inquiry activities indicated that 58.1% were spent in analysis, 19.1% in experiment, 18.4% in hypothesis formation, and 4.4% in observation.

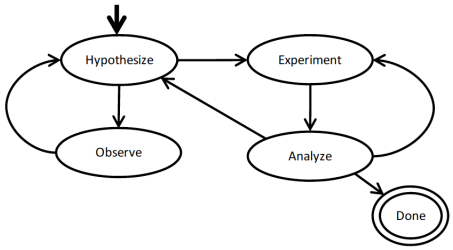


Figure 1: Intended learning paths during scientific inquiry.

As indicated in Figure 1, the intended learning process contains many possible loops while students progress in their scientific inquiry. Figure 2 and Figure 3 show respectively discovered process models from the transactions log using 100% of the events and sequences, and the top 70% most frequent events and sequences. From the visual comparison of the process model for 100% of the data (Figure 2), and the intended process of Figure 1, it is clear that there is a lot of variability in students transitioning between inquiry steps, given that the model is mostly disjunctive, with sequences resulting from loops. However, after leaving out the 30% most infrequent events and event sequences from the data, we discover a process model, Figure 3, that has some resemblance to the intended inquiry process, representing explicitly the sequence of hypothesize to experiment or analyze. Notice that the observation inquiry step is not part of the model because of the low frequency of its related events, which indicates a difference with the intended learning process, or more accurately, a tendency by the students to avoid the observation stage.

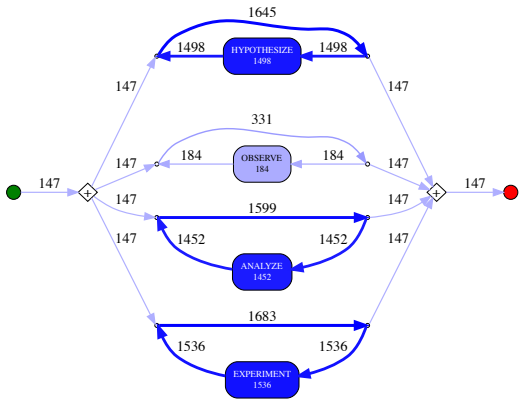


Figure 2: Process model using 100% of events and sequences (from top to bottom: hypothesize, observe, analyse, experiment).

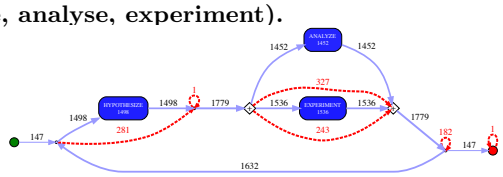


Figure 3: Process model using 70% of most frequent events and sequences (from left to right: hypothesize, analyse (top), experiment (bottom)).

Another element of interest was the sequence of problems students address during their inquiry. The overall distribution of student activities within those problems were relatively balanced with 30.7% in "container size", 24.9% in "amount of substance", 23.0% in "level of heat", and 21.4% in "cover status". Figure 4 shows a process model including 100% of events and event sequences. The process model clearly indicates a bias towards starting from the container size problem, followed by equivalent choices from the three other problems. This is likely a consequence of the the container size being the default value at the start of the inquiry session, which is a restriction on the student self-regulated learning processes.

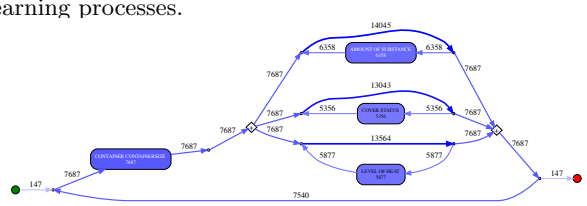


Figure 4: Process model of problems sequence using 100% of events and event sequences (from left to right: container size, amount of substance (top), cover status (middle), level of heat (bottom)).

Interestingly though, one would expect that the inquiry steps would be grouped (follow each other closely) within each problem. An inspection of a process model for an event classifier including the combination of both inquiry steps (hypothesize, observe, experiment, analyze) and problems (container size, amount of substance,level of heat, cover status) with 100% of events and sequences reveals only three groups of steps and not four as one would expect. In Figure 5, 1) the leftmost group is focused on inquiry steps applied to container size, and amount of substance, 2) the middle group to level of heat, amount of substance, and cover status, and 3) the rightmost group to cover status. This distribution of steps indicates that the four problems were not explored completely independently by the students, which manifest a strategy to explore concurrently the effect of different factors. However, this strategy might be different when comparing students with good and poor results and should be explored in a subsequent analysis.

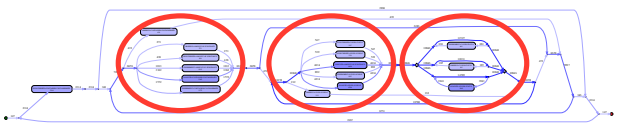


Figure 5: Three groups of problems and inquiry steps combination sequences.

3.2 Sequence classification

The second phase of our study was to explore the potential of sequential pattern mining in the identification of the level of skill exhibited by each student. Since sequences of student activity in the data set were not explicitly labelled as "skilled", "unskilled", etc., we considered two other metrics to measure skill exhibited: 1) number of times the student got an answer wrong, and 2) total time taken to complete the experiments. We used leave-one-out cross validation, applying our sequence classification learning algorithms on the training set and attempting to classify each test student as having either the high/low number of incorrect answers, or high/low time to complete, depending on the test.

Figure 6 shows the results of classifying students as "high number of incorrect steps". Success of the classifiers are measured by likelihood ratio (LR), which indicates how much more likely a positive example will be classified as positive than a negative example. The left-hand chart shows the success in classifying whether a student is in the bottom 50% in terms of number of incorrect answers, for varying maximum sequence size. Thus, a maximum sequence size of 1 represents the case where sequential relations are not considered, and only the presence/absence of certain actions are used for the classification. Observe that the LR is close to 1 in this case, meaning that we are no more likely to classify a positive case as positive or negative. The LR then increases steeply by 230% to 2.3 as sequences of size 2 are considered, before levelling off at about 1.75 for size 3 and greater. The right-hand chart then demonstrates how the classifier improves as we use sequences (max size 4) to classify students into the categories of worst 50%, 40%, 30%, 20% and 10%. Figure 7 depicts the results similarly for classifying students as "long time to complete". While not as dramatic, the positive effect of utilizing sequential information is demonstrated here as well.

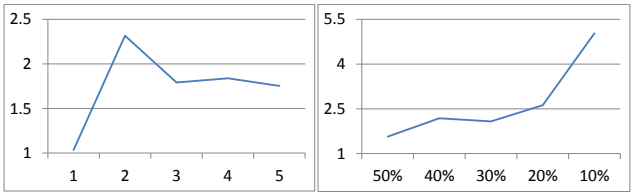


Figure 6: LR for classifying as "high number of incorrect steps".

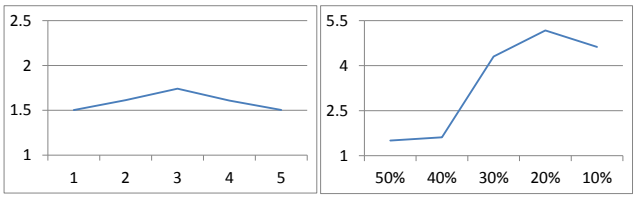


Figure 7: LR for classifying as "long time to complete".

4. CONCLUSION

One of the main thrusts within the Learning Performance Support Systems program (LPSS) at the National Research Council Canada seeks to advance and apply educational data mining to model and support self-regulated learning in heterogeneous environments of learning content, activities, and social networks. The program is at an early stage of development and our initial position points towards a complementary use of latent knowledge estimation and performance prediction methods [3], and sequence mining methods. In order to support the validity of our argument that sequential data analytics holds great potential for the analysis of student knowledge and skill acquisition, we demonstrated the application of discovery process mining and sequence mining in classifying students according to success using a data set of semi-structured learning activities [17, 18, 19] taken from the Pittsburgh Science of Learning Center DataShop [8].

Using process mining tools we were able to discover in-

quiry learning patterns in relationships with inquiry learning steps, learning problems, and a combination of those. Our analysis showed some differences between the semi-structured process intended by the developers of the learning environment and the actual processes followed by the students. We also showed that process mining techniques allow for the discovery of learning processes, and that considering sequences of events as features we can improve classification by up to 230% over considering single, non-sequential events. Given the learning process patterns discovered in the initial analysis of the students inquiry activity log, the next process mining discovery analysis will be to compare the inquiry processes of students having low and high correct outcomes.

5. ACKNOWLEDGEMENT

We would like to thank the Pittsburgh Science of Learning Center for providing the data supporting this analysis. We used the 'Science Sim State Change January 2010' data set accessed via the PSLC DataShop [8]. We thank Ken Koedinger from Carnegie Mellon for his help in choosing this data set. This work is part of the National Research Council Canada program Learning and Performance Support Systems (LPSS), which addresses training, development and performance support in all industry sectors, including education, oil and gas, policing, military and medical devices.

6. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th Int'l Conference on Data Engineering*, pages 3–14. IEEE, 1995.

[2] C. M Antunes and A. L. Oliveira. Temporal data mining: An overview. In *KDD workshop on temporal data mining*, pages 1–13, 2001.

[3] R. S. Baker and A. T. Corbett. Assessment of robust learning with educational data mining. *Research and Practice in Assessment*, 9:38–50, 2014.

[4] R. Bergenthum, J. Desel, A. Harrer, and S. Mauser. Modeling and mining of learnflows. In K. Jensen, S. Donatelli, and J. Kleijn, editors, *LNCS Transactions on Petri Nets and Other Models of Concurrency*. Springer, Springer-Verlag : Berlin Heidelberg, 2012.

[5] M. C. Desmarais and R. S. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modelling and User-adapted Interaction*, 22:9–38, 2012.

[6] H. Duong, L. Zhu, Y. Wang, and N. T. Heffernan. A prediction model that uses the sequence of attempts and hints to better predict knowledge: "better to attempt the problem first, rather than ask for a hint". In *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013* [6], pages 316–317.

[7] J. S. Kinnebrew, K.M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5:190–219, 2009.

[8] K.R. Koedinger, R.S.J.d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. *A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining*. CRC Press, Boca Raton, FL, 2010.

[9] S. Laxman and P. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

[10] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst. Process and deviation exploration with inductive visual miner. In *In Twelve International Conf. on Business Process Management, Accepted Demonstration 46*, Eindhoven, Netherlands, 2014.

[11] N. Lesh, M. J. Zaki, and M. Ogihara. Mining features for sequence classification. In *Proc. of the fifth ACM SIGKDD international conf. on Knowledge discovery and data mining*, pages 342–346. ACM, 1999.

[12] N. Mamoulis. Temporal data mining. In Ling Liu and M Tamer Özsu, editors, *Encyclopedia of Database Systems*. Springer, New York, NY, 2009.

[13] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences extended abstract. In *Proceedings the first Conference on Knowledge Discovery and Data Mining*, pages 210–215, 1995.

[14] R. Martinez, K. Yacef, J. Kay, A. Al-Qaraghuli, and A. Kharrufa. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *Proc. of the Fourth Int'l Conference on Educational Data Mining*, Eindhoven, Netherlands, 2011.

[15] M. Pechenizkiy, N. Trcka, P. De Bra, and P Toledo. Currim: Curriculum mining. In *Proceedings of the 4th International Conference on Educational Data Mining*, pages 216–217, 2012.

[16] D. Perera, J. Kay, I. Koprinska, K. Yasef, and O. Zaiane. Clustering and sequential pattern mining to support team learning. *IEEE Transactions on Knowledge and Data Engineering*, 21:759–772, 2009.

[17] M. Sao Pedro, R. Baker, and J. Gobert. Improving construct validity yields better models of systematic inquiry, even with less information. In J. Masthoff, B. Mobasher, M. Desmarais, and R. Nkambou, editors, *Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization*, pages 249–260, Montreal, Canada, 2012.

[18] M. Sao Pedro, R. Baker, and J. Gobert. What different kinds of stratification can reveal about the generalizability of data-mined skill assessment models. In J. Masthoff, B. Mobasher, M. Desmarais, and R. Nkambou, editors, *Proceedings of the 3rd Conference on Learning Analytics and Knowledge*, Leuven, Belgium, 2013.

[19] M. Sao Pedro, R. Baker, J. Gobert, O. Montalvo, and A. Nakama. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, 23:1–39, 2013.

[20] M.A. Sao Pedro. *Real-time Assessment, Prediction, and Scaffolding of Middle School Students' Data Collection Skills within Physical Science Simulations*, 2013.

[21] W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.

Desirable Difficulty and Other Predictors of Effective Item Orderings

Steven Tang Hannah Gogel Elizabeth McBride Zachary A. Pardos
University of California, Berkeley
Tolman Hall
Berkeley, CA, USA
{steventang, hgogel, bethmcbr, pardos} @berkeley.edu

ABSTRACT
Online adaptive tutoring systems are increasingly being used in classrooms as a way to provide guided learning for students. Such tutors have the potential to provide tailored feedback based on specific student needs and misunderstandings. Bayesian knowledge tracing (BKT) is used to model student knowledge when knowledge is assumed to be changing throughout a single assessment period. The basic BKT model assumes that the chance a student transitions from "not knowing" to "knowing" after each item is the same, with each item in the tutor considered a learning opportunity. It could be the case, however, that learning is actually context sensitive; context in our analysis is the order in which the items were administered. In this paper, we use BKT models to find such context sensitive transition probabilities in a mathematics tutoring system and offer a methodology to test the significance of our model based findings. We employ cross validation techniques to find models where including item ordering context improves predictive capability compared to the base BKT models. We then use regression testing to try to find features that may predict the effectiveness of an item ordering.

Keywords
Item Ordering, Bayesian Knowledge Tracing, Item Difficulty

1. INTRODUCTION
Online adaptive tutors are increasingly being used in classrooms as supplements to traditional instruction. Some systems, such as the ASSISTments [4] platform used for middle school math subjects, provide scaffolding or hints to students upon request or when the student answers a question incorrectly. In this paper, we focus on employing the Bayesian knowledge tracing (BKT) model of student learning but with the hypothesis that learning could be *context sensitive*. In this case, the context is the order that items of a particular skill are administered in.

2. BACKGROUND
2.1 ASSISTments Data
The data set analyzed in this paper comes from use of the ASSISTments platform in AY 2012-2013. The data set is publicly available and is rich with information that has been mined by other research projects [7] [9]. In this paper, we focus on the *Skill Builder* sequences used in ASSISTments, where a problem set consists of items given in a random order, generated from a set of templates. Items generated from these templates are assumed to be answerable with knowledge of a single underlying knowledge component (KC). For example, one problem set might contain three item templates. Each template can be populated with a set of numbers to generate an item; thus many different items can be derived from a single template. The number of templates per problem set varies; in this paper, we look at problem sets with between 2 and 6 templates. The number of items delivered to the student depends on the student's performance; in the Skill Builder set, mastery is assumed to occur after three consecutive correct responses. Each template in a Skill Builder sequence has an associated method of assistance; it is either a *hint* template or a *scaffolding* template. Scaffolding templates are bundled with a set of simpler questions to guide the student through the ideas in the item, while hint templates have guiding statements available to assist the students (usually the final hint provides the exact answer to the item).

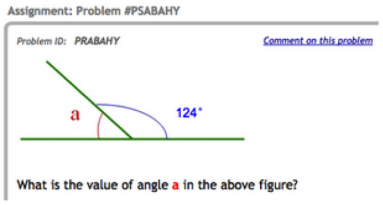


Figure 1: Example of an item in the ASSISTments database

3. METHODS AND ANALYSIS
3.1 Bayesian Knowledge Tracing
Bayesian knowledge tracing [3] assumes a binary representation of student knowledge. Figure 2 depicts a BKT model representation as a hidden Markov model (HMM). The basic BKT model is shown inside the dashed portion of the figure. O_1 through O_4 are binary indicators of correctness at opportunities 1 through 4. K_1 through K_4 represent the latent knowledge of the KC (assumed to be 0 or 1) at opportunities 1 through 4. In between each K_i and K_{i+1}, there is an arrow representing a probability of *transition*, or learning. Guess and slip parameters can be assumed to be equal among all items or can be item-specific [10].

3.2 Item Ordering Effects
The Skill Builder sequences in the ASSISTments platform pick from a set of templates at random to generate items for

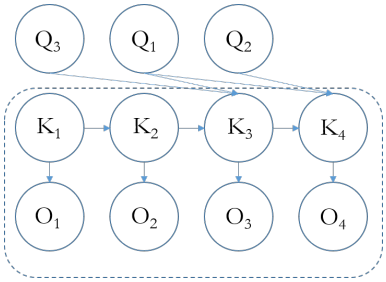


Figure 2: BKT Model. The dashed portion represents the basic BKT model, and the Q nodes represent the item order modification.

the student. However, it is our hypothesis that there may exist pedagogically more advantageous orderings of problems than the default random orders. Data mining and learning analytics techniques have been used to create process models and determine the most effective order of events for learners in online science education [8], as well as for finding patterns where students exhibited patterns of self-regulated learning [6]. Investigating the effects of item ordering can help both researchers and teachers, bridging the gap between educational theory and practice.

The BKT model could be extended to model a transition probability per particular item ordering. For example, one student might receive items from templates in the order of (3, 1, 2, ...) while another student might receive items from templates in the order of (1, 3, 2, ...). Over a number of such permutations, the BKT model could estimate a separate transition probability associated with items in the order (3, 1) as opposed to (1, 3). Figure 2 depicts how this new model might be formulated as an HMM, where items in the order of (3, 1, 2) are seen by the student. Note that the probability of knowledge at K_3 is influenced by seeing question 3 followed by question 1. Other students will be given items in different and random orders, allowing for all possible combinations of item order pairs to be analyzed. This model is drawn from work by Pardos and Heffernan [9]. We extend this work by finding significant improvements in predictive accuracy with the item order model by looking at the mean absolute errors produced by both the basic BKT and the item order model.

3.3 BKT model fitting

Among the Skill Builder response sets (SBs) from the 2012-2013 ASSISTments data set, we only looked at sets with more than 2000 student responses, more than 250 students, and between 2 and 6 (inclusive) templates. There were 112 Skill Builders that met these criteria, with 130,496 student response streams and 606,948 responses. Two BKT models, estimated using the XBKT code base, were fit to each of the 112 SBs. The first model was standard BKT (baseline), where every item was assumed to have the same transition probability. In our standard BKT model, every template type was allowed to have its own guess and slip parameters. The second model allowed for both different guess and slips per template and different transition probabilities based on the previous two items administered. We enabled different guess and slips per template for our baseline model so that any difference between models would be attributed to the different item order learning transitions. Additionally, we modeled a transition probability for each template specifically when that template was the first item administered in the sequence.

3.4 CV prediction to identify item orders of interest

To obtain statistical confidence in the generalization of a certain item ordering to unobserved students, we performed 5-fold cross validation (CV) on the data. This process starts by fitting both base and item order BKT models on a randomly selected 80% of student response data, and then using the trained models to predict student responses in the held out 20%, called the test set.

By comparing the predicted responses to the actual responses, Mean Absolute Errors (MAE) were obtained for both the base and the item order models. The error rates were then compared using a paired t-test for each possible item order. Out of the 1789 possible item orders among all Skill Builder problem sets, 605 item orders were found to have statistically significant error differences between the two predictive models at the .05 level. Among the 605 item orders, 157 had their responses predicted better by the base BKT model (by an average rate of .0138), while the remaining 448 item orders had their responses predicted better when using the item order model (by an average rate of .0173). It is important to note that the item orders in this section include ordering situations where the same template is administered twice in a row. The result that a portion of the item orders had better response prediction when using the base BKT model is not surprising, considering that each addition of a single new template to an SB increases the number of potential item orders dramatically. Thus, as the number of templates increases, the number of responses per item order decreases, resulting in less data per parameter for the model to learn from. The occurrence of 448 item orders whose responses were better predicted by the item order model suggests that the item order model could be able to uncover effective (or ineffective) item orderings.

Figure 3 shows the distribution of learn rates from both the basic and the item order BKT models. In the basic BKT model, a learn rate represents the rate at which a student is expected to learn (if they did not already know it) the latent knowledge component after seeing any item. In the item order BKT model, learn rates are modeled per item order pair, thus representing the rate a student is expected to learn a knowledge component after seeing a particular order of two items. The combination of the item order model with the cross validation approach provides a procedure that can determine when the item order model provides more accurate predictions compared to the base BKT model. Such a procedure can reveal when an item ordering might be considered effective or ineffective.

3.5 Regression analysis

Regression analyses (212,858 student responses) were run on the 448 item orders found to be significantly better fitting from the cross validation approach in order to find predictors of the item order learn rates. For the regression analyses,

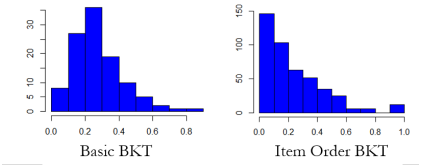


Figure 3: Distribution of learn rates

we extracted template level features from both templates in an item order. Features included are: average time to first response (milliseconds), percent correct on first problem attempt, average number of attempts, problem type (text response or radio button/multiple choice), difference in time to first response between Template A and Template B (where Template A is the first item in an ordering), difference in percent correct between Template A and Template B, whether Template A offered hints or scaffolding as assistance, and the *individual* learn rates for Templates A and B.

In our first model, stepwise regression was used regressing item order learn rate on these features (R^2=.17, F = 46.19, p<.01). The only features that were found to be significant at the .05 level were the learn rate of Template A, which had a negative effect on item order learn rate for the pair (β = -0.13, p = .01), and the learn rate of Template B, which had a positive effect (β = .502, p < .01) in the model.

Our second model only included item orderings where Template A was a scaffolding problem (R^2=.37, F = 11.47, p < .01). All of the features from the first model were included except for problem type due to lack of variation. Features unique to scaffolding problems were added as potential predictors: problem type of the associated sub-questions and percentage of scaffolding problems (including sub-questions) answered correctly. Average attempts on Template A (β = .93, p < .01) and the learn rate for Template B (β = .58, p < .01) had a positive effect on the item order learn rate. When the scaffolding for Template A consisted of text responses, the learn rate of the ordering decreased (β = -.13, p < .01).

The third model was fit using only orders where Template A was a hint item (R^2=.22, F = 20.94, p < .01). Hint features included percentage of students who went through all the hints on Template A and average amount of template hints seen. Average number of attempts on Template A (β = .27, p < .01), average milliseconds to first response on Template A (β = < .01, p = .03), percentage of students who accessed all of the hints on Template A (β = .71, p < .01), learn rate of Template A (β = -0.16, p <.01), and the learn rate for Template B (β = .43, p < .01) were significant predictors.

Regression analyses were also conducted to look for feature predictors of individual template learn rates for the 321 individual templates included in these 448 orderings. Percent correct on the template (β = .31, SE = .1, p < .01) and the item requiring a text response (β = .14, SE = .04, p < .01) were significant predictors (R^2=.06, F = 10.84, p < .01).

The primary unexpected result from the regression findings is that a *lower* learn rate of Template A predicts a higher learn rate for the ordering. It is important to note that this effect may be due to constraints in our current model. The individual learn rate of Template A is calculated when Template A occurs as the first item in a problem set presented to a student. That Template A is also included as part of an item ordering pair made up of the first and second items in the administered problem set. If the learn parameter for Template A is high, the knowledge component is already known (and has already been learned) by the time we consider the learn rate for the ordering including Template A. However, this phenomenon does not occur for template B of the item ordering, as Template B would not be the first template seen by the student in this case. In order to alleviate the discrepancy between the correlations, single template learn rates should be calculated from all template occurrences throughout administration in future work.

3.6 Desirable difficulty

In previous proof-of-concept work [11], a qualitative analysis was performed to examine what might make certain item orderings more effective than other item orderings. One feature of item pairs that became obvious was that not all items had exactly the same level of difficulty. In addition, some effective orderings contain a harder item first whereas other effective orderings contain an easier item first. One potential hypothesis that can help explain this difference in item ordering and difficulty is that of "desirable difficulties". In a series of studies, Bjork and colleagues determined that some challenges to performance during learning activities may actually contribute to greater learning [1] [2] [5]. By introducing "desirable difficulties" that help learners engage in the active processing of information, learning tasks that may be perceived as challenging or inefficient may prove more beneficial in the long run than those completed with high fluency.

In the case of item orderings where the first problem is more difficult than the second, the first (more difficult) problem may introduce a desirable difficulty, leading the student to learn more than they would with an easier problem. This learning then carries over into the second problem in the pair, thus leading to a higher overall rate of learning. This hypothesis works towards explaining our finding that a lower learn rate of the first template predicts a higher learn rate for an item ordering. When the first problem is easier than the second, this might be an instance where the material is better learned through a gentler or simpler introduction, as perhaps the second problem might be more difficult than is "desirable". In this case, a student would not properly learn from the more difficult problem unless it were preceded by an easier problem that would serve as a scaffold.

Using data from the BKT model to examine this hypothesis, we looked at how the difference between prior knowledge (at the start of an SB) and the percent correct on a template (as a proxy for template difficulty) compared to the probability of learning using regression. Finding *no difference* between a student's prior knowledge and the percent correct for a given template might show when an item has an "appropriate" difficulty. In this case, the difficulty of the item closely matches the prior knowledge of the student. Pedagogically, for an item to help the student learn, the difference between the student's prior knowledge and the item difficulty should be negative; in other words, the difficulty of the item should

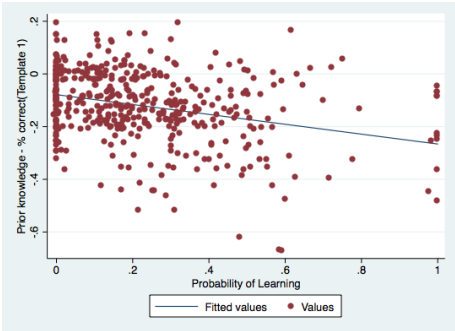


Figure 4: Scatterplot using template A data

be above the level of the student's prior knowledge to promote learning.

Regressing the difference between prior knowledge and item difficulty (percent correct) on the probability of learning showed statistical significance at the 0.01 level. This statistical significance held when using the difference between prior knowledge at the beginning of an SB and the percent correct on the first item in a pair (Template A), as well as the difference between prior knowledge and the percent correct for the second item in the pair (Template B). Using the percent correct for Template A to find the difference between the student's prior knowledge and the item difficulty had a correlation of -0.3039 with the probability of learning, while using Template B had a -0.2146 correlation with the probability of learning. These correlations are both relatively high, showing enough relationship between the variables to warrant further exploration in this area.

Similar to the correlations, the regressions were also run using percent correct from Template A and from Template B in the difference between prior knowledge and item difficulty. For Template A the coefficient for regressing the difference between prior knowledge and item difficulty (percent correct) on the probability of learning was -0.187 (R^2 =0.09, F=45.39); using template B, the coefficient was -0.120 (R^2= 0.046, F=21.53). The negative correlations, as well as negative coefficients in each of the regressions, show that the more negative the difference between prior knowledge and item difficulty becomes (the larger the difference between these two variables in the right direction for a "desirable difficulty"), the greater the probability of learning becomes. A scatterplot showing the relationship between these variables can be seen in Figure 4.

4. LIMITATIONS AND FUTURE WORK

The findings from this paper suggest that the item order BKT model combined with the use of a cross-validation technique show promise in uncovering learning mechanisms not apparent when just the base BKT model is used. The cross-validation approach confirmed that some item order models had better predictive capabilities compared to the base BKT models. Thus, statistically reliable suggestions can be made about item order delivery, and more research into item ordering is warranted, especially using such a cross-validation approach.

The results from the regression were somewhat surprising, where a lower individual learn rate from the first template in an ordering predicted a higher overall learn rate for the ordering. We hypothesize that this could be due to a constraint in our item order model, where individual learn rates of templates were modeled using only instances of that item when it appeared as the first item in a sequence. This hypothesis can be investigated in future research using a modified item order model.

5. REFERENCES

[1] M. Anderson, J. Neely, E. Bjork, and R. Bjork. *Memory, Chapter 6: Interference and inhibition in memory retrieval.* Academic Press, 1996.

[2] R. W. Christina and R. A. Bjork. *In D. Druckman and R. A. Bjork (Eds.), In the mind's eye: Enhancing human performance: Optimizing long-term retention and transfer.* Washington, DC: National Academy Press, 1991.

[3] A. T. Corbett and J. R. Anderson. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *The Journal of User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[4] M. Feng, N. Heffernan, and K. R. Koedinger. Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *The Journal of User Modeling and User-Adapted Interaction*, 19:243–266.

[5] V. Halamish and R. Bjork. When does testing enhance retention? A distribution-based interpretation of retrieval as a memory modifier. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37.

[6] K. L. J.S. Kinnebrew and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *JEDM-Journal of Educational Data Mining*, 4:190–219, 2013.

[7] J. Ocumpaugh, R. Baker, G. S., N. Heffernan, and C. Heffernan. Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*, 45(3):487–501, 2014.

[8] L. M. P. Reimann and M. Bannert. e-Research and learning theory: What do sequence and process mining methods contribute? *British Journal of Educational Technology*, 45:528–540, 2014.

[9] Z. A. Pardos and N. T. Heffernan. Determining the significance of item order in randomized problem sets. In *Proc. of the 2nd International Conference on Educational Data Mining*, 2009.

[10] Z. A. Pardos and N. T. Heffernan. KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP)*, 2010.

[11] S. Tang, E. McBride, H. Gogel, and Z. A. Pardos. Item ordering effects with qualitative explanations using online adaptive tutoring data. In *Proceedings of Works-in-progress at the second ACM conference on Learning@ scale*, 2015.

Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities

Ran Liu
Human-Computer Interaction Institute
Carnegie Mellon University
ranliu@cmu.edu

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
koedinger@cmu.edu

ABSTRACT

A growing body of research suggests that accounting for student-specific variability in educational data can improve modeling accuracy and may have implications for individualizing instruction. The Additive Factors Model (AFM), a logistic regression model used to fit educational data and discover/refine skill models of learning, contains a parameter that individualizes for overall student ability but not for student learning rate. Here, we show that adding a per-student learning rate parameter to AFM overall does not improve predictive accuracy. In contrast, classifying students into three "learning rate" groups using residual error patterns, and adding a per-group learning rate parameter to AFM, substantially and consistently improves predictive accuracy across 8 datasets spanning the domains of Geometry, Algebra, English grammar, and Statistics. In a subset of datasets for which there are pre- and post-test data, we observe a systematic relationship between learning rate group and pre-to-post-test gains. This suggests there is both predictive power and external validity in modeling these distinct learning rate groups.

Keywords

Student learning rate, learning curves, Additive Factors Model

1. INTRODUCTION

A growing body of research suggests that accounting for student-specific variability in statistical models of educational data can yield prediction improvements and may potentially inform instruction. The majority of work investigating the effects of student-specific parameters [6, 10, 11, 15] has been done in the context of a class of models called Bayesian Knowledge Tracing (BKT), a special case of using Hidden Markov Models to model student knowledge as a latent variable.

Logistic regression is another popular method for modeling educational data. The Additive Factors Model (AFM) [4] is one instantiation of logistic regression that was developed with the primary intention of evaluating, discovering, and refining *knowledge component (KC) models* (also referred to as Q-matrices). In contrast to *statistical models* of educational data, KC models define the knowledge components (e.g., skills, concepts, facts) on which estimates of students' knowledge are based. AFM has parameters modeling KC difficulty, KC learning rate, and individual student ability, but it does not have a parameter for individual student *learning rate*.

Recent work extending BKT models [15] suggests that better predictive accuracy is achieved by adding parameters that accommodate different learning rates for different students. Here, we investigate two different extensions of AFM that model student learning rate variability. The first model (AFM+StudRate) adds a per-student learning rate parameter to AFM, dramatically increasing the number of parameters in the model. We find some evidence that this model overfits the training data. For the second model (AFM+GroupRate), we introduce a method of classifying students into learning rate groups. We then add a per-group, rather than per-student, learning rate parameter to AFM and show that this model significantly outperforms regular AFM in predictive accuracy across 8 datasets spanning various domains.

Importantly, we move beyond simply evaluating the models in terms of their predictive accuracy to assess the external validity of the additional parameters. We show that they relate significantly to post-test outcomes. Validation and interpretation of statistical model parameter fits are a critical step towards successfully bridging EDM, the science of learning, and instruction.

1.1 The Additive Factors Model

AFM is a logistic regression model that extends item response theory by incorporating a growth or learning term. This statistical model (Equation 1) gives the probability p_{ij} that a student i will get a problem step j correct based on the student's baseline ability (θ_i), the baseline difficulty (β_k) of the required knowledge components or KCs on that problem step (Q_{jk}), and the improvement (γ_k) in each of the required KCs with each additional practice opportunity multiplied by the number of practice opportunities (T_{ik}) the student has had with that KC prior to the current problem step [4].

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k\in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik}) \quad (1)$$

AFM accommodates some individualization with the student ability parameter but makes the simplifying assumption that students learn at the same *rate*, since the original purpose of AFM was to refine KC models [4]. Here, we investigate whether extensions of AFM can accommodate variability in student learning rates and provide meaningful information about learning rate differences.

2. IDENTIFYING AND MODELING LEARNING RATE VARIATION

To explore adding learning rate variation to AFM, we created two new models extending AFM. The first model (AFM+StudRate) adds a per-student learning rate parameter, and the second model (AFM+GroupRate) adds a per-group learning rate parameter whereby membership among the three groups is determined using the method described in Section 2.1.

2.1 Student classification method

To classify students, we sought to identify those who improve—with each practice opportunity—more (or less) so than would be predicted by traditional AFM, which has a per-KC rate parameter that already accounts for the learning rate variability that is predicted by the KCs present at each opportunity. To do so, we examined the patterns in residual errors across opportunity counts after the data are fit with traditional AFM. A student whose learning curve is steeper than that predicted by AFM will exhibit

systematically increasing residual errors; i.e., residuals will correlate positively with opportunity count. Conversely, a student whose performance consistently increases *less* per opportunity than AFM predicts will exhibit a negative correlation between residual error and opportunity count.

To leverage this feature of residual error to classify students, we first fit the baseline AFM model to a full dataset (all students and KCs). Then, for each individual student, deviance residuals were computed, comparing the AFM model prediction against the actual data. Correlation coefficient cut-offs were set for each dataset at r > 0.1 for the "steep" learning-curve group and r < -0.1 for the "flat/declining" learning-curve group. Based on exploratory analyses, we selected the most stringent cut-off that yielded reasonable group sizes (approximately 50% students classified into either the steep or flat groups). The remaining students, whose learning curves were reasonably captured by the per-KC learning rates specified in AFM, were classified into a third "regular" group.

2.2 AFM+StudRate and AFM+GroupRate

The model that extends AFM by adding a per-student learning rate (AFM+StudRate) is given in Equation 2. It contains the parameters of traditional AFM with an additional parameter capturing the improvement (δ_i) by each student with every additional practice opportunity. Here, T_{ik} represents the practice opportunity count of a given KC required for a problem step *j*.

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k\in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik} + \delta_i T_{ik}) \qquad (2)$$

The model that extends AFM by adding a per-group learning rate (AFM+GroupRate) is given in Equation 3.

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \theta_i + \sum_{k\in KCs} Q_{jk}(\beta_k + \gamma_k T_{ik} + \delta_c S_{ic} T_{ik}) \qquad (3)$$

It uses the same parameters as AFM+StudRate except that each student's improvement rate with each additional practice opportunity (δ_c) is derived from a per-group rate (and thus can only take on one of three different values). Each student's group membership is specified by S_{ic}, which takes on a value of 1 when the student *i* belongs to group *c* and a value of 0 otherwise.

3. EVALUATING MODELS FOR FIT AND PREDICTIVE ACCURACY

3.1 Datasets

To test these statistical models on real educational data and to compare their predictive accuracies, we applied them across 8 datasets from DataShop [8]: Geometry Area 96-97, Cog Model Discovery Experiment Spring 2010, Cog Model Discovery Experiment Spring 2011, Cog Model Discovery Experiment Fall 2011, Assistments Math 2008-2009 Symb-DFA, Self Explanation sch_a3329ee9 Winter 2008 CL, IWT Self-Explanation Study 1 Spring 2009, and Statistical Reasoning and Practice - Fall 2009. These span a variety of content domains: Geometry, Equation solving, Story problems, English grammar, and Statistics. All of these datasets are publicly available at http://pslcdatashop.org.

We selected datasets that had already undergone significant KC model refinement via both manual and automated methods [9].

3.2 Methods

Each dataset was pre-processed based on the single-skilled KC model that achieved the best item-stratified CV performance according to values reported on DataShop. Table 1 lists the names of the KC models used and the number of KCs in each model. The three AFM models were implemented in R with student ability (θ_i), KC difficulty (β_k), and all learning rate parameters modeled as random effects, since many datasets used here were characterized by non-uniform sparsity in student-KC pairings, due to the mastery-based adaptive nature of the tutors from which the data originate. Modeling the parameters as random effects also reduces the likelihood of over-fitting the data by keeping their estimates close to zero.

The sparsity found in mastery-based datasets is particularly extreme at high opportunity counts, and this introduces noise to our classification method, which is dependent on good resolution across opportunity counts. Thus, we employed a conservative and systematic opportunity count cut-off method prior to analyses. The number of observations at each opportunity count was totaled for each student. Counts at which the average observations per student was less than 1 *and* the number of observations for any single student was 1 or fewer were excluded. In other words, at the excluded opportunity counts, no student had more than 1 *total* observation, and the majority of students did not have any. This excluded a very small percentage of total observations; the percent of observations retained are reported in the "Opp Cut-off" column of Table 1. In addition, our grouping technique required at least 5 observations in order to run the residual-by-opportunity correlations, so students who performed fewer than 5 total problem steps were excluded from the analyses. The left-most column of Table 1 reports the number of students included (with the original N in parentheses).

Models were evaluated for each dataset using Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and cross-validation measures. Two types of cross-validation (CV) were assessed: item-stratified CV, in which different random folds contain different problem steps, and student-stratified CV, in which different random folds contain different students (i.e., the model is tested on "unseen" students). Due to the random nature of the folding process, we repeated ten runs of each type of 10-fold CV, and the mean RMSEs across each run were used to compute the overall means and standard errors (in parentheses) reported in Table 2. Any CV results in which AFM+StudRate or AFM+GroupRate significantly outperforms regular AFM (as assessed by p<0.05 in a paired t-test between mean RMSEs across the 10 runs) are denoted with stars.

3.3 Results

The results of fitting the three statistical models to all 8 datasets are summarized in the right-most columns of Table 1.

AFM with a per-student learning rate fails to perform consistently better than regular AFM either across metrics within any dataset or across datasets. With an extra parameter per student, AFM+StudRate naturally fits training data better, but the evaluation metrics indicate over-fitting that is likely idiosyncratic (i.e., resulting in parameter estimates that will not generalize well to "unseen" items or students). Even for the AIC metric, which incorporates a smaller penalty for extra parameters than BIC, AFM+StudRate is better than regular AFM for only half of the datasets and only slightly so. By BIC, it is better than regular AFM in only one dataset. Cross-validation reveals that AFM+StudRate fails to achieve significantly lower RMSEs than regular AFM in 14 of 16 cases.

In contrast, AFM+GroupRate performs best on *all* 8 datasets by AIC, BIC, and item-stratified CV measures. It also performs the best on the majority of datasets (6 out of 8) by student-stratified CV. The superior performance according to student-stratified CV is particularly notable, because the predictions are made on data from "unseen" students. That is, no student information (not even group membership) is available for the data in the test set. The

fact that AFM+GroupRate performs better than regular AFM implies that this model is successfully capturing some student-level variability that produces better, cleaner KC parameters. This is not true for AFM+StudRate, which did not achieve significantly better student-stratified CV for any dataset.

4. RELATIONSHIP TO PRE-POST GAINS

Predictive accuracy is often used as a proxy for quality in EDM models. Assessing the validity of these student groups beyond relevance to model-fitting is equally, if not more, important. To do so, we investigated the relationship between group membership and post-test outcomes. Four of the datasets tested in Section 3 contained pre/post-test data that were accessible via DataShop: the three geometry Cog Discovery datasets and the IWT 1 dataset.

For each dataset we ran a simple regression with both pre-test score and per-group coefficients (from fitting AFM+GroupRate) as predictors of post-test score. Even after taking into account the variance explained by pre-test scores, learning rate group membership predicts post-test scores significantly for Cog Discovery Spring 2010 ($p<0.001$), Cog Discovery Fall 2011 ($p=0.016$), and Cog Discovery Spring 2011 ($p<0.001$), and marginally significantly for IWT 1 ($p=0.077$). These results suggest that group classification predicts unique variance in post-test outcomes and is thus a valid and interpretable construct.

5. DISCUSSION

5.1 Conclusions and implications

In the present work, we investigated two extensions of AFM that incorporated learning rate variation: adding a per-student learning rate parameter (AFM+StudRate) and adding a per-group learning rate parameter (AFM+GroupRate). AFM+StudRate overall did not significantly improve upon regular AFM, according to predictive accuracy metrics. In contrast, the residual-based student grouping method we developed seems to capture meaningful differences in learning rate variations. The groups have internal validity: adding a per-group learning rate to AFM improved predictive accuracy across all datasets based on the vast majority of fit metrics. They also have external validity: per-group rate

Table 1. *Dataset details and predictive accuracy metrics for each of the three statistical models fit to datasets. The percent of observations retained for analyses are shown in parentheses underneath opportunity cut-off values. Item- and student-stratified CV values are mean RMSEs over 10 separate runs of 10-fold cross validation, with standard errors in parentheses. Stars denote models with significantly better cross-validation performance (at p<0.05 in paired t-tests of RMSE values across CV runs) than regular AFM. The best-performing models by each metric are bolded.*

Dataset [Domain] # Students	KC Model (# KCs)	Opp Cut-off	Statistical Model	AIC	BIC	Item-Strat CV RMSE	Student-Strat CV RMSE
Geometry 1996-97 [Geometry] N = 56 (of 59)	LFASearchAIC WholeModel3 (18)	27 (99.22%)	AFM	5039.7	5072.4	.3996 (.0003)	**.4063 (.001)**
			+StudRate	5043.8	5080.5	.3991 (.0004)	.4063 (.001)
			+GroupRate	**4999.2**	**5038.4**	**.3975 (.0003)***	.4068 (.001)
Cog Discovery Spring 2010 [Geometry] N = 123 (of 123)	KTskills.Mcontext.single.sep.ind.areas (42)	80 (99.72%)	AFM	29208.5	29251.7	.3238 (.00003)	.3319 (.0001)
			+StudRate	29160.8	29221.3	.3232 (.00002)*	.3318 (.0001)
			+GroupRate	**29030.1**	**29081.9**	**.3230 (.00002)***	**.3317 (.0001)***
Cog Discovery Spring 2011 [Geometry] N = 65 (of 69)	KTracedSkills.matched.Fall2011 (7)	30 (99.3%)	AFM	4099.7	4131.5	.3877 (.0002)	.4025 (.0004)
			+StudRate	4101.4	4146.0	.3879 (.0002)	.4025 (.0004)
			+GroupRate	**4077.3**	**4115.3**	**.3856 (.0002)***	**.4017 (.0004)***
Cog Discovery Fall 2011 [Geometry] N = 103 (of 103)	KTracedSkills.Concatenated (15)	26 (97.87%)	AFM	3175.9	3208.2	.3104 (.0003)	**.3194 (.0003)**
			+StudRate	3177.8	3223.0	.3108 (.0003)	.3198 (.0003)
			+GroupRate	**3155.6**	**3194.3**	**.3090 (.0002)***	.3198 (.0003)
Assistments Symb-DFA [Story Problems] N = 318 (of 318)	Main.LFASearch Model0 (4)	11 (98.81%)	AFM	6013.1	6046.0	.4265 (.0006)	.47008 (.0001)
			+StudRate	6016.9	6062.9	.4267 (.0006)	.47008 (.0001)
			+GroupRate	**5793.2**	**5832.7**	**.4166 (.0006)***	**.47005 (.0001)**
Self-Explanation Winter 2008 [Equation Solving] N = 70 (of 71)	LFASearchAIC Model.r2 (19)	49 (98.78%)	AFM	6201.8	6235.6	.3905 (.0002)	.4140 (.0005)
			+StudRate	6201.4	6248.8	.3906 (.0002)	.4141 (.0006)
			+GroupRate	**6158.9**	**6199.5**	**.3889 (.0002)***	**.4127 (.0005)***
IWT 1 Spring 2009 [English Grammar] N = 120 (of 120)	LFASearchAIC WholeModel1 (26)	11 (98.64%)	AFM	6820.8	6854.7	.4134 (.0003)	.4392 (.0002)
			+StudRate	6815.2	6862.7	.4128 (.0003)*	.4392 (.0002)
			+GroupRate	**6752.9**	**6793.6**	**.4099 (.0002)***	**.4389 (.0002)***
Statistics Fall 2009 [Statistics] N = 52 (of 52)	LFASearchAIC Model0 (16)	30 (99.81%)	AFM	2967.8	2999.4	.3090 (.0032)	.3250 (.0003)
			+StudRate	2965.5	3009.8	.3105 (.0031)	.3250 (.0004)
			+GroupRate	**2935.5**	**2973.5**	**.3085 (.0029)***	**.3248 (.0003)***

coefficients significantly predict each group's post-test outcomes, controlling for pre-test.

Despite the focus of the AFM+GroupRate model on student-level differences, adding the per-group rate parameter produces more accurate estimates of KC parameters, based on the model's superior performance in student-stratified CV for the vast majority of datasets. The only information the model gets for fitting test data in student-stratified CV ("unseen" students whom the model has no information about with respect to ability, learning rate, or group) are the KC parameters. For this reason, AFM+GroupRate may be useful for data-driven refinement of KC parameters, which in turn has implications for instruction (e.g., parameter-setting in Knowledge Tracing based cognitive tutors [14]).

Compared to other statistical models extending AFM (Performance Factors Analysis [12], Instructional Factors Analysis [5], Recent Performance Factors Analysis [7]), AFM+GroupRate adds relatively few parameters (only three) to AFM but achieves consistent and substantive improvements in prediction. These three parameters' coefficient estimates are consistently interpretable (the per-group learning rates are ordered according to intuitions about each group's learning curve steepness), and the model avoids overloading on the interpretation of parameters.

We conducted extensive post-hoc analyses to interpret what the three learning groups actually reveal about student behavior and did not find evidence that the groups detect learning speed as an inherent trait, per se. For example, high ability students did not tend to be in the "steep" group, and low ability students did not tend to be in the "flat" group. Rather, the amount of improvement per opportunity seems to differ, more generally, depending on where the learner is on his/her *true* learning curve for any given skill. That is, the improvement per opportunity may be different for the earliest opportunities on a skill than for much later opportunities on a skill. Different students' learning curves within cognitive tutor data may vary because they start using the cognitive tutor at different points of their true learning curves for any given skill, depending on their experience with that skill prior to tutor use. We found evidence supporting this notion in post-hoc analyses. Considered in conjunction with the lack of evidence for a per-student learning rate, our findings contradict the intuitive notion that some students naturally learn faster than others.

5.2 Limitations and future work

The present results somewhat conflict with a finding from [15] that adding a per-student learning rate parameter to BKT yields substantial improvements in model fit, though we note that that report did not provide an interpretation nor any external validity evidence. We did not observe a benefit when adding a per-student learning rate parameter to AFM. Further work to compare these per-student parameter estimates across AFM and BKT and to externally validate the estimates from individualized BKT will provide insight into this issue.

Based on our post-hoc analyses, classification into the "flat/declining" group seems to capture high-ability students who descend into noisy performance at late opportunity counts (indicating boredom and/or "gaming the system" [2]) and low-ability students who never seem to improve ("wheel spinners" [1]). It would be interesting to validate this by seeing whether the detectors in [1] and [2] yield the same students when tested within the present datasets.

Another avenue for future investigation is to assess the degree to which different learning rate groups would benefit optimally from different KC models, via KC model search (as in [13]).

6. ACKNOWLEDGMENTS

We thank Carnegie Learning, Inc. for providing the majority of the datasets that were analyzed, Christopher MacLellan for insightful discussion, IES for support to RL (training grant #R305B110003), and NSF for support to LearnLab (#SBE-0836012).

7. REFERENCES

[1] Beck, J.E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. AIED, 431-440.

[2] Baker, R.S.J.d., Corbett, A.T., Roll, I., & Koedinger, K.R. (2008). Developing a generalizable detector of when students game the system. User Modeling and User-Adapted Interaction, 18(3), 287-314.

[3] Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., & Graesser, A.C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. International Journal of Human-Computer Studies, 68(4), 223-241.

[4] Cen, H., Koedinger, K.R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. Intelligent Tutoring Systems, 164-175.

[5] Chi, M., Koedinger, K.R., Gordon, G., Jordan, P., & VanLehn, K. (2011). Instructional factors analysis: A cognitive model for multiple instructional interventions. 4th International Conference on EDM.

[6] Corbett, A.T., & Anderson, J.R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278.

[7] Galyardt, A., & Goldin, I. M. (accepted). Move your lamp post: Recent data reflects learner knowledge better than older data. Journal of Educational Data Mining.

[8] Koedinger, K.R., Baker, R.S., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

[9] Koedinger, K.R., McLaughlin, E.A., & Stamper, J.C. (2012). Automated Student Model Improvement. 5th International Conference on EDM.

[10] Lee, J.I., & Brunskill, E. (2012). The Impact on Individualizing Student Models on Necessary Practice Opportunities. 5th International Conference on EDM.

[11] Pardos, Z.A., & Heffernan, N.T. (2010). Modeling individualization in a bayesian networks implementation of knowledge tracing. User Modeling, Adaptation, and Personalization, 255-266.

[12] Pavlik, P.I., Cen, H., & Koedinger, K.R. (2009). Performance factors analysis–a new alternative to knowledge tracing. AIED, 531–538.

[13] Rafferty, A.N., & Yudelson, M. (2007). Applying learning factors analysis to build stereotypic student models. Frontiers in Artificial Intelligence and Applications, 158, 697.

[14] Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. (2007). The Cognitive Tutor: Applied research in mathematics education. Psychonomics Bulletin & Review, 14(2), 249-255.

[15] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. (2013). Individualized bayesian knowledge tracing models. AIED, 171-180.

Evaluating The Relevance of Educational Videos using BKT and Big Data

Zachary MacHardy
UC Berkeley
354 Hearst Memorial Mining Building
Berkeley, CA 94720
zmmachar@cs.berkeley.edu

Zachary A. Pardos
UC Berkeley
4641 Tolman Hall
Berkeley, CA 94720
zp@berkeley.edu

ABSTRACT
Along with the advent of MOOCs and other online learning platforms such as Khan Academy, the role of online education has continued to grow in relation to that of traditional on-campus instruction. Rather than tackle the problem of evaluating large educational units such as entire online courses, this paper approaches a smaller problem: exploring a framework for evaluating more granular educational units, in this case, short educational videos. We have chosen to leverage an adaptation of traditional Bayesian Knowledge Tracing (BKT), intended to incorporate the usage of video content in addition to assessment activity. By exploring the change in predictive error when alternately including or omitting video activity, we suggest a metric for determining the relevance of videos to associated assessments. To validate our hypothesis and demonstrate the application of our proposed methods we use data obtained from both the popular Khan Academy website and two MOOCs offered by Stanford University in the summer of 2014.

Keywords
knowledge tracing, educational videos, instructional technology, bayesian inference, online education

1. INTRODUCTION
As the relative importance of MOOCs and other online learning platforms such as Khan Academy has increased, so has the importance of verifiably sound online pedagogy increased apace. While many of the lessons learned through a long history of research on the traditional classroom are applicable to the online environment, many indicators available during traditional instruction are not present for a designer of online material. In order to address the need for scalable and reproduceable evaluation, we hypothesize that by relating the use of materials and performance on subsequent assessment items, we can construct a metric to evaluate the relevance of those videos, without needing to resort to comparative studies.

To model student interactions with educational material and improvement over time, we have chosen to use an adaptation of Bayesian Knowledge Tracing (BKT), a technique developed and used with Intelligent Tutoring Systems (ITS) but which has been applied outside of that domain as well. We seek to incorporate behavior, such as video observation, which falls beyond the purview of attempting assessment items. We contrast this extended model with a simpler one excluding resource usage in order to discover whether videos contribute to model accuracy, and if some models benefit more than others.

Our ultimate goal is not to produce high predictive accuracy for the purposes of predicting students' latent knowledge, but rather to provide a quantitative framework for evaluating video resources. We set out first to prove that there is a reduction of predictive error when incorporating video resources into BKT analysis, in order to validate the inclusion of such observations. Second, we propose a metric based on a combination of both the delta in error between models using and eschewing video data and the learn rate associated with a particular video, in order to foreground both those which appear most relevant, as well as those which may need attention.

2. RELATED WORK
2.1 Bayesian Knowledge Tracing
Bayesian Knowledge Tracing [1] is used extensively in computer-assisted instruction environments, intended to approximate mastery learning. The model in its most basic form is defined by four parameters: $P(L_0)$, the prior probability that a student has mastered a particular KC, or knowledge component; $P(S)$, the probability a student who knows a concept will get an associated question wrong, or 'slip'; $P(G)$, the probability that a student who does not know a concept will correctly 'guess' the correct answer; and $P(T)$ the probability that a student who does not know a particular KC will learn it after a given observation. Through a process of Bayesian inference, an observed correct or incorrect response to an assessment item can be used to calculate a posterior probability that a student has mastered the KC. Using this posterior and $P(T)$ as described above, a new prior is calculated, accounting for the probability that the KC was learned between observations. This process is then repeated, using the updated estimate, for each subsequent observation.

We chose to use BKT as a modeling framework as it is well-studied and possesses relatively well understood properties, with parameters which are intuitively interpretable and therefore potentially actionable. Additional work has been done to extend this basic model of BKT to incorporate individualized parameters, based on factors depending both upon individual student properties (see e.g. [7], [2]), as well as properties of particular assessment items within a knowledge component [8].

Source	Total Events	Distinct KCs
Khan	353,202	176
Economics	689,709	94
Statistics	337,428	70

Table 1: Properties of the three sources

2.2 Online Course Resources

There has been a fair amount of research devoted to studying the efficacy of videos, forums, and other study aids offered in online educational contexts. Past work has typically focused on issues such as student attrition, student interaction, and building student-facing recommender systems. For example, Yang et al. described a framework for helping students sift through the the large volume of forum discussion posts in order to find content relevant to them [10]. Similar efforts have been made to provide recommendations for more general content, using methods such as social media analysis and reinforcement learning [5] [9].

Relative to the research on student perception and experience in the MOOC context, little attention has been paid to that of the instructor. That is not to say that such work has been absent. Guo et al. [3] and Kim et. al [4] offer guidance for the construction of videos used in MOOCs. Explorations of the application of Item Response theory in a MOOC environment [6] similarly offer instructors guidance in evaluating the efficacy of their assessments using traditional methods. Yousef et al. constructs an inventory of features, pedagogical and technological, which contribute to a sense of course quality. [11]. Yet there remains a relative paucity of research on the quantitative assessment of content outside of the scope of assessment items.

3. DATA

In order to demonstrate the generalizability of our results, we leveraged three sources of event log data. Two of our datasets were taken from Stanford Online courses run using the edX platform: 'Statistics and Medicine' and 'Principles of Economics.' The third was taken from the popular Khan Academy Website. See table 1 for details.

The data we obtained from Khan Academy contains observation events collected over about two years, from June 2012 to February 2014, while both edX courses were offered from June to September of 2014. Assessment items in Khan are categorized hierarchically as part of a larger 'exercise' representing a particular skill, and further as a member of a 'problem type,' describing the template used to generate a specific problem, while exercises from edX are categorized as individual problems. For the sake of simplicity we have chosen to consider each exercise as a separate knowledge component (KC) for the purposes of training BKT models.

For both the Khan and edX data, there was not an immediately available canonical mapping between videos and associated problems. By scanning the logs of learner activity and using a metric combining chronological proximity of use as well as frequency of associated observation, we produced a mapping between videos and their related KCs. Because our goal was not to produce a generative procedure for semantically associating log events, we chose our method

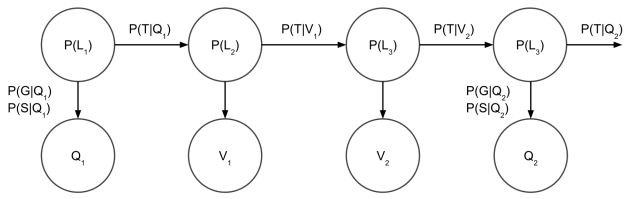


Figure 1: The Template-Videos Model

to be sufficiently successful without introducing unnecessary complexity. However, this does introduce possible sources of error in terms of both overlooked and spuriously constructed mappings.

4. METHODS

Though the previous section describes the fundamentals of Bayesian Knowledge Tracing, we employ several extensions to the model. First, and for all models used in evaluation, we condition $P(G)$ and $P(S)$ for each observation on which specific problem template is observed, to model varying template difficulty. We will refer to this model as 'Standard BKT'.

Second, we similarly condition the transition probability $P(T)$ on the observed problem template, generating a second distinct but still video-free 'Template' model. We include this model for the Khan data for the sake of completeness, but note that there is only a single template for each edX problem in the data and thus the results of this extension are omitted for both the 'Statistics and Medicine' and 'Principles of Economics' cases

Third, we extend our model to incorporate video observations, conditioning $P(T)$ either on the specific template observed or the specific video, generating the 'Template Videos' model. The presence of a video observation functions similarly to that of a problem attempt, save that as there is no associated student response to be considered, a video is associated only with a unique $P(T)$. We simplify the 'Template Videos' into a fourth 'Template 1 Video' model, conditioning $P(T)$ only on the presence of either a video or a question, but not the specific identity of the resource observed.

All models were trained and evaluated using 5-fold cross validation. For each model above, one BKT model was trained for each of the knowledge components. For each model, for each fold, each of the KC models was randomly initialized and trained using Expectation Maximization (EM) algorithm to minimize the log likelihood of the observed events 25 times, with the maximally likely resulting model chosen for that model-fold-model tuple. The metric used to compare the four models is the root mean squared error (RMSE) taken across all five folds.

5. RESULTS AND DISCUSSION

Tables 2, 3, and 4 describe the results of running the data through the three analytical models. In each case, the 'Template Videos' and 'Template 1 Video' models tended to perform best, while the 'Template' model, using the Khan Academy

data, showed no significant difference from the baseline distribution. The significance test is performed across the distribution of RMSE across each of the KC models in each data-set.

Model	Mean RMSE	Significance
Pct. Correct	.4930	.0000*
Standard BKT	.3824	—-
Template	.3824	.9448
Template Videos	.3810	.0253*
Template 1 Video	.3811	.0061*

Table 2: Khan Academy

Model	Mean RMSE	Significance
Pct. Correct	.6243	.0000*
Standard BKT	.3824	—-
Template Videos	.3715	.0000*
Template 1 Video	.3716	.0000*

Table 3: Principles of Economics

Model	Mean RMSE	Significance
Pct. Correct	.5551	.0000*
Standard BKT	.3711	—-
Template Videos	.3638	.0000*
Template 1 Video	.3642	.0000*

Table 4: Statistics and Medicine

Though the tables reflect changes in RMSE aggregated over all KC models, not all models benefited evenly from the inclusion of video resources. Among the Khan data 77 of 193 KCs saw more then a trivial amount of reduction in error, while in Statistics and Medicine and Economics, the bulk of the improvement could be seen in 57 of the 94 and 44 out of 70 models, respectively. This asymmetry of improvement is an expected behavior of the system. Intuitively, in the case that a particular video resource is either not helpful or actively harmful to a student in solving a particular problem or set of problems, this would be reflected in the trained model as additional noise, leaving the overall RMSE unaffected at best.

Rather, the presence of a statistically significant, though perhaps small, decrease in predictive error in some models is indicative of the soundness of the hypothesis that considering video usage can offer useful information.

5.1 Highest and Lowest Performing Models

In order to gain an intuition for why some models were better described by the inclusion of resources, we chose to consider a selection of the best and worst performers from each data set under the 'Template-Videos' condition. By examining what properties might explain the performance of each model, we seek insight into what sort of videos appear to offer the greatest benefits to student performance.

For the highest performing models in the Khan data, the videos appeared highly relevant to their associated exercises, often demonstrating solutions in the Khan interface. For example, 'The Fundamental Theorem of Arithmetic,' explains the manipulation of a bespoke tool created for that particular exercise, showing the completion of a practice problem using that tool.

For the low performing Khan models the possible sources of error mirror the effects seen in the high performing cases. 'Scalar Matrix Multiplication' and 'Linear Inequalities', for example, present video explanation very differently than their related videos and involve customized input fields, which may have been a source of trouble.

Though the Principles of Economics and Statistics in Medicine edX courses are formatted very differently than the lessons of Khan academy, the distinctions between the best and worst models are similar. In both cases, the best videos in the data-set are, while less compellingly visually similar than the Khan examples, pointedly related to the subsequent assessments. Additionally, most of the associated assessments allowed students only one attempt, explaining the particularly strong reduction in error when including video information.

Perhaps most interesting is that one of the best predicted models is the ninth question on the final exam of the 'Statistics and Medicine' course. The content of this question is nearly identical to content of the video from a couple of weeks previous, 'Practice Interpreting Linear Regression Results.' It is therefore unsurprising to find that the video, while not explicitly grouped with the exam, is associated with a very strong learn parameter; students who sought out the video succeed significantly more often on the assessment.

Two of the videos related to the worst models in the Economics set, 'The Spending Allocation Model', and 'The Fed and the Money Supply' are both relatively long, each over fifteen minutes. Despite their length, each video dwells only briefly on the subject concerned in the assessment, spending most of their running time on other topics, with the pertinent sections easy to skip or miss. Another worst performer is one of the first videos in the course, associated with a quiz with nearly a 90% correctness rate.

Intuitively, an unhelpful video does not contribute to a predictive model, simply adding additional complexity and noise. By measuring which videos do and do not contribute constructively to predictive accuracy, it may be possible to detect which videos might be most appropriately suggested as helpful for a learner, and which need revision. In particular, such results could be useful to an instructor or course manager in navigating what to improve and what to keep when iterating on a course between offerings.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated that the inclusion of video observations in a KT model can offer information relevant to predicting student behavior, not only in one dataset, but generalizably across multiple domains. Though the effect size is small, the statistically significant decrease in error under the 'Template 1 Video' and 'Template Videos' conditions across the three data-sets considered is an encouraging sign. It is indicative that there is information to be gleaned from a learner's use of video resources. Further,

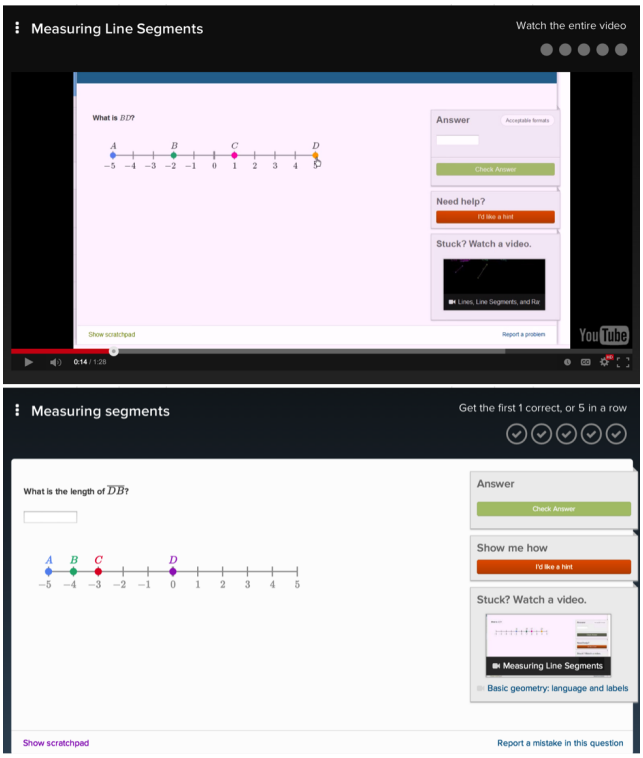


Figure 2: Videos from Khan Academy contributing maximally to model accuracy tended to closely mirror subsequent assessments

as suggested by our investigation of some of the superlative models, it is possible that the delta in error generated by a given model, coupled with the associated P(T) for a video within that model, could be a useful metric for evaluating video relevance.

One piece missing from this analysis is a canonical association of videos to exercises. Though we generated and used a set of associations, we may have lost information in the process. Another avenue worth pursuing is the possibility that some users would benefit strongly from video resources while others may not. To that end, it would be useful to examine potential reductions in error that might be made by individualizing parameters to each KC-Student pair.

An important caveat of this analysis is to note that our results do not speak to a general 'quality' of a video, and indeed that is perhaps beyond the scope of a quantitative analysis. A video rated poorly by our metrics need not necessarily be a bad video, merely unrelated or unhelpful for a subsequent assessment task. The importance of this particular property is a matter of educational policy, and thus beyond the scope of this paper. Our goal is not to supplant the role of instructor decisions in course management, only to support them.

7. 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, Dec. 1994.

[2] R. S. d Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[3] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 41–50, New York, NY, USA, 2014. ACM.

[4] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller. Understanding in-video dropouts and interaction peaks inonline lecture videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 31–40, New York, NY, USA, 2014. ACM.

[5] D. Kravvaris, G. Ntanis, and K. L. Kermanidis. Studying massive open online courses: recommendation in social media. In *Proceedings of the 17th Panhellenic Conference on Informatics*, pages 272–278. ACM, 2013.

[6] J. P. Meyer and S. Zhu. Fair and equitable measurement of student learning in moocs: An introduction to item response theory, scale linking, and score equating. *Research & Practice in Assessment*, 8(1):26–39, 2013.

[7] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[8] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, pages 243–254. Springer, 2011.

[9] V. Raghuveer, B. Tripathy, T. Singh, and S. Khanna. Reinforcement learning approach towards effective content recommendation in mooc environments. In *MOOC, Innovation and Technology in Education (MITE), 2014 IEEE International Conference on*, pages 285–289. IEEE, 2014.

[10] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.

[11] A. M. F. Yousef, M. A. Chatti, U. Schroeder, and M. Wosnitza. What drives a successful mooc? an empirical examination of criteria to assure design quality of moocs. In *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference On*, pages 44–48. IEEE, 2014.

Measuring Problem Solving Skills in Plants vs. Zombies 2

Valerie J. Shute
Florida State University
1114 West Call Street
Tallahassee, FL 32306
vshute@fsu.edu

Gregory R. Moore
Florida State University
1114 West Call Street
Tallahassee, FL 32306
grm13@my.fsu.edu

Lubin Wang
Florida State University
1114 West Call Street
Tallahassee, FL 32306
lw10e@fsu.edu

ABSTRACT

We are using stealth assessment, embedded in *Plants vs. Zombies 2*, to measure middle-school students' problem solving skills. This project started by developing a problem solving competency model based on a thorough review of the literature. Next, we identified relevant in-game indicators that would provide evidence about students' levels on the various problem-solving facets. Our problem solving model was implemented in the game via Bayesian networks. To validate the stealth assessment, we ran a small pilot study to collect data from students who played our game-based assessment and completed an external problem solving measure (*MicroDYN*). Preliminary results indicate that problem solving estimates derived from the game significantly correlate with the external measure, suggesting that our stealth assessment is valid. Our next steps include running a larger validation study (in progress) and developing tools to help educators interpret the results of the assessment.

Keywords

Stealth Assessment, Problem Solving, Game-Based Learning, Bayesian Networks

1. INTRODUCTION

In this paper, we describe the design, development, and preliminary validation of an assessment embedded in a video game to measure the problem solving skills of middle school students. After providing a brief background on stealth assessment and problem solving skills, we describe the game (*Plants vs. Zombies 2*) used to implement our stealth assessment, and discuss why it is good vehicle for assessing problem solving skills. Afterwards, we present the in-game indicators (i.e., gameplay evidence) of problem solving, describing how we decided on these indicators and how the indicators are used to collect data about the in-game actions of players. While discussing the indicators, we show how the evidence is used in a Bayesian network to produce an overall estimate for students' problem solving skills. We then discuss the results of a pilot validation study, which show that our stealth assessment estimate of problem solving significantly correlates with an external measure of problem solving (*MicroDYN*). We conclude with the next steps in developing the assessment and practical applications of this work.

2. BACKGROUND

2.1 Stealth Assessment

Good games are engaging, and engagement is important for learning. The challenge is validly and reliably measuring learning in games without disrupting engagement, and then leveraging that information to bolster learning. For the past 6-7 years, we have been researching various ways to embed valid assessments directly into games with a technology called *stealth assessment* (e.g., [15, 16, 20]). Stealth assessment is grounded in an assessment design framework called evidence-centered design (ECD) [10]. In general, the main purpose of any assessment is to collect information that will allow the assessor to make valid inferences about what people know, can do, and to what degree (collectively referred to as "competencies" in this paper). ECD defines a framework that consists of several conceptual and computational models that work in concert. The framework requires an assessor to: (a) define the claims to be made about learners' competencies, (b) establish what constitutes valid evidence of a claim, and (c) determine the nature and form of tasks or situations that will elicit that evidence.

Stealth assessment complements ECD by determining specific gameplay behaviors (specified in the evidence model and referred to as indicators) and linking them to the competency model [19]. As students interact with tasks/problems in a game during the solution process (see Figure 1), they are providing a continuous stream of data (captured in a log file, arrow 1) that is analyzed by the evidence model (arrow 2). The results of this analysis are data (e.g., scores) that are passed to the competency model, which statistically updates the claims about relevant competencies in the student model (arrow 3).

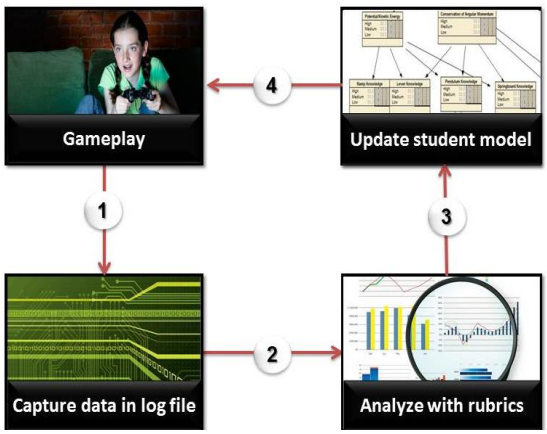


Figure 1. Stealth assessment cycle.

The ECD approach, combined with stealth assessment, provides a framework for developing assessment tasks that are explicitly

linked to claims about personal competencies via an evidentiary chain (i.e., valid arguments that connect task performance to competency estimates), and are thus valid for their intended purposes. The estimates of competency levels can also be used diagnostically and formatively to provide adaptively selected levels, feedback, and other forms of learning support to students as they continue to engage in gameplay (arrow 4). Given the dynamic nature of stealth assessment, it is not surprising that it promises advantages, such as measuring learner competencies continually, adjusting task difficulty or challenge in light of learner performance, and providing ongoing feedback.

Examples of stealth assessment prototypes, designed to measure a range of knowledge and skills—from systems thinking to creative problem solving to causal reasoning—can be found in relation to the following games: *Taiga Park* [18], *Oblivion* [20], and *World of Goo* [17], respectively. For the game *Physics Playground* (formerly Newton's Playground, see [19]), three stealth assessments were created and evaluated in relation to the validity and reliability of the assessments, student learning, and student enjoyment (see [21]). The stealth assessments correlated with associated external validated measures for construct validity and demonstrated reliabilities around .85 (i.e., using intraclass correlations among the in-game measures such as number of gold trophies received for various objects created). Furthermore, students (167 middle school students) significantly improved on an external physics test (administered before and after gameplay) despite no instruction in the game. Students also enjoyed playing the game (reporting a mean of 4 on a 5-point scale in where 1 = strongly dislike and 5 = strongly like).

Next, we briefly review our focal competency for this project—problem solving skills—and discuss the natural fit between this construct and particular video games (i.e., action, puzzle solving, simulation, and strategy games).

2.2 Problem Solving Skills

Problem solving has been studied by researchers for many decades (e.g., [3, 7, 11]). It is generally defined as any goal-directed sequence of cognitive operations [1] and is seen as one of the most important cognitive skills in any profession, as well as in everyday life [7]. Mayer and Wittrock [9] identified several characteristics of problem solving: (a) it is a cognitive process; (b) it is goal directed; and (c) the complexity (and hence difficulty) of the problem depends on one's current knowledge and skills.

In 1984, Bransford and Stein [2] integrated the collection of research at that time and came up with the IDEAL problem solving model. Each letter of IDEAL stands for an important part of the problem solving process: *Identify* problems and opportunities; *define* alternative goals; *explore* possible strategies; *anticipate* outcomes and act on the strategies; and *look* back and *learn*. Gick [4] presented a simplified model of the problem-solving process, which included constructing a representation, searching for a solution, implementing the solution, and monitoring the solution. Recent research suggests that there are two main facets of problem-solving skills: rule identification and rule application [14, 23]. "Rules" are the principles that govern the procedures, conduct, or actions in a problem-solving context. Rule identification involves acquiring knowledge of the problem-solving environment, while rule application involves controlling the environment by applying that knowledge.

Can problem solving skills be improved with practice? Polya [12] argued that people are not born with problem-solving skills. Rather, people cultivate these skills when they have opportunities to solve problems. Researchers have long argued that a central point of education should be to teach people to become better problem solvers [1, 13]. However, there is a gap between problems in formal education and those that exist in real life. Jonassen [6] noted that the problems students encounter in school are mostly well-defined, which contrasts with real-world problems that tend to be messy, with multiple possible solutions. Moreover, many problem-solving strategies that are taught in school entail a "cookbook" type of memorization and result in functional fixedness, which can obstruct students' ability to solve problems for which they have not been specifically trained. Additionally, this pedagogy can stunt students' epistemological development, preventing them from developing their own knowledge-seeking skills [8]. This is where good digital games—which have a set of goals and complicated scenarios that require the player to generate new knowledge—come in. Researchers (e.g., [22]) have argued that playing well-designed video games can promote problem-solving skills because games require constant interaction between the player and the game, usually in the context of solving many interesting and progressively more difficult problems. However, empirical research examining the effects of video games on problem-solving skills is still sparse. Our research begins to fill this gap.

3. PRESENT WORK

3.1 The Game

We are using a slightly modified version of the game *Plants vs. Zombies 2* (Popcap Games and Electronic Arts) as the vehicle for our problem solving assessment. In *Plants vs. Zombies 2* (*PvZ2*), players must plant a variety of special plants on their lawn to prevent zombies from reaching their house. Each of these plants has different attributes. For example, some plants (offensive ones) attack zombies directly, while other plants (defensive ones) slow down zombies to give the player more time to attack the zombies. A few plants generate "sun," an in-game resource needed to purchase more plants. The challenge of the game comes from determining which plants to use and where to place them in order to defeat all zombies in each level of the game.

We chose *PvZ2* as our assessment environment for two main reasons. First, we are able to alter the game because of our association with the Glasslab. Glasslab has access to the source code for *PvZ2*, so we can make direct changes to the game as needed (e.g., the particular information to be collected in the log files). This is important because it allows us to build stealth assessments directly into the game itself and to make alterations to the design of the game if needed. Second, *PvZ2* requires players to apply problem solving skills. Thus, our stealth assessment will be able to collect data relevant to problem solving and estimate learners' levels (e.g., low, medium, high) on the facets and problem solving as a whole. However, because problem solving is not easily measured, we cannot assess it directly. We instead need to define directly observable, in-game indicators of problem solving and its associated facets.

3.2 Problem Solving Model

Based on a review of the literature, we built a problem solving competency model. We divided problem solving into four facets: (a) analyzing givens and constraints, (b) planning a solution

pathway, (c) using tools and resources effectively, and (d) monitoring and evaluating progress. We then identified relevant in-game indicators of the four facets (see Section 3.3 for details). The rubrics for scoring each indicator and the statistical links between the indicators and the competency model variables comprise the evidence model. The competency and evidence models are implemented together in Bayesian networks. We created a unique Bayes net for each game level (42 total) because many indicators do not apply in every level and simple networks make computations more efficient. In the Bayes nets, the overall problem solving variable, each facet, and the associated indicators are nodes that influence each other. Each of the nodes has multiple potential states and a probability distribution that defines the likely true state of the variable. The Bayes nets accumulate data from the indicators and propagate this data throughout the network by updating the probability distributions. In this way, the indicators influence our estimates of the student's problem solving competency and its associated facets dynamically.

3.3 Indicators of Problem Solving

In line with the stealth assessment process, we defined indicators for each of the four facets of problem solving by identifying observable actions that would provide evidence per facet. This was an iterative process which began by brainstorming a large list of potential indicators. After listing all potential indicators, we evaluated each one for (a) *relevance* to their associated facets and (b) the *feasibility* of being implemented in the game. We then removed indicators that were not closely related to the facets or were too difficult or vague to implement. We repeated this process of adding, evaluating, and deleting indicators until we were satisfied with the list of indicators.

In total, there are 32 indicators for our game-based assessment: 7 for analyzing givens and constraints, 7 for planning a solution pathway, 14 for using tools and resources effectively, and 4 for monitoring and evaluating progress. Examples of indicators for each facet are shown in Table 1.

Table 1. Examples of indicators for each problem solving facet

Facet	Examples of Indicators
Analyzing Givens & Constraints	• Plants > 3 Sunflowers before the second wave of zombies arrives • Selects plants off the conveyor belt before it becomes full
Planning a Solution Pathway	• Places sun producers in the back, offensive plants in the middle, and defensive plants up front • Plants Twin Sunflowers or uses plant food on (Twin) Sunflowers in levels that require the production of X sun
Using Tools and Resources Effectively	• Uses plant food when there are > 5 zombies in the yard *or* zombies are getting close to the house (within 2 squares) • Damages > 3 zombies when firing a Coconut Cannon
Monitoring and Evaluating Progress	• Shovels Sunflowers in the back and replaces them with offensive plants when the ratio of zombies to plants exceeds 2:1

3.4 Preliminary Findings

To test the validity of the stealth assessment of problem solving skills, we recruited ten undergraduate students to play *PvZ2* for 90 minutes, as well as complete an external measure of problem solving — MicroDYN [5], a computer-based test in which participants analyzed the relationships between variables in a system and manipulated those variables to achieve a desired state. This comprised our pilot validation study. We correlated the MicroDYN scores with our stealth assessment estimates of problem solving skill to test for construct validity. The results suggest that our game-based assessment is significantly correlated with MicroDYN (r = .74, p = .03). These preliminary findings suggest that our problem solving stealth assessment is valid, but needs to be further tested with a larger sample size. We are currently running a larger validation study with 200 middle-school students and will have the results from that study in time for the EDM conference.

3.5 Limitations

There are several methodological issues with this pilot validation study. First, the sample of students was very small. Second, the participants were not from the target population of our assessment. This pilot was done with undergraduate students, but our target audience is middle school students. It is unclear if similar results will be seen with our target audience. However, middle school students do enjoy playing PvZ2 and our external measure (MicroDYN) has been successfully tested with that age group. Finally, the participants had a very limited amount of time to play the game in the small pilot study. Ninety minutes is only enough time to play about 15-20 of the game's levels. To improve the validity and reliability of the stealth assessment, players need to engage in gameplay for a longer period of time and over multiple sessions.

4. NEXT STEPS

This work is still in its early stages and we have a lot to do before it can have a meaningful impact on education. We are currently running a validation study with 200 middle school students. These students are playing *PvZ2* over three days, one hour per day. On the fourth day, the students complete MicroDYN [5] and a demographic questionnaire. For every 30 students who complete the study, we are examining the results to see if adjustments need to be made to our Bayes nets. This provides us with multiple opportunities to adjust our Bayes nets throughout the course of the validation study. Thus, this larger, ongoing study will help us to create a more valid and reliable assessment.

Our long term goal is to implement the *PvZ2* game-based assessment in middle school classrooms to help educators improve students' problem solving abilities. As part of this effort, we are teaming with Glasslab to create a dashboard that allows educators to easily interpret the results of the assessment — overall and at the individual facet level. The development of this dashboard and other tools to aid the game's implementation will occur alongside our ongoing validation study.

This focus on the validity and practicality of our game-based problem solving assessment makes it much more likely that the assessment will be both accurate and useful in classroom settings. Students can be assessed on problem solving, a key cognitive skill, in an engaging environment that presents rich problem solving situations and can parse complex patterns of students'

actions. Teachers get a valuable tool that will allow them to pinpoint students' abilities in various aspects of problem solving and, in turn, help each student improve their problem solving skills. These benefits stem from our use of evidence-centered design, which gives a framework for creating valid assessments, and stealth assessment, which gives us the ability to invisibly embed such assessments into complex learning environments such as games. By embracing evidence-centered design and stealth assessment, other researchers can also create complex and engaging assessments that meet their specific needs.

5. ACKNOWLEDGMENTS

We would like to thank our colleagues at GlassLab for supporting our work assessing problem solving in Plants vs. Zombies 2—specifically Jessica Lindl, Liz Kline, Michelle Riconscente, Ben Dapkiewicz, and Michael John. We also thank Weinan Zhao for his great programming assistance, as well as Sam Greiff and Katarina Krkovic for letting us use *MicroDYN*.

6. REFERENCES

[1] Anderson, J. R. 1980. *Cognitive psychology and its implications.* Freeman, New York, NY.

[2] Bransford, J. and Stein, B.S. 1984. *The IDEAL problem solver: A guide for improving thinking, learning, and creativity.* W. H. Freeman, New York, NY.

[3] Gagné, R. M. 1959. Problem solving and thinking. *Annual Review of Psychology*. 10, 147-172.

[4] Gick, M. L. 1986. Problem-solving strategies. *Educational Psychologist*, 21, 99-120.

[5] Greiff, S. and Funke, J. 2009. Measuring complex problem solving: The MicroDYN approach. In *The transition to computer-based assessment: New approaches to skills assessment and implications for large-scale testing,* F. Scheuermann and J. Björnsson, Eds. Office for Official Publications of the European Communities, Luxembourg, Luxembourg, 157-163.

[6] Jonassen, D. H. 2000. Toward a design theory of problem solving. *Educational Technology Research and Development*. 48, 4, 63-85.

[7] Jonassen, D. 2003. Using cognitive tools to represent problems. *Journal of Research on Technology in Education.* 35, 3, 362-381.

[8] Jonassen, D. H., Marra, R., and Palmer, B. 2004. Epistemological development: An implicit entailment of constructivist learning environments. In *Curriculum, plans, and processes of instructional design: International perspectives,* N. M. Seel and S. Dikjstra, Eds. Lawrence Erlbaum Associates, Mahwah, NJ, 75-88.

[9] Mayer, R. E. and Wittrock, M. C. 1996. Problem-solving transfer. *Handbook of educational psychology,* D. C. Berliner and R. C. Calfee, Eds. Macmillan Library Reference, New York, NY, 47-62.

[10] Mislevy, R. J., Steinberg, L. S., and Almond, R. G. 2003. On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives.* 1, 1, 3-62.

[11] Newell, A. and Shaw, J. C. 1958. Elements of a theory of human problem solving. *Psychological Review.* 65, 3, 151-166.

[12] Polya, G. 1945. *How to solve it: A new aspect of mathematical method.* Princeton University Press, Princeton, NJ.

[13] Ruscio, A. M. and Amabile, T. M. 1999. Effects of instructional style on problem-solving creativity. *Creativity Research Journal.* 12, 251-266.

[14] Schweizer, F., Wüstenberg, S., and Greiff, S. 2013. Validity of the MicroDYN approach: Complex problem solving predicts school grades beyond working memory capacity. *Learning and Individual Differences.* 24, 42-52.

[15] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. In *Computer games and instruction,* S. Tobias and J. D. Fletcher, Eds. Information Age Publishers, Charlotte, NC, 503-524.

[16] Shute, V. J. and Ke, F. 2012. Games, learning, and assessment. In *Assessment in game-based learning: Foundations, innovations, and perspectives*, D. Ifenthaler, D. Eseryel, and X. Ge, Eds. Springer, New York, NY, 43-58.

[17] Shute, V. J. and Kim, Y. J. 2011. Does playing the World of Goo facilitate learning? In *Design research on learning and thinking in educational settings: Enhancing intellectual growth and functioning,* D. Y. Dai, Ed. Routledge Books, New York, NY, 359-387.

[18] Shute, V. J., Masduki, I., and Donmez, O. 2010. Conceptual framework for modeling, assessing, and supporting competencies within game environments. *Technology, Instruction, Cognition, and Learning.* 8, 2, 137-161.

[19] Shute, V. J. and Ventura, M. 2013. *Measuring and supporting learning in games: Stealth assessment.* The MIT Press, Cambridge, MA.

[20] Shute, V. J., Ventura, M., Bauer, M. I., and Zapata-Rivera, D. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In *Serious games: Mechanisms and effects,* U. Ritterfeld, M. Cody, and P. Vorderer, Eds. Routledge, Taylor and Francis, Mahwah, NJ, 295-321.

[21] Shute, V. J., Ventura, M., and Kim, Y. J. 2013. Assessment and learning of qualitative physics in Newton's Playground. *The Journal of Educational Research*. 106, 423-430.

[22] Van Eck, R. 2006. Building intelligent learning games. In *Games and simulations in online learning: Research & development frameworks,* D. Gibson, C. Aldrich, and M. Prensky, Eds. Idea Group, Hershey, PA.

[23] Wüstenberg, S., Greiff, S., and Funke, J. 2012. Complex problem solving — more than reasoning? *Intelligence.* 40, 1-14.

Strategic game moves mediate implicit science learning

Elizabeth Rowe
EdGE at TERC
Cambridge, MA
elizabeth_rowe@terc.edu

Ryan S. Baker
Teachers College
Columbia University
New York, NY
baker2@tc.columbia.edu

Jodi Asbell-Clarke
EdGE at TERC
Cambridge, MA
jodi_asbell-clarke@terc.edu

ABSTRACT

Educational games have the potential to be innovative forms of learning assessment, by allowing us to not just study their knowledge but the process that takes students to that knowledge. This paper examines the mediating role of players' moves in digital games on changes in their pre-post classroom measures of implicit science learning. We applied automated detectors of strategic moves, built and validated from game log data combined with coded videos of gameplay of 69 students, to a new and larger sample of gameplay data. These data were collected as part of national implementation study of the physical science game, *Impulse.* This study compared 213 students in 21 classrooms that only played the game and 180 students in 18 classrooms in where the players' teacher used game examples to bridge the implicit science learning in the game with explicit science content covered in class. We analyzed how learning outcomes between conditions were associated with six strategic moves students made during gameplay. Three of the strategic moves observed are consistent with an implicit understanding of Newton's First Law, the other three strategic moves were not. Path analyses suggest the mediating role of strategic moves on students' implicit science learning is different between the two conditions.

Keywords

Game-based science learning; Discovery with models; Automated detectors; Predictive modeling;

1. INTRODUCTION

Digital games are garnering increasing attention as potential learning environments as the volume of research increases indicating games may foster scientific inquiry, problem-solving, and public participation in breakthrough scientific discoveries [1]. Because nearly all youth and many adults participate in Internet-based games [2], educators and researchers are trying to tap this pervasive vehicle for learning and assessment environments for the 21st century [3].

Our research group studies how games can be used to improve learning of fundamental high-school science concepts (e.g. Newton's laws of motion). Our games use popular game mechanics embedded in accurate scientific simulations so that through engaging gameplay, players are interacting with digitized versions of the laws of nature and the principles of science. We hypothesize that as players dwell in scientific phenomena, repeatedly grappling with increasingly complex instantiations of the physical laws, they build and solidify their implicit knowledge over time.

It is not our intent that these games *teach* science content explicitly, but rather that they engage the learner with scientific phenomena allow them to build their implicit understandings about these phenomena through gameplay. To measure implicit learning in games, we built automated detectors of strategies we saw players using in the games [4, 5]. Thus, we address the question: *Do learners' strategic moves in the game correspond to increased implicit understanding of the science content outside the game?*

We also examine the role of the teacher in game-based learning. As Jim Gee points out, games rely on what he refers to as the Big "G" Game – the surrounding interactions that arise because of and support the game [6]. Post-game debriefing and discussions connecting gameplay with classroom learning are critical in helping students apply and transfer learning that takes place in games [7]. Our research attempts to capture the strategies players develop during gameplay that may reveal implicit knowledge, so that we can help educators seize and leverage that implicit learning to support explicit classroom learning.

Success in this approach will result in a new way to think about game-based assessments, starting not from prescribed learning outcomes, but from watching what types of strategy development actually take place. The final step of this research, reported in this paper, is to examine the extent to which strategic moves used while playing *Impulse* mediate changes in classroom measures of students' understanding of the same science content.

2. THE GAME: *IMPULSE*

The game *Impulse* is built for the web and wireless devices. Impulse challenges players use an impulse (a click or touch on the screen) to move their ball to a goal without crashing into any other (ambient) balls on the screen. All the balls have mass and obey Newton's laws of motion. As the levels of the game increase, more ambient balls are introduced, with varying mass.

Impulse is an attempt by designers to immerse a player in what is known to physicists as a n-body simulator. We hypothesize that by having to predict the motions of the particles, and their reactions to the force imparted by the impulse, the player will build implicit knowledge of forces and motions (Figure 1) that we could measure through data mining.

The first 20 levels of the game introduce players to 4 particles of different mass, providing 5 levels of experience with each of the 4

Figure 1: Impulse game

particles; across these 5 levels, the number of particles in the game space increases from 1 to eventually 10. Beginning in Level 21, players encounter particles with different masses simultaneously. As players reach higher levels with greater numbers and variety of masses of particles, they need to "study" the particles' behavior to predict the motion of particles so that they can guide their particle to the goal, not run out of energy, and avoid collision with other particles.

3. STRATEGIC MOVES

Our research attempts to capture and automatically assess the range of strategies players develop during gameplay. We identified a set of 6 strategic moves that we observe players making in the game *Impulse* (Table 1). Three of these strategic moves are theorized to constitute evidence of implicit understandings of Newton's First Law: each particle will keep moving on its path without an impulse or force from another particle. The remaining three strategic moves reflect an understanding of the game mechanic, but are not considered strong evidence of implicit understanding of Newton's First Law.

Table 1. Strategic moves and coding definitions

Strategic Move	Coding Definition
*Float	The player particle was not acted upon for more than 1 second
Toward goal	The learner intended to move the player particle toward the goal
*Stop/slow down	The learner intended to stop or slow the motion of the player particle
*Player path clear	The learner intended to move non-player particles to keep the path of the player particle clear
Goal clear	The learner intended to move non-player particles to keep the goal clear
Buffer	The learner intended to create a buffer between the player and other particles to avoid collision

*Evidence of implicit understanding of Newton's First Law

Video data was collected from 69 high school students, to develop automated detectors of these strategies. Every click in randomly selected, three-minute video segments, one per student, was coded for these strategic moves, with every player action in these video segments coded as to which strategy it represented. Two coders coded ten videos with Kappa values exceeding 0.70 for all of these strategic moves [4, 5].

We built classifiers to infer the ground truth labels created by the video coders. For each player action a set of 66 features of that action were automatically distilled, including the time since the last player action and the distance between the player particle and goal. These features were then aggregated at the click level to map to the labels provided by the video coders [6]. Classifiers were created using J48 decision trees within RapidMiner 5.3 that mapped the student behaviors in the features distilled from the clickstream data to the training labels, cross-validating at the student level. All detectors discussed here had cross-validated Kappas between 0.51 and 0.86 and A' between 0.78 and 0.97 [6].

4. IMPLEMENTATION STUDY

Having developed these detectors of student strategic moves, we then collected a much larger data set to be able to study the relationship between in-game strategic moves, pedagogical practices, and learning outcomes. To this end, we conducted an implementation study [8] to examine the conjecture that implicit learning in game play can help prepare students for classroom learning.

Forty-two teachers were assigned to one of three groups (14 per group). Teachers could include a maximum of three sections of an individual class. Of the 42 teachers who initially agreed to participate, 23 teachers completed the study (55 percent), resulting in this final sample with complete data:

Bridge: 180 students in 18 classes in which 8 teachers incorporated game examples to bridge game play and science content

Game Only: 213 students in 21 classes in which 10 teachers encouraged students to play the game, but provided no in-class interaction around the game

Control: 108 students in 11 classes in which 5 teachers taught the science content as they normally do, without games.

Students took pre-post online assessments with six items, three dealing with Newton's First Law and three dealing with Newton's Second Law. All items were written to be answerable with an intuitive understanding of the physics concepts and were piloted with think-aloud interviews. Both assessments had a maximum of 10 points possible. Assessment scores were standardized as Z-scores and all coefficients are reported in effect sizes.

Hierarchical linear modeling of data from the 23 teachers (50 classes) shows a significant positive effect of the Bridge and Game Only groups compared to the Control group on student's post-assessment scores after accounting for pre-assessment scores [8]. This group effect, however, was significantly moderated by whether or not the class was a Honors/AP class (Figure 2). There was also a significant main effect for gender, with female students receiving lower post-scores than male students.

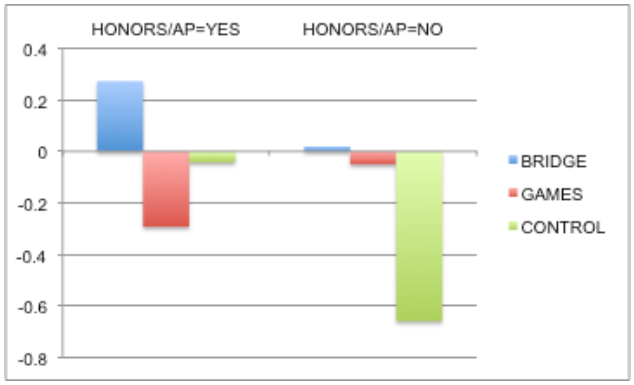


Figure 2: Predicted post-assessment scores across study conditions in Honors/AP classes versus non-Honors/AP classes (y-axis=standard deviations from the mean post-score, accounting for all components of the HLM model) [8].

The group effect was significant among students in non-Honors/AP classes. Among students in Honors/AP classes, Bridge students performed better than Game only students but not Control students. These results, while intriguing, tell us that the Bridge condition was generally best, but do not explain why Bridge was better. Did the teachers in the Bridge condition promote learning separate from the game? Or did it actually drive different behavior within *Impulse*, making the game a more valuable learning experience?

5. THE ROLE PLAYED BY IN-GAME STRATEGIC BEHAVIOR

The final step in this research, and the specific contribution novel to this paper, is to connect in-game measures of implicit science learning with external measures of those concepts. Specifically, we hypothesize that strategic moves consistent with an implicit understanding of Newton's First Law will mediate changes in these external assessments, whereas the other strategic moves will not be associated with changes in the pre-post assessments.

5.1 Apply Automated Detectors

We applied the automated detectors built with the sample of 69 students to this larger sample of gameplay data from 393 students to detect when learners used each type of strategic move. The detectors were applied to every student action during the entire duration of gameplay, 1.01 million actions in total. The same log data features were automatically distilled for this entire data set as for the initial creation of the models. Then this data was inputted into RapidMiner 5.3, along with the previously generated W-J48 decision trees model files, in order to apply the trees to the data. The result was a prediction for every click, for each of the relevant strategic moves in Table 1, of the detector's confidence that strategy was being used. Every learner action in this game was thereby annotated with an estimated probability that the learner was using each of the strategic moves.

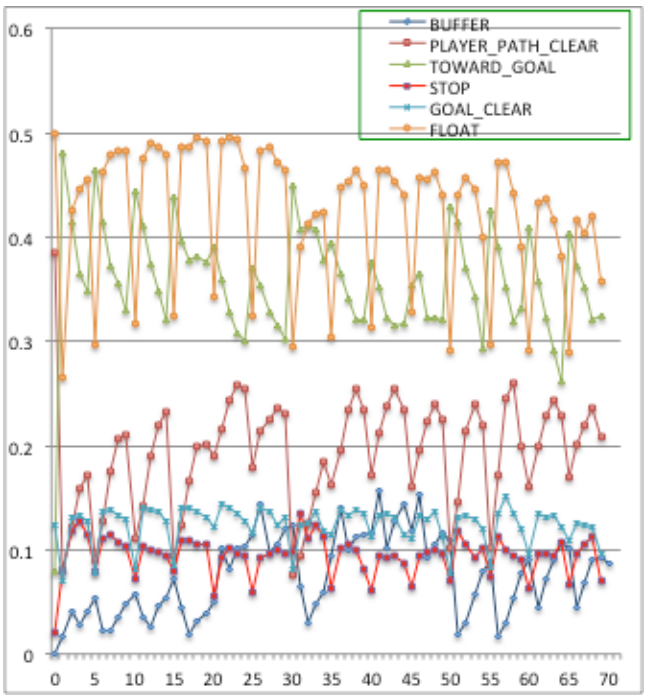


Figure 3: Average probability for each strategic move (y-axis) by game level (x-axis)

Figure 3 shows the average probability for each strategic move at each game level. The most prevalent strategic moves were Toward Goal and Float, with Float being evidence for implicit understanding of Newton's First Law. The least common strategic moves were Stop/Slow Down (evidence for implicit understanding) and Buffer. Float reflects the absence of activity (on the player particle in the time prior to the click and can co-occur with any other strategic move. Stop/Slow Down, in contrast, reflects a deliberate attempt by the player to stop or slow down the motion of the player particle. Float and Stop/Slow Down both reflect understandings of Newton's First Law e.g., a mass will keep moving until acted upon by a force, but the float strategy is a passive move and the stop strategy is an active move.

Figure 3 also shows evidence of shifts in behavior every 5 levels. The cyclical patterns in this data correspond with the planned transitions in the game. Every 5 levels, the game reduces the difficulty level of the game when a new challenge (e.g., particle with a different mass, two particles with different masses) is introduced, by decreasing the number of particles in the space (a decrease in gameplay challenge which balances for the increase in conceptual challenge). However, the reduction in the number of particles makes it more likely a player will simply push the particle toward the goal, leading to corresponding declines in all of the other strategies. Overall, as the number of particles in the game space increases, the average probability of using the simple Toward Goal strategy declines while the probabilities of using the other strategies increase.

5.2 Path Models

Path models were built to estimate the mediating role of each strategic move between prior achievement and post assessment scores using SmartPLS [9]. As pre-assessment scores and Honors/AP enrollment were significantly correlated, they were combined into a single latent variable labeled 'Prior Achievement'. Separate path models were created for the Bridge and Game Only conditions (Figures 4 and 5). The standardized coefficients appear on the paths and the adjusted R^2 values appear in the circles. T-values were calculated using a bootstrapping process with 1000 samples.

Among students in Bridge classrooms, the use of the Buffer strategy significantly mediates the impact of prior achievement and gender on post-scores (adjusted R^2 = 0.151, p=0.005). This suggests using the Buffer strategy enhanced Bridge student's understanding of the concepts, beyond what is accounted for their prior levels of achievement. In Game Only classrooms, student use of the Buffer (adjusted R2 = 0.095, p=0.018), Stop (adjusted R^2 = 0..149, p<0.001), and Float (adjusted R^2 = 0.109, p=0.031), strategic moves significantly mediate the relationship between prior achievement & gender on post-scores. In these classes with no teacher scaffolding of the gameplay, use of the Buffer and Float strategies enhanced student's understanding, but use of the Stop strategy diminished their understanding.

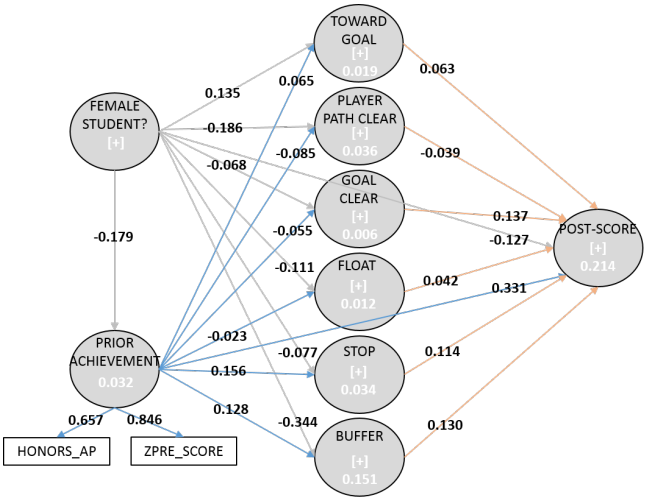


Figure 4: Full path model—Bridge Classrooms

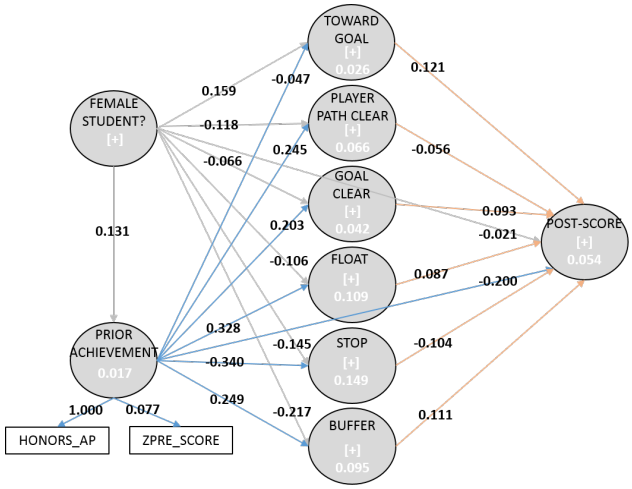


Figure 5: Full path model—Game Only Classrooms

In Bridge and Game Only classrooms, once gender differences in strategic moves are taken into account, the impact of being a Female student on post-scores is no longer significant (coeff=-0.127 in Bridge classrooms, p=0.108; coeff=-0.021 in Game Only classrooms, p=0.759). This suggests the gender main effect found in the HLM analyses may be entirely attributable to gender differences in gameplay. One potential explanation is that while females play games at equal rates as males [2], the types of games they play and the amount of time they spend doing so may vary. Success in a rapid-fire, reaction time educational game like *Impulse* may require gameplay skills more congruent with games more popular among males (e.g, first person shooters) than the social, puzzle, and role-playing games females tend to prefer [2].

6. DISCUSSION

It is noteworthy that two of the three strategies we anticipated reflecting an implicit understanding of Newton's First Law were significant mediators in Game Only classrooms. Player Path Clear, a strategic move applied to non-player particles, may not have been a significant mediator because it is likely to co-occur with Float, a strategic move applied to the player particle. By contrast, the other strategies were not significant mediators with one exception: Buffer, the simultaneous use of force on more than one particle when the particles were in close proximity to each other. Sometimes those forces were in direct opposition to the other particles (i.e., simultaneous use of the Stop strategy), while other times they were not. While Buffer was not a strategic move we a priori identified as consistent with an understanding of Newton's First Law, these results suggest it plays a mediating role similar to Stop and Float. Use of the Buffer strategy was associated with higher post-scores in Bridge and Game Only classrooms.

The negative mediating relationship of the Stop strategy in Game Only classrooms is consistent with the HLM findings shown in Figure 2, where students in Honors-AP classes did not perform on the post-assessment as well as students in non-Honors/AP classes [8]. This lack of use of the Stop strategy is consistent with the lack of understanding of Newton's Laws exhibited on the pre-post assessments. This suggests that learners who already have a basic understanding of the scientific concepts may not be aided by the game as a sole intervention. Their improvement in science understanding is enhanced when the game and the teacher bridge materials are used together. These results reinforce the importance of teachers providing bridges between gameplay and science content.

This paper also makes an important contribution to the space of problems that can be addressed by EDM. Many projects have attempted to detect strategic behavior in online learning. This project, by detecting strategic behavior explicitly connected to core concepts, and modeling how different classroom activities influence in-game behavior, shows how EDM methods can bridge understanding of the relationship between what students learn in class, and how they behave online. As such, we are able to see the concrete impact of classroom activity on gameplay behavior, and to measure its scope and manifestations.

In the long term, then, this combination of methods – automated detectors, path analysis, and classroom studies – creates the potential to make EDM useful for investigating interventions not just online, but in classroom settings as well.

7. ACKNOWLEDGMENTS

We are grateful for NSF/EHR/DRK12 grant #1119144 and our research group, EdGE at TERC, which includes Erin Bardar, Teon Edwards, Jamie Larsen, Barbara MacEachern, Katie McGrath, and Emily Kasman. Our evaluators, the New Knowledge Organization, helped establish the reliability of video coding.

8. REFERENCES

[1] Cooper, S., et al. (2010). Predicting protein structures with a multiplayer online game. *Nature, 466*(7307), 756-760.

[2] Lenhart, A., et al. (2010) *Social Media & Mobile Internet Use Among Teens and Young Adults*. Washington, DC: Pew Internet & American Life Project.

[3] National Research Council. (2011). *Learning Science Through Computer Games and Simulations. Committee on Science Learning: Computer Games, Simulations, and Education*. M. Honey & M. Hilton (Eds.). Washington, DC: National Academies Press.

[4] Rowe, E., Baker, R., & Asbell-Clarke, J., Kasman, E., & Hawkins, W. (2014). Building automated detectors of gameplay strategies to measure implicit science learning. *Proceedings of the International Conference on Educational Data Mining*, 337-338.

[5] Rowe, E., Baker, R. & Asbell-Clarke, J. (in press). Serious games analytics to measure implicit science learning. To appear in C.S. Loh, Y. Sheng, D. Ifenhelter (Eds). *Serious Games Analytics*. New York: Springer.

[6] Gee, J. (2008). Learning and Games. In K. Salen (Ed). *The Ecology of Games: Connecting Youth, Games, and Learning*. Cambridge, MA: The MIT Press. 21–40.

[7] Ke, F. (2009). A qualitative meta-analysis of computer games as learning tools. In R. E. Furdig (Ed.), *Handbook of Research on Effective Electronic Gaming in Education* (pp. 1–32). New York: IGI Global.

[8] Rowe, E., Asbell-Clarke, J., Bardar, E., Kasman, E., & MacEachern, B. (2014, June). *Crossing the Bridge: Connecting Game-Based Implicit Science Learning to the Classroom.* Paper presented at the 10th annual meeting of Games+Learning+Society in Madison, WI.

[9] Ringle, C. M., Wende, S., & Becker, J.M. (2015). SmartPLS 3. Boenningstedt: SmartPLS GmbH, http://www.smartpls.com.

Predicting learning-related emotions from students' textual classroom feedback via Twitter

Nabeela Altrabsheh
School of Computing
Lion Terrace
University of Portsmouth
nabeela.altrabsheh@port.ac.uk

Mihaela Cocea
School of Computing
Lion Terrace
University of Portsmouth
mihaela.cocea@port.ac.uk

Sanaz Fallahkhair
School of Computing
Lion Terrace
University of Portsmouth
sanaz.fallahkhair@port.ac.uk

ABSTRACT
Teachers/lecturers typically adapt their teaching to respond to students' emotions, e.g. provide more examples when they think the students are confused. While getting a feel of the students' emotions is easier in small settings, it is much more difficult in larger groups. In these larger settings textual feedback from students could provide information about learning-related emotions that students experience. Prediction of emotions from text, however, is known to be a difficult problem due to language ambiguity. While prediction of general emotions from text has been reported in the literature, very little attention has been given to prediction of learning-related emotions. In this paper we report several experiments for predicting emotions related to learning using machine learning techniques and n-grams as features, and discuss their performance. The results indicate that some emotions can be distinguished more easily then others.

Keywords
Emotion prediction from text, Machine learning, Learning-related emotions

1. INTRODUCTION
Detecting emotions is important in the learning process [4]. Positive emotions may increase students' interest in learning, increase engagement in the classroom and motivate students [4]. Additionaly, students who are happy generally are more motivated to accomplish their learning goals.

Sentiment analysis research has grown considerably in the last decade, mainly due to the availability of rich text resources such as social networking sites, blogs and microblogs, and product reviews. Despite the name of this area, sentiment analysis is mostly focused on detection of polarity (negative or positive sentiment) rather than specific emotions. Thus, there is relatively little research on the prediction of specific emotions from text [2, 3], with even fewer reports of such research in education [9]. Moreover, from these studies (both within the educational field and outside of it), an even smaller number use machine learning to predict emotion from text, e.g. [2, 3, 9].

In this paper we focus on the prediction of emotions relevant for learning from students' textual feedback via Twitter in a classroom context using machine learning techniques. To investigate the prediction of the identified emotions from text, we experiment with several preprocessing methods, n-gram features, and machine learning techniques.

2. RELATED RESEARCH
There are four main steps to create predictive models from text with machine learning: preprocessing the data, selecting the features, applying the machine learning techniques and evaluating the results.

Preprocessing the data involves preparing the data and cleaning it from unwanted elements which may negatively affect the performance of the machine learning techniques. Some of the general preprocessing techniques used with basic text are: tokenization, convert text to lower or upper case, remove punctuation, remove numbers and, remove stop words [8].

Preprocessing Twitter data requires additional techniques due to the presence of emoticons, hashtags and chat language. Some of the Twitter-specific data preprocessing techniques from previous research [8, 11] are: removing hashtags, removing URLs, removing retweets, identifying emoticons, removing user mentions in tweets, removing Twitter special characters, and slang/chat language handling.

In relation to specific emotions detection, both general preprocessing techniques and Twitter-related preprocessing techniques have been used, e.g. removal of stop words and stemming [3], removing URLs [5], and tokenization [5].

Feature selection refers to the process of selecting relevant features for the particular prediction problem, while eliminating the features that are redundant or irrelevant. In prediction problems where the data is in the form of text, the most common features are n-grams [7]. The most commonly used n-gram for emotion detection is unigrams (one word) [7]. In contrast, there are very few studies investi-

gating the use of bigrams (two words) and trigrams (three words) in emotion prediction. However bigrams and trigrams has been used in sentiment analysis of tweets [7]. In this paper, we investigate the influence of these different n-grams and their combination on emotion detection.

Various *machine learning techniques* have been used for polarity and emotions prediction from text. In our experiments we used classifiers previously shown to work well [9]: Naive Bayes (NB), Multinomial Naive Bayes (MNB), Complement Naive Bayes (CNB), Support Vector Machines (SVM), Maximum Entropy (ME), Sequential Minimal Optimization (SMO), and Random Forest (RF).

Previous research on emotions related to learning indicates a variety of emotions experienced by learners [6]. In previous research [1], we identified from the literature a number of common emotions that are associated with learning: amused, anxiety, appreciation, awkward, bored, confusion, disappointed, embarrassed, engagement, enthusiasm, excitement, frustration, happy, motivated, proud, relief, satisfaction, shame and uninterested.

3. DATA CORPUS

The data was collected from lectures taught in English in Jordanian universities on different topics: calculus, English communication skills, database, engineering, molecular biology, chemistry, physics, science, contemporary history of the world and architecture.

Twitter was used to collect students feedback, opinions, and feelings about the lecture. For each tweet, they were asked to choose one emotion from a set of emotions provided, i.e. the 19 emotions listed in the previous section. Although tweets were used the language was formal and did not include chat language or slang, however, they did include emoticons and hashtags.

A total number of 1522 tweets were collected with their corresponding emotion label. There was one label per feedback. Some of the emotions appeared more frequently than others. The most frequent emotions that were used in our research were: Bored (336), Amused (216), Frustration (213), Excitement (178), Enthusiasm (176), Anxiety (130), Confusion (73), and Engagement (67). The least frequent ones were discarded due to insufficient data for training and testing machine learning algorithms: Happy (32), Satisfaction (31), Appreciation (26), Embarrased (18), Dissapointed (12), Uninterested (4), Proud (3), Relief (3), Shame (2), Awkward (1), and Motivated (1).

4. PREDICTION OF EMOTIONS FROM STUDENTS' FEEDBACK

Two different preprocessing levels were experimented with: (a) high preprocessing, which includes: tokenization, convert text to lower case, remove punctuation, remove numbers, remove stop words, remove hashtags, remove URLs, remove retweets, remove user mentions in tweets, and remove Twitter special characters; (b) low processing, which includes: tokenization, convert text to lower case, and remove stop words.

The high preprocessing was only used for one of the models which contained all the emotions combined, due to the low results that it led to in comparison with the low level of preprocessing for this model. Consequently, for the other models only the low preprocessing was experimented with.

The negative influence of preprocessing on the performance of the models indicates that information that is typically discarded for polarity prediction has value for the identification of specific emotions, as for example in the case of punctuation [11].

We experimented with different n-grams, i.e. unigrams, bigrams, and trigrams, and all combinations between them to find which n-gram or combination of n-grams leads to the best performance for the different models. The features that were experimented with are: Unigrams (UNI); Bigrams (BI); Trigrams (TRI); Unigrams and Bigrams combined; Unigrams and Trigrams combined; Bigrams and Trigrams combined; and Unigrams, Bigrams, and Trigrams combined.

We used the classifiers mentioned previously in section 2 due to their common use in previous research. Additionally, we used two common kernels for SVM: radial basis (RB) and linear (LIN) kernel.

We experimented with all the emotions combined and then subtracted, in turn, the emotion with the lowest number of instances. The total number of models experimented with was 16 models, which are: 7 emotions (All except engagement) + other (8 classes); 6 emotions (7 emotions except confused) + other (7 classes); 5 emotions (6 emotions except anxiety) + other (6 classes); 4 emotions (5 emotions except enthusiasm) + other (5 classes); 3 emotions (4 emotions except excitement) + other (4 classes); 2 Emotions (Amused, Bored) + other (3 classes); and each emotion + other (2 classes).

All the models were tested using 10-fold cross-validation; the accuracy and the error rate were used to assess the overall performance of the classifiers, while the precision, recall, and F-score were used to assess the ability of the classifiers to correctly identify the specific emotion(s).

The results indicate that the models with a single emotion perform better than the multi-emotion models in terms of accuracy, although one has to bare in mind that the baseline for multi-class models is lower than the baseline for 2-class models.

The results show that two classifiers performed best in terms of accuracy: the Support Vector Machine with Radial Basis kernel (RB), mainly for the 2-class models, and Sequential Minimal Optimization (SMO), mainly for the multi-class models. In term of features, unigrams and trigrams were found to lead to the best performance for the 2-class models, while unigrams combined with bigrams and trigrams led to the best performance for the multi-class models.

Despite the fact that accuracy can be useful in predicting the models performance, it does not indicate how well a classifier can predict specific emotions. As the recall indicates the percentage of correctly identified instances for a class of in-

Table 1: Highest recall for each model

Model	Tecbnique	N-gram	Accuracy	Error rate	Precision	Recall	F-score
ALL Preprocessed	ME	UNI+BI+TRI	0.32	0.68	0.34	0.33	0.33
ALL W/O Preprocessing	ME	UNI+BI	0.32	0.68	0.33	0.32	0.32
7 Emotions+ other	NB	BI+TRI	0.26	0.74	0.24	0.25	0.25
6 Emotions+ other	MNB	UNI	0.27	0.73	0.27	0.26	0.27
5 Emotions+ other	MNB	UNI+TRI	0.25	0.75	0.32	0.32	0.32
4 Emotions+ other	MNB	BI	0.26	0.74	0.29	0.38	0.33
3 Emotions + other	ME	UNI+BI+TRI	0.51	0.49	0.43	0.36	0.39
2 Emotions+ other	ME	UNI+BI+TRI	0.57	0.43	0.40	0.51	0.45
Amused	CNB	TRI	0.49	0.51	0.19	0.70	0.30
Anxiety	CNB	TRI	0.45	0.55	0.12	0.77	0.21
Bored	CNB	TRI	0.44	0.56	0.28	0.85	0.42
Confused	CNB	TRI	0.28	0.72	0.06	0.81	0.11
Engagement	CNB	TRI	0.24	0.76	0.04	0.68	0.08
Enthuisiasm	CNB	TRI	0.36	0.64	0.14	0.76	0.24
Excitement	CNB	TRI	0.37	0.63	0.15	0.86	0.26
Frustration	CNB	TRI	0.40	0.60	0.19	0.84	0.31

Table 2: Best overall models for identification of specific emotions

Model	Tecbnique	N-gram	Accuracy	Error rate	Precision	Recall	F-score
Amused	CNB	Bi+Tri	0.64	0.36	0.24	0.62	0.35
Bored	CNB	UNI+BI+TRI	0.71	0.29	0.43	0.63	0.51
Excitement	CNB	UNI+TRI	0.64	0.36	0.21	0.64	0.32

terest, it can be used to assess the ability of the classifiers to predict emotions; in addition, precision can indicate where the identification problems occur.

For most of the models with the highest accuracy, the recall is extremely low or even 0% in some cases. In addition, precision is also low for most of the models (with a few exceptions). For instance in the "engagement + other" model where the accuracy is 95% and the precision, recall, and F-score are (0-0.05)% for the emotion class. This indicates that the high accuracy is due to the correct identification of the "other" class rather than the correct identification of emotion(s).

Table 1 displays the best experimental results when focusing on the recall, i.e. the correct identification of the emotion(s). In terms of machine learning techniques, Complement Naive Bayes (CNB) performs best for half of the models, which could be explain by the ability of this technique to compensate for uneven class sizes. In terms of features, trigrams led to the best performance in the 2-class models, while unigrams combined with bigrams and trigrams led to the best performance in the multi-class models.

The fact that the models with high recall rates have low accuracy and low precision values indicates that many instances of the "other" class are wrongly classified as indicating particular emotions. In other words, although the classifiers have a higher sensitivity for the emotion classes, they are not precise in distinguishing the "other" class from the emotion class(es).

When looking at the overall picture and the balance of the evaluation metrics considered (i.e. accuracy, error rate, precision and recall), some of the models stand out – these are presented in Table 2. We found that the best classifier is Complement Naive Bayes (CNB). When looking at the features, one can notice that different combinations of n-grams led to the best performance for different classifiers. This indicates that a combination of various n-grams instead of a single n-gram is useful for the prediction of specific emotions and should be investigated further.

It is not surprising that the best performing models are for the emotions for which we had larger number of instances (see section 3), i.e. bored, amused and excitement. Interestingly, the models for excitement performed better that the ones for frustration, although there were more instances for frustration than for excitement.

From previous research studies focusing on the prediction of emotions using machine learning techniques, only one study was conducted in an educational context [9]. This research used part-of-speech (POS) tags as features, and more specifically, they experimented with the combination of the following part-of-speech tags: verb, adverb, adjective and noun. They evaluated their models using precision, recall, and F-score and found that Random Forest performed better than the other classifiers with a weighted average F-score at 0.638. Similar to our research they found that the recall score was higher than the precision. From the emotions that we identified as relevant for learning from previous literature, they only looked at anxiety, for which they obtained a precision value of 0.6 using a LogitBoot classifier. However, this re-

search was conducted on Chinese text, which has different characteristics and structures compared with English text. Moreover, the research was based on text from online chats and discussion groups. Furthermore, they used in their approach an affective words base (i.e. lexicon), where each affective word had a number associated with its degree of reflection of a particular emotion.

Outside the educational domain, there are very few studies that looked at the prediction of specific emotions from text only, which are described below.

One study, which used unigrams and a experimented with a multi-class model with 5 emotions [3], found that the Naive Bayes and Support Vector Machine classifiers performed well, leading to an accuracy of 67%. This data, however, is not representative for other types of text expressing emotions, as indicated by the low accuracy, i.e. less than 35%, of these models on test sets with other data. Similarly to the research described above, they also experimented with lexicons for specific emotions.

Another study which used unigrams as a feature and machine learning looked at predicting the presence of emotion versus the lack of emotion [2]; they obtained a maximum accuracy of 74%. However, they did not discuss the performance in terms of identifying the presence of emotion (i.e recall for the emotion). They have also used lexicons with emotion-related words.

However, very few studies investigated the use of other n-grams. Youn and Purver [10] investigated the prediction of emotions from the Chinese microblog service Sina Weibo; in their experiments they found that the models with bigrams and trigrams outperformed the models using unigrams. Similarly, our results showed that using all of the n-grams (i.e. unigrams, bigrams, and trigrams) combined led to the best identification of emotions for the multi-emotion models. Additionally, we found that trigrams led to the best identification of emotions for the 2-class models.

While it is difficult to compare the performance of our models with previous work given the variations in different experimental set-ups (e.g. data origin, language, choice of emotions, choice of features and the use of lexicons), one aspect that seems to be prevalent in previous research is the used of lexicons. Consequently, in out future work, we will investigate the use of such an affective word base for education and its effect on the prediction models.

5. CONCLUSIONS AND FUTURE WORK

In this paper we conducted several experiments with the purpose to investigate the prediction of specific emotions related to learning from students' textual classroom feedback. We focused on several learning emotions which were found to be relevant from previous literature: Amused, Anxiety, Bored, Confusion, Engagement, Enthusiasm, Excitement, and Frustration. We experimented with several preprocessing and machine learning techniques, and also with different combinations of n-gram features.

The models were evaluated using 10-fold cross-validation and using the following evaluation metrics: accuracy, error rate, precision, recall, and F-score. The best performing models were obtained for three particular emotions using 2-class models: amused, bored and excitement. The best classifier was Complement Naive Bayes (CNB). A combination in n-grams led to the best performance in most models.

In future work we will investigate the influence on prediction of a learning-related emotion lexicon; we will also investigate the relation between learning emotions and polarity.

6. REFERENCES

[1] N. Altrabsheh, M. Cocea, and S. Fallahkhair. Predicting students' emotions using machine learning techniques. In *The 17th International Conference on Artificial Intelligence in Education*, 2015. forthcoming.

[2] S. Aman and S. Szpakowicz. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, pages 196–205. Springer, 2007.

[3] T. Danisman and A. Alpkocak. Feeler: Emotion classification of text using vector space model. In *Convention Communication, Interaction and Social Intelligence*, volume 1, pages 53–59, 2008.

[4] S. D'Mello, T. Jackson, S. Craig, et al. Autotutor detects and responds to learners affective and cognitive states. In *Workshop on Emotional and Cognitive Issues at the International Conference on Intelligent Tutoring Systems*, 2008.

[5] F. Keshtkar and D. Inkpen. A corpus-based method for extracting paraphrases of emotion terms. In *Proceedings of the NAACL HLT Workshop on Computational approaches to Analysis and Generation of emotion in Text*, pages 35–44, 2010.

[6] B. Kort, R. Reilly, and R. W. Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies*, pages 43–436. IEEE Computer Society, 2001.

[7] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, volume 10, pages 1320–1326, 2010.

[8] A. Pak and P. Paroubek. Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, volume 5, pages 436–439, 2010.

[9] F. Tian, P. Gao, L. Li, W. Zhang, H. Liang, Y. Qian, and R. Zhao. Recognizing and regulating e-learners' emotions based on interactive chinese texts in e-learning systems. *Knowledge-Based Systems*, 55:148–164, 2014.

[10] Z. Yuan and M. Purver. Predicting emotion labels for chinese microblog texts. In *The 1st International Workshop on Sentiment Discovery from Affective Data*, volume 40, 2012.

[11] S. Zhao, L. Zhong, J. Wickramasuriya, and V. Vasudevan. Analyzing twitter for social TV: Sentiment extraction for sports. In *Proceedings of the 2nd International Workshop on Future of Television*, volume 2, pages 11–18, 2011.

Video-Based Affect Detection in Noninteractive Learning Environments

Yuxuan Chen
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
ychen18@nd.edu

Nigel Bosch
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
pbosch1@nd.edu

Sidney D'Mello
University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556, USA
sdmello@nd.edu

ABSTRACT

The current paper explores possible solutions to the problem of detecting affective states from facial expressions during text/diagram comprehension, a context devoid of interactive events that can be used to infer affect. These data present an interesting challenge for face-based affect detection because likely locations of affective facial expressions within videos of students' faces are entirely unknown. In the current study, students engaged in a text/diagram comprehension activity after which they self-reported their levels of confusion, frustration, and engagement. Data were chosen from various locations within the videos, and texture-based facial features were extracted to build affect detectors. Varying amounts of data were used as well to determine an appropriate window of data to analyze for each affect detector. Detector performance was measured using Area Under the ROC Curve (AUC), where chance level is .5 and perfect classification is 1. Confusion (AUC = .637), engagement (AUC = .554), and frustration (AUC = .609) were detected at above-chance levels. Prospects for improving the method of finding likely positions of affective states are also discussed.

Keywords

Affect detection; facial expression recognition; reading

1. INTRODUCTION

Educational activities like playing educational games [9], interacting with a computerized tutor [4], and comprehending text [13] have been linked to affective experiences that potentially play important roles in the learning process. Thus, automatically detecting and responding to specific affective states can be a useful technique for improving educational software [5]. A wide variety of approaches have been used to detect students' emotions and tailor instruction to their affective needs (see [8] and [5] for reviews). Affect detection is a core challenge that needs to be addressed before affect-sensitive instructional strategies can be devised.

Affect detection during interactions with educational technologies are a widely studied problem. The two most common approaches involve the use of interaction data (e.g., clicks, response times) from log files (called sensor-free detection as reviewed in [1]) and the use of physiological/behavioral sensors, such as webcams, electrodermal sensors, posture sensors, and so on (called sensor-based affect detection as reviewed in [3]). As an illustrative example, Kai et al. [11] built both interaction-based and video-based affect detectors while students played an educational game called Physics Playground [14]. Their data included affect labels corresponding to specific moments in the learning session (provided by human observers in real-time). The metric of performance was A', a close approximation of Area Under the ROC Curve (AUC), where A' = .5 is chance level and 1 is perfect classification. They were able to detect affective states at levels above chance: confusion (A' = .588 for interaction-based, .622 for face-based), engaged concentration (A' = .586 interaction, .658 face), and frustration (A' = .559 interaction, .632 face).

The aforementioned study highlights two commonalities of affect detection during learning from educational software. First, the software is typically interactive in nature, thereby providing considerable opportunities for external events (e.g., a new problem, submission of a response, system feedback, a hint) to trigger affective states. Information on these events and students' responses to these events provide valuable information to guide affect detection. Second, the data (log-files, videos, etc) used to build affect detectors is accompanied by affect labels corresponding to specific moments in a learning session. This allows label-based segmentation of the data stream and affords pinpointing the sections of the data stream for affect detection (typically windows of 10-20 seconds before the labels; e.g., [9]).

Data in some educational contexts are not well suited to creating affect detectors. For example, in self-paced reading tasks there are not necessarily many key events that are likely to trigger affective responses, unlike many educational activities where there is frequent feedback and interaction. Similarly, not all educational experiences include labeled-data that can be used to pinpoint the temporal location of affective states. For example, students might self-report their affective states *after* reading an entire passage or viewing an online lecture. This raises the additional challenge of how to segment the data stream for affect detection.

The present paper involves affect detection in the context of a noninteractive, but everyday learning task, involving mechanical reasoning from illustrated texts [7]. Students were presented with a complete text passage with an associated diagram for two minutes of study. Students self-reported their affective states after each a two minute study session, rather than any specific moment in the session. This data raised many challenges. First, interaction data was non-existent as there are no page turns or other navigation features that can be used to gain information about student behaviors. Due to the lack of interaction information, we use facial features extracted from videos of students' faces to detect affective states as they processed the text/diagram. Second,

without predictable events in the task that could trigger affective states and without affect labels during the study session, the position within a video where facial expressions of affective states are likely to occur is unknown. Rather than analyzing the entire video, knowing the location of affective states is important because the duration of affective experiences can be short and the facial expressions associated with affective states can be even shorter [2,6]. To address this problem we explore affect detection using different data window sizes and window positions within face videos to determine where displays of affect tend to occur and how long they last.

We also studied the role of learning goals on affect detection performance. Specifically, students studied the illustrated texts under two different instructional conditions. The first was to simply learn about a mechanical device (general instructions). This was followed by a focused goal that either directed students to review key components of the device or to pinpoint a particular problem with the device (specific instructions). We anticipate differences in affect detection results between the two types of instructions because they are expected to engender different levels of processing. Thus, we also build separate detectors for the two types of instructional goals to determine if there was a notable difference in detection performance.

Our main approach consisted of applying machine learning techniques to build detectors of confusion, engagement, and frustration with features extracted from facial videos using CERT [12], which is a well validated computer vision tool for extracting texture-based facial features. Detection results with different window sizes and positions show both the potential and the difficulty of detecting affective states from face videos when little is known about when displays of affect might likely occur. The data in this study come from studying instructional texts with illustration, and as such is representative of potential real-world education scenarios. Thus, determining how to detect affective states in this context is important for improving computerized education systems.

2. METHOD

Data Collection. Data were collected from 88 college students from the Psychology subject pool at a large public university in the mid-South. These students from diverse backgrounds were asked to study illustrated texts about four everyday devices: an electric bell, a toaster, a car temperature gauge, and a cylinder lock. The illustrated texts were taken from Macaulay's book, *The Way Things Work* (1988), with text order counterbalanced across participants. Each of the general and specific study instructions lasted for two minutes. Videos of the students' faces were recorded with webcams mounted on the computer monitors. Upon completion of each two-minute study session, students rated their levels of engagement, confusion, and frustration on scales of 1 (very little) to 6 (very much). Students studied all four devices with device order counterbalanced across students, thereby resulting in 704 videos (88 students × 4 devices × 2 study goals per device).

Three students' videos were discarded due to recording errors, which resulted in 680 usable videos. These videos were then analyzed using CERT, which computed the likelihoods of occurrence for facial action units (AUs) in every video frame. Large outliers in AU likelihoods were found in the last two seconds of most videos, which are probably the result of students posture shifts in response to the end of the session. The last 2 seconds were removed to compensate for these anomalies, so each video was then exactly 1 minute 58 seconds long.

Feature Engineering. CERT was able to detect 20 different AUs as well as unilateral (one side of the face only) AUs, head orientation, and nose position. From the CERT data, windows of eight different sizes (2, 3, 6, 9, 12, 15, 20, and 30 seconds) were generated. For each size, windows were drawn from the beginning, middle, and end of each video. If the window came from the beginning or the end of the video, the margin from the beginning or the end was equal to the length of the window. Figure 1 illustrates examples of windows created in this manner.

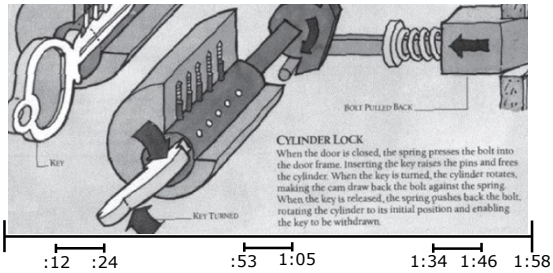


Figure 1. Positions of 12-second windows during the task.

The AU data of the windows were standardized within each student. This was followed by feature generation, in which the median, maximum, and standard deviation of the frame-level AU likelihoods were computed within each window and used as features. Some windows had less than one second of valid data, largely because the camera could not capture the student's face when they moved too much, leaned outside the camera's field of view, or when the face was occluded due to gestures. These windows were removed from the dataset, as we assumed that an affective facial expression would usually be longer than one second. Features exhibiting high multicollinearity (variance inflation factor > 5) were removed.

Supervised Classification. The features obtained above were used to construct classification models using the Waikato Environment for Knowledge Analysis (WEKA), a machine learning tool.

The classification task comprised binary high vs. low affect ratings for confusion, frustration, and boredom. The medians of the engagement, confusion, and frustration ratings on the 1-6 scale were 4, 2, and 1, respectively. We used a median split to discretize the affect ratings into "low" and "high", discarding the median instances except in the case of frustration where the median was 1. For frustration 1 was used as the "low" label.

For model validation, leave several out student-level cross-validation was applied. The training data were randomly chosen from two thirds of the students. RELIEF-F feature ranking was used to select the most diagnostic features on the training data only. The data of the remaining students were used to test the generalizability of the classifiers. Each model was trained and tested for 150 iterations with random students selected for training and testing each iteration to reduce random sampling error. Fifteen different classifiers were applied to help determine which among the eight window sizes tended to work best. Regression analysis was also explored, though the resulted models showed little promise and will not be discussed further.

3. RESULTS

The best classification models that merged videos recorded during both general and specific study instructions are listed in Table 1. The AUCs for confusion and frustration were well above chance, whereas the AUC for engagement was only slightly higher than chance level.

Table 1. Overview of results when general and specific instructional videos were combined.

Affective State	Classifier	AUC	Accuracy	No. Instances	No. Features	Window Size
Confusion	Updateable Naïve Bayes	0.637	62%	352	65	9 seconds
Engagement	AdaBoostM1	0.554	55%	403	49	20 seconds
Frustration	AdaBoostM1	0.609	64%	356	39	6 seconds

It should be noted that there were fewer than 680 instances (the total number of usable videos) for these classification models. This was largely because instances that captured less than a second of data were eliminated and the median splits that were performed to ascertain "low" and "high" values resulted in the loss of instances with affect ratings at the median.

General vs. Specific Study Instructions. The best AUCs for each video type are in Table 2. We note that for engagement, AUCs for individual general-instruction and specific-instruction models were higher than when the videos were combined. However, for confusion and frustration, it seems that the best AUCs are mostly equivalent across both individual videos and combined videos.

Table 2. Comparison of classification performance (AUC) for models using only explanation, only review, or both types of data.

Affective State	General	Specific	Both
Confusion	0.664	0.606	0.637
Engagement	0.610	0.580	0.554
Frustration	0.600	0.620	0.609

Window Position. The best AUCs (for combined models) with respect to the three window positions (i.e., beginning, middle, and end) are shown in Figure 2. Clear patterns stand out for confusion and frustration. The windows taken from the beginning of the videos seem to be more effective for confusion than those taken from the middle or the end of the videos, whereas the windows drawn from the end of the videos may best capture frustration. There is no clear pattern for engagement.

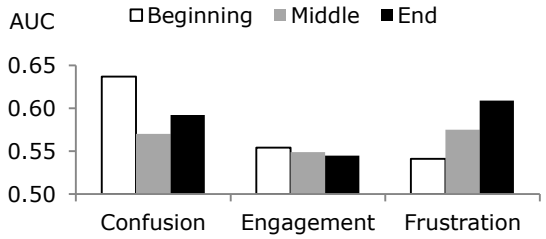


Figure 2. AUC of models using data from different positions within videos.

Window Size. Figure 3 shows the best AUCs as a function of window size for the combined models. The window position was held constant as the best window position for each affective state as noted in Figure 2. Confusion and frustration again show interesting patterns. AUC peaks at a certain window size where classification is much more successful than the surrounding window sizes. The peaks for the AUCs of confusion and frustration both occur when the window size is relatively small (9 seconds for frustration and 6 seconds for confusion). Conversely the window size seems to have no notable relationship with AUC for engagement.

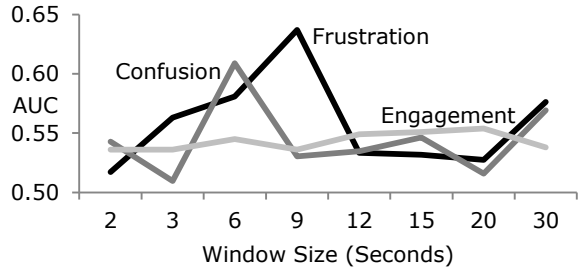


Figure 3. AUC of models as window size varies.

4. DISCUSSION

The novelty of the contributions in this paper stems from the differences between data in this study and previous affect detection work. Facial expressions of affect are often related to events in an interface (e.g., feedback, new problems), but the present study tracked affect in a noninteractive study activity – comprehension from illustrated texts. Affect labels used for detection in this study were given as retrospective judgments covering an entire 2-minute study period, so they do not provide any information about the appropriate position in the video to search for facial expressions. Thus the position of potential facial expressions in the face videos is entirely unknown. Unlike related studies with affect labels not tied to specific moments in a learning session (e.g., [10]), the current research used a subset of data from the session rather than considering all data in the session. This approach was chosen to better capture the brief nature of affective facial expressions. In the remainder of the section we discuss our main findings, and highlight limitations and avenues for future work.

Main Findings. The results above show that confusion and frustration ratings of the students can be detected with greater accuracy than the engagement rating, but that detection was successful above chance for all three affective states despite the difficulty of identifying a brief affective facial expression within the videos. However, if we split the general-instruction videos from the specific-instruction videos, the engagement rating may be better modeled, especially for the general videos. For confusion, a 9-second window at the beginning of the video worked best for classification; for frustration, a 6-second window at the end of the video was best. There were no clear patterns with respect to window position or window size for engagement.

The results suggest that when given a video with the occurrences of different affects unknown, affect ratings for confusion, frustration, and potentially engagement can still be well modeled. Smaller window sizes such as 6 or 9 seconds can be a good start to find such best models for confusion and frustration, which parallels the results in previous research [2]. Also, clips taken from the beginning of the video may yield good models for confusion, and those taken from the end of the video may work well for frustration. This seems to suggest that students' facial expressions at the beginning of the 2-minute study session can potentially indicate how confused they think they are in the end,

and that their facial features at the end of a session may provide evidence as to how frustrated they rate themselves to be. It seems that when students confront a specific task, their first impression or assessment of the difficulties and intricacies of the task can last until the end of the task. As they try to understand new concepts or to tackle problems, they experience the details of the task that they might not have known before. This may be why at the end of the task, whether they completely absorb the concepts or solve the problems, they may still feel frustrated and challenged and such emotions can be detected by analyzing facial expressions.

The reasons why engagement detection is a difficult task in this context may be due to differences in facial expressions of engagement between the general and specific study periods. It is possible that students' definitions of engagement may be linked to the particular task they are working on. General and specific study periods may be essentially different tasks, the former requiring students to intake new concepts and the latter challenging students to focus on specific aspects of concepts they have learned. Thus students may experience and display engagement differently between the two study periods, which may explain why model performance improved when each period was analyzed independently.

Limitations and Future Work. The results were promising, but there are a few limitations to this research. First, the number of videos was rather low and around 30% of the windows had to be discarded due to difficulties in registering the face (mostly due to hand-over face gestures). Also, the videos for the research were only 2 minutes long. If the window size is 30 seconds, trimming off the beginning and end 30 seconds from a video indicates that we only have one minute left for the video and the segments taken from this video can be overlapping, which is not ideal. Further research should consider a greater number of longer videos, which would allow a more thorough search of window positions and window sizes, as well as a test of the generalizability of our results to longer learning sessions.

In addition, we adopted a rather arbitrary approach of searching the start, middle, and end of each video to identify diagnostic affect expressions. In future work, we will delve more deeply into the data we already have. The feature selections of models will be examined to determine if different AUs are selected for different parts of the videos. Additionally, different methods will be applied to search for positions in the videos where affective facial expressions occur. For example, we may utilize the 9-second window size to perform a random sampling across all videos, taking segments from random positions within each video to offer more insight into how facial expressions can be leveraged for affect detection. It may also be possible to develop techniques for finding the optimal window position on a per-video basis, for example by searching for peaks or valleys in calculated features, and using windows of data specific to each video.

Concluding Remarks. In summary, this paper introduces a potential method to detect students' affective states in non-interactive instructional contexts when the locations and durations of affective facial expressions are unknown. Much work remains to be done to improve these techniques, but our results show that detecting affective states with these challenging data is certainly possible, highlighting the importance of correctly identifying the position and length of windows of data within each video.

5. ACKNOWLEDGEMENTS

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

6. REFERENCES

1. Baker, R. and Ocumpaugh, J. Interaction-Based Affect Detection in Educational Software. In R. Calvo, S. D'Mello, J. Gratch and A. Kappas, eds., *The Oxford Handbook of Affective Computing*. New York: Oxford University Press, 2015, 233–245.
2. Bosch, N., D'Mello, S., Baker, R., et al. 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 (2015), 379–388.
3. Calvo, R.A. and D'Mello, S. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing 1*, 1 (2010), 18–37.
4. Craig, S., Graesser, A., Sullins, J., and Gholson, B. Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media 29*, 3 (2004), 241–250.
5. D'Mello, S., Blanchard, N., Baker, R., Ocumpaugh, J., and Brawner, K. I feel your pain: A selective review of affect-sensitive instructional strategies. In R. Sottilare, A. Graesser, X. Hu and B. Goldberg, eds., *Design Recommendations for Intelligent Tutoring Systems - Volume 2: Instructional Management*. 2014, 35–48.
6. D'Mello, S. and Graesser, A. The half-life of cognitive-affective states during complex learning. *Cognition & Emotion 25*, 7 (2011), 1299–1308.
7. D'Mello, S. and Graesser, A. Inducing and tracking confusion and cognitive disequilibrium with breakdown scenarios. *Acta Psychologica*, 151 (2014), 106–116.
8. D'Mello, S. and Kory, J. Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. *Proceedings of the 14th ACM international conference on Multimodal interaction*, ACM (2012), 31–38.
9. Graesser, A., Chipman, P., Leeming, F., and Biedenbach, S. Deep learning and emotion in serious games. *Serious games: Mechanisms and effects*, (2009), 83–102.
10. Grafsgaard, J.F., Wiggins, J.B., Boyer, K.E., Wiebe, E.N., and Lester, J.C. Automatically Recognizing Facial Expression: Predicting Engagement and Frustration. *Proceedings of the 6th International Conference on Educational Data Mining*, (2013).
11. Kai, S., Paquette, L., Baker, R., et al. Comparison of face-based and interaction-based affect detectors in physics playground. *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, International Educational Data Mining Society (in press).
12. Littlewort, G., Whitehill, J., Wu, T., et al. The computer expression recognition toolbox (CERT). *2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011)*, (2011), 298–305.
13. Mills, C., Bosch, N., Graesser, A., and D'Mello, S. To quit or not to quit: predicting future behavioral disengagement from reading patterns. *Proceedings of the 12th International Conference on Intelligent Tutoring Systems (ITS 2014)*, Switzerland: Springer International Publishing (2014), 19–28.
14. Shute, V.J., Ventura, M., and Kim, Y.J. Assessment and learning of qualitative physics in Newton's Playground. *The Journal of Educational Research 106*, 6 (2013), 423–430.

Modeling Classroom Discourse: Do Models that Predict Dialogic Instruction Properties Generalize across Populations?

Borhan Samei[1] Andrew M. Olney[1] Sean Kelly[2] Martin Nystrand[3]
Sidney D'Mello[4] Nathan Blanchard[4] Art Graesser[1]
[1] University of Memphis [2] University of Pittsburgh [3] University of Wisconsin [4] University of Notre Dame
bsamei@memphis.edu

ABSTRACT
It has previously been shown that the effective use of dialogic instruction has a positive impact on student achievement. In this study, we investigate whether linguistic features used to classify properties of classroom discourse generalize across different subpopulations. Results showed that the machine learned models perform equally well when trained and validated on different subpopulations. Correlation-Based Feature Subset evaluation revealed an inclusion relationship between different subsets in terms of their most predictive features.

Keywords
Classroom Discourse, Machine Learning, Authenticity, Uptake

1. INTRODUCTION
Previous research on classroom instruction has shown the positive influence of dialogic instruction on student achievement [2]. Dialogic instruction is a classroom discourse strategy based on the free and open exchange of ideas between teachers and students. It is hypothesized that dialogic instruction improves achievement by increasing student engagement in classrooms [3, 5].

Previous efforts to carefully quantify teachers' use of dialogic instruction include three major studies by Nystrand and colleagues [6]. Nystrand et al.'s approach included coding discourse moves with a focus on the nature of question events, which are defined by the discourse context preceding and following a question. Question events include the question along with the response and optional evaluation/follow up. They follow a pattern that mirrors the well-known initiation response, and evaluation sequence (IRE). This coding scheme treats questions as sites of interaction and takes into account the response and evaluation. As a result, the questions alone do not uniquely determine the dialogic properties of the event; instead, they create a context through which dialogic properties may be realized.

In this research, question events were coded with five properties that were hypothesized to relate dialogic instruction to student achievement: authenticity, uptake, level of evaluation, cognitive level, and question source. However, Nystrand and Gamoran found that among these variables, authenticity and uptake were the most strongly related to student achievement [2, 8]. A question is defined as having authenticity when the asker does not have a pre-scripted answer, i.e. an open-ended question, which creates a context for students to contribute to an open ended discussion. Uptake occurs when one asks a question about something that another person has said previously. When teachers exhibit uptake, they incorporate student contributions into the discussion, potentially encouraging additional student contributions.

Question properties were live-coded by observers in Nystrand et al.'s study, a time-consuming and expensive process requiring trained classroom observers. To facilitate research into dialogic instruction, we recently developed a machine learning model to investigate the extent to which question properties can be automatically coded [9]. This previous study showed that machine learned models can predict authenticity and uptake as accurately as human experts in a setting where the questions are presented without the preceding and following context, which was the information available to the machine learned model.

Machine learned models, often referred to as *predictors or classifiers,* are sensitive to the properties of the data set on which they are trained. However, in order to perform large scale analysis, these models must be applicable to new, larger, and more diverse data. An important question in this work is whether the models systematically vary their predictions with different subpopulations in the data (e.g. different demographics). This systematic variation, essentially bias, could lead to incorrect predictions and flawed conclusions when the model is applied to a sample drawn from the same subpopulation as opposed to different subpopulations and indeed any sample where the individuals are spatially or temporally correlated may potentially have problems of generalizability.

Some recent research has focused on examining generalizability of EDM models. For example, Baker and Gowda studied the difference in student behaviors associated with disengagement in urban, suburban, and rural schools and found that urban students went off-task more often and exhibited significantly more careless behaviors than students in the rural and suburban schools [1]. Furthermore, Ocumpaugh et al, found that models trained on a population drawn primarily from one demographic grouping (rural, urban, or suburban) do not always generalize to populations drawn primarily from the other demographic groupings [7]. Generalization can sometimes occur across seemingly distinct contexts. For example, San Pedro et al. (2011) found that their models of detecting student carelessness were generalizable among different tutor interfaces (i.e. with and without an embodied conversational agent), as well as different school settings (i.e. Philippine high school and US middle school) [10].

In this paper we investigated the generalizability of two previously developed models for predicting authenticity and uptake in classroom discourse [9].

2. METHOD
We trained and tested our models using data collected from the Partnership for Literacy Study (Partnership). The data set consists of question events as recorded by the classroom observers. Partnership was a study of professional development, instruction, and literacy outcomes in middle school, in which 120 classrooms in 21 schools were observed twice in the fall and twice in the spring

over two years. The Partnership data set consists of observational data which were coded using the CLASS 4.24 computer-based data coding program [9]. Inter-rater agreement was approximately 80% on question properties with observation-level inter-rater correlations averaging approximately .95 [6].

Some of the teachers received special training in the first year and their classes were observed again in the second year. We used teacher training to split the data into Pre-training (N=7082) and Post-training (N=13655) groups. The school location was coded into categories of large and mid-size central city, urban fringe of mid-size city, small town rural outside MSA (metropolitan statistical area), and rural inside MSA. Based on the number of data points in each category, we split the data in two categories: Urban (i.e. Mid-size and Large Central City, N=13126) vs. Non-urban (the rest of categories, N=10911). Table 1 shows the distribution of authenticity and uptake across the different splits.

Table 1. Proportion of Authenticity and Uptake in different subsets and the full data set.

Category	% Authenticity	% Uptake
Non-urban : Urban	54 : 47	23 : 20
Pre-training : Post-training	39 : 52	15 : 24
Full-set	50	21

As seen in Table 1, authentic questions were more frequent than uptake in general, and the Non-urban group had higher rates of both authenticity and uptake than Urban. Overall the distribution of authenticity and uptake was similar among Non-urban, Post-training, and Full-set. Pre-training had the lowest rate of authenticity and uptake compared to others. It is also worth noting that teacher training was apparently quite effective at increasing both authenticity and uptake, as shown by the increase from Pre- to Post-training.

Based on our previous work on automating coding the questions with authenticity and uptake [9], we applied machine learning to train separate classifiers for authenticity and uptake on each of the above subsets. The models use linguistic features utilized in the classification of question types [8], including parts of speech, manually constructed bags of words (e.g., causal antecedent words), and positional information.

Most of the features are binary and indicate the presence/absence of certain keywords or part of speech tags in the question. Other features include attributes that show the position of the target keyword in the question in addition to presence/absence using four values: middle, beginning, end, and none. For example, if a question consisted of four words, e.g. "word1 word2 word3 word4" the position of "word1" is captured as beginning and "word4" as end, furthermore "word2" and "word3" are both captured as middle and if there were only two words in the question, we consider the first one as the beginning and the other as the end.

An example of a feature is causal consequent words, which include "outcomes," "results," "effects," etc. Similarly, procedural words are defined as a set of keywords including "plan," "scheme," "design," etc. Moreover, part of speech tags, such as determiner, noun, pronoun, adjective, adverb, and verb, and certain words such as "What," "How," and "Why," were also included in the feature set. More complete descriptions and justifications of these features for question classification can be found in the mentioned references.

We first trained models on each subset and evaluated their performance using 10-fold cross validation within the subset. Next, we tested generalizability by training on one subset and testing on its dual. For example, a model trained on Urban subset was tested on the Non-urban subset and vice versa. Moreover, the models trained on the full set of data were also tested on each subset. This methodology allows for the following contrasts. First, cross validation within a subset establishes a reasonable upper bound on performance since training and testing instances, while distinct, still come from the same subset. Second, training on one subset and testing on its dual subset establishes a reasonable lower bound on performance, since accuracy would be determined by shared features between the subsets rather than by distinctive properties to each subset. Training on the full data set and testing on subsets (thus training and testing on those subsets) allows similar comparisons of bias. For example, if training on the full set and testing on set A has higher accuracy than testing on set B, we may hypothesize that the features of the full model are better aligned with the features of A, or the prevalence of category distribution in the full set better matches that of A.

3. RESULTS & DISCUSSION

We first trained separate models to predict authenticity and uptake and evaluated the models using on 10-fold cross validation for each subset. For each category (e.g. Urban, Non-urban, etc.) separate decision tree models were trained and evaluated using WEKA [4]. The models for predicting uptake were trained on a random subsample of the data to obtain an even (50-50) distribution. Table 2 shows the performance of the models along with the performance of a model trained on the full set of data.

Table 2. Performance of the decision tree models trained on different data subsets using 10-fold cross validation.

Training Data	Authenticity		Uptake	
	Accuracy	*Kappa*	*Accuracy*	*Kappa*
Non-urban	0.61	0.21	0.59	0.19
Urban	0.62	0.24	0.60	0.20
Pre-training	0.64	0.24	0.61	0.23
Post-training	0.63	0.26	0.61	0.22
Full-set [9]	0.64	0.28	0.62	0.24

As seen in Table 2, the models on different splits show comparable performances, where the maximum difference on their accuracy is 0.03 (3%). To examine performance of these models and their generalizability across different subsets, we trained models on one subset and tested on its dual subset, e.g. Urban – Non-Urban. In Table 3, the performance of each model is tested on its dual. Additionally, the models trained on full set of data are tested on different subsets.

Table 3. Generalizability of models on different splits of data (trained on one tested on other).

Train	Test	Authenticity	Uptake
		Accuracy	*Accuracy*
Non-urban	Urban	0.60	0.63
Urban	Non-urban	0.62	0.62

Full-set	Non-urban	0.70	0.68
Full-set	Urban	0.68	0.68
Pre-training	Post-training	0.59	0.62
Post-training	Pre-training	0.60	0.64
Full-set	Pre-training	0.70	0.68
Full-set	Post-training	0.72	0.67

In Table 3, training on one subset and testing on its dual is never more than 2 percentage points away from the reverse. Thus the results are fairly stable. However there are several patterns of differences of interest. First, accuracy for the authenticity models when trained on Urban and tested on Non-urban is slightly higher than when trained on Non-urban and tested on Urban, however the uptake model performs slightly better when trained on Non-urban and tested on Urban than the reverse. Moreover, uptake and authenticity accuracy were higher for models trained on Post-training and tested on Pre-training compared to the reverse.

These results show that the model's performance when trained on one subset and tested on its dual is comparable to the results presented in Table 2. These results suggest that Pre-training and Non-urban are more likely to be proper subsets of Post-training and Urban respectively than the reverse. In other words, Post-training and Urban models may (by virtue of having better training data for their duals) include features that are effective on Pre-training and Urban, however this could also be due to the base rate or prevalence of authenticity and uptake in these subsets which needs further investigation.

In order to further examine the models, we compared the confusion matrices to illustrate the bias/prevalence of the models. Using the confusion matrices of models presented in Table 2 (i.e., 10-fold cross validated), we subtracted the confusion matrix when training on the Full-set from the others (Figures 1 and 2.) The resulting matrices represent the extent to which the confusion matrix of a model is different from the baseline model (i.e. Full-set). Each of the confusion matrices were separately proportionalized (before subtraction) by size of the corresponding subset to make the values comparable. Positive values in the figures indicate that the associated category occurred more often in the subset than in the Full-set. Likewise negative values mean that the category occurred less often in the subset than the Full-set.

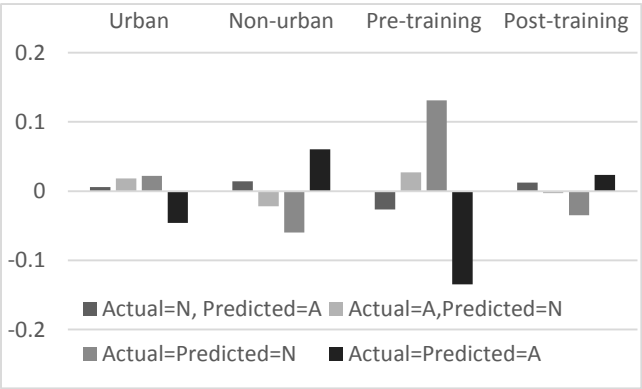


Figure 1. Normalized distance of confusion matrices of Authenticity models on subsets from full-set (A=Authenticity, N= Non-authentic).

It is seen in Figure 1 that the Urban and Post-training authenticity models are the most similar to the Full-set model because their differences with the Full-set are close to zero. This suggests that these models are not biased with respect to the Full-set. However, the Non-urban and Pre-training have larger differences with the Full-set model. Non-urban and Post-training subsets have more true-positives (Actual=Predicted=A) and less true-negatives (Actual=Predicted=N) than the Full-set while the opposite is true for Urban and Pre-training. This contrast in true-positive and true-negatives creates a trade-off in the models which previously appeared to be consistent. Specifically, Figure 1 reveals that Pre-training is more biased towards predicting N (non-authentic instances) than A (authentic instances) which may be due to the fact that there are fewer authentic instances than non-authentic in the Pre-training subset (39% vs. 50%, see Table 1). Conversely, the Non-urban model is biased towards A at the expense of N reflecting the higher distribution of A in the Non-urban subset (54% vs. 50%, see Table 1). Overall, the trade-off between true-positive and true-negative is symmetric which explains why the overall accuracy of the models is not particularly affected despite the differences in error patterns.

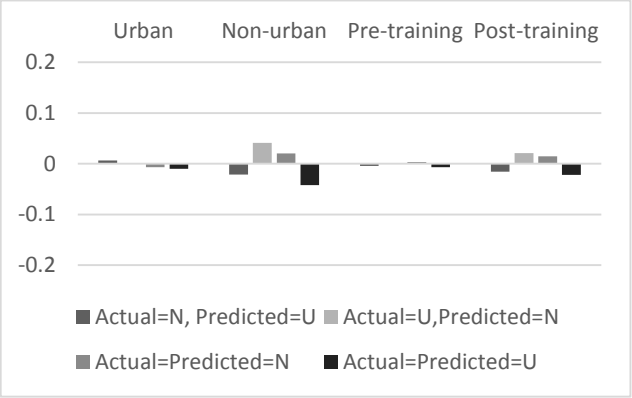


Figure 2. Normalized distance of confusion matrices of Uptake models on subsets from full-set (U=Uptake, N= Non-uptake).

Similar to Figure1, Figure 2 shows the distance between confusion matrices of uptake models. The overall distance of uptake models on subsets compared to Full-set is lower than the distance of authenticity models. Note that the uptake models were trained and 10-fold cross validated on a subsample with an even distribution (50-50) which removes the effect of prevalence on the models. Notably, the Non-urban model sacrifices more true-positives at the expense of false-negatives which explains the lower accuracy of Non-urban in predicting uptake (59% vs. 62%, see Table 2) while the rest of models are very close to the Full-set and hold a balanced tradeoff between true-positive and true-negative.

We examined the models in more detail using Correlation-Based Feature Subset evaluation (CFS). Specifically, we analyzed the frequency of each CFS feature to determine the most important CFS features for each subset. Table 4 shows the CFS results for each model. The features are presented in groups to show whether they were common between the models (shared) or exclusively included in one model only.

Table 4. CFS results, most predictive features of each model grouped based on inclusion.

Models	Authenticity	Uptake
	Urban & Non-Urban	
Shared	Wh, What	Why
Urban only	Be, Judgmental, Enablement	Neg, Pron, Causal_Antecedent
Non-urban only		Disjunction
	Pre-training & Post-training	
Shared	Judgmental, What	Neg, Metacog, Pron, Judgemental, Why
Pre-training only	Comparison	What
Post-training only	Be, Wh, Enablement	Modal, No, Causal_Antecedent

Although the models show similar performance, the most predictive features of each model is different, as seen in Table 4. However there are also marked commonalities among the groups. The features for authenticity on the Non-urban subset, for instance, are fully included in the Urban authenticity subset. Thus this analysis further supports the interpretation of inclusion suggested by the pattern of results in Table 3.

Similarly most of the features of pre-training are included in the post training features, which implies that although teachers' language changed after they received training, the result was that their linguistic behavior broadened with training such that their pre-training behavior was still evident.

4. CONCLUSION

We investigated the generalizability of previously presented models that predict authenticity and uptake in classroom discourse. Overall the results showed that the proposed models' performance is consistent among different subsets of the data set. However, we also found that some subpopulations were potentially more representative of the nature of dialogic instruction than others, making them better for classifier training.

The inclusion relationship between our subsets was investigated by comparing the confusion matrices of our models which revealed that authenticity models of supersets (i.e. Urban and Post-training) were closer to the full-set model than their duals. The consistent accuracy of the models on different subsets was attributed to the tradeoff between true-positive and true-negative predictions which was also explained by the prevalence and bias of the subsets towards one category.

We plan to apply our model to new data which is being collected currently. The proposed models will be applied with the ultimate goal of recording and coding classroom interaction in a fully automatic way and generating statistical reports to show effective instructional strategies. While the models proposed in this paper showed generalizability, another direction of future work is to improve the accuracy by adjusting current features and adding new predictive features to our models.

5. ACKNOWLEDGMENTS

This research was supported by the Institute of Education Sciences (R305A130030) and the National Science Foundation (IIS 1352207). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of these sponsoring agencies.

6. REFERENCES

[1] Baker, R. S., & Gowda, S. M. 2010. An Analysis of the Differences in the Frequency of Students' Disengagement in Urban, Rural, and Suburban High Schools: *Proceedings of the 3rd International Conference on Educational Data Mining*, 11-20.

[2] Gamoran, A., & Nystrand, M. 1991. Background and instructional effects on achievement in eighth-grade English and social studies: *Journal of Research on Adolescence, 1*(3), 277-300.

[3] Gamoran, A., & Nystrand, M. 1992. Taking students seriously: *Student engagement and achievement in American secondary schools*, 40-61.

[4] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The WEKA data mining software: an update: *ACM SIGKDD explorations newsletter, 11*(1), 10-18.

[5] Kelly, S. 2007. Classroom discourse and the distribution of student engagement: *Social Psychology of Education, 10*(3), 331-352.

[6] Nystrand, M., & Gamoran, A. 1997. The big picture: Language and learning in hundreds of English lessons: *Opening dialogue*, 30-74.

[7] Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., & Heffernan, C. 2014. Population validity for Educational Data Mining models: A case study in affect detection: *British Journal of Educational Technology, 45*(3), 487-501.

[8] Olney, A., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. 2003. Utterance classification in AutoTutor: *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, 1-8.

[9] Samei, B., Olney, A., Kelly, S., Nystrand, M., D'Mello, S., Blanchard, N., Graesser, A. 2014. Domain Independent Assessment of Dialogic Properties of Classroom Discourse. *Proceedings of the 7th International Conference on Educational Data Mining*, 233-236.

[10] San Pedro, M. O., d Baker, R. S., & Rodrigo, M. M. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics: *Artificial Intelligence in Education*, 304-311.

Breaking Off Engagement: Readers' Disengagement as a Function of Reader and Text Characteristics

Patricia J. Goedecke,[1] Daqi Dong,[1] Genghu Shi,[1] Shi Feng,[1] Evan Risko,[2] Andrew M. Olney[1] Sidney K. D'Mello,[3] Arthur C. Graesser[1]

[1]University of Memphis
The Institute for Intelligent Systems
Memphis Tennessee, USA
1 901-678-2000
trish.goedecke@gmail.com

[2]University of Waterloo
Department of Psychology
Waterloo Ontario, Canada
1 519-888-4567
efrisko@uwaterloo.ca

[3]University of Notre Dame
Department of Psychology
Notre Dame Indiana, USA
1 574-631-322
sdmello@nd.edu

ABSTRACT

Engagement during reading can be measured by the amount of time readers invest in the reading process. It is hypothesized that disengagement is marked by a decrease in time investment as compared with the demands made on the reader by the text. In this study, self-paced reading times for screens of text were predicted by a text complexity score called formality; formality scores increase with cohesion, informational content/genre, syntactic complexity, and word abstractness as measured by the Coh-Metrix text-analysis program. Cognitive decoupling is defined as the difference between actual reading times and reading times predicted by text formality. Decoupling patterns were found to differ as a function of the serial position of the screens of text and the text genre (i.e., informational, persuasive, and narrative) but surprisingly not as a function of reader characteristics (reading speed and comprehension). This underscores the importance of mining text characteristics in addition to individual differences and task constraints in understanding engagement during reading.

Keywords

Coh-Metrix; comprehension; decoupling; engagement; formality; genre; mind wandering; reader characteristics; reading; text characteristics.

1. INTRODUCTION

Engagement during reading is essential for comprehension and learning [1]. Methods for gauging engagement include measuring time invested in the reading process and eye tracking [2-5]. We hypothesize that when mind wandering or other forms of disengagement occur, there is a marked decrease in time allocation; text characteristics then have little impact on reading times. The disjoint relationship between textual demands and time investment is termed decoupling. Cognitive decoupling is defined as the difference between actual reading times and reading times predicted by text characteristics.

This study investigates how engagement changes as a reader progresses through screens of text in moderately lengthy documents. Changes are expected to be moderated by characteristics of reader and text. Relevant reader characteristics included overall reading speed and comprehension; text characteristics included text difficulty and genre.

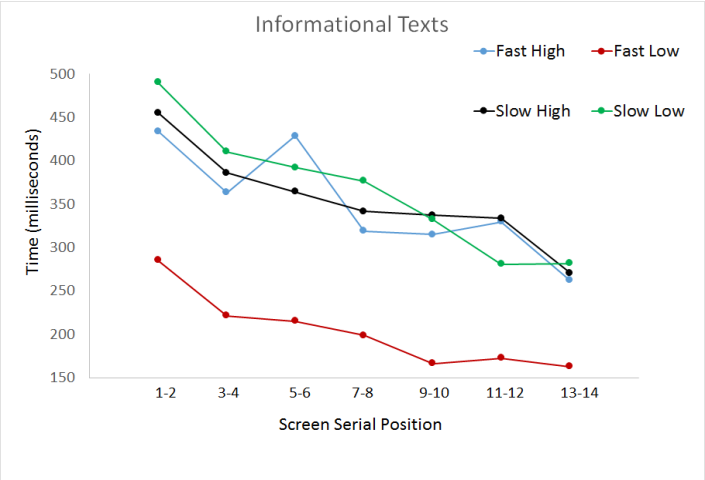
1.1 Text Difficulty

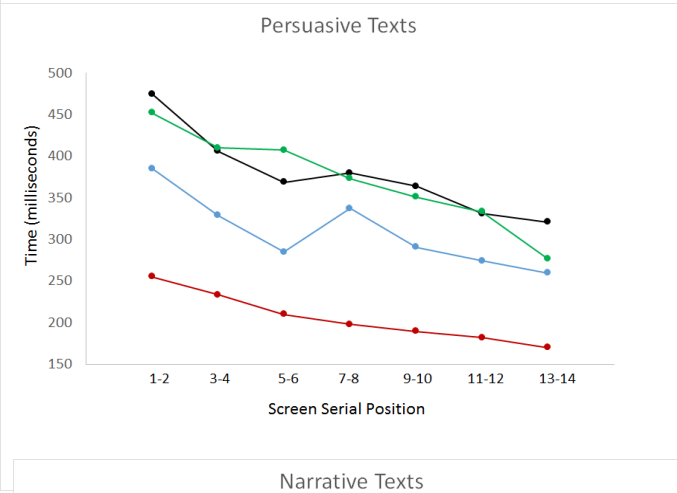
Text difficulty can been scaled in a variety of ways, validated by predicting grade levels of text and performance on psychometric tests of comprehension [6]. The Flesch-Kincaid Grade Level formula is a readability assessment based on word length and sentence length [7]. The Coh-Metrix tool analyzes text on multiple levels of language and discourse using computational linguistics techniques [8, 9]. Graesser et al [10] have introduced formality as a composite measure of text difficulty based on Coh-Metrix higher order principal components. Formality has a high correlation (0.72) with Flesch-Kincaid Grade Level. Discourse formality is calculated as a mean of five Coh-Metrix principal components having positive values for increasing levels of difficulty. These include: (1) referential cohesion; (2) deep (causal) cohesion; (3) informational content; (4) syntactic complexity and (5) word abstractness. Normative values (z-scores) for these 5 factors and formality are based on the TASA corpus. These norms are used to compute difficulty scores on new texts that researchers wish to analyze.

1.2 Genre and Order of Information

Genre is a discourse feature that is expected to influence engagement as well as text difficulty. Narrative texts are considered the most intrinsically engaging genre for most readers; and least difficult, compared with informational texts [6], [9], [11, 12]. Persuasive texts lie in-between narrative and informational text in expected difficulty and engagement.

The order of information presented in the text is also expected to influence engagement as well as text complexity. Readers begin engaged with a text, but may eventually lose interest and disengage as the text progresses. Research is needed to document the time allocated to texts at different points in the text. Interestingly, basic research questions have not yet been investigated at a fine grained level. Available research has only compared mind wandering as a function of texts that vary in difficulty as entire texts and these





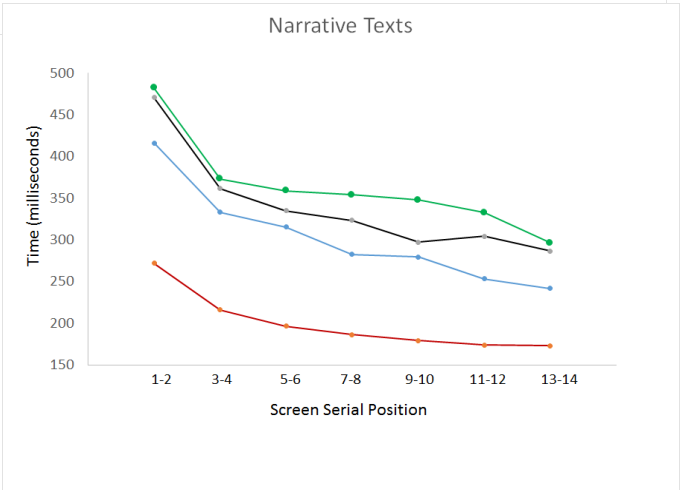


Figure 1. Reading Time per Word as a Function of Screen Serial Position, Segregated by Genre and Reader Type

studies are not consistent with respect to mind wandering increasing or decreasing with text difficulty [13].

1.3 Decoupling

Cognitive decoupling is a discrepancy between textual demands and the time a participant invests in reading a text. Decoupling increases as a function of the readers' disengagement with the text. Decoupling in this study is measured as the difference between actual reading times and times predicted by text characteristics. We interpret positive decoupling scores to indicate that a participant is investing more time in reading a text than the text characteristics demand. According to our assumptions, negative values of decoupling represent a participant investing less time than text characteristics' demands. The Coh-Metrix formality z-scores were used to measure text difficulty of a text, as normalized by the TASA corpus. Analogously, the reading time for each text segment was normalized through z-scores for individual readers on the mean reading time per word for the text segment under consideration (compared with the other text segments for that individual). Decoupling is normalized reading times for a particular person minus the normalized text difficulty based on the TASA corpus.

We predict that decoupling scores will become more negative or less positive as a reader progresses through a text, corresponding with a decrease in engagement. However, previous research [14] has not identified the shape of this decreasing function for different categories of texts and readers. These effects are predicted to be moderated by reader characteristics and genre.

2. METHODS

This study had 254 participants in two groups: 128 participated online via Mechanical Turk; 126 undergraduate Psychology students participated in a lab study.

Participants were classified according to reading time and comprehension using the Nelson Denny assessment with median split criteria. Participants read one text from each of three genres in counterbalanced order; texts assigned were randomly sampled from 24 informational, 24 persuasive, and 25 narrative texts. Following reading, participants wrote a 75-100 word summary of each text; then rated the familiarity, value, and interest for each text.

Participants used the spacebar to advance through each screen, providing reading time measurements Self-paced reading times were measured as average time per word in milliseconds for each screen of text. The number of words per screen ranged from 79 to 131, with a mean of 88.8 and a standard deviation of 11.0. The number of screens ranged from 10 to 23 per text.

3. RESULTS

3.1 Word Reading Times as Function of Text and Reader Characteristics

Mean reading times per word are presented as a function of serial position of screens of text, through position 14. Figure 1 shows times for informational (1a), persuasive (1b), and narrative texts (1c). Participants are segregated into slow versus fast readers and high versus low comprehenders.

In Figure 1, reading time functions are similar for readers with differing comprehension levels and reading speeds. We fit linear functions to each reader's times as a function of serial position, performing an ANOVA on the slopes. As expected, the slopes were negative, reflecting serial reading time decreases. A significant effect appeared in the Genre x Reading Time x Comprehension ANOVA: the slopes were lower for fast than slow readers, $F(1, 748) = 16.54$, $p < .001$. Intercepts were lower for fast readers, $F(1, 748) = 153.93$, $p < .001$. No other significant effects or interactions appeared, indicating individual differences had minimal impact on raw reading time functions. Predicted reading time per word on a page RT' follows the function: RT' (milliseconds per word) = 536 -10 * serial position (SP) of screen.

There did appear to be a dip in early serial positions and then a leveling off. Therefore we fit a quadratic equation to the reading time data. When averaging over the reader groups, the resulting

predictive equation was RT' = 409 + -23* SP + 88*SP^2. The improvement in the quadratic equation over the linear function was small when fitting curves to mean data points, R^2 = 0.97 versus 0.88, respectively. Moreover, the only coefficient that showed any differences in the Genre x Reading Time X Comprehension ANOVA was the intercept, which was lower for faster readers, F (1, 748) = 79.95, p < .001 In summary, the raw reading times showed decreases over serial position and a slight quadratic trend, but did not unveil differences in genre or individual differences.

3.2 Formality as a Function of Text Formality and Genre

It is possible that the above trends in decreasing reading times over serial position could be explained by characteristics of the text, as opposed to the readers' strategies (implicit or explicit) in allocation of reading time. We conducted an analysis of formality scores as a function of serial position, segregating the three text genres. These formality scores are plotted in Figure 2 for serial positions 1-14. The slopes for each genre were essentially flat as a function of serial position, with mean slopes of 0.00, 0.07, and 0.11 for informational, persuasive, and narrative texts, respectively. Therefore, decreasing trends in reading times cannot be attributed to systematic changes in text characteristics over serial positions.

In contrast, formality scores differed by genre, as consistent in previous studies [10]. The mean formality scores were 0.18, 0.09, and -0.26 for informational, persuasive, and narrative texts, respectively. These differences were significantly different, p < .001, showing the predicted ordering of informational > persuasion > narrative. Therefore, text characteristics varied over genre but not serial position.

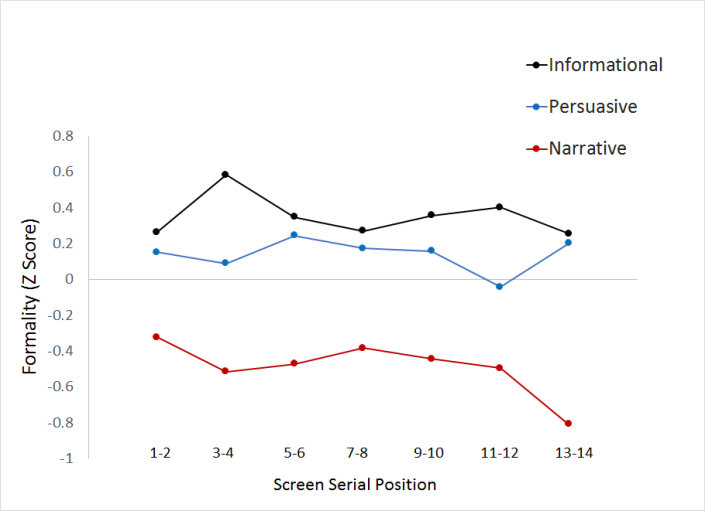


Figure 2. Formality as a Function of Screen Position, Segregated by Genre

3.3 Decoupling as a Function of Genre, Serial Position, and Reader Characteristics

It is possible that decoupling, rather than raw reading times, provides a more sensitive approach to analyzing disengagement. Figure 3 shows the decoupling scores for informational (3a), persuasive (3b), and narrative texts (3c). The participants are segregated into slow versus fast readers and high versus low comprehenders. As in the raw reading times, there did appear to be a dip in early serial positions and then a leveling off with a slow descent. The only exception was a slight upward trend for the narrative texts at the very end. When we fit a linear function to all of the participants for all of the texts, the best fit regression line yielded an R^2 =.63. A quadratic equation had a significant increase in variance explained of R^2 =.88. The best fit function was Decoupling' = 0.835 -0.204*SP + 0.010*SP^2. When we conducted a Genre x Reading Time x Comprehension ANOVA, there was only one significant effect. There was a significant effect of genre for the three coefficients in the quadratic function: F (2, 748) = 36.37; F (2, 748) = 8.46, p < .001, F (2, 748) = 11.00, all p < .001. There were no significant individual differences (reading speed or comprehension) and no interactions.

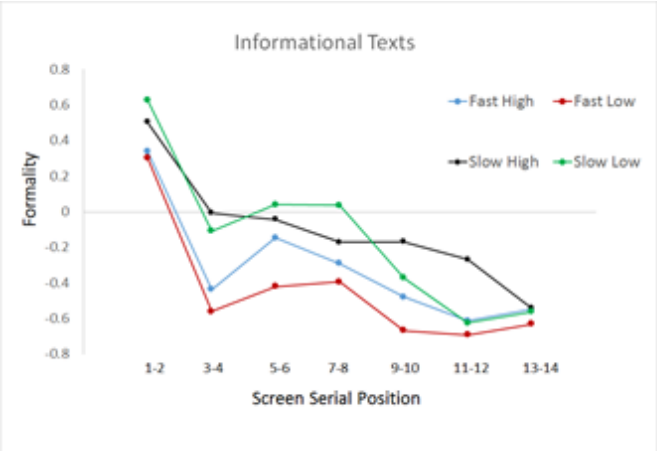
4. DISCUSSION

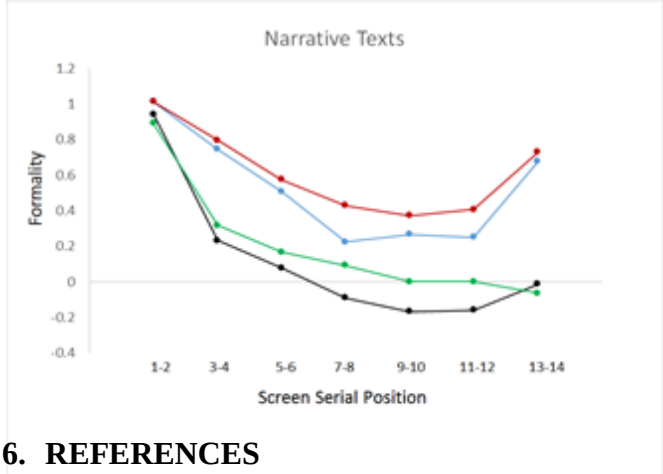
This study has revealed how reading times and cognitive decoupling are significantly influenced by text characteristics, namely genre and the serial position of information in the text. The pattern of results showed higher engagement (reflected in decoupling scores) in the first few screens of text and a subsequent decrease over the serial position of the screens. The deepest engagement is in the first 200-400 words, then noticeably decreases and slowly decreases thereafter (aside from an interesting upsweep for narrative texts). The quadratic function captures this trend and shows a better fit than a linear trend. It is of course strategically wise to pay attention to the early text segments because that is a critical point when the situation model is set up [11, 14], and the reader can make judgments whether the text is interesting or important to continue reading [1]. It is important to acknowledge that text difficulty is not comparatively high in early text segments, as shown in Figure 2, so increased time allocation at the beginning of a text cannot be attributed to text difficulty.

Regarding decoupling scores, text formality and difficulty show the following trend compatible with previous research using Coh-Metrix [2, 10]: informational > persuasive > narrative. However, cognitive decoupling showed the opposite ordering, such that readers tended to over allocate reading times to narrative text and under-allocate for the difficult informational text. In essence, there was a tendency to have lower engagement when the text was more difficult. The role of text difficulty has also been found to predict mind-wandering during text comprehension [13, 15] and listening to lectures [16], but the jury is still out as to (a) whether mind wandering is more prevalent in discourse that is very easy or very difficult and (b) what level of discourse analysis is most diagnostic of mind-wandering. Future research awaits an analysis of the impact on decoupling as computed via a deviation between reading time and formality and mind wandering.

5. ACKNOWLEDGMENTS

The research on was supported by the National Science Foundation (1108845) and the Institute of Education Sciences (R305C120001), Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF or IES.





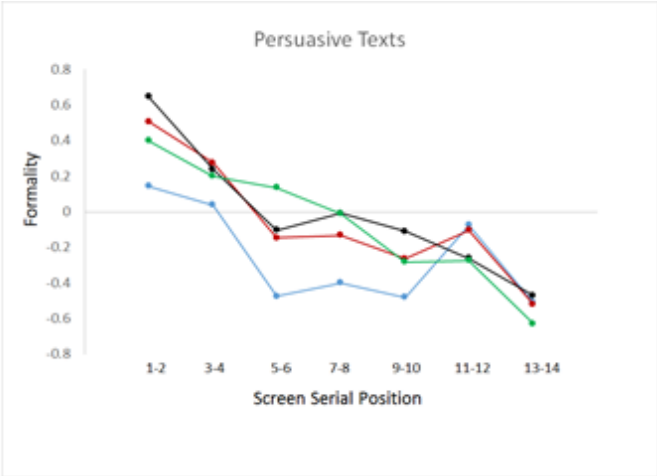


Figure 3. Decoupling by Formality as a Function of Screen Position, Segregated by Reader Type

6. REFERENCES

[1] Guthrie, J.T., S.L. Klauda, and A.N. Ho. 2013. Modeling the relationships among reading instruction, motivation, engagement, and achievement for adolescents. *Reading Research Quarterly*. 48, 1, 9-26.

[2] Franklin, M.S., J. Smallwood, and J.W. Schooler. 2011. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin & Review*. 18, 5, 992-997.

[3] Nguyen, K.-V., et al. 2014. Gotcha! Catching Kids during Mindless Reading. *Scientific Studies of Reading*. 18, 4, 274-290.

[4] Ainley, M., S. Hidi, and D. Berndorff. 2002. Interest, learning, and the psychological processes that mediate their relationship. *Journal of Educational Psychology*. 94, 3, 545.

[5] Fulmer, S.M., et al. 2015. Interest-based text preference moderates the effect of text difficulty on engagement and learning. *Contemporary Educational Psychology*. 41, 98-110.

[6] Nelson, J., et al. 2012. Measures of text difficulty: Testing their predictive value for grade levels and student performance. *Council of Chief State School Officers*, Washington, DC.

[7] Klare, G.R. 1974. Assessing readability. *Reading Research Quarterly*. 62-102.

[8] McNamara, D.S., et al. 2014. Automated evaluation of text and discourse with Coh-Metrix. *Cambridge University Press*.

[9] Graesser, A.C., D.S. McNamara, and J.M. Kulikowich. 2011. Coh-metrix providing multilevel analyses of text characteristics. *Educational Researcher*. 40, 5, 223-234.

[10] Graesser, A.C., et al. 2014. Coh-Metrix measures text characteristics at multiple levels of language and discourse. *The Elementary School Journal*. 115, 2, 210-229.

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

[12] McNamara, D.S., Graesser, A. C., and Louwerse, M. M. 2013. Sources of text difficulty: Across the ages and genres. *Assessing Reading in the 21st Century: Aligning and Applying Advances in the Reading and Measurement Sciences*, J.P.S.E.A., Editor. R&L Education: Lanham, MD.

[13] Feng, S., S. D'Mello, and A.C. Graesser. 2013. Mind wandering while reading easy and difficult texts. *Psychonomic Bulletin and Review*. 20, 3, 586-592.

[14] Kintsch, W. 1998. *Comprehension: A paradigm for cognition*. Cambridge university press.

[15] Mills, C., et al. 2014. To quit or not to quit: Predicting future behavioral disengagement from reading patterns. *Intelligent Tutoring Systems*. Springer.

[16] Medimorecc, M.A., Pavlik, P., Olney, A., Graesser, A.C., and Risko, E.F. The language of instruction: Compensating for challenge in lectures. *Journal of Educational Psychology*, in press

Semantic Similarity Graphs of Mathematics Word Problems: Can Terminology Detection Help?

Rogers Jeffrey Leo John
Center for Computational Learning Systems
Columbia University
New York, NY, USA
rl2689@columbia.edu

Rebecca J. Passonneau
Center for Computational Learning Systems
Columbia University
New York, NY, USA
becky@ccls.columbia.edu

Thomas S. McTavish
Center for Digital Data, Analytics & Adaptive Learning
Pearson
Austin, TX, USA
tom.mctavish@pearson.com

ABSTRACT

Curricula often lack metadata to characterize the relatedness of concepts. To investigate automatic methods for generating relatedness metadata for a mathematics curriculum, we first address the task of identifying which terms in the vocabulary from mathematics word problems are associated with the curriculum. High chance-adjusted interannotator agreement on manual identification of math terms was achieved by considering terms in their contexts. These terms represent 13% of the vocabulary in one seventh grade mathematics text. Six classification algorithms were compared to classify math terms for this text. To avoid overfitting to this curriculum, we relied on a small number of features that exploit external knowledge sources.

1. INTRODUCTION

Curricula often lack metadata to characterize the relatedness of concepts. Our ultimate goal is to develop methods for automatic generation of knowledge graphs for mathematics from existing curricula. Towards that end, we develop a representation for math word problems that allows us to measure similarities between problems, based on the math terminology they share [14]. In this paper, we present our methods to automatically identify the math terms. While mathematics is a highly structured domain with many sources that define terms, we found no single source that captured the mathematics terms as used in the context of this curriculum. Furthermore, several terms that occur in the word problems, such as *independent*, *chances*, and *set*, are polysemous, but occur more frequently in a "mathematical" sense. We therefore annotated the full vocabulary as "math" or "non-math" based on the predominant usage in the curriculum, and found high agreement among annotators. We then tested six methods for automatic classification.

The vocabulary items to be classified were represented using a small number of features based on glossaries, web search, and corpus statistics. Only 13% of the terms in our vocabulary were labeled as "math." Such data skew is challenging for many machine learning methods. To address the class imbalance, we used ensembles of weak learners and support vector machines (SVMs), weighting errors on the "math" class more heavily. We found that SVMs were our best classifiers. The automated methods presented here can enhance existing math curricula with domain knowledge graphs of content similarity among word problems.

2. RELATED WORK

Adaptive learning environments (ALEs) have shown promising results for mathematics and other STEM subjects [18, 5, 1], even when compared with human tutors [24]. For ALE's, the domain model is typically created anew but automated methods have been applied [3] [25]. The latter build concept maps from handbooks about SCORM standards, based on hand-constructed patterns to match dependency parses, then use the concept maps to build ontologies. Our work also derives semantic knowledge from text, aimed at representing semantic relations among mathematics word problems.

Automated methods have also been used in construction of educational domain models for assessments [20], standards [9], and targeted prerequisites for learners [13]. Various approaches have been used to represent domain knowledge, including semantic networks with frames and production rules [23], or model-tracing architectures to identify problem-solving steps students take, including incorrect ones [2]. Model-tracing, inherently reactive, has been extended with tutorial actions to pro-actively guide students [12]. Other approaches to automatically generate metadata require existing domain ontologies [22]. Our goal is to develop a network of relations among problems that could be used proactively by ALEs or teachers to move students through the curriculum in a way that promotes optimal learning.

To represent mathematics word problems, we create a bag-of-words (BOW) vector for math words using methods similar to terminology identification [10]. In separate work, we use this vector to create similarity networks among problems [14]. A range of methods have been used to identify terms in product reviews [6], concepts in semi-structured data [4], technical language in patents [15], or domain-specific terminology in general [21]. Much of this work deals with identi-

Chap.	Sec.	Exer.	Text
2	1	19	The table shows a **proportional** relationship between x and y. Complete the table.
9	1	11	Solve the **inequality** x + 1 < 4. Then graph the solutions.

Figure 1: Sample word problems.

fication of multi-word noun-noun compounds of a technical nature, and ranking them. In contrast, the secondary school math terminology has few compounds, includes a mix of different parts of speech, and is non-technical. As in [21, 6, 4], we rely on relative frequency ratio [8] to distinguish the frequencies of words in our corpus from their frequencies in a large background corpus. Unlike most of this work, apart from [15], we developed annotation guidelines and measured interannotator agreement. We find an agreement of 0.81 among three annotators using Krippendorff's α (see below), compared to 0.76 (Fleiss's κ; a similar metric) in [15].

3. DATA: MATHEMATICS EXERCISES

The data consists of 3000 word problems from a Grade 7 mathematics curriculum. The problems, which can incorporate images, tables, and graphs, are instantiated through templates. Figure 1 shows two problem exercises from chapters 2 and 9, with words that evoke math concepts in boldface. Note that a template, $x\{+|-\}X\{< \mid >\}Y$, randomly generates instances such as $x+4>9$ or $x+1<4$. Depending on the number of instance variables and constraints, a template may generate a bounded or nearly limitless number of instances. In addition to the exercise itself, which may contain a few steps that are typically solved via multiple choice or fill-in-the-blank, learners are able to select a more detailed guided solution, or to view the steps to solve a sample problem instance. We created an XML parser to extract the text from the exercises, the guided solutions, and sample problems. The vocabulary analysis is based on the extracted text.

4. ANNOTATION AND RELIABILITY

At 4,495 words (not lemmatized), the curriculum's vocabulary is relatively small. Removal of typical stopwords leaves 4,283 words. An additional 103 words, while not typical stop words, have very high frequency across problems (e.g., *amount, answer, compare*) and are not likely to be useful for measuring semantic similarity among problems.

The terms we are interested in are those that are characteristic of the concepts the students should know to demonstrate mastery of the curriculum. The three co-authors, working independently, each labeled an initial sample of 100 words as math, non-math and other, based on initial guidelines. Because pairwise agreement can be high when a chance-adjusted agreement coefficient is low (the so-called paradox of kappa [11]), agreement was measured using both pairwise agreement and Krippendorff's Alpha [16], a metric that factors out chance agreement. Initially, pairwise agreement was 0.93, but Alpha was 0.54, which is rather low. The low chance-adjusted agreement was mainly due to inconsistency among annotators in looking at the contexts in which words were used, and also due to borderline cases. We wrote more explicit guidelines with examples (4 pages), then labeled two additional samples of 100 words each, computing agreement on each sample before proceeding to the next. On the second and third samples, pairwise agreement was 0.92 in both cases, and Alpha was 0.83 and 0.81. Given the high agreement and consistency across the second and third samples, we determined the labeling to be reliable. One of the co-authors labeled the remainder of the vocabulary, yielding 3832 words labeled as non-math, 571 as *math* and 92 as *other*. Only the words labeled as *math* and *non-math* were used to train the classifier.

1. Wolfram Mathworld
2. About.com: mathematics
3. Math domains in Google search results
4. Math domains in Bing search results
5. Digits math glossary
6. Relative frequency ratio

Figure 2: Features to represent vocabulary

5. CLASSIFICATION EXPERIMENTS

This section reports results from a suite of classification algorithms applied to the labeled data. To represent the vocabulary for the learner, we engineered features based on search and glossary information, and on a corpus-based metric. Two challenges for the classification were infrequency of the positive class (high data skew), and apparent non-linearity of the class separation. Of six learning algorithms, those that had best performance were most suited to these learning challenges, as described further below.

5.1 Feature Representation

We constructed a feature vector representation for the words with the 6 features listed in Figure 2. All feature values were scaled to be in the range of 0 to 1.

For the first two features listed in Figure 2, we used the functionality of Google Custom Search that permits customized searches to user-specified domains. For the first feature we queried mathworld.wolfram.com, and for the second we queried math.about.com. The value for each of these features consists of the total number of query returns, which can be arbitrarily large.

Google Custom Search can also be configured so that for the top ten returns to a query, each return consists of a triple with the url, a list of text snippets containing the term at that url, and the page title at that url. For the third feature listed in Figure 2, we query the web using this functionality, and calculate the feature value based on the triples for the top ten returns. Each time *math*, *mathematics*, or *arithmetic* occurs at least once in each element of a triple, a counter is incremented. The maximum value is thus 30.

Bing is a Microsoft search engine with an interface through which queries can be made programmatically. The interface returns the top 50 search results. Like Google searches, each result contains the relevant URL, snippets, and title of the page. As in the Google search feature, for the fourth feature in Figure 2, a counter was incremented whenever *math*,

Table 1: Classification Results

Classifier	Precision	Recall	Fscore	Sensitivity	Specificity	G-Mean
adaboost	0.89	0.90	0.89	0.42	0.97	0.64
bagging	**0.90**	**0.91**	**0.90**	0.41	**0.98**	0.63
rand-forest	**0.90**	**0.91**	**0.90**	0.45	0.97	0.66
SVM-poly	0.89	0.86	0.87	**0.68**	0.89	**0.78**
SVM-RBF	0.89	0.87	0.88	**0.68**	0.90	**0.79**
logistic regression	0.89	0.90	0.88	0.31	**0.98**	0.56

mathematics, or *arithmetic* occurred at least once in a triple element. Values are in [0,150].

The mathematics curriculum has an associated glossary of 246 math terms. It includes simple terms, e.g., "sphere," and compound terms, e.g., "associative property of multiplication." The glossary was expanded with the individual words in compound terms, excluding stop words. Thus for the compound term "associative property of multiplication", the words *associative*, *property* and *multiplication* were added. In this way, the glossary was expanded to 516 terms. A boolean feature value was used here to indicate exact occurrence of a word in the glossary.

Relative frequency ratio (RFR) measures relative frequency of a term in reference to a contrastive background corpus [6, 8]. The frequency of a word w_i in a corpus C, expressed as $FR(w_i, C)$, is its count normalized by size of the corpus. For a domain specific corpus, e.g., a mathematics text, the frequency of domain-specific terms should be higher than in a large, background corpus. The formula for RFR is:

$$RFR(w_i) = \frac{FR(w_i, DC)}{FR(w_i, BC)} \quad (1)$$

where DC is the domain corpus and BC is the background corpus. We tested RFR with two background corpora: the Open American National Corpus (OANC: N=22 $\times 10^6$) and English Gigaword, Fifth Edition (N=4,033 $\times 10^6$). Unsurprisingly, we found that the size of the background corpus is critical to the precision of the RFR measures. When we ranked Digits words by RFR scores using Gigaword, 306 of the words labeled as "math" occur in the top 1,000 words compared with 248 using OANC. Therefore we used Gigaword as the background corpus.

5.2 Classification

The labeled data was randomly split into a training set with 75% of the vocabulary (3301 terms) and a test set with 25% of the vocabulary (1101 terms). Using logistic regression, classification results yielded an overall precision of 0.87 and a recall of 0.88, compared with 0.78 precision and 0.25 recall for the *math* class. The low recall of math terms can be attributed to high class imbalance, where only 13% of terms are in the *math* class. Linear SVM also yielded poor results, suggesting that the classes cannot be linearly separated. To address the class imbalance, we use class weights for SVM, where we use polynomial and RBF kernels to address the non-linearity. Ensembles of weak learners also help with non-linearity. For each of three ensemble methods, *Boosting*, *Bagging* and *Random Forests*, we used 1000 *Decision Trees*.

Evaluation results are reported using precision, recall, f-measure, and g-mean [17]. The latter, the geometric mean of accuracy on the positive class (recall, or sensitivity) and accuracy on the negative class (specificity), is high when both accuracies are high and their difference is small. It is particularly useful when there are no criteria for constructing a cost matrix for errors in sensitivity versus specificity.

For the SVM classifiers, we used C=10,000. For the polynomial kernel, the degree was 4 and the class weights assigned to the math and non-math classes were 270 and 1350 respectively. For the SVM with the RBF Kernel, class weights were set to 200 and 1100.

6. RESULTS AND DISCUSSION

Table 1 shows the results for the six classification experiments. All the classifiers had high accuracy, due to the high class imbalance favoring non-math words. Accuracy on the math words (sensitivity), however, was relatively low for all but the SVM learners. The ensemble methods had higher precision on the math words ($\geq$ 0.78) but low sensitivity (0.41-0.46). The SVM learners had lower precision (about 0.5) and higher sensitivity (0.68). The logistic regression had very high precision on the math words (0.81) but very low sensitivity (0.32). For g-mean, all the classifiers had values above 0.50, indicating respectable peformance. The two SVM learners, however, had the highest g-means: 0.78 (polynomial kernel) and 0.79 (RBF kernel).

Manual error analysis of math words that were incorrectly classified by multiple learners indicated that many of the errors were due to polysemous words that have one or more non-math senses that occur with non-neglible frequency. This includes words like *point*, *dependent*, and *trial*. In WordNet [19], for example, *point* used as a noun has twenty-five senses, and fourteen senses used as a verb. Future work on the classification task will include investigation of features commonly used for coarse-grained word sense disambiguation, where accuracies of 88% have been achieved using lexical, syntactic and topical features [7] so that we can apply the same methods to new curricula.

7. CONCLUSIONS

The vocabulary classification task we address, to identify vocabulary that characterizes the semantics of a curriculum, differs from standard terminology detection, where the focus is on highly technical compound terms. It also differs from word sense disambiguation in that we are interested in binary classification of senses, based on the use of terms for a given curriculum. We have shown that human annotators can achieve very high pairwise and chance-adusted agreement. To avoid overfitting to a given curriculm, the features we used draw on external knowledge sources such as glossaries, web search and large background corpora. With relatively few such features and choice of an appropriate learning

algorithm, we achieve very high accuracy and good sensitivity, despite the small proportion of the positive class.

8. ACKNOWLEDGMENTS

We would like to thank the Research and Innovation Network at Pearson for support of this work.

9. REFERENCES

[1] V. Aleven, A. Ogan, O. Popescu, C. Torrey, and K. R. Koedinger. Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In J. Lester, R. M. Vicari, and F. Paraguaca, editors, *Intelligent Tutoring Systems: Seventh International Conference, ITS 2004*, pages 443–454. Springer, 2004.

[2] J. R. Anderson and R. Pelletier. A developmental system for model-tracing tutors. In L. Birnbaum, editor, *The International Conference on the Learning Sciences*, pages 1–8. Association for the Advancement of Computing in Education, Charlottesville, VA, 1991.

[3] L. Aroyo, P. Dolog, G.-J. Houben, M. Kravcik, A. Naeve, M. Nilsson, F. Wild, and others. Interoperability in personalized adaptive learning. *Educational Technology & Society*, 9(2):4–18, 2006.

[4] T. Atapattu, K. Falkner, and N. Falkner. Automated extraction of semantic concepts from semi-structured data: Supporting computer-based education through the analysis of lecture notes. In S. W. Liddle, K.-D. Schewe, A. M. Tjoa, and X. Zhou, editors, *Database and Expert Systems Applications*, number 7446 in Lecture Notes in Computer Science, pages 161–175. Springer Berlin Heidelberg, 2012.

[5] C. R. Beal, I. M. Arroyo, P. R. Cohen, and B. P. Woolf. Evaluation of AnimalWatch: An intelligent tutoring system for arithmetic and fractions. *Journal of Interactive Online Learning*, 9(1):64–77, 2010.

[6] J. Broß and H. Ehrig. Terminology extraction approaches for product aspect detection in customer reviews. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 222–230, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[7] J. F. Cai, W. S. Lee, and Y. W. Teh. NUS-ML: Improving word sense disambiguation using topic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 07)*, pages 249–252, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[8] F. J. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447, 1993.

[9] H. Devaul, A. R. Diekema, and J. Ostwald. Computer-assisted assignment of educational standards using natural language processing. *Journal of the American Society for Information Science and Technology*, 62(2):395–405, 2011.

[10] P. Drouin, N. Grabar, T. Hamon, and K. Kageura, editors. *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, August 2014.

[11] A. R. Feinstein and D. V. Cicchetti. High agreement but low Kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.

[12] N. T. Heffernan, K. R. Koedinger, and L. Razzaq. Expanding the model-tracing architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education*, 18:153–178, 2008.

[13] S. Jain and J. Pareek. Automatic extraction of prerequisites and learning outcome from learning material. *International Journal of Metadata, Semantics and Ontologies*, 8(2):145–154, Jan. 2013.

[14] R. J. L. John, T. S. McTavish, and R. J. Passonneau. Semantic graphs for mathematics word problems based on mathematics terminology. In *WS-1: Graph-based Educational Data Mining (G-EDM 2015)*, 2015.

[15] A. Judea, H. Schütze, and S. Bruegmann. Unsupervised training set generation for automatic acquisition of technical terminology in patents. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 290–300, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[16] K. Krippendorff. *Content analysis*. Sage Publications, Beverly Hills, CA, 1980.

[17] M. Kubat, Robert, and S. Matwin. When negative examples abound. In *Proceedings of the 9th European Conference on Machine Learning*, ECML '97, pages 146–153, London, UK, UK, 1997. Springer-Verlag.

[18] E. Melis, E. Andres, J. Budenbender, A. Frischauf, G. Goduadze, P. Libbrecht, M. Pollet, and C. Ullrich. Activemath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12:385–407, 2001.

[19] G. A. Miller. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[20] R. J. Mislevy, J. T. Behrens, K. E. Dicerbo, and R. Levy. Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *JEDM - Journal of Educational Data Mining*, 4(1):11–48, Oct. 2012.

[21] Y. Park, R. J. Byrd, and B. K. Boguraev. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[22] D. Roy, S. Sarkar, and S. Ghose. Automatic extraction of pedagogic metadata from learning content. *Int. J. Artif. Intell. Ed.*, 18(2):97–118, Apr. 2008.

[23] S. Stankov, M. Rosić, Žitko, and A. Grubišić. TEx-Sys model for building intelligent tutoring systems. *Computers and Education*, 51:1017–1036, 2008.

[24] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[25] A. Zouaq and R. Nkambou. Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies*, 1(1):49–62, Jan. 2008.

An analysis of peer-submitted and peer-reviewed answer rationales, in an asynchronous Peer Instruction based learning environment

Sameer Bhatnagar
Polytechnique Montreal

Michel Desmarais
Polytechnique Montreal

Chris Whittaker
Dawson College

Nathaniel Lasry
John Abbott College

Michael Dugdale
John Abbott College

Elizabeth S. Charles
Dawson College

ABSTRACT

This paper reports on an analyis of data from a novel *Peer Instruction* application, named DALITE. The Peer Instruction paradigm is well suited to take advantage of peer-input in web-based learning environments. DALITE implements an asynchronous instantiation of peer instruction: after submitting their answer to a multiple-choice question, students are asked to write a rationale for their choice. Then, they can compare their answer to other students' answers, and are asked to choose the best peer-submitted rationale among those displayed. We engaged in an analysis of student behaviour and learning outcomes in the DALITE learning environment. Specifically, we focus our investigation on the relationship between student proficiency, how students change their answers after reading each others' writings, and the peer-votes they earn in DALITE. Key results include i) peer-votes earned is a significant predictors of success in the course; ii) there are no significant differences between strong and weak students in how often they switch from the correct answer to a wrong answer after consulting peer-rationales, or vice versa; iii) even though males outscore females in conceptual physics questions, females earn as many votes from their peers as males do for the content they produce when justifying their answer choices.

Keywords

peer instruction, exploratory data analysis

1. INTRODUCTION

Active learning encompasses a broad movement in modern pedagogical practices, including any activities which engage the student as a part of the learning process, instead of passively receiving information during a traditional lecture. Such activities should encourage the student to read, write and discuss classroom content, as well as engage in higher order thinking tasks, such as synthesis and evaluation [1]. Active, cooperative, and collaborative learning practices have been shown to yield greater learning gains in science in engineering [8]. With the growing presence of on-line learning through instructional videos and accompanying readings, there is place for web-based activities which promote the same higher-order learning processes as those being used in more active classrooms.

This is where our research group found the need to develop the Distributed Active Learning Technology Integrated Environment (DALITE). The teacher-researchers in our group wanted a web-based homework system which would go beyond simply asking students for the answers to conceptual questions, by asking them to express the reasoning behind their thinking. This learning environment was meant to capture some of the higher-order thinking processes students engage in when reasoning about new concepts. DALITE is a system that would provide data on the mechanism of conceptual change, through the writings of students, as well as their evaluation of each other's work. What has emerged is an open source system which is being used in classrooms by learning science researchers who are also teachers.

Thus far, it has produced a dataset which can reveal new insights from the data on student production and consultation of answer rationales. Previous analysis of our work has already shown that students who use DALITE in college level physics classrooms do as well as those who use other on-line homework environments [2]. In the current study we analyze how the data on the production of rationales and the voting patterns can yield novel indicators of success and other characteristics of students.

This paper will begin with a description of the related field of Peer Instruction. The DALITE platform will then be described, as well as the most recent dataset collected. The focus of the analysis and results will be on the relationship between student proficiency, how students change their answers after reading each others' writings, and how many votes they earn for what they write. Finally we will discuss the potential and challenges that lie ahead, especially as student models are integrated into the DALITE system.

2. RELATED WORK

2.1 Peer Instruction

Peer instruction is a classroom practice popularized by Eric Mazur of Harvard University [3]. In its most common instantiation, the classroom script goes as follows:

1. The teacher displays a multiple choice question to the whole class, and asks everyone to reflect, and individually choose what they think is the correct answer. This is typically done by giving each student a handheld clicker, which transmits the answer to a receiver plugged into the teacher's computer.
2. The teacher displays a bar chart showing the distributions of answer choices for the whole class. The students are then prompted to discuss their answer choice with their peers for several minutes, after which they are given the opportunity to answer the question again using their clicker.
3. The teacher shows the new distribution of answers. Typically, after the peer discussion, there is a major shift towards the correct answer.

Making this a regular practice in class has been shown to yield higher learning gains [7] and lower dropout rates [4] compared to conventional, teacher-centered, lecture style courses. However it is very difficult to capture what is actually happening during the student discussions. What is actually being said to convince someone to change their answer (or at least change their rationale for their answer choice)? How does that relate to cognitive theories of learning? DALITE collects information exchanged in written form through Peer Instruction features embedded within a web based learning environment, namely answer rationales and votes. The information hereby collected allows us to better address the above questions empirically.

3. THE DALITE PLATFORM

DALITE is a web-based drill and practice platform that contains introductory level physics problems. It has an interface for the student to work on physics problems, and a teacher interface to manage the learning content.

3.1 Student interface

Students log into DALITE, and work on an assignment which typically contains four to six multiple choice questions. For each question, there are three screens they must flip through, each with the following structure:

1. The question is displayed, and the student selects one of the multiple choice answers. They are then prompted to write a couple of sentences that explain why they selected their answer choice. These little paragraphs will from now on be referred to as "rationales".
2. Once a rationale is given, the system presents two columns: one for their answer choice, and one for another choice to the question. Each column contains four rationales, written by previous students. The aim is to give students a chance to reflect on their thinking by providing them with an opportunity to compare and contrast other rationales and change their mind. The student is prompted to read the rationales from the two columns, and decide whether they would like to keep their choice, or switch. What's more, the student is asked to choose one rationale out of the ones displayed that they best like. They can also simply cast an "empty ballot", in effect saying that none of the other students' rationales were convincing. This up-voting process is anonymous.
3. The third screen recaps everything that just happened: the question is shown, alongside their two answer choices (one from each of the previous two screens). What's more, the rationale they originally wrote is reflected back to them, right next to a rationale written by an expert for the correct answer.

3.2 Teacher Interface

When teachers login to the system, they can:

- upload new questions to the database. This requires that the question be of multiple choice format. The teacher must specify the correct answer, with a rationale justifying that answer choice. The teacher must also identify a "second best answer", which would be used for the second column of the second screen (described above) should the student answer correctly on their first attempt. Teachers can also add "tags" to the question, which describe the content of the question.
- build new assignments based on questions already in the system.
- observe the results of assignments done by their students. The current reporting tool gives the teacher a mini grade-book for each assignment, where each student is a row, and each question is described by two columns: one for the student's first answer, and one for their second answer. Teachers can quickly get a sense of where the students are getting confused, as cells are coded green for the correct answer, and red for the incorrect answer. Transitions from red to green are signs that the rationales in the database are doing their job of convincing students to move away from the wrong answer, while transitions from green to red show that the students' conceptual understanding is shallow.

4. THE DATASET

Although DALITE has been in use for the last five years, it was during the Fall semester of 2013 that a comprehensive dataset was collected in a systematic manner over the entire term. The cohort was comprised of 144 students, spread out in five groups, taught by four different teachers, across three colleges. The system was used to teach freshman year, calculus -based Newtonian Mechanics. This is at a level equivalent to grade 12 in high school in the US and other Canadian provinces.

4.1 Data from within DALITE

Over the course of the semester, 80 question items were assigned by the different teachers, 40 of which were completed by at least half of the entire cohort, providing data on over 7000 student-item pairs.

Each student-item pair in the dataset includes the initial answer, the rationale, and the final answer. A separate table

in the database keeps a count of how many peer-votes are earned by any given rationale.

4.2 Data from classrooms

For each student in the five experimental groups, as well as one control group (which did not use DALITE), the following data was collected inside their classrooms over the course of the semester:

Pre-Post FCI The Force Concept Inventory (FCI)[5], is a questionnaire of 30 conceptual questions about the Newtonian concept of force. The exact same questionnaire was administered on the first day of class, and then again on the last day of class, for each of the groups, in order to compare the learning gain between the DALITE users and students who did not use DALITE. The item-by-item results of this questionnaire can be compared to a FCI dataset which holds the results of more than 13000 students from across Canada and the U.S.

Midterm & Final Exam Grades The Newtonian Mechanics course commonly has three major themes: Kinematics, Dynamics, and Laws of Conservation. This lines up with the three midterms for which each student's grade is recorded. Finally, for each student, the final exam grade is broken down by the result on the multiple choice section (typically more conceptual questions, and hence more similar to DALITE), and the long-answer section (typically computations and problem-solving).

5. RESULTS

During the Fall 2013 study, four experimental groups were assigned DALITE specifically as homework for their students. Following are the key results:

Student Success How well students succeeded on DALITE questions had 0.50 and 0.60 correlations with their performance on the conceptual, multiple choice part of their final exam, and the post-semester FCI questionnaire, respectively. This provides some measure of the reliability of this relatively new homework system.

Also a linear model was fit to predict a student's final grade based on statistics from their DALITE account. The fraction of questions students answered correctly out of those they attempted, as well as the total number of votes they accumulated, were both significant predictors of their final grade in the course ($R^2 = 0.24$, $p<0.001$). This predictive power of DALITE emerges as early as after the first third of the course, meaning the teacher can get early indicators of which students are at risk for the midterm.

In a related line of questioning, the data was partitioned by gender of the students. Male students did significantly better than female counterparts in all measures of conceptual understanding from the classroom (pre-term FCI score, pre-post term gain on FCI, conceptual questions on final exam). This is in line with previous work looking into the gender gap in introductory physics [6]. This gap was found in the DALITE data as well, with males getting 20% more of the questions items right ($p<0.001$).

Patterns in how students change their answer choices Over the course of the semester, students who started with the right answer, only switched to the wrong one 1 out of 10 times. However, when they started with the wrong answer, they switched to the correct answer 3 out of 10 times after reading their peers' rationales. This gives some measure of overall quality of the rationales currently in the database: the rationales to the wrong answers are not highly persuasive, and there are at least some rationales for the correct answers which can convince students to change their minds when they are wrong.

Factors affecting answer change When the data was separated into quartiles for the final course grade, it was found that strong students were as likely as weaker students to switch from the right answer to the wrong answer. In addition, the converse was also true: weaker students were as capable of switching to the right answer when they got it wrong on their first attempt. There was some effect herein due to the teacher: the experimental groups that regularly discussed DALITE homework in class, were significantly more likely to change their answer when in DALITE. In the group that used DALITE purely as extra homework, answer switches were much less likely ($p<0.001$).This may indicate that the students who are reminded that the system is a valuable tool, are more engaged with the system, and take the time to more carefully read each others' rationales.

The well known gender gap mentioned, males outscoring females in conceptual physics questions, interestingly disappears if we measure correctness based on the second attempt: female students choose the wrong answer 20% more often on their first attempt, but after reading peer-written rationales, they identify the correct choice just as often as males.

Who amasses more peer votes? Students from the stronger half of the cohort earned, on average, more than two times as many votes as those from the bottom half. What's surprising is that this pattern holds true for the wrong answers as well: even when the strong students are wrong, they are twice as convincing as their weaker peers. This is especially relevant in light of the fact that 1/3 of all the votes cast over the term were for rationales to wrong answer choices. In parallel to this finding, when we looked only at rationales justifying the correct answer choice, it was found that weak students earned as many votes as their stronger colleagues. This seems to indicate that even if a student did not perform as well on tests, when they were right on a particular conceptual question, they were able to justify their understanding as well as stronger students.

The gender gap discussed earlier, was also lost when looking specifically at the voting data. Even though males achieve higher grades on conceptual questions, females of all strengths earn as many votes for their rationales as the males. This tends to indicate that females produce content justifying their understanding

that is as valued by their peers as rationales written by males.

6. DISCUSSION

The key results described above show the potential for DALITE to be an effective tool for teachers to probe their students' deeper understanding of concepts in physics, and identify students at risk of failing midterms and final exams. The data on how students change their answers based on the writings of their peers, and which rationales they vote for, may give teachers and researchers insight on what words can trigger conceptual change in different types of students. Finally, the data shows that students who may not perform as well on summative evaluations, are still able to produce valuable content when justifying their understanding.

7. FUTURE WORK

Future directions of research on this project include capturing not just which rationales got voted for, but who is casting the votes, and in what context. The goal is to explore what features in student written text have an impact on changing peer conceptions of scientific concepts. Do students learn from stronger students, or only those within their Vygotskian zone of proximal development [10].

Another important direction would include collaborative filtering techniques, which are traditionally applied to recommender systems, such as in the e-commerce setting, where a users-by-item ratings matrix is used to predict what items new users would most likely enjoy. Recently such techniques have been applied in the context of educational data mining, where the matrix is now student-by-item performance, and factorization leads to estimates of the probability of another student getting a new item correct [9]. With the ratings data collected, the system may be able to deliver individualized rationales to different learners with the same misconceptions to the same question item. What is most promising is how this open-source tool creates a venue for learning science researchers to ask questions regarding higher-order learning processes, such as evaluation and synthesis, and for the EDM community to test-drive different text mining techniques in a real classroom setting.

8. ACKNOWLEDGMENTS

The strength of the DALITE platform resides in the database of student rationales, so the students who have used this platform for learning must be thanked for providing this rich set of data. This work has been funded through the Programme d'Aide à la Recherche sur l'Éducation et l'Apprentissage(PARÉA), administered by the Ministère d'Éducation et Loisirs de Quebec.

9. ADDITIONAL AUTHORS

Kevin Lenton, Vanier College

10. REFERENCES

[1] C. C. Bonwell and J. A. Eison. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991.

[2] E. Charles-Woods, C. Whittaker, M. Dugdale, N. Lasry, K. Lenton, and S. Bhatnagar. Beyond and within classroom walls: Designing principled pedagogical tools for students and faculty uptake. In *Computer Supported Collaborative Learning (in press)*, 2015.

[3] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977, 2001.

[4] A. P. Fagen, C. H. Crouch, and E. Mazur. Peer instruction: Results from a range of classrooms. *The Physics Teacher*, 40(4):206–209, 2002.

[5] D. Hestenes, M. Wells, and G. Swackhamer. Force concept inventory. *The physics teacher*, 30(3):141–158, 1992.

[6] L. E. Kost, S. J. Pollock, and N. D. Finkelstein. Characterizing the gender gap in introductory physics. *Physical Review Special Topics-Physics Education Research*, 5(1):010101, 2009.

[7] N. Lasry, E. Mazur, and J. Watkins. Peer instruction: From harvard to the two-year college. *American Journal of Physics*, 76(11):1066–1069, 2008.

[8] M. Prince. Does active learning work? a review of the research. *Journal of engineering education*, 93(3):223–231, 2004.

[9] N. Thai-Nghe, L. Drumond, T. Horváth, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme. Factorization techniques for predicting student performance. *Educational Recommender Systems and Technologies: Practices and Challenges*, pages 129–153, 2011.

[10] L. Vygotsky. Interaction between learning and development. *Readings on the development of children*, pages 34–41, 1978.

Learning Analytics Platform, towards an open scalable streaming solution for education

Nicholas Lewkow
McGraw-Hill Education
281 Summer St.
Boston, MA
nicholas.lewkow
@mheducation.com

Neil Zimmerman
McGraw-Hill Education
281 Summer St.
Boston, MA
neil.zimmerman
@mheducation.com

Mark Riedesel
McGraw-Hill Education
281 Summer St.
Boston, MA
mark.riedesel
@mheducation.com

Alfred Essa
McGraw-Hill Education
281 Summer St.
Boston, MA
alfred.essa
@mheducation.com

ABSTRACT

Next generation digital learning environments require delivering *just-in-time feedback* to learners and those who support them. Unlike traditional business intelligence environments, streaming data requires resilient infrastructure that can move data at scale from heterogeneous data sources, process the data quickly for use across several data pipelines, and serve the data to a variety of applications. As a solution to this problem, we have designed and deployed into production the Learning Analytics Platform (LAP), which can ingest data from different education systems using standardized IMS Caliper events. The education events are triggered by student and instructor activity within Caliper instrumented learning systems. Once sent to the LAP, events are transformed and stored in a data store where they can be used for student, educator, and administrator visualizations as well as education driven analytics research. Two McGraw-Hill Education platforms, Connect, used for higher education, and Engrade, for K-12, are currently instrumented to send the LAP event data which in turn feeds visualizations for educational insight. Future plans for the LAP include collection of education event data from a wide variety of proprietary and open source education platforms, computational engines for predictive analytics, and an open API for third-party analytics using LAP data.

Keywords

Learning Analytics, Event Processing, Heterogeneous Data, Streaming Data, Parallel Architecture

1. INTRODUCTION

It is the goal of next generation digital learning systems to use big data and analytics to advance learning outcomes. These next generation systems should be able to provide just-in-time feedback to students and educators with an aim to increase the efficiency and effectiveness of digital education. Further, with the increasing instrumentation of all digital media, digital learning environments should be instrumented in a way that allows important education data to be collected in a standardized fashion for both real-time and after-the-fact (batched) data analysis. This task requires large scale processing of streaming data utilizing massively parallel architectures which may ingest, store, and analyze data in real-time.

Our solution to this problem is the Learning Analytics Platform (LAP). The LAP is designed to ingest educational data from present and future education platforms in the form of standardized events using the IMS Caliper spec [2]. Two existing education platforms, Connect for higher education and Engrade for K-12, have already been instrumented to create and ship Caliper education events to the LAP. Once ingested by the LAP, the data is transformed and stored for building of visualizations used for educational reports [4]. These 'insights' include several real-time statistics for students and educators including time-spent, outcomes, submission times near due dates, attendance, and class performance comparisons. Additionally, messages indicating negative trends, such as repeatedly starting assignments near or after the due date, are also presented to the user.

To meet the requirement of providing just-in-time feedback to students and educators, an analytics platform must also be have a parallel architecture which may effectively ingest streaming data. There are several proposed requirements for a streaming data architecture, including the ability to handle data imperfections, generate predictable outcomes, and to guarantee data safety and availability [5]. Additionally, the architecture should be automatically scalable and fault tolerant for both software and hardware failures, particularly, it must not lose any event data under any circum-

stance. Our architecture has the additional requirements of ingesting data from heterogeneous sources and performing data transformations using different data pipelines. The LAP fulfills all of the above requirements in its current version with further refinements and additions planned for the near future.

Details about the LAP are discussed in the following sections including information about the standardized data format which was used, IMS Caliper events, as well as specifics on the LAP architecture and performance. Information regarding future versions of the LAP is also discussed followed by concluding remarks.

2. STANDARDIZED CALIPER EVENTS

With the continual adoption of digital education systems a global standard for educational event data, which is generated from a large diversity of heterogeneous systems, has become increasingly sought after. While there has been advancement in this area by the Tin Can API [1], the IMS Global Learning Consortium has proposed a schema-driven solution to this problem with their Caliper event spec [2]. JSON-LD (linked data) is used for the Caliper events as a way to link specific, normalized fields within a set of events [3]. Using the Caliper events, data from heterogeneous learning systems can be created, transmitted, and collected for analysis in a global and standardized fashion.

Caliper events strive to create a generalized framework that can be utilized by all types of learning events ranging from a student using an interactive education tool, such as a learning game, to an educator recording attendance in their class. The Caliper events are based on the data triple of **"Actor" - "Verb" - "Object"**. As an example, the event for a student submitting an assessment (homework, quiz, test, etc.) would have the **Actor** be the student, the **Verb** be the submission of an assessment, and the **Object** would contain information on the assessment being submitted, potentially how long it took to complete.

In order to utilize the Caliper event spec, learning platforms must be instrumented to create events when actions occur by either students or educators. Currently the Connect and Engrade systems are instrumented to to create Caliper events when actions occur, and to send the events to the LAP. Instrumentation is unique to the system in question and greatly depends on how that system's data is stored. In the case of Connect, Caliper events are created through a series of database triggers when actions are taken by students. The database triggers are automated to create the Caliper events using tables from their system databases when new information is passed to the system from a user. Future plans include instrumentation of several new external systems, allowing for increasingly rich data in the LAP for analysis and visualization.

3. ARCHITECTURE

Increasing instrumentation of sensors and digital media requires streaming analytics architectures to analyze data in real-time. Attention was paid to the development and design of the LAP architecture to ensure that it met all requirements of a parallel streaming system containing both a data store and an analytics engine. Key features include auto-scaling fundamental architecture components with varying load, fault tolerance for both hardware and software failures, and the ability to process data and have it available for output API access immediately.

In the simplest of descriptions, the LAP is designed to:

1. Receive learning events from external applications through an ingestion API.
2. Send raw events directly to long-term storage.
3. Validate each event for expected fields and types.
4. Process the events, which requires application dependent data transformations.
5. Store those events in the data store according to application dependent schema.
6. Query the data store and perform transformations and aggregations as needed for the output API.

Figure 1 displays a high level view of the LAP architecture with learning events from three separate external applications coming into the LAP. Once the events are received by the LAP they are transformed according to an application specific schema and stored in the data store. When a user requests an insight visualization, the LAP output API is called by the external application. This triggers the results and analytics service to query the data store, aggregate and transform the data as needed, and pass it to the insight service where the data is used to build the appropriate insight visualization which is then passed to the user.

The ingestion API and collection service are configured to receive IMS Caliper events sent from external applications through sensors which have to be implemented in those external applications. To maximize performance over high-overhead HTTP, several events are sent simultaneously in an event container. The number of events sent in a single container can range from one to tens of thousands. Once a container of events is received it is immediately sent to a long term storage system for backup purposes. The container is then opened and the events within are validated, transformed into an application specific schema, and sent to the data store.

The data store utilizes the non-relational database MongoDB, which allows for great flexibility in data model schema. Events from external applications are not guaranteed to come into the LAP in chronological order so the data model schemas were developed to create a deterministic data store state from events coming in arbitrary order. The data model employed for the current two applications consists of a two schema model; one document type holds the student level data and the other holds the class level data. Building of the insight visualizations then requires the results service querying N student documents for a class with N students, and 1 document per class for a total of $N + 1$ documents.

The insight service is built external to the LAP to provide the user with the requested analytical visualization. Once the data store is queried and the appropriated documents are

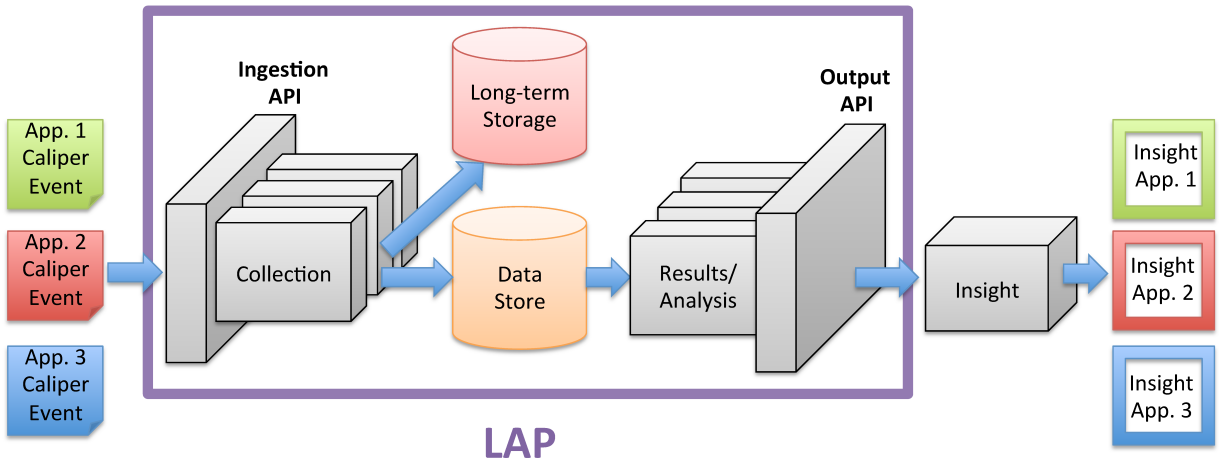


Figure 1: High level view of the LAP architecture. Learning events are passed into the LAP through the ingestion API and visualizations 'insights' are produced through the insight service calling the LAP's output API. Internally, the LAP consists of a collection service, a data store, long-term storage, and a results service, which also performs analytics. Several instances of the collection and results services run in parallel.

returned, they are aggregated, transformed, and returned to the output API to provide the client with data to build the visualization. This process is very lightweight and fast, allowing for quick feedback and response to the user who has just finished an assessment.

The technology used for the LAP was chosen to be lightweight, reliable, and adaptable for future system iterations. The collection, results, and insight services were all built using Node.js which allows for asynchronous communication run in parallel across several service instances. The analytical visualizations were built using JavaScript with AngularJS and D3.js frameworks which provide fast, responsive, and interactive interfaces that are customizable. The data store was built using MongoDB with a three member replica set, allowing for highly available data and low latency access for both read and write operations. Amazon Web Services (AWS) were used to host the LAP utilizing load balancing technologies routed to EC2 instances. An AWS S3 bucket was used for long-term storage of the raw event containers. Details pertaining to the performance of the LAP as well as future plans for the system follow.

4. PERFORMANCE

Performance is very important to the success of the LAP. It is imperative that the LAP be able to ingest data from several external systems during their peak times simultaneously in a manner which does not delay the real-time analysis on the output of the system.

In testing the performance of the LAP, two main points within the system were identified as the potential bottlenecks. The first of these points is the ingestion of events from external applications into the LAP while the second is the querying, aggregation, and analysis done by the results service prior to returning data to the output API.

For the current two external systems sending data to the LAP, our anticipated peak load is around 0.1 MB/sec. While this load is not particularly large, future plans include ingesting events from many more external systems, so the desired performance should easily be at least ten times larger at around 1 MB/sec. To test the performance from event reception to data store insertion, an automated script was developed which creates 10 threads with each sending a series of containers with varying numbers of events to the LAP running three instances of the collection service. The processing time, from data being sent from an external application to be inserted into the data store, was then measured as a function of number of events per container. Figure 2 displays the results of this test with collection rates as MB/sec and the number of events per container ranging from 1 to 1000. The results shown in Figure 2 are informative for a few reasons. First, it is clear that high collection rates into the system requires more than one event to be sent per container. In particular, the LAP can process 0.1 MB/sec or more if the events are sent with at least 4 per container. To reach our desired bandwidth of ten times our current peak, 1 MB/sec, requires sending about 75 events per container or more with three collection instances. The second realization from Figure 2 is that the collection rate of the system somewhat flattens out between 500 and 1000 events per container, making it less efficient to process these larger containers. This effect ultimately has to be weighed against the network bandwidths in sending events from external applications and has not yet been tested. It should also be mentioned again that these performance tests were done with three instances of the collection service. Increased rates could always be achieved by increasing the number of collection service instances, but these tests were done to determine collection rates for a static number of instances.

The second potential bottleneck in the LAP is the querying of the results service and the load on the output API. Currently the LAP is in a trial mode with tens of thousands of users. For this relatively low number of users the load on the output API is not a major concern and detailed performance testing has not yet been done. The implementation

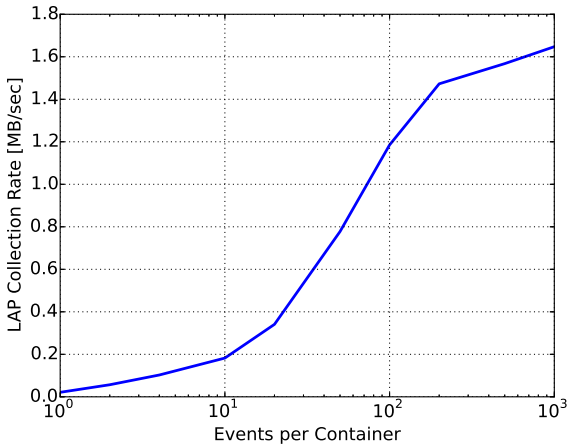


Figure 2: The collection rate of Caliper events, in [MB/sec], processed by the LAP is shown as a function of number of events per container. 10 parallel threads were used to send data to the LAP during this test, simulating the load from several external systems sending events simultaneously. The system tested had 3 instances of the collection service operating in parallel.

of auto-scaling within AWS, however, should easily handle large spikes in front end usage during peak hours.

5. FUTURE VERSIONS

The current LAP is the first iteration of a production system. It supports two external education systems, can support more than ten times the anticipated peak load, and has a modest analytics layer within its architecture. The next version of the LAP is currently being developed with the goal of supporting many more external systems with totals of tens of millions of users. In addition to increased user load, the future plans for the LAP include a substantial computational layer, opening up the possibility for richer analytics, as well as an open API for third party analytics to be done using LAP data.

The initial success of the current version of the LAP has led to plans for instrumentation of several new educational systems so their data may be ingested by the LAP. To be able to handle the increased numbers of users with the LAP, several changes and additions are needed. The most drastic of these changes is switching to a completely AWS system, fully utilizing Amazon cloud technologies [6]. Moving the LAP to a full AWS stack will allow for massive scalability and the ability to store data, perform analytics, and give support to millions of students, educators, and researchers. Further, future versions of the LAP will also have the ability to perform more advanced analytics including predictive analytics, machine learning algorithms, and large distributed calculations and aggregations. Incorporating a distributed calculation layer into the LAP will allow for a richer set of analytics to be performed and thus give the ability for deeper insight into large educational data sets.

Implementation of an open API for data consumption and analytics by third parties is also planned for the LAP. One of the intended features of the LAP is the lack of PII data held within the data store. The de-identified of the LAP data allows for third parties to ingest and do analysis on our data without concern for privacy. Creating an open API for the LAP will help push the fields of learning analytics and educational science by allowing researchers greater access to student and educator data.

6. CONCLUSIONS

We have built a platform able to ingest, store, and analyze data from external learning applications in a scalable fashion. Two existing applications, Connect and Engrade have been instrumented to create and send standardized Caliper learning events to the LAP. Once received, the learning events are transformed within customized data pipelines and stored within a fast data store, implemented using non-relational MongoDB. Analysis is done on the stored learning events, creating visualizations of educational insight for students and educators. The architecture of the LAP allows for just-in-time feedback with the insight visualizations to its users. The current version of the LAP is able to process and store education events ten times faster than is required for peak usage by the current two applications interfaced with the LAP. Future versions are planed for the LAP and will include a complete backend stack which is hosted by AWS and able to auto-scale across the entire platform. Additionally, an advanced computational layer and open API are planed for future versions of the LAP. It is our vision that present and future iterations of the LAP may provide the analysis, high quality educational data, and predictive analytics to drive the next generation of education.

7. ACKNOWLEDGMENTS

This paper is based on work supported by the McGraw-Hill Education Digital Platform Group (MHE DPG). We would like to extend our appreciation for all the help and the informational support provided by the Analytics team at DGP and the MHE CDO Stephen Laster. Despite provided support, any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect positions or policies of the company.

8. REFERENCES

[1] Experience API Version 1.0.0. Technical report, Advanced Distributed Learning, 2013.

[2] Learning Measurement for Analytics. Technical report, IMS GLOBAL Learning Consortium, 2013.

[3] JSON for Linking Data, 2015. http://json-ld.org/.

[4] J. Feild. Improving student performance using nudge analytics. In *Educational Data Mining*, 2015.

[5] M. Stonebraker, U. Cetintemel, and S. Zdonik. The 8 Requirements of Real-Time Stream Processing. *SIGMOD Record*, 34(4), 2005.

[6] E. Swensson. Big Data Analytics Options on AWS. 2014.

Improving Student Performance Using Nudge Analytics

Jacqueline Feild
McGraw-Hill Education
281 Summer Street
Boston, MA USA
jacqueline.feild@mheducation.com

ABSTRACT
Providing students with continuous and personalized feedback on their performance is an important part of encouraging self regulated learning. As part of our higher education platform, we built a set of data visualizations to provide feedback to students on their assignment performance. These visualizations give students information about how they are doing compared to the rest of the class, and allow them to compare the time they spent on assignments across their courses. Included in the feedback are 'nudges' which provide guidance on how students might improve their performance by adjusting when they start or submit assignments. In order to understand what nudges to provide to students, we analyzed historical data from over 1.4 million students on over 27 million assignment submissions to find student performance trends. The data confirmed that student performance significantly decreases when assignments are started on the same day they are due and when they are submitted after the due date. We used these findings and the past and current performance of each student to display nudges relevant for them in their visualizations, highlighting actionable strategies for improving future performance.

Keywords
self-regulated learning; data visualization; data mining

1. INTRODUCTION
Self-regulation is a trait very often associated with highly effective learners [6, 1]. Feedback is an important part of the process of self-regulation, as it allows students to evaluate their performance, to decide what actions might improve their future performance and to make adjustments to their learning processes [3, 4]. Feedback can be provided in a variety of ways, but it is especially effective when it is personalized and given in near real-time. In this paper, we describe a set of data visualizations we incorporated into our higher education platform, Connect, to provide students with exactly this kind of continuous, easy to understand feedback on their assignments to encourage the development of self-regulated learning.

Specifically, these visualizations allow students to see how they are doing on assignments as soon as they are graded. In two easy-to-understand visuals they can see trends in their performance over the semester, compare their performance to the rest of the class, and compare the time they spent on each assignment across courses. In addition to this information, we use 'nudge analytics' to provide personalized messages to encourage students toward actions that might improve future performance based on patterns in historical data [2, 5]. The word 'nudge' means to encourage someone to do something, and nudge messages are an unobtrusive way to push students toward better behavior, while leaving the choice to change up to them.

To find relevant nudges, we performed exploratory data analysis on eight months of student submissions to our higher education platform, including over 1.4 million unique students and over 27 million assignments. Our goal was to find trends in the data that identify factors that lead to decreased performance for most students. In this paper, we explore the assignment submission trends by day of semester, day of week, hour of day, and started and submitted time.

2. CONNECT INSIGHT FOR STUDENTS
McGraw-Hill Education offers a teaching and learning environment, called 'Connect', for higher education. This environment allows instructors and students to manage assessments, and access ebooks and other instructional materials. As part of Connect, we built a set of visualizations called 'Insight' to help students understand their performance on assignments. These visualizations provide important feedback to students as soon as assignments are graded in a way that is easy to understand. The interactive nature allows students to make decisions about what actions might improve their future performance.

The example visualization in Figure 1 answers the question, 'How am I progressing?' and shows a student their scores on assignments in a particular class over time. The yellow trend line shows the students scores and the blue trend line shows the class average on assignments. Clicking each data point opens a right-hand panel with more details, including the nudges toward better performance if applicable. In the following sections, we will describe our analysis for determining these messages in more detail.

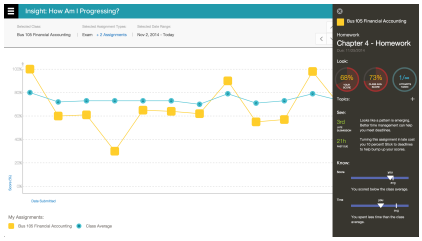


Figure 1: A visualization to answer the question 'How am I progressing?'

3. EXPLORATORY DATA ANALYSIS

In this section, we describe our analysis of eight months of historical data from assignment submissions. The goal of this analysis is to find trends in student behavior that negatively influence performance. This will help us identify the nudges that are supported by the data, and can be used to encourage students towards performance increasing behaviors.

We used historical data collected by Connect during the spring and summer semesters of 2014. This included data for 80,000 class sections taught by 29,000 instructors to 1,400,000 students. The result is over 27 million assignment submissions.

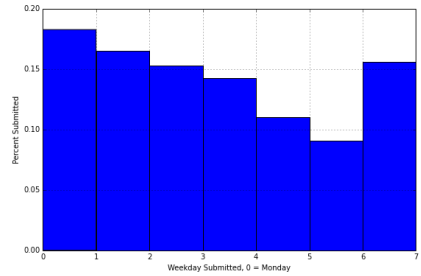
The data we used for our analysis was given to us by the Connect team from their database designed for end users, and it was not optimized for analytics. Instead, we used the existing fields for assignment submissions, including the assignment type (homework, quiz, exam, etc), start date, completion date, due date and outcome. From this data we computed a number of derived fields, including the hour of the day, day of the week and, day of the semester an assignment was submitted. We also computed the number of minutes before the due date each assignment was started and submitted.

Given these attributes, we focused our analysis on trends in assignment started and assignment submitted times. In the following sections we explore the assignment submission trends by day of semester, day of week, hour of day, and started and submitted time.

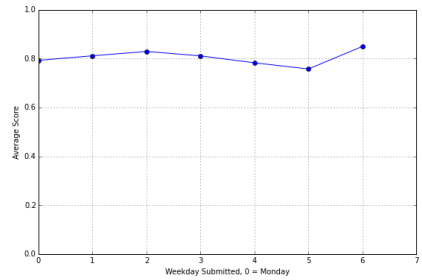
3.1 Day of the Semester

First we asked the question, does performance decrease during the semester? To start, we looked at the percent of assignments submitted on each day in our data set. This shows an interesting repeating pattern of the highest number of submissions on Monday and the lowest number of submissions on Saturday. It also shows a drop in submission volume in the middle of the spring semester, which can likely be explained by the week long spring break that occurs during this time period. Other than this decrease, submission volume remains consistent over both the spring semester and summer semester.

In order to understand student performance, we looked at the average score for assignments submitted on each day in our data set. We see a trend of decreasing performance toward the end of the spring semester (starting just before day 100). We see a similar downward trend for scores toward the end of the summer semester as well.



(a) A histogram of the percent of assignments submitted on each day of the week



(b) A plot of the average score for all submissions for each day of the week.

Figure 2

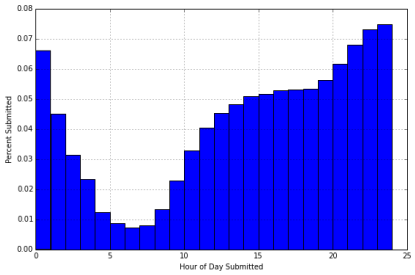
Unfortunately, there is not a clear delineation between the spring semester and the summer semester, and between the summer semester and the following fall semester, as different schools schedule classes over different time periods. Information on when classes start and end is not included in our data set, so further research is needed to confirm that this trend exists on a normalized data set.

3.2 Day of the Week

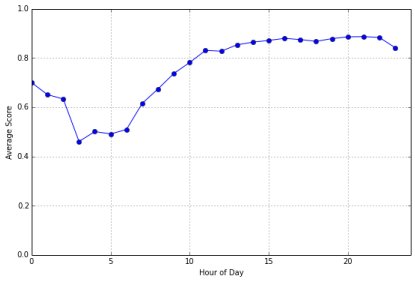
The previous analysis showed that performance decreased toward the end of the semester, but we also want to know, does performance decrease on any day of the week? Figure 2a shows the percent of assignments submitted on each day of the week. This confirms what we saw in the previous section, that the most number of assignments are submitted on Monday, while the least number of assignments are submitted on Saturday. Figure 2b shows the average score for assignments submitted on each day of the week. As expected, this shows that there is no performance advantage to submitting on a particular day of the week.

3.3 Hour of the Day

Following this analysis of scores over the semester and week, the obvious next question to explore was, does performance decrease when assignments are submitted at particular times of the day? Figure 3a shows the percent of assignment submitted during each hour of the day. This shows that most assignments are submitted between 12pm and 12am, with an increase around 8pm. While submissions do decrease in the early morning hours, there are still many submissions between 12am and 8am.



(a) A histogram of the percent of assignments submitted during each hour of the day



(b) A plot of the average score for all submissions during each hour of the day

Figure 3

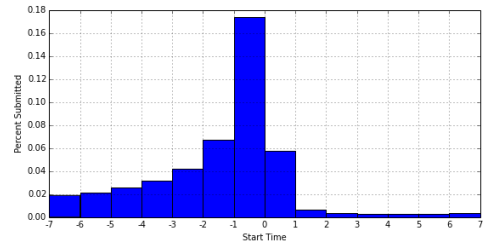
Figure 3b shows the average score for assignments submitted during each hour of the day. The average score is similar between 10am and 11pm, but steadily decreases from 11pm to 6am before increasing again. The decrease in score is significant, going from an average score of 89 at the peak hour to an average score of 46 at the lowest hour.

Unfortunately, this data represents students in many different timezones, but the the date fields are all represented in Eastern local time, where the platform servers are located. This means that we cannot draw the conclusion that submitting in the early morning hours leads to lower scores from these plots. Further work is required to obtain student time zone information and to clean the data by adjusting dates to each students local time.

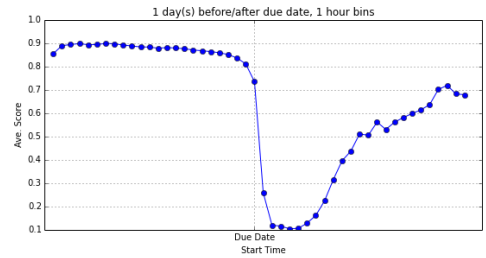
3.4 Start and Submit Time

We also looked at when a student started and submitted assignments in relation to the due date to answer the question, does assignment start time or submission time affect performance? Figure 4a is a histogram showing the percent of assignments started each day before and after the due date. The zero on the x-axis represents the deadline, so the bar between -1 and 0 represents all of the assignments that were started the same day they were due. The interesting trend in this plot is that most late assignments are started the day after the due date. This means that most late assignment could be avoided without drastic behavioral changes.

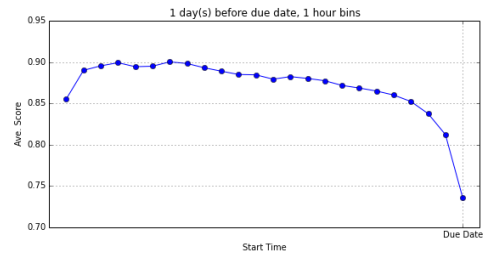
Figure 4b shows the average grade for assignments started at different points before and after the due date. The due date is in the center, and each data point to the left and right represents a 1-hour range. So the data point at the due date represents the average score of all of the assignments started



(a) A histogram of the percent of assignments started each day before and after the due date. The zero on the x-axis represents the deadline.



(b) A zoomed in version of (b) where each data point represents a 1-hour range.



(c) A zoomed in version of the left half of (c)

Figure 4

within the last hour before the due date. The point just to the left of the due date represents all of the assignments started between 1 and 2 hours ahead of the due date, and so on. In total, the plot shows one day before and after the due date. This shows that there is a decrease in average score as assignments are started closer to the deadline.

For a more detailed view, Figure 4c is a zoomed in view of Figure 4b , showing just the 24 hour window before the due date. This makes it clear that average scores significantly decrease from a high around 90 to just below 75 when started within an hour of the due date.

Plots for submit time show similar trends and are omitted due to space constraints.

The previous analysis was done using our complete data set, but we also wanted to explore whether these trends hold for each assignment type. We reproduced the plots in Figure 4 for all 14 assignment types used by our platform. We found that these trends hold for homework, quiz and exam assignments, but not all assignment types. One example where it does not hold is for LearnSmart assignments. This is most likely because Learnsmart assignments are used to drill students on a set of topics, and take a shorter period

of time to complete. Students can start them the day they are due and have plenty of time to complete satisfactorily.

4. DETERMINING DATA-DRIVEN NUDGES

We used this exploratory analysis to determine the nudge messages used in our visualizations. Based on the analysis above, conclusions could not be drawn about the day of the semester or hour of the day an assignment is submitted without further data collection and research, so these messages come from the trends seen in our exploration of start and submission time. It is clear that average scores decrease significantly as assignments are started and submitted closer to the due date and after the due date. Messages to students about when to start and when to submit are both similar in spirit, so we decided to focus our messages on starting early and avoiding submission after the due date.

We include four types of messages in our visualizations. When a student submits an assignment after the due date, they see the following message:

'Turning this assignment in late cost you <x> points! Stick to deadlines to help bump up your scores.'

and the amount of time the assignment is late is displayed in the right hand panel. When there are multiple late submissions over the semester, they will also be shown how many have been submitted late and the following additional message:

'Looks like a pattern is emerging. Better time management can help you meet deadlines.'

We also have a pair of messages focusing on starting assignments early. When students start a homework, quiz or exam within one day of the due date and they do not receive a score of 90 or better, they will receive the following message:

'Starting more than one day before the due date could result in better grades. Give yourself more time!'

If they repeatedly start assignments late, then they will see how many assignments have been started late and the following additional message:

'Late starts can lead to lower scores. Start assignments early and give yourself more time to perform better.'

These messages are designed to nudge students toward actions that will improve their performance. By providing explicit feedback about how many points they lost by submitting late, when they started assignments relative to the due date, and highlighting repeating behaviors, these message encourage students to evaluate their current actions and provide suggestions for adjusting their behavior to increase future performance on assignments.

5. CONCLUSIONS AND FUTURE WORK

In this paper we present an exploratory analysis of assignment submission data to find trends in student behavior that lead to increased performance. The data confirmed that student performance significantly decreases when assignments are started on the same day they are due and when they are submitted after the due date. We use these trends to develop data-driven nudges for students, which encourage behaviors that will help them achieve higher scores on assignments. Students see these messages when they start assignments on the same day as the due date, submit after the due date or repeatedly start or submit assignments late. These nudges are incorporated into a set of visualizations as part of our higher education platform, aimed at providing continuous, personalized feedback to students on their assignments and encouraging self-regulated learning through highlighting actionable strategies for increasing performance.

Our analysis revealed several promising avenues for future research. First, it would be interesting to understand why there are two particular assignment types that are submitted much less often than other types. This information could be used to encourage students to complete these specific assignment types or to alert instructors that these assignments are not being completed at an alarming rate and perhaps help them adjust their course to encourage completion.

We also saw potential trends in the analysis of the day of the semester assignments are completed, but we need to collect data on course start and end dates in order to clean the data set. This could lead to nudge messages reminding students to submit work as the semester progresses and scores tend to decrease. Similarly, we need to collect time zone information for each student so dates can be adjusted to local time for the analysis of the hour of the day assignments are submitted. This could lead to messages that remind students that submitting work in the early morning hours tends to lead to decreased performance.

In addition to these areas of future work, it would be interesting to do a long-term study looking at the affects of using our platform with nudge messages to understand how it affects student behaviors compared to a system that does not provide nudge messages.

6. ACKNOWLEDGMENTS

This paper is based on work supported by the McGraw-Hill Education Digital Platform Group (MHE DPG). Despite provided support, any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect positions or policies of the company.

7. REFERENCES

[1] D. L. Butler and P. H. Winne. Feedback and self-regulated learning: A theoretical synthesis. *Review of educational research*, 65(3):245–281, 1995.

[2] C. Carmean and P. Mizzi. The case for nudge analytics. *Educause Quarterly*, 33(4):n4, 2010.

[3] J. Hattie and H. Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

[4] D. J. Nicol and D. Macfarlane-Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2):199–218, 2006.

[5] B. Wildavsky. Nudge nation: A new way to prod students into and through college. *ES Select*, 2013.

[6] B. J. Zimmerman. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1):3–17, 1990.

Educational Reports That Scale Across Users and Data

Rob Rolleston
PARC, a Xerox Company
128-29E 800 Phillips Rd
Webster, NY 14580
1-585-422-0712
Rob.Rolleston@xerox.com

Richard Howe
PARC, a Xerox Company
128-29E 800 Phillips Rd
Webster, NY 14580
1-585-422-5283
Richard.Howe@xerox.com

Mary Ann Sprague
PARC, a Xerox Company
128-29E 800 Phillips Rd
Webster, NY 14580
1-585-422-2913
MaryAnn.Sprague@xerox.com

ABSTRACT
The field of education is undergoing fundamental change with the growing use of data. Fine-scale data collection at the item-response level is now possible. Xerox has developed a system that bridges the paper-to-digital divide by providing the well-established and easy-to-use paper interface to students, but digitizes the responses for scoring, validating, reporting, and managing data using a range of digital technologies. The Ignite™ system supports written responses, shading, connecting lines, multiple choice selections, and other question types. For some users, monitoring is at a very fine-grain level in both time and skill, while for others the data is used for more summative evaluations and strategic planning; one users' details may be another user's overview. All the reports presented in this document use the same basic atomic data elements and associated meta-data. The hierarchical nature of the organization of users requires that these atomic elements be combined in different ways for specialized visual representations, dependent upon the needs of the user.

Keywords

Visualization in Education, Application, Data Transformation and Representations, Field Studies, Ethnography.

1. INTRODUCTION
Technology and regulations regarding education have increased the availability of student data as well as the need to track student performance over time. No Child Left Behind [7], Race-to-the-Top [9], and the Common Core State Standards [3] are all efforts within the United States that have endeavored to make the tracking of student learning growth more measurable. Despite the increase in the access to, and need for data and data analysis, the abilities for many educators to make use of available data has not kept pace with the need [5]. Data analysis requires knowledge and tools to which not all educators have easy access [13]. Enabling data to be visualized in ways that are familiar to educators will help encourage the use of student data to inform student learning and instruction. This paper provides one example of how reporting student data in a user friendly form, for many levels of users, can help educators to find more effective uses for their student data.

2. BACKGROUND

2.1 Teachers Changing Their Instruction and Their Needs
Teachers have begun transitioning from curriculum-based instruction to student-centered instruction, which shifts the focus to assessing students at the beginning, middle and end of an instructional unit. In this way, teachers learn what the students already know about a particular subject from the start, where to focus needed instruction, and collect data throughout the process of their learning growth.

To help support the shift to utilizing student data to inform day-to-day instruction in ways that fit more closely into educator's current work processes, Xerox has created the Xerox Ignite™ Educator Support System [16]. Ignite™ is a web-based teacher tool for printing, scanning and scoring a variety of hand-marked student work and also manages the student data and produces personalized reports. Student work is generally an assessment (e.g. a quiz or test).

The item-response level of information is defined as an atomic unit: "A student is presented an item on a date by a teacher and provides some type of response." Each part of the atomic element contains additional metadata. All the reports present views of the same underlying data, but with differing levels of aggregation, dependent upon the needs of the user. This paper describes these differing user requirements and how a set of consistent and connected graphical reports can scale across the needs of these different users and their needs for data.

3. RELATED WORK
Data and data mining usage in the education domain (educational data mining or EDM) is relatively new. The field has grown rapidly for just over a decade [1].

There is a desire that the use of data will foster improvements at all levels of education. The desire for data-driven improvement in learning is countered by a concern that the use of data by itself will lead to too much of a focus on testing rather than teaching [5]. Over the years the focus of research has moved more into the field of prediction [2], and it may be that the real value will come over time when enough longitudinal data is available.

Public educational institutions have a hierarchical nature. In the United States primary schools there is a hierarchy of superintendents, principals, team leaders, and classroom teachers all making decisions. This hierarchy of users share common tasks including the analysis and visualization of data, providing feedback to support instructors, recommendations for students, and grouping of students, among others. The hierarchical nature of the users within the educational organization presents interesting challenges in both EDM [2] and in the use of the data. Teachers want easy-to-use systems, with a "desire to see assessment results at the level of subscales (groups of test items) related to specific standards and at the level of individual items in order to tailor instructions." [6] "Decisions are made at all levels of school organization. The superintendent makes decisions concerning a school district's goals and strategies. Then principals make tactical decisions concerning those goals and strategies to accomplish them in relation to their own buildings. Department heads and team leaders then make curricular and operational decisions to carry out the day-to-day activities of a department or unit. And, finally, classroom teachers make decisions in their classrooms".

Others have investigated the use of visualization in higher education situations with limited success [8]. The use of on-line learning tools has led to visualizations of curriculum [4], the design of models of student learning [10], the use of graph structures to understand

patterns of student enrolment [14] and the use of a small set of static and interactive visualizations of user data [11].

4. USERS AND USER REQUIREMENTS

4.1 Ethnographic Study of Pilot Deployment

During a technical pilot of the Xerox Ignite™ application, four elementary schools in two school districts participated. An ethnographic study was conducted to observe teachers' processes to identify challenges with using the pilot tool, and to collect user requirements and needs for the system. Observations and open-ended interviews [12] were the primary data collection methods to study teachers' assessment practices. Many teachers expressed a desire to know how their students were performing in the skills being taught, and wanted to understand how well the students understood the skills.

Talks with principals and school district administrators, including district data specialists, uncovered another level of requirements [29] relating to trend analysis, student growth over time, class-to-class and school-to-school comparisons, and progress monitoring. Principals and administrators expressed a need to see student data at the grade level, reaching from single classrooms, to all classrooms for a single grade in a single building, to entire buildings or the entire district.

Just as the assessments are used for different purposes, the users of reports have different needs. The users have been segmented into three major groups: A single teacher & class, principal or lead teacher with several classes within a school, and a district administrator looking across multiple schools in the district.

4.2 Single Teacher / Class

A teacher working with a class, or a single student in a class, is the lowest level of granularity within the current scenario. In this situation a teacher has one of two major goals: assessing the success or direction of a lesson or helping a single student.

To assess the success of a lesson plan, a teacher needs to see the class average, but also details about the mastery of different skills within that teaching unit. In Ignite™, the teacher can group and sort the questions on an assessment report according to the metadata related to the skills connected to each question.

To help an individual student, the teacher, student, and often parent need fine-scale information about specific mastery of skills. The teacher must be able to communicate with the student and parent on specific problem areas that need immediate work.

The data markers for this type of user must reveal information about the individual student, and the specific question or skill. Reports must reveal the data at the level of the basic atomic data unit, to the level of every item and response by a student to that item.

4.3 Several Classes within a Grade or School

A school principal or lead teacher within a grade or subject area needs a middle level of data aggregation. A principal is in charge of an entire school building, typically covering several grades. A lead teacher is typically focused on a single grade, or a single subject within a grade. Users at this level are typically looking at the overall progress of a cohort and the management of class or group affiliations of students. The goal is to best assign students to classes or groups and to insure that these classes or groups are on track to meet marking period and yearly goals.

The data markers for this type of user need to reveal information about the class statistics and summary information about individual students. Reports for this level can also reveal the data so that cohorts can be compared, and individual outliers within classes or groups can be identified. Skill proficiency can be shown across time and across multiple classes within a given group giving a wider view of proficiency trends. The skills are grouped at the larger unit or quarterly time intervals, and not at the individual skill code level.

4.4 Across Schools within a District

District administrators analyse data to determine long-term trends and comparisons, to report to state agencies, and to evaluate curriculum. Users at this level are more focused on summative and high-stakes assessments. These users look across schools, and compare their own district with other districts in the area and those with similar social and economic demographics.

The data markers for this class of user need to reveal information about the overall aggregate status of a district or school, with visual markers at the grade, teacher, or building level. Information about individual students is not identified or required. Trends for skills are most often limited to subject and grade level expectations.

5. DESIGN CONSIDERATIONS

Design choices were made in response to the user requirements discussed above. These design choices fell into two major areas: The general usability or workflow, and the visual attributes of the reports.

5.1 Report Selection Workflow

To produce a report, the user needs to select both a data set and a report type. These selections are not independent; as not all data can be rendered as any report, and each report needs an appropriate set of data. A linear sequential method was developed to guide the user through the report selection process.

The logical concept of the selection process is shown in Figure 1. The first step is the top row of the selection space where the user specifies "Who?" i.e. the target user and student aggregation level of the report. The second step is the left most column of the selection space where the user specifies "What?" i.e. how many assessments are to be viewed in the report. The third step of the selection process defines "When?" i.e. is this report for a single instance, or does it cover multiple recurrences. These three linear sequential steps allow the user to navigate simply through a three-dimensional specification space ending with a choice of just a few different eligible reports.

(2)What?	(1)Who? (3)When?	Teacher / Class	Principal or Lead Teacher / Several Classes	District/ Multiple Schools
Single Assessment	Once	• Matrix • Table • Image	• Matrix • Bar	• Distribution
	Over Time	• Matrix • Line	• Grouped Bar • Lines	• Lines
Portfolio of Assessments	Once	• Bar	• Bar/ Scatter	• Distribution
	Over Time	• Lines	• Lines	

Figure 1 - Report selection table

The layout of the report selection table shown in Figure 1 also aids in understanding the automatic aggregation of data according to user and data selection. The different user views for the case of aggregating data about a single assessment given once are shown in Figure 2. In all cases, that same single assessment is chosen, but the report is different depending upon the scope of classes selected. A teacher meeting with a single student is most interested in the report

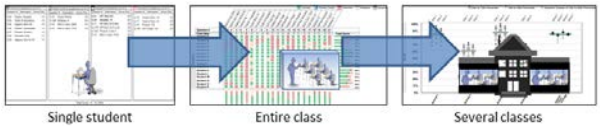


Figure 2 Flow of data aggregation and report type for a single assessment given once. In moving from left to right the question detail is lost and the higher level averages are presented.

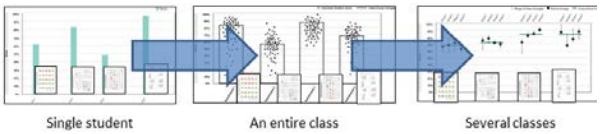


Figure 3 Flow of data aggregation and report type for a portfolio of four assessments. In moving from left to right the student specific information is lost and higher level distributions are presented.

on the left; a table report that provides detailed information about each question. A teacher assessing the progress of an entire class is most interested in the report in the middle; a matrix heat map. A district administrator, looking at several classes is most interested in the report on the right; a distribution report that provides information about school and class averages, and some notion of the distribution of scores within each class and range of scores within each school.

The different user views for the case of aggregating data about a portfolio of assessments are shown in Figure 3. In all cases, the same set of assessments is chosen, but the reports are different depending upon the scope of classes selected. A teacher meeting with a student and parent is most interested in the report on the left; a single bar chart report that provides a student's performance across a range of skills, at a single point in time. A group of teachers meeting to discuss the grouping of students is most interested in the report in the middle; a bar chart report that provides information about the class average, but also the score of each student. A district administrator, looking at several classes is most interested in the report on the right; a distribution report that provides information about school and class averages, and some notion of the range of scores within a class.

6. REPORT DESIGNS

There are 6 different basic report styles: (1) image, (2) table, (3) matrix heat map, (4) line, (5) bar, and (6) distribution. Where appropriate, report styles were customizable to properly display the defined data set selected by the user.

6.1 Image

The image report is an image of the scanned and validated assessment. This report is the only portrait mode report, and has no header or footer information to maximize the actual assessment image region. Two examples of this report type are in Figure 4.

The visual representation exactly matches the physical paper version of the assessment.

The green highlight areas are those questions that were validated by the teacher as correct. Responses that were validated as incorrect were shown with a red overlay, skipped responses with a yellow

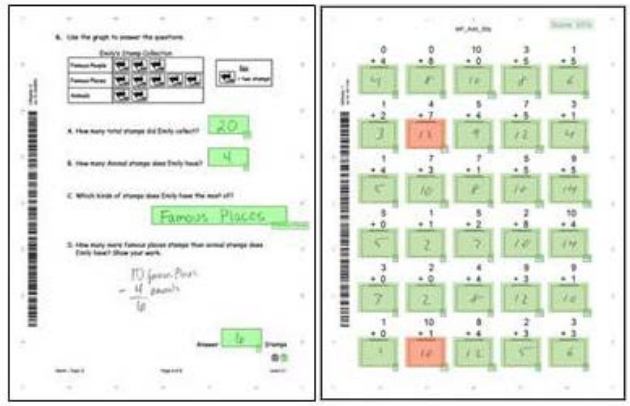


Figure 4 - Image reports where green indicates the area of a correct answer and red is an incorrect answer

overlay, and those validated for partial credit were shown in blue. The color coding served as a pre-attentive signal to the correctness of the student's answer.

This report is used by a teacher in individual consultations with a student and/or parent. It is a record of the marks the student made on the assessment, and how the teacher validated each question.

6.2 Table

The table report is a listing of the correct, partial, skipped, and incorrect responses to a single assessment, by a single student. The questions are placed in the column corresponding to the scored and validated response and include the question number, the question description (obtained from metadata), the number of points the student earned for the question, and the total number of points possible on the question. The data is presented at the fine-scale atomic level, with the use of metadata. By using the metadata about the question, the report is applicable to any type of question. If the question description metadata is used to encode specific skill information, then a quick visual scan down the list reveals common skills that have appeared in a single column, or the columns could be sorted according to some value(s) in the metadata.

This report is used by teachers with students and parents. It is easy for students to understand that their goal is to make all the questions appear in the left-most (correct) column.

6.3 Matrix heat map

The matrix heat map report is a visual summary of the responses to each question on an assessment, by each student. The data for each student appears in a row, and the data for each question appears in a column. At each intersection point there is a graphical representation of the student's response to that question.

Sorting the rows and columns of the matrix provides the user with a quick visual assessment of several different stories [15]. By sorting the rows in order of student score, the teacher is able to quickly discern who has mastered the skills, and who has not as well as identifying which areas where most students are struggling. By sorting the columns in order of the correct number of student responses, or sorting by the metadata associated with questions, the teacher can quickly see what groups of questions or skill sets were successfully mastered by the class and which were not. If a particular set of question were not mastered by the class, the teacher now has this additional piece of information and can decide upon the need to re-teach a particular set of skills to that subgroup of students.

6.4 Bar

The Bar Chart is used in several instances. For consistency, all bar charts use the vertical axis to represent score. The horizontal axis can be categorical values of assessments, students, classes, or dates. The

bars can be singular or grouped, depending upon the amount of data to be displayed.

6.4.1 Comparing Students or Classes over time

The collection of scores for a single assessment repeated over time can be represented using a bar chart. In this case, the x-axis is a categorical list of students or classes. There is a group of bars for each student or class, and a bar of each instance in time.

6.4.2 Portfolio of Assessments

The portfolio of assessment scores for a single student, a single class, or all the classes within a school all use similar representation. The common representation provides users with a common mental model for scaling across such aggregations. In all these cases, the x-axis is a categorical list of assessments.

For the case of a single student, there is one bar for each assessment.

For the case of a single class, there is one bar for each class average score, and the distribution of student scores are overlaid as a jittered scatter plot, where the x-jitter is bounded to the width of the corresponding bar

6.5 Line

The line chart is used only for trends over time. For consistency, all line charts use the vertical axis to represent score. For a single assessment, the score can be absolute or percentage, and may be aggregated at the student, class, or school level. In all cases, the x-axis is a categorical list of times, e.g. the date each assessment was given. These reports are used to view the progress over time of one or more assessments.

When viewing the portfolio of results over time, each line represents a different assessment. The points on the line are the relative score achieved on the given assessment at that point in time. The score can be a single student, aggregated over a class, or over a school.

6.6 Distribution

Reports showing a statistical summary of distributions typically are used only by district level users. These types of reports use the greatest amount of aggregation of atomic data elements. For consistency, all distribution charts use the vertical axis to represent score. The x-axis is a categorical list of assessments or schools. The markers are groupings of box and whisker plots.

7. CONCLUSION

We designed and deployed a system that implements a large and comprehensive set of reports for use by educators at different levels. The system was designed based on ethnographic studies and iterative participatory feedback from users, as well as subject matter experts on staff. The novel aspect of this work was the creation of a complete system that bridges the paper-digital divide, offers views into the data at different levels of granularity and aggregation, and scales to match a user's needs and work processes while preserving similarity in the selection workflow and report design.

The reports all used the same underlying fine-grain data element at the item-response level by each student, but aggregate the data differently dependent on the user's needs. Users included teachers-students-parents, lead teachers or principals, and district level administrators. These users had needs that required scalability thru several levels of data aggregation.

Report designs focused on the re-use of basic design concepts across the different visual representations, thus allowing users to more easily traverse the report space by learning a common set of patterns and styles. The report selection process follows a linear progression of selecting the collection of students, the collection of item-responses, the time, and finally any visual representation options for the data.

8. ACKNOWLEDGMENTS

The authors wish to thank the Ignite™ school support team of Todd Conrow, Deb Drago-Leaf, and Lori Schirmer for their critiques, suggestions and detailed knowledge of what it is really like to be in the classroom.

9. REFERENCES

[1] A. Peña-Ayala. (2013). "Educational data mining: A survey and a data mining-based analysis of recent works," Expert Systems with Applications. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.08.042.

[2] R.S.J.D. Baker and K. Yacef, "The State of Educational Data Mining in 2009: A Review and Future Visions," *Journal of Educational Data Mining*, Article 1, Vol. 1, No. 1, Fall 2009.

[3] "Common Core State Standards Initiative." [Online]. Available: http://www.corestandards.org/2011.

[4] M. Ida, "Web Service and Visualization for Higher Education Information Providing Service," 2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS), 16-18 July 2010, pp. 415 – 418. Available: 10.1109/ICSESS.2010.5552349.

[5] M.M. Kennedy, "Data Use by Teachers: Productive Improvement or Panacea?" *Education Policy Center at Michigan State University, Working Paper #19*, May 20, 2011.

[6] B. Means, C. Padilla, and L. Gallagher, "Use of Education Data at the Local Level, From Accountability to Instructional Improvement," *U.S. Dept. of Education*, January 2010. Available: https://www2.ed.gov/rschstat/eval/tech/use-of-education-data/use-of-education-data.pdf.

[7] "No Child Left Behind" [Online]. Available: http://www2.ed.gov/nclb/landing.jhtml.

[8] M. Pinto, R. Raposo, and F. Ramos, "Comparison of Emerging Information Visualization Tools for Higher Education," *2012 16th International Conference on Information Visualisation*. Available: 10.1109/IV.2012.27.

[9] Race to the Top, http://www2.ed.gov/programs/racetothetop/index.html

[10] M. Sasakura and S. Yamasaki, "A Framework for Adaptive e-Learning Systems in Higher Education with Information Visualization," 11th International Conference Information Visualization, 2007.

[11] K. Silius, A.-M. Tervakari, and M. Kailanto, "Visualizations of User Data in a Social Media Enhanced Web-based Environment in Higher Education, "*2013 IEEE Global Engineering Education Conference (EDUCON)*, pp. 893 – 899.

[12] M.A. Sprague and M. Fuhrmann, "Ethnography Supports Changes to Student-Centered Instruction," *TQR 4th Annual Conference*, January 18, 2013.

[13] J.C. Wayman, "Involving Teachers in Data-Driven Decision Making: Using Computer Data Systems to Support Teacher Inquiry and Reflection," *JOURNAL OF EDUCATION FOR STUDENTS PLACED AT RISK*, 10(3), 2005, pp. 295–308.

[14] D. Wortman and P. Rheingans, "Visualizing Trends in Student Performance Across Computer Science Courses, "Proceedings *of the 38th SIGCSE technical symposium on Computer science education*, 2007, pp. 430 – 434.

[15] Journal of Educational Data Mining, http://www.educationaldatamining.org/JEDM/index.php/JEDM/issue/archive

[16] Ignite, http://www.acs-inc.com/health-and-human-services/education-services/ignite-teacher-tool.aspx

Mining Login Data For Actionable Student Insight

Lalitha Agnihotri
McGraw-Hill Education
lalitha.agnihotri
@mheducation.com

Ani Aghababyan
McGraw-Hill Education
ani.aghababyan
@mheducation.com

Shirin Mojarad
McGraw-Hill Education
shirin.mojarad
@mheducation.com

Mark Riedesel
McGraw-Hill Education

Alfred Essa
McGraw-Hill Education

ABSTRACT

Student login data is a key resource for gaining insight into their learning experience. However, the scale and the complexity of this data necessitate a thorough exploration to identify potential actionable insights, thus rendering it less valuable compared to student achievement data. To compensate for the underestimation of login data importance, in this paper we performed an exploratory data analysis of a large educational dataset consisting of 100 million instances of login data from 1.5 million unique students who attempted 783 thousand assignments. The data were from a McGraw-Hill Education web-based assessment platforms called *Connect*. Different data mining methods were employed to answers our initial questions regarding students' login behavior. Most of the findings were consistent with the intuitive expectations of student login patterns such as a considerable decline of activity on Saturdays, a visible peak on Sunday evenings, a high activity in September and February, and an increased activity toward later hours of the day. However, we also discovered an unexpected result while investigating the effects of the login activity, the performance scores, and the attempts. Surprisingly, this analysis showed a high positive correlation between login activity and performance scores, only up to a certain threshold. This provided us a new hypothesis on student groupings, which we explored through a cluster analysis. As a result of our exploratory efforts, a significant amount of patterns emerged that not only confirmed previously set forth expectations but also provided us new hypotheses, which can be leveraged to improve student outcomes.

Keywords

Exploratory data mining, assessment platform, clustering, log data, pattern & trend mining

1. INTRODUCTION

An increasing number of higher education institutions are incorporating online course management platforms, which creates a tremendous opportunity for monitoring learners' academic activity. These web-based learning environments capture immense amounts of login data that could be used for student monitoring and profiling ([11]). Educational literature suggests that monitoring students' academic activity is a key to a more effective and higher quality education ([2], [3], [7], [8]). Furthermore, research shows that college students would benefit from opportunities of introspection and cognitive monitoring of their progress in order to engage in careful academic planning ([1]). Hence, given its scale, these login data are a promising resource for shedding light onto students' academic behavior.

In this paper we explore login data from a McGraw-Hill Education's (henceforth MHE) web-based assessment platform. These data can serve as a basis for instructors' personalized intervention programs and feedback for student efforts toward self-regulated learning. While interest in login data analysis has been continuously increasing, there is no standardized way of analyzing this type of data ([9]) due to diversity of the data and uniqueness of research questions. Hence, we conducted exploratory data analysis without setting a priori limitations or hypotheses on our data. In Sections 2 through 4, we discuss our methods with detailed descriptions and their findings. Section 5 contains discussions about our results and conclusions along with future work.

2. METHODS

2.1 Participants and Materials

Our research data is collected via one of the MHE assessment platforms called Connect (`http://connect.mheducation.com`). Connect is a higher education web-based assessment and assignment platform, which provides students an online environment to do their coursework and logs user activity in order to provide feedback and support to its user needs.

In this paper we explored 100 million instances of user login data obtained from Connect between June of 2013 and June of 2014. For this analysis, we used data such as students' login dates, total number of logins, number of attempts on an assignment and assignment score. Depending on the analysis, some of these data were aggregated based on time or grouped by the unique students.

2.2 Procedures and Methods

To extract the necessary data for our analyses, we used Oracle's procedural language extension for SQL (i.e., PL/SQL) [4] and Python programming language [13], along with the necessary Python libraries to query, wrangle, clean, plot, and explore our login data. Our data contains the following attributes: student related data (e.g., student ID, student logins) and assessment related data (e.g., number of attempts, assessment score, number of attempts).

3. LOGIN BEHAVIOR ANALYTICS

3.1 Login Behavior

In this section we investigated the trends related to student logins. Figure 1 visualizes the overall pattern of student logins over the days of the week. The red line shows the average number of logins for any given day. This analysis validates the expected pattern of decreasing activity on Saturdays and increasing activity on Sunday evenings. This shows students' tendency to stay away from their homework assignments on the weekend until late Sunday when they attempt to prepare for the week. This finding is not surprising, in fact, it confirms the intuitive expectation of student academic activity on weekends vs. weekdays. If investigated further (i.e., A/B testing), this information could provide a basis for notifying students with customized and timely recommendations via Connect.

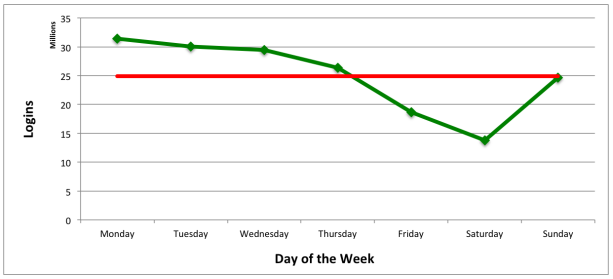


Figure 1: Logins by the day of the week. X-axis = Day of the week from Monday to Sunday; y-axis = Logins (in millions).

Next, in Figure 2 we investigated the number of logins per day. While the overall pattern of logins increasing in Fall through Spring and decreasing in Summer seemed very reasonable, the significant spike in Spring of 2014 seemed out of ordinary. To understand this unusual pattern, we requested more information from the Connect marketing team who explained that the spike in the Spring of 2014 is congruent with the new marketing effort making Connect assignments mandatory portion of students' coursework. This finding provided a data grounded confirmation of Connect team's marketing efforts.

4. PATTERN MINING & STUDENT PROFILING

For the analyses in this section, we used the average number of logins per assignment (henceforth logins), the average score per student (henceforth score), and the average attempt per assignment (henceforth attempt). In this section, we present our analysis of comparing the student login data with students' scores on assignments.

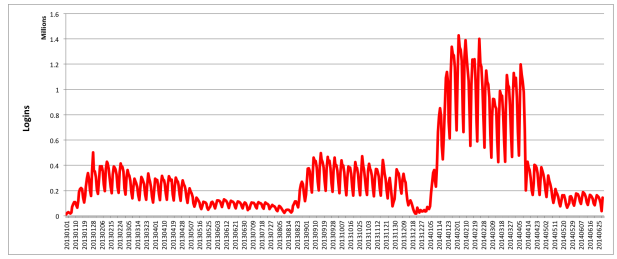


Figure 2: Logins by the month. X-axis = Days in months from 01/01/2013 to 06/25/2014; y-axis = Logins (in millions).

4.1 Login vs. Score Trends

To continue our data explorations, we decided to further investigate the potential patterns in the student login and student assignment score data.

4.1.1 Data Preparation

For this analysis, we looked at a total of 1.5 million users' assignments scored between June 2013 and June 2014. For each user, score, login and total number of attempts were normalized against users' total number of activities. Further, we eliminated some of the outliers by excluding the users with 1 or no attempts and eliminated users with more than average 50 logins which removed 100,000 users' data. On average, students login 5.5 times, have 1.03 attempts and have a score of 53% per activity.

4.1.2 Data Analysis

We plotted student logins per assignment vs. student's median score (see the green line in Figure 3). In this plot, we used the median score instead of the mean of the scores in order to account for the high variability of the distribution of scores. This figure shows that student median score grows as the number of logins increases. However, after a certain threshold, the score tends to decrease as the number of logins per assignment increases, thus showing the counter-productivity of the login activity. This contradicts to the intuitive assumption that more logins result in a better academic performance.

To further explore the relationship between login and scores, we performed a piecewise linear regression to identify possible segments in the data. Fitting a single regression line, the standard error (SE) of estimate with one regression line was σ_{est}=18. The SE for a model with two regression lines resulted in σ_{est}=12.5. We also tried fitting three regression lines through, which resulted in a higher SE of σ_{est}=16.8. Therefore, we used a model with two regression lines (see Figure 3). This resulted in a break at i=4 (i.e., Segment 1 = 0:4 and Segment 2 = 5:50). This suggests two distinct segments in the data. In the first segment, as the number of logins increase, the performance improves (slope = 6.48; correlation = 0.99). However, after a certain threshold, 4 logins, the scores plateaus, and gradually decrease as the logins increase (slope = -0.45; correlation = -0.93). This hypothesis is further explored in the next section through cluster analysis.

4.2 Student profiling

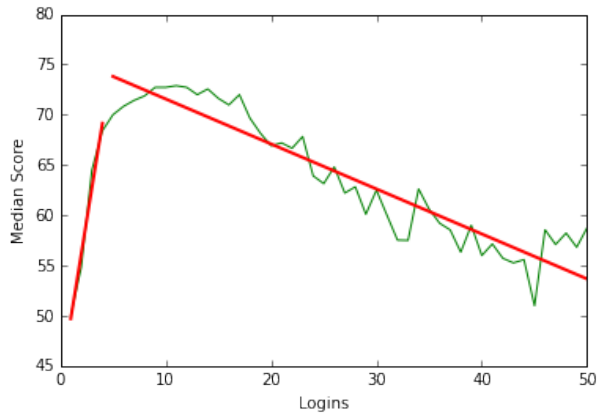


Figure 3: Piecewise linear model. X-axis = Number of logins per assignment; y-axis = Median score.

4.2.1 *K-Means Clustering Method*

Following the hypthesis formed in the previous section, we explored student login patterns through k-means clustering. In k-means clustering, data is partitioned into k clusters where each observation is assigned to the cluster with the nearest mean ([6]). The clustering process starts by choosing k random observations as initial cluster centroids. Thereafter, each observation is assigned to the nearest centroid and the new centroids are recalculated using the average of the data points in each cluster. We selected Euclidean distance as the distance metric in k-means clustering ([5]) where within-cluster sum of squares (hereafter, WCSS) is the cost function. Representing the data as a set of N observations $\{x_1, x_2, , x_n\}$, where each observation is a D-dimensional vector of D attributes, k-means clustering partitions N observations into k clusters $\{c_1, c_2, , c_k\}$ where WCSS is minimized as:

$$argmin \sum_{k=1}^{K} \sum_{X \in c_k} \parallel X - \mu_k \parallel^2$$

where μk is the mean of points in c_k. To accommodate the scale of our dataset, we have selected k-means clustering method due to its computational speed and efficiency compared to hierarchical clustering. In addition, k-means clustering is a robust approach, which results in non-overlapping clusters that are very easy to interpret. We have used the Elbow method ([12]) to identify the optimal number of clusters. In this method, average WCSS is measured as the number of clusters increase. Having more clusters results in smaller distances from centroids and hence a smaller average WCSS. However, the amount of drop is not constant as the number of clusters increase and the decrease in average WCSS flattens at a certain k value. This value, called the elbow metric, creates a break in the elbow graph and is a good measure for identifying optimal number of clusters.

4.2.2 *Clustering Results*

In this analysis, we used the same data aggregations for students' login, score and attempts as described in the beginning of this section to explore student groupings according to their login behavior. The elbow method is used to decide an optimum number of clusters. Figure 4 shows the average WCSS value as the number of clusters increases from 1 to 9. The graph nearly flattens after k equals to three, thus suggesting 3 as the optimal number of clusters.

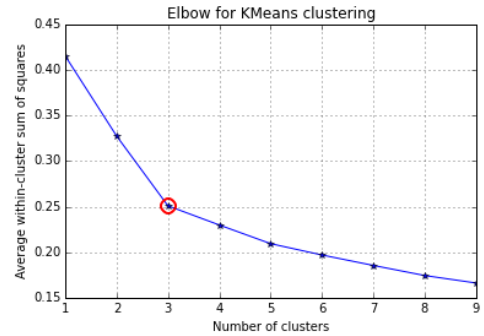


Figure 4: Elbow metric. k=3; x-axis = Number of clusters; y-axis = Average WCSS.

We used Scikit-learn python library ([10]) to implement k-means clustering. Figure 5 shows a 3D scatter plot of the three attributes used to cluster the data where the data points are colored by the cluster labels. Figure 5 shows

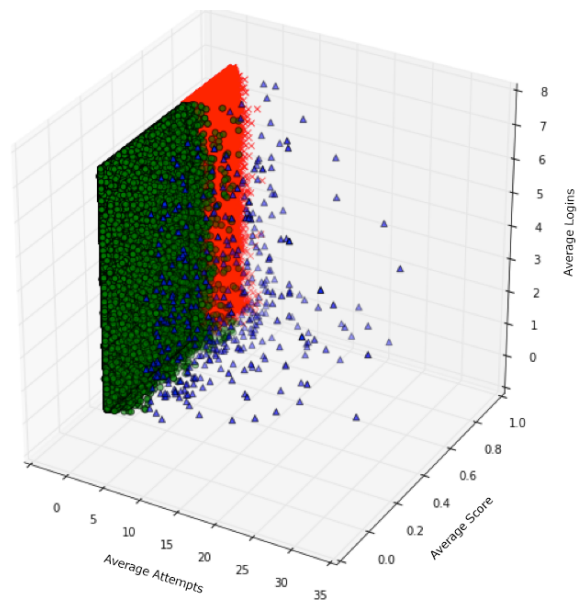


Figure 5: 3D scatter plot. Cluster 1 (red) = High Achievers; Cluster 2 (green) = Low Achievers; Cluster 3 (blue) = Persistent Students; Attempts = x axis; Logins = y axis; Score = z axis.

three sets of distinct student login profiles. The Cluster 1 (red), whom we label as *High Achievers*, represent a group of students with a low number of attempts, a medium number of logins, and a high score. The Cluster 2 (green), whom we label as *Low Achievers*, is the group with a medium number of attempts, and low number of both logins and score. Finally, the Cluster 3 (blue), whom we label as *Persistent Students*, is the most distinct group with a high number of both attempts and logins, and a medium score. To quantify

this information, in Table 1 we have tabulated the count, the mean and the standard deviation of these three attributes across each of the three clusters. In addition, we have simplified this content in Table 2.

Table 1: Cluster Statistics. Total = number of observations. SD = standard deviation.

		Average Attempts	Average Score	Average Login
Cluster 1 (High Achievers)	Count	1097675	1097675	1097675
	Mean	1.03	0.84	8.51
	SD	0.19	0.37	58.07
Cluster 2 (Low Achievers)	Count	780405	780405	780405
	Mean	1.05	0.24	6.03
	SD	0.25	0.17	21.33
Cluster 3 (Persistent Students)	Count	1220	1220	1220
	Mean	9.82	0.56	35.65
	SD	6.83	0.32	62.13

Table 2: Student groups based on cluster statistics.

	Login	Attempt	Score
High Achiever	Medium	Low	High
Low Achiever	Low	Medium	Low
Persistent Student	High	High	Medium

Table 2 shows that Cluster 1 (high achievers) includes students with the highest score among the three clusters. Low achievers, Cluster 2, stand out with a very low score and a low number of logins. This shows a relationship between the low logins and the low performance scores in students with very high or very low scores. However, students with medium score have very high average logins and high average attempts per activity. This fluctuation between average score and login indicates a non-linear and non-trivial relationship between student behavior (number of logins and attempts) and performance.

5. CONCLUSION & DISCUSSION

In this paper we explored student login data collected from MHE's Connect higher education platform. The investigation of student login activity reveals a non-linear relationship between student activity and performance. Piecewise linear regression revealed that students who do better on their assignments tend to login more. However, if a student logs in 5 or more times per assignment, their performance tends to plateau and then deteriorate. Thus, it would be beneficial for the instructor to intervene at this point as it might indicate that the student has not grasped the concepts required for the assignment. Finally, investigating student login behavior led to identifying three distinct groups of students: high achievers who login just optimum number of times to get high score, low achievers, who login very rarely and tend not to do well, and persistent students who show grit in their efforts to succeed by logging in and attempting the most but still perform less than high achievers. The educational value of such finding is in identifying and encouraging certain activity behaviors that are correlated with good performance.

Future work will be concentrating on factors such as the variability in the students' scores based on the due date of the assignments, time spent on assignments, potential recommendations or instructors actions and effectiveness of these recommendations via A/B testing. Finally, we will be attempting to join students academic performance gathered from Connect to their performance or other institutional or demographic data in order to predict student academic success.

6. ACKNOWLEDGEMENTS

This paper is based on work supported by McGraw-Hill Education. We would like to extend our appreciation for all the informational support provided by the Connect Team, the research support provided by the MHE CDO Stephen Laster, and the Analytics team at DGP. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect positions or policies of the company.

References

[1] N. E. Commander and B. D. Smith. Learning logs: A tool for cognitive monitoring. *Journal of Adolescent & Adult Literacy*, 39(6):446–453, 1996.

[2] K. Cotton. Classroom questioning. *School improvement research series*, 3, 2001.

[3] R. DuFour. Professional learning communities. 1998.

[4] S. Feuerstein and B. Pribyl. *Oracle pl/sql Programming.* " O'Reilly Media, Inc.", 2005.

[5] J. C. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

[6] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[7] K. Leithwood, K. Seashore Louis, S. Anderson, K. Wahlstrom, et al. Review of research: How leadership influences student learning. 2004.

[8] R. Mazza and V. Dimitrova. Visualising student tracking data to support instructors in web-based distance education. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 154–161. ACM, 2004.

[9] M. Muehlenbrock. Automatic action analysis in an interactive learning environment. In *The 12th international conference on artificial intelligence in education, AIED*, pages 73–80, 2005.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[12] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[13] G. Van Rossum and F. L. Drake. *The python language reference manual.* Network Theory Ltd., 2011.

Building Models to Predict Hint-or-Attempt Actions of Students

Francisco Enrique Vicente Castro **Seth Adjei** **Tyler Colombo** **Neil Heffernan**
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA 01609, USA
{fgcastro, saadjei, tjcolombo nth}@wpi.edu

ABSTRACT
A great deal of research in educational data mining is geared towards predicting student performance. Bayesian Knowledge Tracing, Performance Factors Analysis, and the different variations of these have been introduced and have had some success at predicting student knowledge. It is worth noting, however, that very little has been done to determine what a student's first course of action will be when dealing with a problem, which may include attempting the problem or asking for help. Even though learner "course of actions" have been studied, it has mostly been used to predict correctness in succeeding problems. In this study, we present initial attempts at building models that utilize student action information: (a) the number of attempts taken and hints requested, and (b) history backtracks of hint request behavior, both of these are used to predict a student's first course of action when working with problems in the ASSISTments tutoring system. Experimental results show that the models have reliable predictive accuracy when predicting students' first course of action on the next problem.

Author Keywords
Educational data mining; intelligent tutoring systems; student modeling; student behavior.

1. INTRODUCTION

Most educational data mining (EDM) research focus on modeling student behavior and performance. Algorithms such as Bayesian Knowledge Tracing [1], and Performance Factors Analysis [4] have been used to achieve this end. In intelligent tutoring systems, it is crucial to be able to understand student behavior to provide better tutoring practices and improved content selection for these systems. Student behavior may provide another means to identify low-knowledge or low-performing students and determine when to proactively intervene. Previous works show that students who are more likely to ask for help on problems learn less and perform less. A study on students' help-seeking behavior in an SQL tutoring system [3] suggests that students who used help very frequently had the lowest learning rate and had shallow learning. A study that used the sequence of attempts and hint requests to predict student correctness found that students who first made attempts on problems performed better than those who requested for help first [2]. The Assistance Model [6] used the number of hints and attempts a student needed to answer a previous question to predict student performance. Gaining the capability to recognize students' need for assistance ahead of time by looking at students' pattern of actions could lead to more proactive interventions, such as identifying prerequisite skills, adapting pedagogical methodologies, or gaining insight on student problem solving methodologies.

With these in mind, we then ask: how do we determine when students will ask for help when using an ITS? On the exploratory level of model development, what information may be useful for developing models that forecast students' need for assistance? In this work, we define two models that use information on problem attempts and help requests used by students in the ASSISTments tutoring system: (1) *Attempt/Hint Count model* (AHC) makes use of information on the number of attempts and hints used by students on a question to predict the occurrence of a help request as the first action on the next problem, and (2) *Hint History model* (HH) makes use of the history of hint request as the first action in preceding questions to predict the occurrence of a help request as the first action on the next problem.

We utilized tabling methods to generate prediction values from the information used by each model. Tabling methods have been found to be effective alternatives for performing predictions using datasets and offer the advantage of being computationally inexpensive and easily expandable to leverage more features into simple models [2, 7].

2. DATASET

The data used in the analysis is from ASSISTments, an online tutoring system maintained at the Worcester Polytechnic Institute that provides tutorial assistance if students make incorrect attempts or ask for help [5]. The dataset is from released ASSISTments data that spans about five months within the 2012-2013 school year, containing

599,368 student log entries. More details about ASSISTments data can be accessed from: https://sites.google.com/site/assistmentsdata/how-to-interpret.

Analysis for the AHC model was done on problem logs with 1 to 5 attempts taken in answering problems, accounting for 98% of all data entries (585,926 rows). Problem entries with 3, 4, and 5 available hints (AvH) were used and these accounted for 70% of the data (415,895 rows). The resulting dataset contains 420 problem sets and 12,966 students, totaling to 299,968 entries. The resulting dataset was separated into problem groups that differed in the number of available hints to avoid comparing the hint request behavior of students who had more opportunities to hint against students with fewer opportunities to do so.

Problem Group	Problem Sets	Students	Dataset entries
3 AvH	285	11,402	169,100
4 AvH	224	10,282	111,754
5 AvH	60	4,724	19,114

Table 1. AHC dataset for each of the problem groups

For the HH model, we selected entries in the dataset where each student sequence had at least 4 rows. The student sequence is the sequence of problems that a student answered. Sequences had to at least have 4 rows for the HH model which looks at the history of hint use, 3 problems prior the next problem. The resulting dataset contained 279,925 entries with 555 problem sets and 12,429 students.

3. STUDENT ACTION MODELS

In ASSISTments, students exhibit varying behaviors when encountering problems: submitting an answer to a problem first ("attempting the problem"), asking for help (hint) first, asking for hints after an initial attempt, alternating between attempts and requests for hints, or continuously attempting a problem until a correct answer has been submitted. These behaviors have likewise been observed in [2].

3.1 Initial Experiments: AHC

The AHC prediction table maps the number of attempts and hints used to the probability that the student attempted or asked for a hint on the next problem. The probability is the percentage of students who asked for a hint on the next problem. Table 2 shows a sample prediction table from training data. Table 3 shows a matching scenario using Table 2. A value under *Hints Taken* in Table 2 such as 2/3 indicates that a student used 2 out of 3 available hints for the problem and values on the first column indicate the count of attempts. Five-fold cross validation was used to train and test the AHC model on the three problem groups. Problem set and student-level analyses were done to see whether the model generalizes across unseen problem sets and students.

3.2 Secondary Experiment: HH

For HH analysis, the prediction table was generated by using the percentage of hint use as first action in three previous problems. Table 4 shows a prediction table from training data. Column labels correspond to the number of times the first action was an attempt on the problem or a hint request. For example, 1H/2A indicates that in three prior problems, a total of 1 hint as first action and 2 attempts as first action were used. Counts of attempts and hints as first action were then generated for each column. In the table, for those who used a total of 2 hints and 1 attempt in three previous problems, there are 3330 instances of attempts and 1833 instances of hint requests as first action on the next problem. *% Hint* is the percentage of instances of hint use within the bin. Problem set and student-level five-fold cross validation was used to train and test the HH model.

Attempts Taken	Hints Taken 0 / 3	1 / 3	2 / 3	3 / 3
1	0.0211	0.1001	0.2213	0.4025
2	0.0261	0.0558	0.0747	0.1105
3	0.0237	0.0447	0.0737	0.0916
4	0.0363	0.0287	0.0743	0.0949
5	0.0132	0.0263	0.0857	0.0912

Table 2. AHC Prediction Table

Student	A_C	H_C	H_T	FANP
92677	1	0	3	0.0211
92680	2	3	3	0.1105

Table 3. Matching scenario using Table 2 (Note: A_C = Attempt Count, H_C = Hint Count, H_T = Hint Total, FANP = First Action Next Problem)

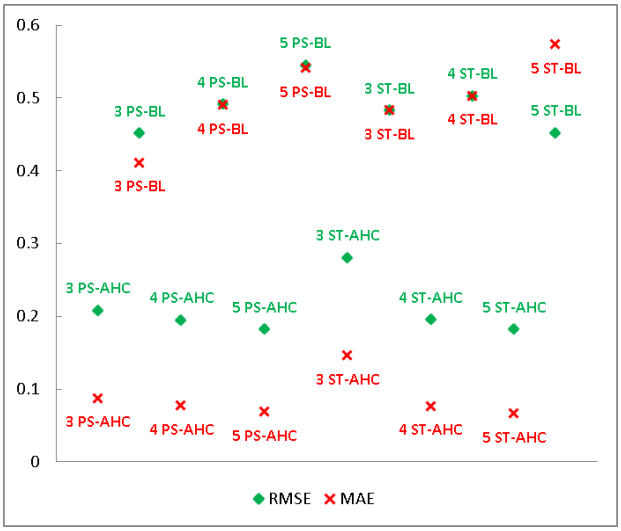
	Previous 3 First Action Hints / Attempts 0H / 3A	1H / 2A	2H / 1A	3H / 0A
# Attempt	111017	17219	3330	683
# Hint	5859	3254	1833	1663
% Hint	0.0501	0.1589	0.3550	0.7089

Table 4. HH Prediction Table

To analyze whether the number of history points affected the predictive power of HH, an additional analysis with four problems prior the next problem was done.

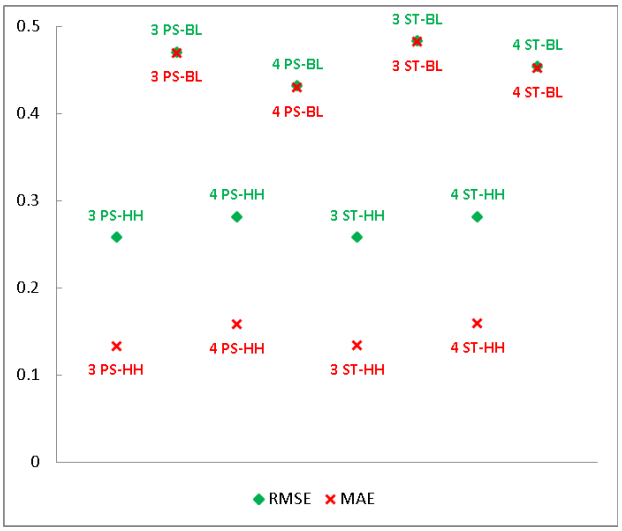
4. RESULTS AND DISCUSSION

The predictive performance of the AHC and HH models were evaluated using root mean squared error (RMSE), mean absolute error (MAE), and area under the ROC curve (AUC). Additionally, a naïve baseline (BL) model was generated for comparison, as we have found no other gold standard model for first-course-of-action prediction to compare our work with. The BL model uses the percentage of hint instances on the students' second action on all problems in the dataset. Table 5 shows a scenario for BL prediction. *Hint %* is the percentage of hint instances in the problem entries, which translates to a prediction on the students' first action on the next problem. If a student's second action on the current problem is a hint, the prediction for FANP is *Hint %*, otherwise, use *Attempt %*. The intuition for this is the hypothesis that students who have greater tendency to ask for hints on succeeding actions may most likely ask for hints in succeeding problems.



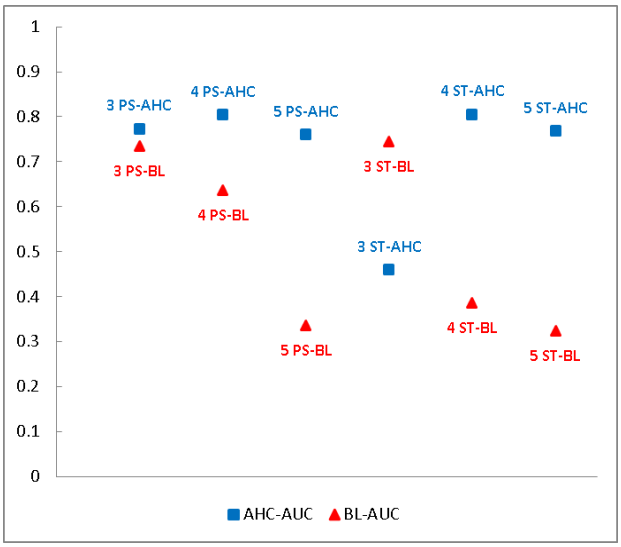
PS	3 AHC	3 BL	4 AHC	4 BL	5 AHC	5 BL
RMSE	0.2075	0.4506	0.1942	0.4910	0.1813	0.5445
MAE	0.0866	0.4104	0.0763	0.4899	0.0677	0.5403
ST	**3 AHC**	**3 BL**	**4 AHC**	**4 BL**	**5 AHC**	**5 BL**
RMSE	0.2799	0.4826	0.1945	0.5023	0.1811	0.4514
MAE	0.1452	0.4821	0.0758	0.5022	0.0653	0.5729

a. RMSE and MAE performance for AHC vs. BL across three problem groups (3, 4, and 5 available hints)



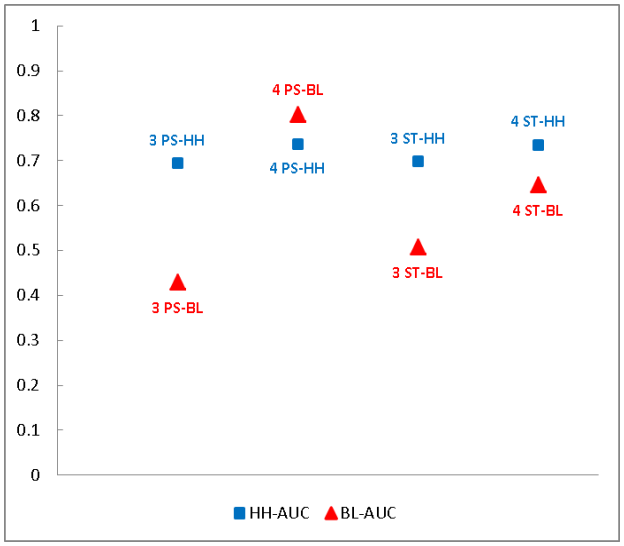
PS	3 HH	3 BL	4 HH	4 BL
RMSE	0.2574	0.4697	0.2809	0.4307
MAE	0.1327	0.4687	0.1572	0.4291
ST	**3 HH**	**3 BL**	**4 HH**	**4 BL**
RMSE	0.2573	0.4821	0.2808	0.4528
MAE	0.1328	0.4810	0.1580	0.4513

b. RMSE and MAE performance for HH vs. BL for 3 and 4 prior problems



PS	3 AHC	3 BL	4 AHC	4 BL	5 AHC	5 BL
AUC	0.7737	0.7332	0.8043	0.6338	0.7602	0.3338
ST	**3 AHC**	**3 BL**	**4 AHC**	**4 BL**	**5 AHC**	**5 BL**
AUC	0.4599	0.7419	0.8056	0.3841	0.7689	0.3223

c. AUC performance for AHC vs. BL across three problem groups (3, 4, and 5 available hints)



PS	3 HH	3 BL	4 HH	4 BL
AUC	0.6936	0.4298	0.7357	0.8026
ST	**3 HH**	**3 BL**	**4 HH**	**4 BL**
AUC	0.6989	0.5071	0.7355	0.6458

d. AUC performance for HH vs. BL for 3 and 4 prior problems

Figure 1. Problem set (PS) and student (ST) level RMSE and MAE performance for AHC, HH, and BL (a and b); Problem set and student level AUC performance for AHC, HH, and BL (c and d).

Problem entries	Hint Count: 2nd Action	Hint % (BL)	Attempt %
2200	852	0.3872	0.6127

Table 5. Sample scenario for BL prediction values

4.1 AHC Analysis

Problem set level findings for both AHC and BL are presented in Figure 1a. AHC consistently outperforms BL across all problem groups in both RMSE and MAE. Lower values for both metrics indicate better model fit. A reliability analysis to compare AHC with BL using a two-tailed paired t-test indicates that the findings are reliably different across all problem groups (p=0). The effectiveness of the model is likewise seen using the AUC metric (Figure 1c). AUC values closer to 1 indicate better model fit. It can be noted that AHC performance in all metrics are closely consistent, suggesting that the model is fairly generalizable across problems with varying numbers of hint availability. Predictive performance using student level analysis for problems with 4 and 5 available hints is fairly consistent across all three metrics; however, the model does not perform as well for problems with 3 available hints, suggesting that AHC may be used to predict the hint request behavior of unseen students, provided there is a high number of opportunities to ask for help. BL performance fails to improve as the number of available hints increase for both problem set and student-level analyses.

4.2 HH Analysis

A problem set level analysis of the HH model across the number of prior history points demonstrates that the HH model maintains a fairly consistent level of predictive performance across all three metrics. While HH significantly outperforms BL in MAE and RMSE, it is outperformed by the latter in AUC for 4 history points. This may be because the ordering of values in BL's predictions is not as close to the actual as those of HH. This situation rarely happens; we may have to try another dataset to confirm this behavior. On a student level analysis, HH outperforms BL across all values of first action prior history points (Figures 1b and 1d). A reliability analysis to compare HH with BL using a two-tailed paired t-test indicates that the findings are reliably different across all prior hint history with p=0. There is a consistency of results for all performance metrics for HH, while BL exhibits more prominent fluctuation in its results, suggesting that the HH model can be feasibly used to predict student hint request behavior for both unseen skills and unseen students, as well as across the number of first action history points with fair reliability.

5. CONTRIBUTION AND FUTURE WORK

Results of the experiments suggest that students' help request behavior can be feasibly predicted from data that are descriptive of student action information. While the methods in this study are a starting point in using action information, we feel that such initiatives are worth discussing for building up further studies in the field. The models provide utility for predicting when students will ask for help, using dataset information on problem attempts and help requests. Both models predicted students' first course of action when answering problems from an ITS with fairly consistent predictive performance and generalizability.

Future improvements to these models may include the accounting of patterns in student actions which may provide a rich source of information for possible prediction of need for assistance by students (partly explored here with the BL model). The dataset used contained other information including student response times and skill difficulty and exploiting these may provide further insight into factors of assistance need to aid in developing a proactive and effective early intervention framework. These models should be tested on other ITS datasets to determine whether these models are consistent across different datasets.

REFERENCES

[1] Corbett, A.T. and Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction 4, 4 (1994), 253–278.

[2] Duong, H., Zhu, L., Wang, Y. and Heffernan, N.T. (2013). A prediction model that uses the sequence of attempts and hints to better predict knowledge: "Better to attempt the problem first, rather than ask for a hint". In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 316-317.

[3] Mathews, M. and Mitrović, T. How Does Students' Help-Seeking Behaviour Affect Learning? In B.P. Woolf, E. Aïmeur, R. Nkambou and S. Lajoie, eds., Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2008, 363–372.

[4] Pavlik, P. I., Cen, H., and Koedinger, K. (2009) Performance Factors Analysis - A New Alternative to Knowledge Tracing. Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK, pp. 531-538.

[5] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K.R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek, M.A., and Rasmussen, K.P. (2005). The Assistment project: Blending assessment and assisting. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) Proceedings of the 12th Artificial Intelligence in Education, Amsterdam: ISO Press. pp. 555-562.

[6] Wang, Y. and Heffernan, N. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. The 24th International FLAIRS Conference.

[7] Wang, Q.Y., Kehrer, P., Pardos, Z. and Heffernan, N. Response Tabling– A simple and practical complement to Knowledge Tracing. KDD 2011 Workshop: Knowledge Discovery in Educational Data.

Modeling Students' Memory for Application in Adaptive Educational Systems

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

ABSTRACT

Human memory has been thoroughly studied and modeled in psychology, but mainly in laboratory setting under simplified conditions. For application in practical adaptive educational systems we need simple and robust models which can cope with aspects like varied prior knowledge or multiple-choice questions. We discuss and evaluate several models of this type. We show that using the extensive data sets collected by online educational systems it is possible to build well calibrated models and get interesting insight, which can be used for improvement of adaptive educational systems.

1. INTRODUCTION

Development of intelligent tutoring system and other adaptive educational systems is often focused on teaching mathematics, physics, and similar domains. The related research in student modeling is thus concerned mainly with modeling skill acquisition. Another interesting area, where adaptability is very useful, is learning of facts [8], particularly in domains with varied prior knowledge like vocabulary, geography, or human anatomy. In this context, modeling of students' memory is important.

Principles of human memory and their consequences for education have been extensively studied in psychology, e.g., [2, 5, 9, 10]. Models developed in the psychological research are not, however, easily applicable in practical implementation of adaptive practice. The purpose of models described in psychological literature is to describe and explain mechanisms of human memory, e.g., the spacing effect [9]. Experiments are done using lab studies under controlled setting, in areas with little prior knowledge, e.g., learning of arbitrary word lists, nonsense syllables, obscure facts, or Japanese vocabulary.

In the context of development of adaptive educational systems, our goal is more pragmatic – we do not need to capture all details of human memory, we need a model which will work well in an adaptive system. A model needs to provide good input for other modules of an adaptive system (e.g., question selection or open learner model). The specific context of our work is an adaptive application `slepemapy.cz` for learning geography [8].

Although we can afford to model memory in a simplified manner, we have to deal with issues like varied prior knowledge, multiple-choice questions (with possibility of guessing), and no control on when students use the system. Compared to laboratory studies online educational systems can easily collect much more extensive data (millions of answers), so we can employ machine learning techniques to find fitting models. Specifically, in our work we use this approach to detect the dependence of memory activation on time from previous answer. The standard approach [9] is to make an assumption about the functional form of such dependence. We learn the function from the data and it turns out to be an S-shaped function which cannot be represented symbolically in a straightforward way. The results also show that there are large differences between learning of facts even in a seemingly compact domain like geography. These results may be useful for improving the behaviour of adaptive educational systems.

2. MODELING

Before we go into the description of models, let us clarify the context of considered models. In previous work [8] we described a modular architecture for an adaptive practice of facts based on three modules: estimation of prior knowledge, estimation of current knowledge, construction of questions. Here we focus on improving the estimation of current knowledge by taking timing between answer into account.

Specifically, we assume the following input: for each student and repeatedly answered fact (e.g., a country in the case of our application), we have an initial estimate of the student's knowledge of the fact and data about a sequence of student's answers. For each answer we consider the correctness of the answer, the type of question (either open question or multiple-choice question with a specified number of options), and time from previous answer (in seconds). For estimating initial activation we use a variant of the Elo rating system [4, 13] as specified in [8]. For purpose of this work this estimation is treated as a black box.

As an output a model provides estimated probability that the next answer will be correct. This output can be used for the adaptive construction of questions (in such a way that

they have appropriate difficulty) [7, 8]. Model parameters can be also used for presenting feedback to students in the form of an open learner model.

2.1 Basic Approach

Student models of learning [3] most commonly use either a binary skill (a typical model of this type is Bayesian Knowledge Tracing) or a continuous skill with probability of correct answer specified by the logistic function of the skill. For modeling memory it is natural to use a continuous skill since memory is build gradually – as opposed, for example, to understanding or insight in mathematics, which may undergo sudden transition from unlearned to learned state as assumed by Bayesian Knowledge Tracing [1]. Modeling based on the logistic function was also previously used for modeling memory [9]. In the following we use the notion of *memory activation* instead of skill.

All models that we consider have the following basic form. Based on the data we estimate memory activation m. Probability that the next answer will be correct is estimated using a logistic function: $P(m) = \frac{1}{1+e^{-m}}$. In the case of multiple-choice question with n options the probability of correct answer is given by the shifted logistic function: $P(m) = \frac{1}{n} + (1 - \frac{1}{n})\frac{1}{1+e^{-m}}$. Note that this functional form is a simplification, since it does not consider the possibility that a student answers correctly by ruling out distractors.

2.2 Computing Memory Activation

A basic model applicable under the outlined approach is a simplified, one-dimensional variant of Performance Factor Analysis (PFA) [11] (originally PFA was formulated in terms of skills and vectors, as it uses multiple knowledge components). In this model the memory activation is given by a linear combination of an initial activation and past successes and failures of a student: $m = \beta + \gamma s + \delta f$, where β is the initial activation, s and f are counts of previous successes and failures of the student, γ and δ are parameters that determine the change of the skill associated with correct and incorrect answers. The basic disadvantage of this simple approach is that it does not consider the time between attempts; in fact it even ignores the order of answers (it uses only the summary number of correct and incorrect answers).

ACT-R model [9, 12] of spacing effects can be considered as an extension of this basic model. In this model the memory activation is estimated as $m = \beta + \log(\sum b_i t_i^{-d_i})$, where the sum is over all previous attempts, values t_i are the ages of previous attempts, values b_i capture the influence of correctness of answers, d_i is the decay rate, which is computed by recursive equations [9]. The model also includes additional modifiers for treating time between sessions. The focus of the model is on modeling the decay rate to capture the spacing effect. Studies using this model [9, 12] did not take into account the probability of guessing and variable initial knowledge of different items (initial activation was either a global constant or a student parameter). In the current work we focus on these factors and for the moment omit modeling of spacing effects.

Another possible extension [8] of the basic PFA model is to combine it with some aspects of the Elo rating system [4, 13]; in the following we denote this version as PFAE (PFA Elo/Extended). The estimated memory activation is updated after each answer as follows:

$$m := \begin{cases} m + \gamma \cdot (1 - P(m)) & \text{if the answer was correct} \\ m + \delta \cdot P(m) & \text{if the answer was incorrect} \end{cases}$$

To include the timing information into this model, we can locally increase the memory activation for the purpose of prediction, i.e., instead of $P(m)$ to use $P(m + f(t))$, where t is the time (in seconds) from the last attempt and f is a *time effect function*. As m denotes memory activation, the value $f(t)$ corresponds to temporal increase in memory activation due to (short) time from previous exposure of an item.

It is natural to use as a time effect function some simple analytic function, but analysis of our data suggests that this approach does not work well. Figure 1 shows calibration analysis for two time effect functions: $f(t) = \frac{w}{t}$ (used in previous work [8]) and $f(t) = 1.6 - 0.1 \log(t)$ (the functional form is based on [9] and fitted to data). We see that neither of these functions leads to well calibrated predictions. Since we were not able to find a simple time effect function that would provide a good fit, we represent the function $f(t)$ as a staircase function with fixed bounds $\vec{b}$ and values $\vec{v}$ which we learn from the data:

$$f(t) = \begin{cases} v_i & \text{if } b_i \leq t < b_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

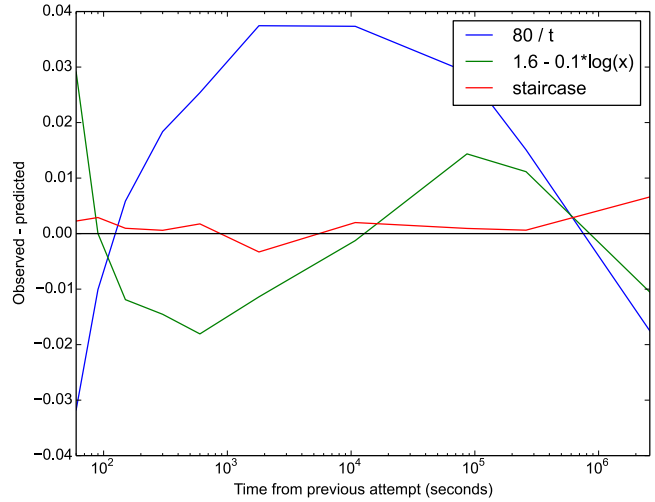


Figure 1: Calibration for the PFAE model with different time effect functions – the y axis shows difference between observed frequency of correct answers and average prediction.

3. EXPERIMENTS

We report experiments with the PFAE model with time effect function. For evaluation we used data from an online system for practicing geography [8] (`slepemapy.cz`). Data were filtered to include only students with at least 20 answers, items (places) with at least 40 answers, and we consider only sequences where a student answered at least 3 questions about an item. For experiments we divided the data into 10 sets, each containing 52,190 sequences of answers.

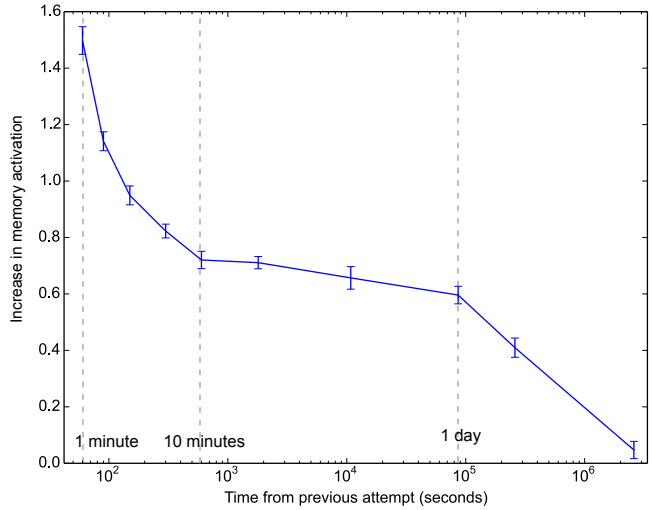


Figure 2: Time effect function – average from 10 independent data sets, error bars show standard deviations of parameter estimates.

3.1 Model Parameters

As the fixed bounds used in the staircase representation of time effect function we have chosen the following values: 0, 60, 90, 150, 300, 600, 1800, 10800, 86400, 259200, 2592000. These values were chosen to be easily interpretable (e.g., 30 minutes, 1 day) and at the same time to have reasonably even distribution of data into individual bins.

The model has the following parameters which we need to estimate from the data: update constants γ, δ and the vector $\vec{v}$ representing the time effect function. To estimate these parameters we use a gradient descent. To evaluate stability of parameter estimates we computed the parameter values for the 10 independent data sets. The results show that the obtained parameters are very stable: $\gamma = 2.290 \pm 0.042$, $\delta = -0.917 \pm 0.018$; values $\vec{v}$ for the representation of time effect function are depicted in Figure 2.

Since our data set is large and parameter estimates are stable, we can afford to do more detailed analysis. Figure 3 shows fitted time effect functions and γ, δ values when the parameters are fitted using only part of the data. Figure 3 A shows that there is quite large difference between parameter values for cases with high and low prior knowledge. This suggests possible improvement to the PFAE model – not just by including more parameters, but also by changing its functional form. However, prior knowledge is not the only factor that plays role. Figure 3 B shows fitted parameters for several types of places. In all of these cases the prior knowledge is low, yet there are still large differences between fitted parameters values. These parameters may contain useful information about students' learning in particular parts of the domain, e.g., data in Figure 3 B illustrate that it is easier to learn states of Germany than provinces of China.

In the case of countries we have enough data to perform parameter fitting for individual places. In this case we fix the time effect function (as learned on the whole data set and reported in Figure 2) and we learn only the γ, δ parameters on

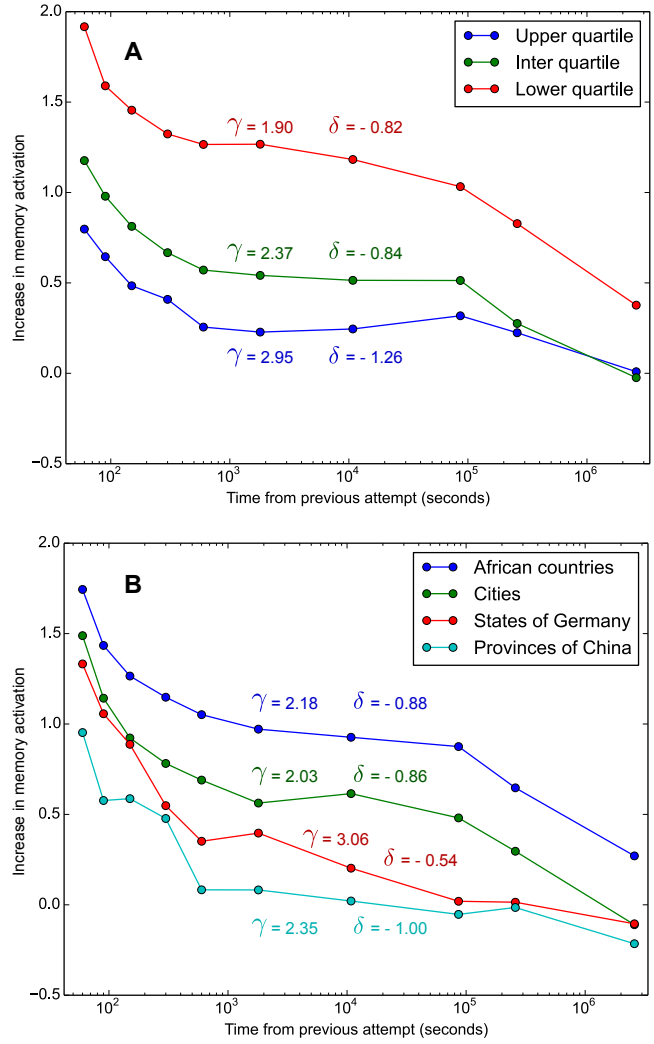


Figure 3: Time effect function and γ, δ parameters fitted to filtered data: A) by estimated prior knowledge, B) by the type of a place.

data for a single place. We use only places for which we have at least 1300 students answering at least 3 questions. The fitted parameter γ is has an interpretable meaning "how easy it is to remember a country". Examples of countries with high γ (>3.3): Western Sahara, Southern Sudan, Vietnam, Egypt, Somalia; countries with low γ (< 1.7): Bulgaria, Romania, Serbia, Moldova. Note that the reported results are clearly dependent on the origin of students using the system – in our case mostly Czech students.

3.2 Accuracy of Predictions

Table 1 show comparison of several model variants with respect to three common performance metrics [14]: root mean square error (RMSE), log-likelihood (LL), and area under the ROC curve (AUC). The results show averages from 10 runs on different training/testing sets. The results are consistent over the three metrics and show that the PFAE models brings quite large improvement over the PFA model. Differences between variants of the PFAE model due to the used time effect function are statistically significant, but other-

Table 1: Comparison of models with respect to three performance metrics.

model	time effect	RMSE	LL	AUC
PFA	–	0.3593	-106517	0.719
PFA	$80/t$	0.353	-103441	0.7195
PFAE	$80/t$	0.3377	-94454	0.757
PFAE	$1.6 - 0.1\log(t)$	0.3367	-93987	0.7591
PFAE	staircase	0.3363	-93642	0.7614

wise rather small. Individual predictions are actually highly correlated (correlation coefficient around 0.97).

4. DISCUSSION

We have evaluated several variants of a model of memory activation in the context of adaptive practice of facts. We proposed a model which incorporates the effect of time from previous answer by a general staircase function, which is learned from data (as opposed to assuming a specific symbolic form of the function). The model is better calibrated than other studied models and provides slightly better predictions. More importantly, the model is simple, parameters are easy to learn from data and robust. The learned function also provides interesting insight into students memory in the particular application – there is fast decrease in memory activation within the first 10 minutes, then the effect is nearly steady for 1 day, after that the activation decreases again.

By performing fine-grained analysis of the data, it is possible to use the model parameters to determine items that are easy or difficult to remember. Such results may be useful for improvement of educational systems, e.g., by offering mnemonics for difficult to remember facts, or by changing the adaptive selection of questions to prefer easy to remember facts at the beginning of a session. Specifically, results reported in Figure 3 suggest that different adaptive behaviour may be useful for learning African countries and provinces of China.

A possible limitation of this study is that the used data do not come from a properly designed and controlled experiment, but from an adaptive system which uses a student model to choose questions [8]. This may potentially cause a bias in the performed analysis. Although it seems unlikely that the reported results would be significantly influenced by this data source, feedback loops between student models and data collection deserve attention [6].

Another simplification of the current work is that we do not consider the feedback provided by the used system when a student answers incorrectly. This feedback clearly has impact on memory activation of the selected wrong answer. This raises a more general question: What is more important for the practical development of adaptive educational systems – proper treatment of principal issues (e.g., spacing effect) or incorporation of practical features into the model (e.g., effect of wrong answers)?

Acknowledgement

The author thanks Vít Stanislav and Jan Papoušek for their work on the `slepemapy.cz` project and for their assistance with the data.

5. REFERENCES

[1] 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.

[2] Peter F Delaney, Peter PJL Verkoeijen, and Arie Spirgel. Spacing and testing effects: A deeply critical, lengthy, and at times discursive review of the literature. *Psychology of learning and motivation*, 53:63–147, 2010.

[3] 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.

[4] Arpad E Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[5] Jeffrey D Karpicke and Henry L Roediger. Repeated retrieval during learning is the key to long-term retention. *Journal of Memory and Language*, 57(2):151–162, 2007.

[6] Juraj Nižnan, Jan Papoušek, and Radek Pelánek. Exploring the role of small differences in predictive accuracy using simulated data. 2015. Submitted.

[7] Jan Papoušek and Radek Pelánek. Impact of adaptive educational system behaviour on student motivation. In *Proc. of Artificial Intelligence in Education (AIED)*, 2015.

[8] 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.

[9] Philip I Pavlik and John R Anderson. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29(4):559–586, 2005.

[10] Philip I Pavlik and John R Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

[11] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *Proc. of Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[12] Philip Pavlik Jr, Thomas Bolster, Sue-Mei Wu, Ken Koedinger, and Brian Macwhinney. Using optimally selected drill practice to train basic facts. In *Intelligent Tutoring Systems*, pages 593–602. Springer, 2008.

[13] Radek Pelánek. Time decay functions and Elo system in student modeling. In *Educational Data Mining (EDM)*, pages 21–27, 2014.

[14] Radek Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 2015. To appear.

Social Facilitation Effects by Pedagogical Conversational Agent: Lexical Network Analysis in an Online Explanation Task

Yugo Hayashi
Ritsumeikan University
56-1 Kitamachi, Toji-in, Kita-ku
Kyoto, 603-8577, Japan

yhayashi@fc.ritsumei.ac.jp

ABSTRACT
The present study investigates web-based learning activities of undergraduate students who generate explanations about a key concept taught in a large-scale classroom. The present study used an online system with Pedagogical Conversational Agent (PCA), asked to explain about the key concept from different points and provided suggestions and requests about how to make explanations, and gave social facilitation prompts such as providing examples by other members in the classroom. A total of 314 learner's text based explanation activities were collected from three different classrooms and were analyzed using the social network analysis methods. The main results from the lexical analysis show that those using the PCAs with social feedback worked harder to use more various types of explanations than those without such feedback. Future directions on how to design online tutoring systems are discussed.

Keywords
Online tutoring; Explanation activities; Social Facilitation; Lexical Network Analysis.

1. INTRODUCTION
Studies on designing intelligent tutoring systems, such as Pedagogical Conversational Agents (PCAs), which autonomously engage in learning activities, have suggested its effective use for learning, much like a human tutor [12, 9, 1]. Still, few studies empirically investigate the use of such technology for large numbers of students in a class and investigate the learner's cognitive processes. The present study investigated the unique designs of the user interface for learners that use an online tutoring system guided by a PCA in three different types of classes. The study especially focused on the use of PCAs in a concept-explanation activity task, where the PCA asked several questions for explanation and provided feedback such as social information about other members who were engaging in the task. We focused on how such feedback can increase the learner's explanation behaviors during such activities.

1.1 Facilitating explanation activities using PCAs
Studies on collaborative problem solving in the field of cognitive science reveal how concepts are understood or learned [3, 5]. Studies have shown that asking reflective questions for clarification to conversational partners is an effective interactional strategy to gain a deeper understanding of a problem or a concept [15, 16]. It has also been demonstrated that the use of strategic utterances, such as asking for explanation or providing suggestions, can stimulate reflective thinking and meta-cognition involved in understanding a concept. Based on these theories, there have been many attempts in the learning sciences to use such methods in classrooms [17, 13]. However, in an actual pedagogical situation, as in a large classroom, it is often difficult for one teacher to monitor learners and supervise their explanations. Recent studies [2, 11] have shown that the use of conversational agents that act as educational companions or tutors can facilitate learning process. Study [10] have shown that using PCAs that provide suggestions about how to make effective explanations can facilitate better motivation and improve task performance. Moreover, in a series of studies by the author, it is shown that the use of PCAs can provide affective feedback and facilitate better outcomes [7, 8, 6]. More specifically, the results show that PCAs with positive emotion motivates the learners to work harder compared to those without any emotional expressions. In this report, the author further investigated the effects of using such PCAs in an online explanation task. The study focused on a classroom of more than one hundred students who were using an online explanation task, where individuals made explanations to the PCA on a one-on-one basis, as an after school work activity. In such activity, the PCA will play the role of questioner and ask the student to explain about the key concept. The learners were students enrolled in a psychology class where their task was to make explanations about a key concept taught in their class, as an after class exercise.

1.2 Using social facilitating effects
One of the important factors that strongly influence human behavior in groups is the effect of the social influence produced by other members. Studies in social psychology have suggested that work efficiency is improved when someone is watching a person, i.e., the presence of an audience facilitates the performance of a task. The impact that an audience has on a task-performing participant is called the "audience effect." Another relevant concept on task efficiency is called "social facilitation theory" [19]. The theory claims that people tend to do better on a task when they are doing it in the presence of other people in a social situation; it implies that personal factors can make people more aware of social evaluation.

Coming back to the present study, even though the students made explanations about a concept to the PCA in a one-on-one situation, it was extremely important that they were aware that they were working in a social situation. Studies in media-psychology have provided much evidence that people lack social awareness in computer-mediated communication, compared to face-to-face communication [4]. Thus, it is effective to give information about the awareness of other learners online and create social

facilitations to make the learners become more active. One of the strong points of using online learning environments is that they are able to collect a huge amount of data from learners. A large database of dialogues of explanation texts may be reused for prompting hints or giving examples to learners who make explanations. It is also effective to provide information about the members who are working on the explanation task in real time or non-real time. If such kinds of feedback are used in online tutoring systems, it may facilitate learners' social awareness, and motivate their explanation activities.

Given all this, the present study investigated the effects of PCAs, which provide information about "other members", along with suggestions and comments about their explanations. The goal of the study is to investigate the how the quality of the learners explanations may change due to the facilitations from a PCA which encourages them to actively explain about key terms that were taught in class. The present study will use social network analysis method to capture the dynamics of diverse explanations during the online task. Unlike standard text analysis methods calculating the frequency of single important key terms that appear in the text, this method enables to detect different key terms that appear simultaneously in one explanation made by the learner. If the learner meets the expectations from the PCA, where it asks the learner to explain the key from various perspectives, different types of key terms should be used during their activity.

2. Method

The study was conducted in three large classes, each consisting of more than hundred students. We constructed an online web system that let learners make text-based explanations about key concepts taught in a psychology class. Students in an undergraduate psychology class used the system, and participated as part of their homework. A total of 30 different key terms (e.g., Gestalt, long-term memory, cognitive dissonance) were selected from the class and randomly assigned to each of the learners based on their IDs. On using the system, they were guided by a PCA that (1) instructed them on what to explain, (2) provided meta-cognitive suggestions, and (3) gave examples about how other members in the classroom made explanations.

2.1 Tutoring system for the experiment

A web-based tutoring system was developed only for the experiment using a web server, a database, and rule-based scripts. It was managed as a member-only system, and learners were required to login to the system for use. As mentioned in the previous section, each student was assigned to work on one randomly selected key term. As they logged into the system, a PCA appeared on the screen and stated the selected key concept, and gave him/her questions about how to explain it. The task was comprised by 17 trials with two major steps in each trial as follows: (a) text-input and, (b) feedback from the PCA.

On the first (Trial 1) and the final trials (Trial 17) the PCA asked the learner to input freely regarding whatever they knew about the key concept. These are taken as pre- and post- tests were they can freely input the messages as a free recall test. Through the 2nd and 16th trials, the learners were given specific questions about what to explain about the keyword. For example, the PCA may ask a series of question such as "How can it be used", "What is it similar to", or "In what period of time you use it" etc. These trials are considered as the explanation/training phase. The PCA also encourages the learner to think on their own way and input individual unique explanations.

On each trial, they were asked to do the following: (1) input explanations and click on the next button, (2) read the provided meta-suggestions from the PCA to make effective explanations, and depending on the experimental condition (explained in the next section), it provided information about other members who also responded for the given key concept.

To facilitate the social presence of the other members and make learners to think in their own way, the study uses two types of prompts. First, the utterances of other learners who had already inputted into the system were used. These messages were presented along with the initials of the person who answered the explanation. This enabled them to be aware how many in the class were working on the same key term. The utterances of other group members were only shown after the learner inputted his/her answers, and so the leaner couldn't simply copy and paste other's explanations during their trial.

2.2 Experiment design and learners

The experiment was conducted in three classes where each class was assigned to an experimental condition. In one class (the baseline condition), all learners were assigned to use PCAs without any social awareness functions or examples of other learners. The PCA only provided back-channel feedback and gave meta-suggestions about how to make explanations more effectively (e.g., Try to think from various viewpoints). These suggestions were compiled from a previous study [7]. In another class (the example condition), the learners were assigned to use the PCAs with additional functions, which provided examples of answers inputted by other members. The third class (the example+ condition) was assigned to those in the example condition with PCAs with additional functions. In other words, they were presented with examples with explanations of others, plus information about the number of members who were assigned to work on that key concept. There were 105 Japanese undergraduates (55 males, 50 females, mean age = 18.26 years) in the baseline condition. In the example condition, there were 105 Japanese undergraduates (55 males, 50 females, mean age = 18.46 years). Finally, in the example+ condition, there were 104 undergraduates (52 males, 52 females, mean age = 18.35 years).

3. RESULTS

3.1 Lexical Network Analysis

The text analysis was comprised by several steps such as (1) morphologically analyzing the text data, (2) developing a dictionary database using a thesaurus, and (3) conducting lexical network analysis to understand the usage of variety of different words during their final explanation. Recently, such social network analysis method is adopted to investigate the usage of important words in collaborative learning [8, 16].

3.1.1 Preprocessing

The recorded texts were broken down into morphemes with the Japanese morphological analysis tool MeCab (Java Sen port: *http://mecab.sourceforge.net* (accessed April 2015)). The objective of the first stage of the analysis was to extract the most frequent morphemes, such as the nouns and verbs through all learners textual inputs. 105,488 morphemes were collected and the most 28 frequent words were chosen as important words for explanations. Those were labeled based on the thesaurus

dictionary database such as: 'presence', 'causal', 'relations', 'actions', 'thought', 'matters', 'case', 'conclude', 'understand', 'analogy', 'predict', 'logic', 'reason', 'hypothesis', 'convergence', 'explanation', 'intention', 'theory', 'relative', 'knowledge', 'explicate', 'transform', 'opposition', 'compliment', 'compare', 'inevitability', 'method', and 'reason' [14].

Additionally, based on the semantic hierarchical structure of the thesaurus, new keywords were added to the dictionary database that were related to the 28 keywords. This was done to capture all the semantically related words to these keywords. As a result, 2,722 new words that have relative meanings to the keywords were registered into the semantic dictionary database.

3.1.2 Network Analysis

Using the semantic dictionary database as training data set, the learners textual inputs were further analyzed. For each trial input, the number of appearing semantic keywords in the dictionary were counted. The data of these semantic key words were then analyzed by adopting the social network analysis method. This method was used to analyze the co-occurrence between keywords, i.e. *capturing the diversity of the types of words that were used during one explanation*. The network was developed based on a bipartite graph of keywords x explanations(trials). Since the PCA provided various questions and enforced them to explain uniquely along with their social feedbacks during their explanation activities(trial 2 to 16), their achievements should be reflected to their explanation activities. Learners should use more different types of key terms in the example+ condition since they are facilitated more strongly to take different perspectives by mentioning about other group members presence. Each node in a network was represented as the semantic category of the keyword that was frequently used during their explanation. The threshold of a node(semantic keyword) determining as frequently used or not was defined based on the comparison by the average of other nodes. The threshold of a node *n* was determined as follows:

$$\theta = \begin{cases} 1(n \geq \bar{n}) \\ 0(n < \bar{n}) \end{cases} \tag{1}$$

On investigating the differences between conditions and over time, the number of links connecting each nodes were calculated. The following equation represents the amount of density where *n* stands for the number of nodes and *l* stands for the number of links:

$$d = \frac{l}{n(n-1)} \tag{2}$$

Table 1 shows the quantitative results of the lexical network analysis. The results suggest that at the pre-test (1st trial), learners had only few connections between nodes, thus indicating that the variations of words were few in terms of semantic categories. On the post-test (17th trial), the connections of nodes increased due to conditions. This shows that learners used more variety of words during explanations in the post-test(17th trial) example+ condition(0.27) than example(0.24) and baseline(0.15) conditions. The results gives us a clear vision of the dynamics of explanations they gave to the agent differ due to the conditions using more social awareness designs.

Table 1. The score of density of each conditions performed by the lexical network analysis.

Conditions	Pre (1st trial)	Post (17th trial)
baseline	0.07	0.15
example	0.06	0.24
example+	0.06	0.27

The analysis above shows that learners were using more different key terms at the same time in each trial. However it lacks in evidence rather if they tried to use different key terms in their post test compared from those in the pre-test. They might have simply used the same words they inputted from their first trial. It is it is important in this learning context that to know if they changed their phrases or tried to use more sophisticated words from the initial state of the explanation activity. Therefore, additional analysis was conducted to investigating the network similarity between the pre(1st) and post(17th) trial. The following correlation index was adopted on calculating the similarity between the two networks.

$$c = \frac{\sum_{i=1}^{784}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{784}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{784}(b_i - \bar{b})^2}} \tag{3}$$

a and *b* stands for the number of nodes in the bipartite graph each pre- and post-test respectively. Figure 1 indicates the results of *c* for each condition.

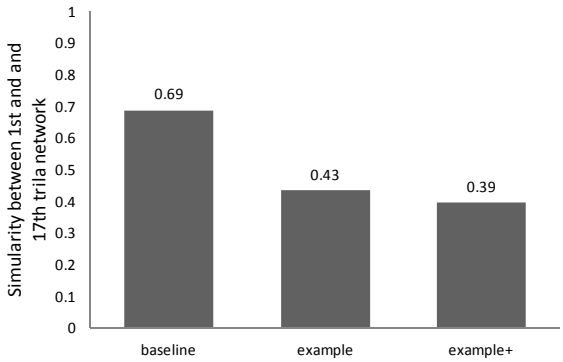


Figure 1. Results of similarity between the pre(1st) and post(17th) trial in each condition

The results indicate that learners in the baseline condition used more similar words from the pre-test on their final post-test explanations(0.69). On the other hand, learners in the example+ and example condition shows that they were using more different key terms compared to those from those in the 1st trial(0.43, 0.39 respectively).

The analysis from the series of analysis indicates that learners with social facilitation (1) used more different key terms simultaneously in their final explanation activities, and (2) those were different from those in the initial explanation activities. This analysis captures a new view from the study of [8] where it did not investigate the changes of the network over time.

4. DISCUSSION AND CONCLUSION

The present study investigated the use of PCAs in an online explanation activity where students were required to make explanations about a key concept. The focus here was to investigate the effects of social facilitations over time, using a large scale database collected during the online explanation task. These social facilitations were provided through a PCA during the learner's explanation activities and they were to enhance the co-presence of other classmates and motivate their activities by encouraging them. In the experiment, students enrolled in three psychology classes used an online explanation system and made explanations to the PCA. They also received comments on how to make effective explanations along with social feedbacks of other classmates. The results of the text analysis show that learners tend to input more important messages simultaneously in the final trial compared to the first trial when they received feedback about other group members (example and example+ condition). This indicates that this type of social feedback can motivate learners to work harder and facilitate effective explanation over time. An interesting point is that even though all the students were told that their answers would not be graded, they still tried harder when they were shown some of the other members' activities. This shows that the effects of the "audience" and "social facilitation" are quite strong in such situations. The results can be interpreted that the situation given to the learner are useful to make the learners aware that their messages could be seen by other in-group members and thus this might have made them work harder in their activities. Another interpretation is that showing others' comments might have allowed learners to avoid negative feelings and thoughts, such as he/she might have inputted something very out of line. As explained earlier in this paper, novice learners have difficulty making explanations to others [5]. Thus, it may be assumed that learners in the baseline condition experienced negative feelings, worrying that they were making mistakes about the text. On the other hand, the use of the examples and the social contexts in the example and example+ conditions may have eased such negative feelings, and thus, increased self-confidence compared to the baseline condition. This study provided implications about how to design effective online tutoring systems, incorporating PCAs with information about other working members, thus providing social facilitation.

5. ACKNOWLEDGMENTS

This work was supported (in part) by 2012 KDDI Foundation Research Grant Program and the Grant-in-Aid for Scientific Research (KAKENHI), The Ministry of Education, Culture, Sports, Science, and Technology, Japan (MEXTGrant), Grant No. 25870910.

6. REFERENCES

[1] Baylor, A.L. and Kim, Y. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education 15*, 1 (2005), 95-115.

[2] Baylor, A.L. and Ryu, J. The API (Agent Persona Instrument) for Assessing Pedagogical Agent Persona. In *Proc. World Conference on Educational Multimedia, Hypermedia and Telecommunications*, D. Lassner and C. McNaught (eds.) (2003), 448-451.

[3] Chi, M., Leeuw, N., Chiu, M. and Lavancher, C. Eliciting self-explanations improves understanding. *Cognitive Science 18*, 3, (1994), 439-477.

[4] Clark, H.H. and Brennan, S.E. Grounding in communication. In B. L. Resnick, M. R. Levine, and D. S. Teasley (Eds.), *Perspectives on socially shared cognition*, APA Press, 1991, 127-149.

[5] Graesser, A. and McNamara, D. Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist 45*, 4 (2010), 234-244.

[6] Hayashi, Y. Learner-support agents for collaborative interaction: A study on affect and communication channels. In *Proc. 10th International Conference on Computer Supported Collaborative Learning* (2013), 232-239.

[7] Hayashi, Y. On pedagogical effects of learner support agents in collaborative interaction. In *Proc. 11th International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science* (Cerri, A. S. and Clancey, B. eds.), Springer-Verlag, 7315 (2012), 22-32.

[8] Hayashi, Y. Explanation activities with a pedagogical agent in an online task: Lexical network analysis. In *Proc. CHI 2015 Works-in-Progress*, 1457-1460.

[9] Heidig, S. and Clarebout, G. Do pedagogical agents make a difference to student motivation and learning? *Educational Research Review 6*, 1 (2011), 27-54.

[10] Holmes, J. Designing agents to support learning by explaining. *Computers & Education 48*, 4 (2007), 523-547.

[11] Kim, Y., Baylor, A.L. and Shen, E. Pedagogical agents as learning companions: The impact of agent emotion and gender. *Journal of Computer Assisted Learning 23*, 3 (2007), 220-234.

[12] Kumar, R. and Rose, C. Architecture for building conversational agents that support collaborative learning. *IEEE Transactions on Learning Technologies 4*, 1 (2011), 21-34.

[13] Miyake, N. Constructive interaction and the interactive process of understanding. *Cognitive Science 10*, (1986), 151-177.

[14] National Institute for Japanese Language and Linguistics, *Bunruigoihyou*, Dainippon publications, (2004). (In Japanese)

[15] Okada, T. and Simon, H. Collaborative discovery in a scientific domain. *Cognitive Science 21*, 2 (1997), 109-146, 1997.

[16] Oshima, J., Matsuzawa, Y., Oshima, R., Chan, C. K. K., & van Aalst, J. Social Network Analysis for Knowledge Building: Establishment of Indicators for Collective Knowledge Advancement. In J. van Aalst, K. Thompson, M. J. Jacobson, & P. Reinmann (Eds.), *The Future of Learning: Proceedings of the 10th International Conference of the Learning Sciences (ICLS2012)*, 2 (2012), 465-466.

[17] Salomon, G. *Distributed Cognition: Psychological and Educational Considerations*. Cambridge University Press, New York, USA, 2001.

[18] Shirouzu, H., Miyake, N. and Masukawa, H. Cognitively active externalization for situated reflection. *Cognitive Science 26*, 4 (2002), 469-501.

[19] Zajonc, R.B. Social facilitation. *Science 149*, (1965) 271-274.

Personalized Education; Solving a Group Formation and Scheduling Problem for Educational Content

Sanaz Bahargam, Dóra Erdős, Azer Bestavros, Evimaria Terzi
Computer Science Department, Boston University, Boston MA
[bahargam, edori, best, evimaria]@cs.bu.edu

ABSTRACT

Wether teaching in a classroom or a Massive Online Open Course it is crucial to present the material in a way that benefits the audience as a *whole*. We identify two important tasks to solve towards this objective; (1.) group students so that they can maximally benefit from peer interaction and (2.) find an optimal schedule of the educational material for each group. Thus, in this paper we solve the problem of team formation and content scheduling for education. Given a time frame d, a set of students $\mathbf{S}$ with their required need to learn different activities $\mathbf{T}$ and given k as the number of desired groups, we study the problem of finding k group of students. The goal is to teach students within time frame d such that their potential for learning is maximized and find the best schedule for each group. We show this problem to be NP-hard and develop a polynomial algorithm for it. We show our algorithm to be effective both on synthetic as well as a real data set. For our experiments we use real data on students' grades in a Computer Science department. As part of our contribution we release a semi-synthetic dataset that mimics the properties of the real data.

Keywords

Team Formation; Clustering; Partitioning; Teams; MOOC

1. INTRODUCTION

Many work has been dedicated on how to improve students' learning outcome. We recognize two substantial conclusions; first, the use of personalized education. By shaping the content and delivery of the lessons to the individual ability and need of each student we can enhance their performance([6, 11, 12]. Second, grouping students; working in teams with their peers helps students to access the material from a different viewpoint as well [7, 4, 13, 1]. In this paper we study the problem of creating personalized educational material for teams of students by taking a computational perspective. To the best of our knowledge we are the first to formally define and study the two problems of team formation and personalized scheduling for teams in the context of education. We present a formal definition for these problems, study their computational complexity and design algorithms for solving them. In addition, we also apply our algorithms to a real dataset obtained from real students. We make our semi-synthetic dataset `BUCSSynth`, generated to faithfully mimic the real student data available on our website.

Related Work: Besides the work on improving students learning outcome, related problems have also been studied in computer science. Topics of interest are team formation [2, 3, 9, 10] and scheduling theory, see [5] for an overview.

2. PRELIMINARIES

We model a student's learning process by a sequence of topics that she learns about. In this sequence topics may appear multiple times, and repetitions of a topic may count with different weights towards the overall benefit of the student. Let $\mathbf{S} = \{s_1, s_2, \ldots, s_n\}$ be a set of students and $\mathbf{T} = \{t_1, t_2, \ldots, t_m\}$ be a set of topics. We assign topics to d timeslots,a *schedule* $\mathcal{A}$ is a collision free assignment of topics to the timeslots. $\mathcal{A}$ can be thought of as an ordered list of (possible multiple occurrences) of the topics. For a topic $t \in \mathbf{T}$ the tuple $\langle t, i\rangle$ denotes the ith occurrence of t in a schedule. The notation $\mathcal{A}[r] = \langle t, i\rangle$ refers to the tuple $\langle t, i\rangle$ that is assigned to timeslot r in $\mathcal{A}$.

For student $s \in \mathbf{S}$ and topic $t \in \mathbf{T}$ the *requirement* $\texttt{req}(s, t)$ is an integer depicting the number of times s needs to learn about t to master its content. We assume that for the first $\texttt{req}(s, t)$ repetitions of t there is some benefit to s from every repetition of t, but for any further repetition there is no additional benefit to s. We call $\mathbf{b}(s, \langle t, i\rangle)$ (Equation (1)) the *benefit* of s from hearing about t for the ith time.

$$\mathbf{b}(s, \langle t, i\rangle) = \begin{cases} \frac{1}{\texttt{req}(s.t)} & \text{if } i \leq \texttt{req}(s, t) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Note that for ease of exposition, we assume that all repetitions of t before $\texttt{req}(s, t)$ carry equal benefit to s. However, the definition and all of our later algorithms could easily be extended to use some other function $\mathbf{b}'(s, \langle t, i\rangle)$.

Given the benefits $\mathbf{b}(s, \langle t, i\rangle)$ there is a natural extension to define the benefit $\mathbf{B}(s, \mathcal{A})$ that s gains from schedule $\mathcal{A}$. This benefit is simply a summation over all timeslots in $\mathcal{A}$,

$$\mathbf{B}(s, \mathcal{A}) = \sum_{r=1}^{d} \mathbf{b}(s, \mathcal{A}[r]) \tag{2}$$

3. THE GROUP SCHEDULE PROBLEM

Given a group of students $P \subseteq \mathbf{S}$ our first task is to find an optimal schedule for P. That is, find a schedule to maximize the *group benefit* $\mathbf{B}(P, \mathcal{A})$ that group P has from $\mathcal{A}$ (Equation (3)).

$$\mathbf{B}(P, \mathcal{A}) = \sum_{s \in P} \sum_{r=1}^{d} \mathbf{b}(s, \mathcal{A}[r]) \tag{3}$$

We call this the GROUP SCHEDULE problem (problem 1).

PROBLEM 1 (GROUP SCHEDULE). *Let $P \subseteq \mathbf{S}$ be a group of students and $\mathbf{T}$ be a set of topics. For every $s \in \mathbf{S}$ and $t \in \mathbf{T}$ let $\texttt{req}(s,t)$ be the requirement of s on t given for every student-topic pair. Find a schedule $\mathcal{A}_P$, such that $\mathbf{B}(P, \mathcal{A}_P)$ is maximized for a deadline d.*

The `Schedule` algorithm. We first give a simple polynomial time algorithm, $\texttt{Schedule}(P, d)$ (Algorithm 1), to solve problem 1. `Schedule` is a greedy algorithm that assigns to every timeslot an instance of the topic with the largest marginal benefit. We say that the *marginal benefit*, $\mathbf{m}(P, \langle t, i \rangle)$, from the ith repetition of t (thus $\langle t, i \rangle$) to P is the increase in the group benefit if $\langle t, i \rangle$ is added to $\mathcal{A}$. The marginal benefit can be computed as the sum of benefits over all students in P as given in Equation (4).

$$\mathbf{m}(P, \langle t, i \rangle) = \sum_{s \in P} \mathbf{b}(s, \langle t, i \rangle) \tag{4}$$

The `Schedule` algorithm is an iterative algorithm with d iterations that in every iteration appends a topic to the schedule $\mathcal{A}_P$. We maintain an array B in which values are marginal benefit of topics t, and an array R that contains a counter for every topic in $\mathcal{A}_P$. In every iteration `Schedule` selects the topic u_t with the largest marginal benefit from B and adds it to $\mathcal{A}_P$ (Lines 5 and 6) . Then it updates marginal benefit of u_t, $B[u_t]$ (Lines 7- 8). It is easy to see that Algorithm 1 yields an optimal schedule for a group P and runs in $O(d(|P| + log|\mathbf{T}|))$.

Algorithm 1 `Schedule` algorithm for computing an optimal schedule $\mathcal{A}_P$ for a group P.

Input: requirements $\texttt{req}(s,t)$ for every $s \in P$ and every topic $t \in \mathbf{T}$, deadline d.
Output: schedule $\mathcal{A}_P$.

1: $\mathcal{A}_P \leftarrow [\,]$
2: $B \leftarrow [\mathbf{m}(P, \langle t, 1 \rangle)]$ for $t \in \mathbf{T}$
3: $R \leftarrow [0]$ for all $t \in \mathbf{T}$
4: **while** $|\mathcal{A}_P| < d$ **do**
5: Find topic u_t with maximum marginal benefit in B
6: $\mathcal{A}_P \leftarrow \langle u_t, R[u_t] \rangle$
7: $R[u_t] + +$
8: Update $B[u_t]$ to $\mathbf{m}(P, \langle t, R[u_t] \rangle)$
9: **end while**

4. THE COHORT SELECTION PROBLEM

The next natural question is, that given a certain teaching capacity K (i.e., there are K teachers or K classrooms available), how to divide students into K groups so that each student benefits the most possible from this arrangement. At a high level we solve an instance of a partition problem; find a K-part partition $\mathcal{P} = P_1 \cup^* P_2 \cup^* \ldots \cup^* P_K$ of students into groups, so that the sum of the group benefits over all groups is maximized. This is the COHORT SELECTION Problem .

PROBLEM 2 (COHORT SELECTION). *Let $\mathbf{S}$ be a set of students and $\mathbf{T}$ be a set of topics. For every $s \in \mathbf{S}$ and $t \in \mathbf{T}$ let $\texttt{req}(s,t)$ be the requirement of s on t that is given. Find a partition $\mathcal{P}$ of students into K groups, such that*

$$\mathbf{B}(\mathcal{P}, d) = \sum_{P \in \mathcal{P}} \mathbf{B}(P, \mathcal{A}_P) \tag{5}$$

is maximized, where $\mathcal{A}_P = \texttt{Schedule}(P, d)$ for every group.

The COHORT SELECTION (Problem 2) is NP-hard as the Catalog Segmentation problem [8] can be reduced to it.

4.1 Partition algorithms.

In this section we introduce `CohPart` (Algorithm 3) as our solution to the COHORT SELECTION problem. The input to Algorithm 3 are the requirements $\texttt{req}(s,t)$, number of groups K and length of the schedule d. The output is a partition $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\}$ of the students and corresponding schedules $\{\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_K\}$ for each group.

`CohPart` first assigns every student to one of the groups in $\mathcal{P}$ at random (Line 3) and an initial optimal schedule for every group is computed (Line 5). Then in every iteration of the algorithm first every student is assigned to the group with the highest benefit schedule for the student (Line 9) and then the group schedules are recomputed (Line 12). The runtime of each iteration is $O(k|\mathbf{S}||\mathbf{T}|)$. In our experiments we observed that our algorithm converges really fast, less than a few tens of iterations.

Algorithm 2 `Benefit` algorithm to compute the benefit for student s from schedule $\mathcal{A}$

Input: requirements $\texttt{req}(s,t)$ for a student $s \in P$ and every topic $t \in \mathbf{T}$ and a single schedule $\mathcal{A}$
Output: $\texttt{Benefit}(s,\mathcal{A})$ Benefit of s from schedule $\mathcal{A}$.

1: $\texttt{Benefit}(s, \mathcal{A}) = 0$
2: **for** all topics $t \in \mathbf{T}$ **do**
3: $\texttt{Benefit}(s, \mathcal{A}) = \texttt{Benefit}(s, \mathcal{A}) + \frac{min(req(s,t), \mathcal{A}[t])}{\mathcal{A}[t]}$
4: **end for**

5. EXPERIMENTS

The goal of these experiments is to gain an understanding of how our clustering algorithm works in terms of performance (objective function) and runtime. Furthermore, we want to understand how the deadline parameter impacts our algorithm. We used a real world dataset, semi synthetic and synthetic datasets. The semi synthetic dataset and the source code to generate it are available in our website. [1] We first explain different datasets and then show how well our algorithm is doing on each dataset.

5.1 Algorithms

We compare `CohPart` to two baseline algorithms.

[1] `http://cs-people.bu.edu/bahargam/edm/`

Algorithm 3 `CohPart` for computing the partition $\mathcal{P}$ based on the benefit of students from schedules.

Input: requirement $\texttt{req}(s,t)$ for every $s \in \mathbf{S}$ and $t \in \mathbf{T}$, number of timeslots d, number of groups K.
Output: partition $\mathcal{P}$.

$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K\}$
$\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$
$i \in_R [1, 2, \ldots, K]$, $\mathcal{P}_i \leftarrow s$ for every $s \in S$
for $i = 1, \ldots, K$ **do**
  $\mathcal{A}_i = \texttt{Schedule}(P_i, d)$
end for
while convergence is achieved **do**
  for all students $s \in \mathbf{S}$ **do**
    $P_i \leftarrow s$, $i = \text{argmax}_{j=1,\ldots,k}\, \texttt{Benefit}(s, \mathcal{A}_j)$
  end for
  for $i = 1, \ldots, K$ **do**
    $\mathcal{A}_i = \texttt{Schedule}(P_i, d)$
  end for
end while

`RandPart`: Partition S at random.

`K_means`: We represent each student s by the $|T|$-dimensional vector $\big(\texttt{req}(s,t_1), \texttt{req}(s,t_2), \ldots, \texttt{req}(s,t_{|T|})\big)$ containing its requirements for each topic. We assign students to groups based on the `K_means` clustering performed on the space of the requirement vectors using Eucledian distance.

`CohPart_S`: We also investigate a speedup version of `CohPart`. We pick a subset of $n' << n$ students $S' \subset S$ at random. We compute the optimal group schedules $\mathcal{A}'_1, \mathcal{A}'_2, \ldots \mathcal{A}'_K$ for S' using `CohPart` and then assign each student $s \in S$ to the group that maximizes $\texttt{Benefit}(s, \mathcal{A}'_i)$.

5.2 Datasets

BUCS data. This dataset consists of grades of real students who majored in CS at Boston University. The data consists of 398 students and 41 courses. Here the courses correspond to topics and letter grades were converted to the requirement of students. That is, grades A – F were converted to $\texttt{req}(s,t)$ such that A = 5 and F = 50. We assumed the number of requirement to master a course for the smartest student is 5 (base parameter). As the ability drops, number of requirement goes up (step parameter). To compute missing requiements, i.e., fill values for missing (student, course) pairs, we used Graded Response Model (GRM). First, using GRM we obtain the ability and difficulty parameters for all students and all courses. Then for each pair of (student, course) in which student s did not take course c, we used the ability of s and difficulty of c to predict the grade of course c for that student.

BUCSSynth data. In order to see how well our algorithm scales to larger datasets, we generated a synthetic data, based on the obtained parameters from GRM. We call this dataset BUCSSynth. From BUCS dataset, we observed that the ability of students follows a normal distribution with $\mu = 1.13$ and $\sigma = 1.41$. Applying GRM to BUCS, we obtained difficulty parameters for 41 courses. In order to obtain difficulties for 100 courses, we used the following:

1. Choose one of the 41 courses at random.
2. Use density estimation, smoothing and then get the CDF of the difficulties.
3. Randomly sample from the CDF to get the difficulties for a new course.

Using these parameters, we generated grades for 2000 students and 100 courses and we transformed grades to number of requirements similar to what we did for BUCS dataset.

Synthetic data. In ground truth dataset we had generated 10 groups of students, each group containing 40 students. For each group we selected 5 courses and assigned requirement randomly to those 5 courses such that the sum of requirement will be equal to the deadline. Then for the remaining 35 courses, we filled number of requirements with random numbers taken from a normal distribution with $\mu = \frac{deadline}{5}$ and $\sigma = 3$. We refer to this dataset as GroundTruth.

We have also generated the requirements for 400 students and 40 courses using Pareto ($\alpha = 2$), Normal ($\mu = 30$ and $\sigma = 5$) and Uniform (in the range of [5,100]) distributions. We refer to this datastes as `pareto`, `normal` and `uniform`.

5.3 Results

All algorithms are implemented in Python 2.7 and all the experiments are run single threaded on a Macbook Air (OS-X 10.9.4, 4GB RAM). We compare our algorithm with `RandPart`and the `K_means`algorithm, the built in k-means function in Scipy library. Each experiment was repeated 5 times and the average results are reported in this section. For sample size in `CohPart_S` algorithm, we set parameter c (explained earlier) to 4 in all experiments.

5.3.1 Results on Real World Datasets

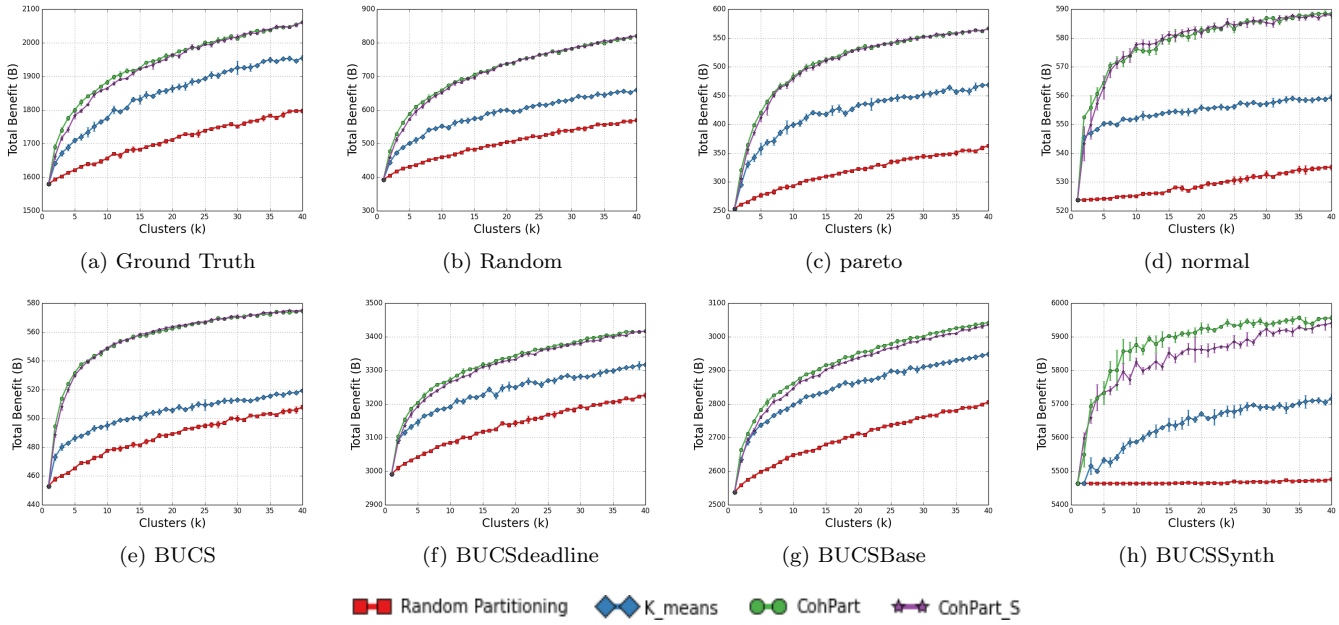
BUCS. The result on the BUSC data is depicted in Figure 1e where each point shows the benefit of all students when partitioning them into K groups. As we see the `RandPart` has the lowest benefit and our algorithm has the best benefit. As the number of clusters increases (having hence fewer students in each cluster), the benefit also increases, means the schedule for those students is more personalized and closer to their individual schedule. In Figure 1f we show that the greater the deadline is, the closer `K_means` gets to our algorithm. But in real life, we do not have enough time to repeat (or teach) all of the courses (for e.g. for preparation before SAT exam). Figure 1f illustrates the case when deadline is equal to the average sum of need vectors for different students.

BUCSBase. We tried different values for base and step parameters (explained earlier) and the result is depicted in Figure 1g when the base and step are equal to 1. The larger is the value of base and step parameter, the better our algorithm performs.

BUCSSynth dataset. We ran our algorithms on on BUCSSynth dataset to see how well our algorithm scales for large number of students. The result is depicted in Figure 1h.

5.3.2 Results on Synthetic Datasets

The result on synthetic data is illustrated in Figure 1a. As we see `CohPart` and `CohPart_S` both are performing well. For all of the courses the mean requirement is close to 10 with standard deviation 3. We expect that students in the same



(a) Ground Truth (b) Random (c) pareto (d) normal

(e) BUCS (f) BUCSdeadline (g) BUCSBase (h) BUCSSynth

Figure 1: Total benefit achieved by different algorithms as a function of the number of groups of students.

group (when generating the data) should be placed in the same cluster after running our algorithm and the schedule should include the selected courses in each group. Students have different requirement values for the selected courses in each group, but the sum of these selected courses is equal to the deadline and our algorithm realized this structure and only considered these selected courses to obtain the schedule. But `K_means` lacked this ability to find the hidden structure. The next studied datasets were `uniform`, `pareto` and `normal` datasets and the results are depicted in Figure 1b, 1c and 1d respectively. For these datasets also our algorithm outperformed `K_means` and `RandPart` .

6. CONCLUSION

In this paper, we highlighted the importance of team formation and scheduling educational materials for students. We suggested a novel clustering algorithm to form different teams and teach the team members based on their abilities. The results we obtained shows that our proposed solution is effective and suggest that we have to consider personalized teaching for students and form more efficient teams.

7. ACKNOWLEDGMENTS

This work was partially supported by NSF Grants: #1430145, #1414119, #1347522, #1239021, #1012798, #1218437, #1253393, #1320542, #1421759.

8. REFERENCES

[1] Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers & Education*, 68(0):199 – 210, 2013.

[2] R. Agrawal, B. Golshan, and E. Terzi. Grouping students in educational settings. In *ACM SIGKDD*, pages 1017–1026, 2014.

[3] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Power in unity: Forming teams in large-scale community systems. In *ACM International Conference on Information and Knowledge Management*, pages 599–608, 2010.

[4] A. Ashman and R. Gillies. *Cooperative Learning: The Social and Intellectual Outcomes of Learning in Groups*. Taylor & Francis, 2003.

[5] P. Brucker. *Scheduling Algorithms*. Springer-Verlag New York, Inc., 3rd edition, 2001.

[6] R. F. Bruner. Repetition is the first principle of all learning. *Social Science Research Network*, 2001.

[7] D. Esposito. Homogeneous and heterogeneous ability grouping: Principal findings and implications for evaluating and designing more effective educational environments. *Review of Educational Research*, 43(2):163–179, 1973.

[8] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. *J. ACM*, pages 263–280, 2004.

[9] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *ACM SIGKDD*, pages 467–476, 2009.

[10] A. Majumder, S. Datta, and K. Naidu. Capacitated team formation problem on social networks. In *ACM SIGKDD*, pages 1005–1013, 2012.

[11] T. P. Novikoff, J. M. Kleinberg, and S. H. Strogatz. Education of a model student. *Proceedings of the National Academy of Sciences*, 109(6):1868–1873, 2012.

[12] A. Segal, Z. Katzir, K. Gal, G. Shani, and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning. 2014.

[13] R. E. Slavin. Ability Grouping and Student Achievement in Elementary Schools: A Best-Evidence Synthesis. *Review of Educational Research*, 57(3):293–336, 1987.

An approach of collaboration analytics in MOOCs using social network analysis and influence diagrams

Antonio R. Anaya
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913986550
arodriguez@dia.uned.es

Jesús G. Boticario
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913989387
jgb@dia.uned.es

Emilio Letón
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913989473
emilio.leton@dia.uned.es

Félix Hernández-del-Olmo
E.T.S.I. Informática - UNED
C/Juan del Rosal, 16
E-28040 Madrid
34 913988345
felixh@dia.uned.es

ABSTRACT
MOOCs pedagogical strategies assume that students construct their own knowledge and collaborate with their mates. Large-scale learners' interaction figures hinder both proper interpretation of learners' needs and prompt remediation actions. To this we describe a preliminary study of a two-step collaboration analysis, which consists of inferring domain-independent indicators on students' relationships obtained from social network analysis and using an influence diagram to warn teachers on students' problematic circumstances to facilitate prompt remediation actions.

Keywords

Collaboration analytics, SNA, influence diagram, collaborative learning

1. INTRODUCTION
Massive open online courses (MOOCs) are stood out as a new pedagogical methodology since they aimed at large-scale participation and open access via the web [1]. In this situation the teacher loses control over the learning process and students should construct their own learning. The students can use the MOOC's communication means to collaborate with their learning mates [2]. In this respect, although the students are to be provided with the tools and services to collaborate, this thus not suffice and frequent and regular analyses of the team process are needed to know whether the collaboration takes place [3]. Moreover, the special large-scale nature of MOOCs hampers teachers when coming to analyze students' communication acts, which drive the collaboration process.

Some researchers have proposed a well-known analysis method, social network analysis (SNA) to minimize the problems commented above [4, 2]. However, in this collaborative learning context some variables, such as emotion and empathy, are out of control [3]. Under these circumstances, analyzing the collaboration process requires to deal with uncertainty [5], which can be tackle with Influence diagrams (ID) [6].

In our research we propose an approach to automatically warn (or recommend [7] teachers on students' problematic collaboration circumstances so that they can readily provide corrective actions when required. Thus, the objectives of the application are: 1) to analyze the collaboration with a transferable analysis method that provides domain-independent collaborative indicators; 2) to minimize the human intervention.

The rest of the paper is organized as follows. First we describe in Section 2 related research, to both SNA in MOOCs and ID in the educational context. In Section 3 we frame the research and educational context in which this work is being applied and an in-depth description of the proposed methodology. We then comment on our preliminary study in Section 4 and finally briefly provide the main conclusions and further planned research in Section 5.

2. Related research
MOOCs offer more leeway to students and thereof features new challenges [8]. In this more crowded and less constrained learning environment it is advisable to use any available technology to analyze the learning process involved. Here technologies such as SNA are starting to be applied with relative success [2].

SNA has been used to identify students who are actively participating in course discussions and thus are potentially at a risk of dropping out [2]. [4] examined and detected, using SNA, communities of users within a large course so that they can be provided with a personalized and social-oriented recommender system. [9] presented an example of a Social Learning Analytics Tool to visualize real-time discussion activities in a MOOC environment.

SNA has been widely applied to study the social aspect of students learning [10]. This way [11] analyzed networks in order to identify the people from whom an individual learns. Here [12] proposed a methodology to analyze students' interactions in a collaborative learning environment, which consists of using SNA to get meaningful statistical indicators, such as the student reputation. [13] emphasized the use of SNA techniques to discover relevant structures in social networks so that the instructors were able to better assess participation.

As the aforementioned approaches we aimed at improving collaborative settings though SNA outcomes in terms of a technology that has proved its usefulness in tackling problems under uncertainty. Moreover, the educational context has been a traditional suitable field where Bayesian networks (BN) have been applied to deal with the inherent uncertainty involved [14]. [15] proposed a course diagram method, based on an ID framework, which can be used by an instructor to design a course structure. The diagram organizes the instructional material and the tests.

3. Towards collaboration analytics in MOOCs
In our education context we proposed to combine two different technologies to analyze the collaboration. Firstly, the SNA obtains indicators from students' interactions, which reflects how students connect with their mates. Secondly, an influence diagram

structures students' indicators as a network, which supports a decision on students' problematic circumstances once an expert, who can be the tutor, tunes the probabilities of the network. The software used for SNA was Gephi[1] and for ID was OpenMarkov[2].

3.1 Social network analysis

To date the most common communication service in MOOCs is forums. SNA has been applied to forums in order to infer the social relationships among users [16]. Here SNA metrics support the inference of social relationship indicators.

Figure 1 shows the SNA diagram resulting from the data of the on-line course that we have used in the preliminary study.

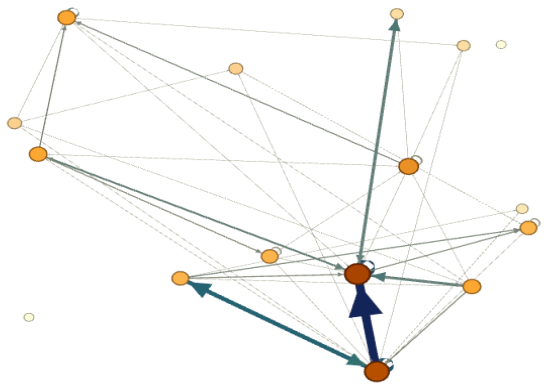


Figure 1. SNA in the preliminary study.

In Figure 1 nodes are students who participated in an online course (see Preliminary study section and communications among students were analyzed through SNA. Within this figure the metric Degree of the nodes is represented as follows: red and big node means high degree, and yellow and small node means low degree. The color and size of the ties mean the weight of the relationship (i.e., number of messages from origin node to destiny node).

We propose the following centrality metrics of the nodes as indicators of the collaboration process:

- **Degree** is the number of ties of one node.
- **In-degree** is the number of ties whose destiny is the node. This indicator is a measure of the node popularity.
- **Out-degree** is the number of ties whose origin is the node. This indicator is a measure of the node sociability.
- **Closeness centrality** is the degree to which an individual is near all other individual in the network. This reflects the ability to access to information by the network members.
- **Betweenness centrality** a measure that quantifies the frequency or number of times that a node acting as a bridge along the shortest path between two other nodes.
- **Eigenvector centrality** is the measure of the importance of a node in the network. Intuitively, the nodes that have a high value of this measure of centrality are connected to many nodes, which are connected also in this sense; therefore, are good candidates to disseminate information.

We use these indictors, because they are well-known in the state-of-the-art research focused on analyzing the position of the students in the network using SNA [16]. These indicators constitute a standard way to measuring network and node features and they can be used in several different context.

3.2 Influence diagrams

IDs provide us with a framework for representing and solving decision problems under uncertainty. As our objective is to maintain a domain independent and general approach of inference IDs include features that are advisable in learning environment as MOOCs, where the collaborative learning is encouraged. The collaboration settings constitute a framework where not all variable are known in advance. In addition, a MOOC is an educational environment where teachers cannot afford the continuous tracking and analysis phases of learners' interactions, which in this case are massive. An ID could help teachers to identify and carried out correction decisions adapted to each student.

We propose an ID where the indicators obtained from the SNA, the centrality metrics commented above, are structured. The network layout of the proposed ID is showed in Figure 2.

In Figure 2 the yellow and round nodes are the variables in the problem. Assessment is the root and hidden variable, which is unknown in future test. The node "Assessment" represents the teacher's assessment of students' collaboration. The ID needs a training dataset with known values of the node "Assessment" to tune the networks probabilities. The other yellow and round nodes are the SNA indicators. The squared node "D" represents the decision, in this case, yes or not. The decision "yes" means a detection of problematic circumstances and the ID supports teacher with a suggestion so that the teacher makes a corrective actions. The node "U" maximizes the decision utility. Notice that the values of the nodes have to be discretized. In order to do the discretization we divided interval values into three groups with equal width. We propose three values: high, medium and low, because these values are easy to understand.

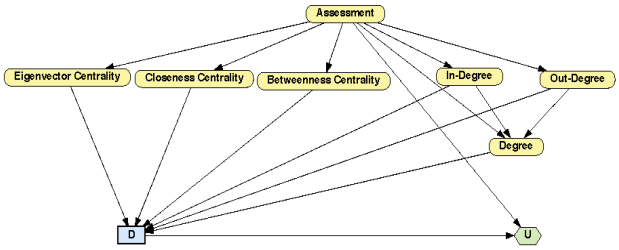


Figure 2. Network of the influence diagram

4. Preliminary study

In the preliminary study we have used data from an on-line course to fine-tune the ID's network. The experience was done with students of the subject "Complexity and Computability" in the forth course of the degree of Computer Systems Engineering at UNED (Spanish National University for Distance Education). In this subject we have mimicked the characteristics of MOOCs, with particular emphasis on the participation on the forum. For that reason we have undertaken a continuous assessment process on the Learning Management System (LMS) forum's interactions

[1] http://gephi.github.io

[2] http://www.openmarkov.org

in order to detect the student level of participation and the recording of a special type of video podcast [17].

We have tested our approach in an online course, which let us make a preliminary proof of concept on the main issues involved, namely tracking and assessing students (16 students). This curse has been designed following the large-scale MOOC's course settings, meaning that it consists of the same video lectures, individual tasks and a communication services that will be ultimately provided [17].

Node Potential: Degree

Relation Type: Table

Assessment	low	low	low	low	low
Out-Degree	low	low	low	middle	middle
In-Degree	low	middle	high	low	middle
high	0	0.333333	0.333333	0	0.333333
middle	0	0.333333	0.333333	0	0.333333
low	1	0.333333	0.333333	1	0.333333

Figure 3: An example of node "Degree" probabilities.

In the fine-tuning process experts can insert knowledge into the ID's network, that is, in the automatic inferring process. Firstly the students should be assessed according to their interactions. It is fairly common that experts decide which students' features, that their interactions have revealed, are the most relevant to be assessed. This knowledge is showed when the assessments are compared with the analysis of students´ interactions, which is independent of expert's assessments. We made the SNA of the students´ interactions and independently an expert assessed the students.

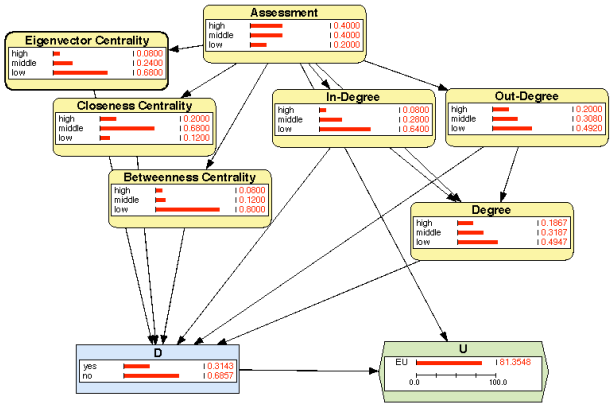


Figure 4: A general perspective of the ID's network results.

Once the students were assessed and we obtained the SNA centrality attributes of each student, we then discretized the data. Then, we were able to measure the probabilities for each case. An example is show in Figure 3. According to the Figure 2, the node "Degree" has three fathers, the nodes "In-Degree", "Out-Degree" and "Assessment". For each possibility of "Degree" value (low, medium or high) we measured the probability according to the values of the father nodes. Figure 3 shows some cases. For instance, when the father node have "low" value, the node "Degree" have "low" value. Because node "Degree" has three father nodes, there are 27 possible cases (the Cartesian product of three variables with three possible values). We made the fine-tuning process with 16 students, thus, we did not have enough data to fine-tune the network completely. We could solve this lack with data from the next experience.

After the probabilities were established for each possible case of each node, the ID was able to infer a decision and the decision utility for each case. Figure 4 shows the general perspective ID's results. It can be seen that the ID advises to recommend only in around one third of cases (In node "D", "yes" is 0.3143 and "no" 0.6857).

We can observe what happens when the ID advises to recommend, i.e., identifies a possible collaboration problematic circumstance. Figure 5 shows the case when the ID advises to recommend. When the ID advises to recommend, the student has low value in the nodes "Degree" and "In-Degree". This informs us that when a recommendation is advisable, the student is not active and her/his classmates ignore her/him. Thus, the ID has identified a problematic collaboration scene, which can be happened over the course. With this information the teacher could make a corrective activity to improve the collaboration process.

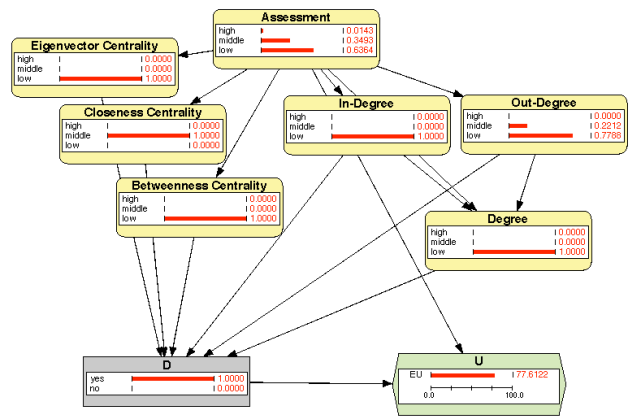


Figure 5. Example: ID advises to recommend.

In addition to the previous analysis, it is possible to calculate the optimal policy of the ID (see Figure 6). Thus, the optimal policy informs about the decision (yes or not) for each combination of nodes values.

Optimal policy: D

Relation Type: Table Reorder variables

Degree	low	low	low	low	low	low	low	low
Eigenvector Centrality	low	low	low	low	low	low	low	low
Betweenness Centrality	low	low	low	low	low	low	low	low
Out-Degree	low	low	low	low	low	low	low	low
Closeness Centrality	low	low	low	middle	middle	middle	high	high
In-Degree	low	middle	high	low	middle	high	low	middle
yes	0	0.5	0.5	1	0	0	0	0
no	1	0.5	0.5	0	1	1	1	1

Figure 6. Optimal policy: all possible decisions of ID.

Figure 6 shows an example of the decisions, yes or not, according to the values (high, middle or low) of the centrality attributes obtained from the SNA. In the preliminary study we had 16 assessed students. The possible cases that the ID can consider mathematically are the Cartesian product of network nodes and the student indicators (a total of 729 cases). Thus, not all the possible cases of the nodes values combination have to be considered by the ID. However, the results (see Figure 6) show that the ID is capable to support with different decisions according to the students SNA centrality attributes values. However, more

interaction data are needed to continue with the ID tuning process. When tuning process is finished, a new student's attributes values from the SNA feed the ID that, in turn, can offer accordingly a new decision (i.e., "yes", suggestion of a corrective action due the possible student's problematic circumstance in the collaboration).

The approach labels students with "yes" (the student needs a recommendation) or "not" (the student does not) and this way guides teachers to identify the student's collaboration problem. Based on this the teacher can create the appropriate recommendation to the student.

5. Conclusions and future work

To facilitate collaborative learning management within MOOCs in this paper we propose a domain independent and transferable approach, which is based on two different technologies: 1) Inferring domain-independent indicators on students' relationships obtained from social network analysis (SNA) in their interactions; 2) From these indicators an ID is used to warn teachers on students' problematic circumstances so they can provide them with prompt remediation actions. Here teachers cannot afford the continuous tracking and analysis phases of learners' interactions, which in this case are massive.

The preliminary results described in this paper confirm that the approach can identify problematic collaboration scenes, although it should be further investigated. Thus, data from more students will be considered, which will be used to tune the ID's network probabilities. Thanks to the approach, the tuning process can be made while the students are participating in the MOOC. Moreover, the final suggestion that is offered to the teacher can also be improved. The suggestion should be easily understandable by any non-expert user so that the analysis process involved won't prevent them from its usage.

The research described in this paper will be further applied within the MAMIPEC project, which aimed to infer and provide affective personalized support to learners in educational contexts [18].

6. ACKNOWLEDGMENTS

Authors would like to thank the MAMIPEC project (TIN2011-29221-C03-01), which has been funded by the Spanish Ministry of Economy and Competence.

7. REFERENCES

[1] Masters, K. (2011). A brief guide to understanding MOOCs. The Internet Journal of Medical Education, 1(2).

[2] Sinha, T. (2014, February). Together we stand, Together we fall, Together we win: Dynamic team formation in massive open online courses. In Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the (pp. 107-112). IEEE.

[3] Johnson, D., Johnson, R. (2004). Cooperation and the use of technology. In: Handbook of research on educational communications and technology. Taylor and Francis Group, pp. 401–424.

[4] Zhuhadar, L. and Butterfield, J. (2014). Analyzing Students Logs in Open Online Courses Using SNA Techniques, Twentieth Americas Conference on Information Systems, Savannah, 2014.

[5] Anaya, A. R., Luque, M., García-Saiz, T. (2013). Recommender system in collaborative learning environment using an influence diagram. Expert Systems with Applications 40, 7193–7202.

[6] Howard, R. A., Matheson, J. E. (1984). Influence diagrams. In: Howard, R. A., Matheson, J. E. (Eds.), Readings on the Principles and Applications of Decision Analysis. Strategic Decisions Group, Menlo Park, CA, pp. 719–762.

[7] Hernández del Olmo, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. Expert Systems with Applications, 35(3), 790-804.

[8] Duque, R., Bravo, C., Ortega, M. (2013). An ontological approach to automating collaboration and interaction analysis in groupware systems. Knowledge-Based Systems 37, 211 – 229.

[9] Schreurs , B., Teplovs, C., Voogd, S. (2014). Social Learning Analytics applied in a MOOC-environment, eLearning Papers, 36, January 2014

[10] Shum, S. B. and Ferguson, R. (2012). Social Learning Analytics. Educational Technology & Society, 15, 3—26

[11] Haythornthwaite, C., & De Laat, M. (2010). Social networks and learning networks: using social network perspectives to understand social learning. Paper presented at the 7th International Conference on Networked Learning, Aalborg, Denmark.

[12] Bratitsis, T., Dimitracopoulou, A., Martínez-Monés, A., Marcos-García, J., Dimitriadis, Y. (2008). Supporting members of a learning community using interaction analysis tools: the example of the Kaleidoscope noe scientific network. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT 2008. pp. 809–813.

[13] Rabbany, R., Takaffoli, M., & Zaïane, O. R. (2011). Analyzing participation of students in online courses using social network analysis techniques. In Proceedings of educational data mining.

[14] Millán, E., Loboda, T., de-la Cruz, J. P. (2010). Bayesian networks for student model engineering. Computers & Education 55, 1663–1683.

[15] Chang, F.-I. (2003). Quantitative analysis on distance learning courseware. Multimedia Tools and Applications 20, 51–65.

[16] Dawson, S. (2008). A Study of the Relationship between Student Social Networks and Sense of Community, Educational Technology & Society, 11, 224—238

[17] Letón, E., Molanes-López, E.M. (2014). Two New Concepts in Video Podcasts: Minimalist Slides and Modular Teaching Mini-videos. 6th International Conference on Computer Supported Education.

[18] Santos, O.C., Salmeron-Majadas, S., Boticario, J.G. (2013). Emotions Detection from Math Exercises by Combining Several Data Sources. AIED 2013, Lecture Notes in Computer Science LNCS/LNAI 7926, 742–745.

On Convergence of Cognitive and Noncognitive Behavior in Collaborative Activity

Luna Bazaldua, Diego A.
Teachers College, Columbia University
dal2159@tc.columbia.edu

Khan, Saad
Educational Testing Services
skhan002@ets.org

von Davier, Alina A.
Educational Testing Services
avondavier@ets.org

Hao, Jiangang
Educational Testing Services
jhao@ets.org

Liu, Lei
Educational Testing Services
lliu@ets.org

Wang, Zuowei
University of Michigan
zwwang@umich.edu

ABSTRACT
We present results from a pilot study to investigate the evidence for convergence and synchrony in cognitive and noncognitive behavior of dyads engaged in a collaborative activity. Our approach utilizes multimodal data including video and participant action log files retrieved from the collaborative activity, an online educational simulation on science topics. The log files captured cognitive behavior including frequency and content of chat messages between dyads, as well system help requests. The video data recorded participant nonverbal behavior that was processed on a frame-by-frame basis using automated facial expression classifiers and coded by trained human raters on high-level noncognitive behaviors including: affect display gestures, engagement, anxiety and curiosity. The data were analyzed at individual and dyad levels and results using hierarchical clustering analysis demonstrate evidence of cognitive and noncognitive behavioral convergence among dyads.

Keywords
Collaborative Assessment, Human-Computer Interaction, Multimodal Data, Noncognitive states, Cluster Analysis

1. INTRODUCTION
Behavioral convergence refers to the unintentional imitation process of gestures, facial expressions, behaviors, moods, postures, or verbal patterns of coparticipants on a range of different time-scales [4, 12]. In literature it has been referred to by a variety of terms e.g., behavioral matching, mimicry, interpersonal coordination, entrainment, interactional synchrony and the Chameleon effect [4, 12, 17, 19]. While previous studies have explored its impact on interpersonal skills, coordinated activity, negotiations, and how individuals influence the behaviors of others [2, 4, 21], little research has focused on finding evidence for behavioral convergence in collaborative activity [24].

Collaboration is a complex activity that constitutes an interplay between *cognitive processes* such as knowledge acquisition, content understanding, action planning, and execution [7, 8, 10, 18, 26] and *noncognitive processes* such as social regulation, adaptability, engagement and social affect, such as boredom, confusion, and frustration [1, 3, 6]. Collaborative activity may take place in face-to-face interactions or through the medium of online distance learning technologies and collaboration platforms [20]. In either context collaboration is more effective when participants are engaged in the task and exhibit behaviors that facilitate interaction [25].

Our hypothesis is that behavioral convergence occurs during collaborative activity and it manifests in both cognitive and noncognitive processes. Based on this premise, we expect that people will tend to synchronize their behaviors (consciously or nonconsciously) while they are engaged in a collaborative activity. To test our hypothesis, a pilot study was conducted involving 12 unique dyads collaborating in an online game-like science assessment: ETS' online collaborative research environment ─the Tetralogue [15, 27]. Multimodal data including video and activity log files of each participating dyad were captured. The log files contain cognitive behavior including frequency and content of chat messages between dyads, as well as system help request (i.e., the participant requests to view educational videos on the subject matter to better answer assessment questions). The video data, on the other hand, recorded participant nonverbal behavior which was analyzed on a frame-by-frame basis using automated facial expression classifiers and annotated by trained human raters on high-level noncognitive behaviors including: affect display gestures, engagement, anxiety, and curiosity. Along with recent studies [17, 20, 24], in this paper we describe one of the first attempts to capture and analyze multimodal data in the context of studying behavioral convergence in collaborative activities.

2. Methodology
2.1 Collaborative Activity Platform
As mentioned earlier, our study used an online collaboration assessment platform: ETS' online collaborative research environment –the Tetralogue. This platform includes a set of multiple-choice items on general science topics, a simulation based assessment, a personality test, and a set of background questionnaires. The simulation task is on geology topics. The simulation-based task was developed as a task for individual test takers who will interact with two avatars and as a collaborative task that requires the collaboration among two human participants and two avatars in order to solve geology problems.

The participants, who may be in different locations, interact through an online chat box and system help requests (selecting to

view educational videos on the subject matter). The main avatar, Dr. Garcia, introduces information on volcanoes, facilitates the simulation, and requires the participants to answer a set of individual and group questions and tasks. A second avatar, Art, takes the role of another student, in order to contrast his information with that produced by the dyad.

The system logs activity data of the participants in structured XML files, which capture participant actions including: identification of the user who performed each action, the number of chat messages, the content of those chat messages, the number of times the participants request additional information on subject matter from the system, the answer selected for each individual and group question, and the time at which each action occurred.

While the dyads interacted with the task, we captured the video of each individual participant. The video data were used for both annotating noncognitive behavior of the participants and automated facial expression analysis (see section 2.3 for further details). It should be noted that the only form of direct communication between the dyads was through the Tetralogue text-based chat interface and the dyads were not able to see or hear each other. Figure 1 illustrates the collaborative activity and data capture while participants interact in the system.

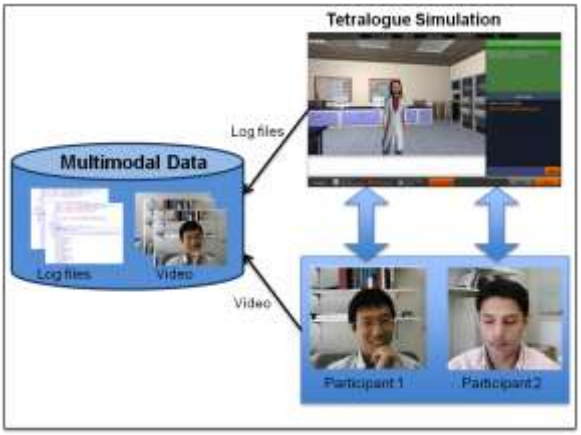


Figure 1. Multimodal data capture including video and action log files while participants engage in collaborative activity on the Tetralogue platform.

2.2 Study Participants and Data Collection

Twenty-four subjects participated in this study and were paired in dyads using random selection. Information about the study was provided to each participant individually and consent forms were obtained from them.

The length of the experiment sessions varied from 15 minutes to 48 minutes, with an average length of 25 minutes. Although there were time variations among sessions, all dyads reviewed the same material and completed the same tasks in Tetralogue. This resulted in approximately 600 minutes of video data and associated participant action log file data. The data stored in the log files were parsed using the 'XML' package [13]. The features extracted from the log files were: number of chat messages sent to the partner and number system helps (viewing educational videos on the subject matter) requested at each stage of the simulation, answer to each individual question, and answer to each group question.

Our focus on "number of messages" and "number of help requests" was driven by former research in the field that associates both features with the performance in learning-oriented tasks, cognitive states, and collaborative interactions [6, 17]. However, more features associated with cognitive activity can be mined from the log files, such as the time length between actions or the content of the chat messages and will be addressed in future studies.

2.3 Video Data Processing and Coding

Facial expression analysis of the video data was performed using the FACET SDK, a commercial version of the Computer Expression Recognition Toolbox [14]. This tool recognizes fine-grained facial features, or facial action units (AUs), described in the Facial Action Coding System [9]. FACET detects human faces in a video frame, locates and tracks facial features, and uses support vector machine based classifiers to output frame-by-frame detection probabilities of a set of facial expressions: anger, joy, contempt and surprise.

In addition, seven trained coders reviewed and coded the videos using the Anvil software [11]. The video data of each participant were assigned to two raters for annotation; however, in three cases there were three raters coding the same video file, and in two cases only a single rater was available for annotation. The raters followed the same coding scheme during the annotation process, which included the next categories: having their hand on their face, expressing engagement, anxiety, or curiosity. As an outcome of the annotation process, the Anvil software produced XML files that were parsed using the 'XML' package [13] in R [22].

Engagement, anxiety, and curiosity were included in the annotation scheme because of the incidence and relevance of these three noncognitive states in simulation games and online learning systems [1, 5]. The coding also included "hand touching face", an affect display gesture that has been linked to affective and cognitive states such as boredom, engagement, and thinking [16].

3. Results

3.1 Behavioral Convergence within Dyads

In order to study evidence of behavioral convergence, features from log files and video data of each of the 24 study participants were represented as a multidimensional behavioral feature vector composed of both the cognitive behaviors: *number_of_messages*, *number_of_help_requests* and the noncognitive behaviors (i.e. fraction of the time each participant exhibited the behavior): *engagment, hand_on_face, anxiety, curiosity, anger, joy, contempt and surprise*.

An agglomerative hierarchical cluster analysis using an average linkage function was performed on an Euclidean distance matrix (i.e., a similarity matrix) computed from the multidimensional behavioral feature data of the study participants. Our hypothesis is that behavioral convergence will manifest in the cognitive and noncognitive features such that members of the same dyad will tend to group together from the beginning of the clustering process (i.e., they will be closer to each other in the feature space than to others).

Figure 2 depicts the dendrogram plot produced from the cluster analysis. In the plot, members of the same dyad are depicted by consecutive numbers and identical color; for instance, the first

dyad includes coparticipants d1.1 and d1.2 colored in red, the second dyad consists of coparticipants d2.1 and d2.2 colored in blue, and so on. The plot shows that participants in 7 of the 12 dyads grouped together in the clustering process (i.e. they were closest to each other in the multidimensional feature space), indicating a high degree of behavioral convergence. Still, some participants (e.g., d10.1 and d4.2) showed a distinctive pattern of values in the variables used to calculate the distances, which prevented them to be grouped with their respective peers.

In addition, we analyzed the similarity matrix of behavioral feature distances for participants within and outside dyads. Behavioral convergence would imply that for dyad members the average distances in feature space is smaller in a statistically significant manner than those of non-dyad members. To study the relative impact of cognitive and noncognitive features we computed two additional similarity matrices: one using exclusively the cognitive features from log files (number of chats messages and number of system help requests) and the other using exclusively noncognitive features produced from the video data (the four facial expression detectors, and the four features from the coding scheme). All features were normalized to present equivalent scaled values between zero and one.

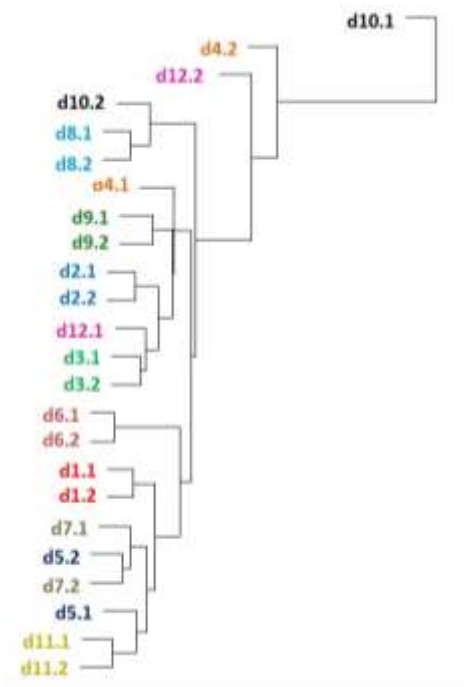


Figure 2. Agglomerative Cluster Dendrogram.

Table 1 shows the mean and standard deviations of feature similarity distances of participants when compared with their dyad partners and others. The results consistently show smaller average distances for the dyads (i.e., members within dyads displayed behavior that was more similar to each other than others), supporting the convergence premise. Additional analysis to test the significance of these differences using the Student's *t*-test demonstrated that using both cognitive and noncognitive features the average distances are statistically significant (t-value = 2.33, df = 11.7, *p*-value < 0.02).

A final analysis was computed on the correlation of the total group scores in the task with the distances of participants with their respective dyad partners and with other users except for their teammate. The group score showed a mild correlation with the distance between dyad members of -0.19 ($s.e._r$ = 0.21). Note that the negative correlation is a consequence of using similarity distances (smaller distance values indicate more convergence) and the group score values (higher values indicate a better performance on the task). Nevertheless, as will be underscored in Section 4, the small sample size in the study produced large standard errors for this correlation estimate and do not imply statistically significant patterns.

Table 1. Average and standard deviation of behavioral feature distances within and outside dyads

Features		Mean	S.D.
Cognitive and noncognitive	**Dyad**	0.57	0.22
	Others	0.73	0.24
Cognitive only	**Dyad**	0.36	0.21
	Others	0.57	0.20
Noncognitive only	**Dyad**	0.41	0.17
	Others	0.41	0.22

4. Discussion and Conclusions

Seminal work from Roschelle [23] in his seminal work made the argument that the crux of learning by collaboration is convergence and showed empirical evidence of the convergence occurring at the linguistic level. Our study provides further empirical evidence of behavioral convergence gleaned from multimodal data. As pointed out in [8], cognitive and noncognitive processes occur simultaneously throughout the collaborative task, and both dimensions cannot be separated in practice. The results from cluster analysis in our experimental study support this idea and the pattern of agglomeration of the participants could be interpreted as evidence of convergence of cognitive and noncognitive states when people interact in a collaborative task.

As reported in table 1, the degree of behavioral similarity within dyads tended to be significantly higher than the similarity between non-dyad members, which is good evidence for behavioral convergence in collaborative interactions [4, 12]. In addition, we observed a mild correlation (of approximately 0.2) between the measure of convergence (i.e., the level of similarity between dyads) and the dyad task scores. This might be interpreted as a scaffolding effect that convergence during interaction can have in group performance outcomes. Similar results were reported in [24], underscoring that specific types of convergence have a positive effect in learning and collaboration.

Further research using these data will address topics such as the synchrony of behavior and noncognitive states between members within dyads, machine learning and classification analyses to detect and predict specific cognitive and noncognitive states from facial action units, and more detailed analysis on the impact of cognitive and noncognitive states on the individual-level and group-level assessment outcomes.

There are certain limitations of this study that should be pointed out. First, the current sample size is small ─24 participants─ despite the rich amount of information gathered from each participant. Second, the current collaboration platform neither allows participants to view each other nor uses face-to-face audio-visual interfaces to communicate. This limits how participants are able to mirror each other's behavior and may also explain why we observed weaker convergence in noncognitive features. Third, the

study has utilized a very limited set of behaviors both cognitive and noncognitive. We aim to extend our behavior feature set and sources of data (e.g., audio data) in future studies as well as utilize the content of participant chat messages to glean features like shared vocabulary, turn-taking etc.

5. REFERENCES

[1] R. Baker, S.K. D'Mello, M.M.T. Rodrigo, & A.C. Graesser. 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] S. Barsade. 2002. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, 47, 644–675.

[3] M. Ben Ammar, M. Neji, A. M. Alimi, & G. Gouardères. 2010. The affective tutoring system. *Expert Systems with Applications*, 37, 4, 3013-3023.

[4] S. Bilakhia, S. Petridis, & M. Pantic. 2013. Audiovisual detection of behavioural mimicry. In *Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction*. 123-128.

[5] R.A. Calvo, & S. D'Mello. 2010. Affect detection: An interdisciplinary review of models, methods, and their applications. *Transactions on Affective Computing*, 1, 1, 18-37.

[6] A.C. Graesser, S.K. D'Mello, S.D. Craig, A. Witherspoon, J. Sullins, B. McDaniel, & B. Gholson. 2008. The relationship between affective states and dialog patterns during interactions with AutoTutor. *Journal of Interactive Learning Research*, 19, 2, 293-312.

[7] S. Greiff. 2012. From interactive to collaborative problem solving: Current issues in the Programme for International Student Assessment. *Review of Psychology*, 19, 2, 111–121.

[8] J. Hao, L. Liu, A.A. von Davier, & P. Kyllonen. 2015. Assessing collaborative problem solving with simulation based task. Paper to be presented at the *11th international conference on computer supported collaborative learning*.

[9] J.C. Hager, P. Ekman, & W.V. Friesen. 2002. *Facial action coding system*. A Human Face, Salt Lake City, UT.

[10] A. Hron, & H.F. Friedrich. 2003. A review of web-based collaborative learning: factors beyond technology. *Journal of Computer Assisted Learning*, 19, 1, 70-79.

[11] M. Kipp. 2012. Multimedia Annotation, Querying and Analysis in ANVIL. In *Multimedia Information Extraction*. M. Maybury, Ed., Wiley-IEEE Computer Society Press, 351-367.

[12] J.L. Lakin, V.E. Jefferis, C.M. Cheng, & T.L. Chartrand. 2003. The chameleon effect as social glue: Evidence for the evolutionary significance of nonconscious mimicry. *Journal of nonverbal behavior*, 27, 3, 145-162.

[13] D. T. Lang. 2013. *XML: Tools for parsing and generating XML within R and S-Plus*. R package version 3.98-1.1. http://CRAN.R-project.org/package=XML.

[14] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, & M. Bartlett. 2011. The computer expression recognition toolbox (CERT). In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*. 298-305.

[15] L. Liu, J. Hao, A.A. von Davier, P. Kyllonen, & D. Zapata-Rivera. In press. A tough nut to crack: Measuring collaborative problem solving. In *Handbook of Research on Computational Tools for Real-World Skill Development*. Y. Rosen, S. Ferrara, & M. Mosharraf, Eds., IGI-Global, Hershey.

[16] M. Mahmoud, & P. Robinson. 2011. Interpreting Hand Over Face Gestures. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 248-255.

[17] R. Martinez, J. Kay, J. Wallace, & K. Yacef. 2011. Modelling and identifying collaborative situations in a collocated multi-display groupware setting. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, 196-204.

[18] H.F. O'Neil, S.H. Chuang, & G.K.W.K. Chung. 2003. Issues in the computer-based assessment of collaborative problem solving. *Assessment in Education*, 10, 361-373.

[19] J.S. Pardo. 2006. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119, 4, 2382-2393.

[20] D. N. Prata, R. Baker, E. Costa, C. P. Rosé, Y. Cui, & A.M.J.B. de Carvalho. 2009. Detecting and Understanding the Impact of Cognitive and Interpersonal Conflict in Computer Supported Collaborative Learning Environments. In *Proceedings of the 2nd International Conference on Educational Data Mining*. 131-140.

[21] A. Pentland. 2008. *Honest Signals: How they shape our world*. MIT Press, Cambridge, MA.

[22] R Core Team. 2012. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[23] J. Roschelle. 1992. Learning by collaborating: Convergent conceptual change. *The journal of the learning sciences*, 2, 3, 235-276.

[24] B. Schneider. 2014. Toward Collaboration Sensing: Multimodal Detection of the Chameleon Effect in Collaborative Learning Settings. In *Proceedings of the 7th International Conference on Educational Data Mining*, 435-437.

[25] A.A. Tawfik, L. Sanchez, & D. Saparova. 2014. The Effects of Case Libraries in Supporting Collaborative Problem-Solving in an Online Learning. *Environment. Technology, Knowledge and Learning*, 19, 3, 337-358.

[26] A.A. von Davier, & P.F. Halpin. 2013. *Collaborative Problem Solving and the Assessment of Cognitive Skills: Psychometric Considerations*. ETS Research Report No.RR-13-41. Educational Testing Service, Princeton, NJ.

[27] D. Zapata-Rivera, T. Jackson, L. Liu, M. Bertling, M. Vezzu, & I.R. Katz. 2014. Assessing science inquiry skills using trialogues. In *Intelligent Tutoring Systems*. S. Trausan-Matu, K. Boyer, M. Crosby, & K. Panourgia, Eds., Springer International Publishing, 625-626.

The Impact of Small Learning Group Composition on Student Engagement and Success in a MOOC

Zhilin Zheng*
Department of Computer Science
Humboldt-Universität zu Berlin
Berlin, Germany
zhilin.zheng@hu-berlin.de

Tim Vogelsang*
iversity GmbH
Bernau bei Berlin, Germany
t.vogelsang@iversity.org

Niels Pinkwart
Department of Computer Science
Humboldt-Universität zu Berlin
Berlin, Germany
niels.pinkwart@hu-berlin.de

ABSTRACT
A commonly known and widely studied problem of massive open online courses (MOOCs) is the high drop-out rate of students. In this paper we propose and analyze the composition of small learning groups as a solution to this problem. In an experiment, we composed such small learning groups in a MOOC context using two methods: Random grouping and grouping by an algorithm that considers selected student criteria. Further, a flipped classroom course was conducted on-campus with a local student group using the MOOC. We compared all three approaches to a control condition using two measures: Drop-out rate and learning performance. The empirical results give an indication, yet no hard evidence, that small groups might reduce student drop-out rates.

Keywords
MOOC; Group Composition; Learning Analytics; Drop-out Rate.

1. INTRODUCTION
MOOC providers, such as Coursera, EdX and iversity, reach course enrolments of up to tens of thousands of students using scalable techniques like lecture videos and quizzes [7]. This massive scale reduces the opportunities for interaction with course instructors. Completion rate, a commonly used (yet debatable) measure of student success, is reported to be less than 13 percent in most MOOCs [3], which has recently attracted extensive studies in order to discover reasons behind this problem [5; 8; 11]. Social connections and collaboration between MOOC students also fall far below expectations. Only 5-10 percent actively participate in course forums [9]. At this point, group formation might help by leading to the creation of informal social ties [4] as well as improving social skills [10].

The composition of small learning groups has already been tested in online learning contexts and local meeting scenarios (i.e. face-to-face groups). In general, self-selected, random and algorithm-based group composition are commonly applied. Algorithm composed groups typically bring together students with either heterogeneous or homogeneous criteria (e.g. based on learning style, personality and demographic information) using technologies such as GT [1] or Swarm Intelligence [2]. Unlike the case with randomly composed or self-selected groups, students' information must be preliminarily collected and then provided to the composition algorithm.

In order to investigate the impact of small learning groups on drop-out rate and learning performance, we conducted a grouping experiment on the iversity.org platform. Specifically, we tested three grouping approaches, all in the same MOOC: 1) automated group composition using an adapted k-means clustering algorithm, accounting for both homogeneous and heterogeneous student criteria; 2) random group composition; and 3) an on-campus flipped classroom approach. This paper describes the results in the three conditions concerning drop-out rates, learning performance and student engagement. The employed algorithm is easy to implement and has low computational costs. In the experiment, we made use of only free and minimal intervention (email) and collaboration methods (email, VoIP, social media). Hence the organizational burden for developers, instructors and students was reduced to a minimum. The experiment is thus scalable and reproducible within many learning environments.

2. METHODOLOGY
2.1 Research Objectives
Empirically, we investigated the following three research questions:

1) Student engagement: Will MOOC students assigned to online groups (without further moderation) be engaged in online collaboration?

2) Drop-out rate: Will random or algorithmic grouping of MOOC students decrease the drop-out rate?

3) Learning performance: Can random or algorithmic grouping lead to higher learning performance, as measured in quizzes and homework scores?

2.2 Experiment Procedure
For conducting the experiment, we chose the second iteration of the course "The Fascination of Crystals and Symmetry", which was offered on the iversity.org platform. This is an introductory course to crystallography held by Dr. Frank Hoffmann (University of Hamburg). Since the course offered open discussion questions, it seemed well suited to engage students in group interaction. It had 3,209 enrollments in total, out of which 771 (i.e. 24.03%) were actively engaged throughout the course.

After the start of the course, 80 percent of the participants received a grouping survey via email asking for information about gender, timezone, language, personality, learning goals (general or in-depth) and their preferred collaboration method (local, email, Facebook, Google+ or Skype). The remaining 20 percent of the course received a motivational survey instead and served as a control condition. One week after the course start, students who provided sufficient answers to the grouping survey were assigned to groups of size 10 by our algorithm and received a second email a few days later. Those who did not respond but had a Facebook account were still randomly assigned to groups. The second email presented the other group members with their personal

* Zhilin Zheng and Tim Vogelsang contributed equally to this work.

descriptions as given in the survey. Further, the email contained a link to the first open discussion question of the course material and a link to their group (if applicable). Students from the control conditions, without or with insufficient grouping survey responses, were not assigned to groups. In addition, the course was held by Dr. Hoffmann as a flipped classroom at the University of Hamburg with approximately 65 students who watched the online lectures at home and met in-class for discussion. Out of these 65 students, 7 used their university account to sign-in to iversity and were anonymously included into our dataset. The other 58 students were not explicitly included. They either used private email addresses or did not sign up to the online course. This (relatively complex, but ecologically valid) assignment procedure of students to seven different conditions is summarized in Figure 1 and Table 1.

Table 1. Student conditions

Condition	Collaboration Method	Description
"Algorithm composed groups" (AlgoCG)	According to preference	Grouping survey, responded sufficiently grouped by algorithm
"Randomly composed groups" (RandCG)	Facebook	Grouping survey, not responded, Facebook user, grouped randomly
"Flipped classroom group" (FlippCG)	Local at University of Hamburg	Attended flipped classroom with the instructor
"No grouping - no answer" (NoG-NA)	none	Grouping survey, not responded, not grouped
"No grouping - insufficient answer" (NoG-IA)	none	Grouping survey, responded insufficiently, not grouped
"No grouping - control group - responsive" (NoG-CG-R)	none	Motivational survey, responded, not grouped
"No grouping - control group - nonresponsive" (NoG-CG-NR)	none	Motivational survey, not responded, not grouped

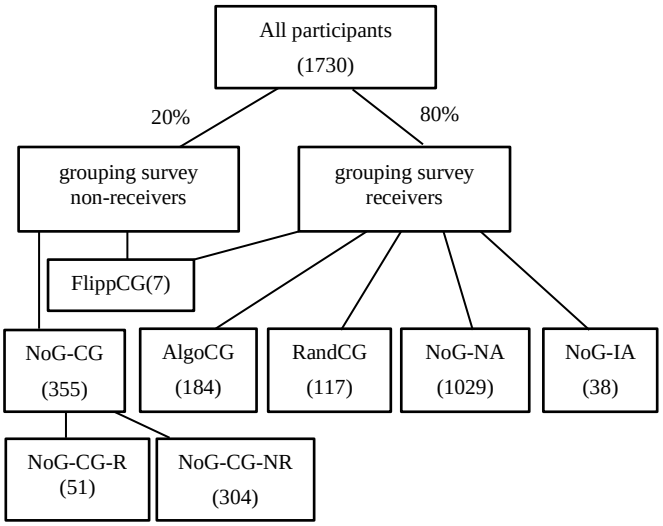


Figure 1. Student conditions with participation numbers.

As a last grouping related intervention, we sent a post grouping survey by the end of the course. This survey was only sent to the 80 percent who had also received the initial grouping survey and contained questions about satisfaction with and intensity of the group work.

3. GROUPING ALGORITHM

In order to create algorithm composed groups (AlgoCG), we used the collected responses from the grouping survey. We first segmented the respondents into five classes according to their collaboration preferences, namely local, email, Facebook, Google+ or Skype. For each class, we extracted each participant's gender, time zone, personality type, learning goal and language for the actual grouping. The task of the algorithm was to compose learning groups consisting of 10 students. Local groups were meant to only contain students from the same cities in order to actually meet up, resulting in very few and small groups qualifying for this option. The main algorithmic challenge was to take into account both heterogeneities (namely gender, personality type and learning goal) and homogeneities (i.e. time zone and language). Concretely, we wanted groups to have e.g. mixed gender, but similar time zone. To solve this optimization problem, we used a k-means clustering algorithm for fixed group sizes, based on [6]. The pseudocode of this algorithm is described in Figure 2 and our implementation in Python is publicly available[1]. In its original form, the algorithm calculates a homogeneity score for a single grouping criterion, like in usual applications of k-means clustering. For our experiment, we modified this algorithm to support multiple criteria and homogeneity as well as heterogeneity at the same time. As a modification, we calculated the group score as the difference between a homogeneity score (on time zone, language and learning goal) and a heterogeneity score (on gender and personality), both of which are actually measured by the Euclidean distance between peers.

```
-----------------------------------------------------------------
Step1: randomly assign students to groups;
Step2: for every group:
            for every student in the current group:
                calculate the possible group scores
                for the student in all the other groups;
                if the student has a higher group score
                in one of the other groups:
                    find the student in the other group with
                    the lowest group score;
                    swap the two students;
Step3: while we are significantly improving the average
        group score, go back to step 2
-----------------------------------------------------------------
```

Figure 2: K-means Clustering for fixed group sizes [6]. Image courtesy: Dirk Uys.

4. EXPERIMENT RESULTS

As a result of our grouping efforts, we composed 22 learning groups in total (4 local meeting groups, 5 Skype groups, 6 Facebook groups, 2 Google+ groups and 5 email groups). The

[1] *https://bitbucket.org/zhilinzheng7/kmeansgrouping*

following sub-sections present three aspects of the experiment results: student engagement, drop-out rate and learning performance.

4.1 Student Engagement

Roughly half of the students enrolled in the course were part of our experiment (1,730 out of 3,209). The other half of the students enrolled after the official start of the course (and, hence, after the start of our experiment), which is a usual pattern for a MOOC.

Overall, the course participants were quite inactive in general, as measured in terms of forum participation. Only 33 students participated in the forum by posting questions, answers or comments. The conducted post grouping survey had nine responses from participants that joined a group. Those respondents spent three hours on average (median one hour) on the group interactions. Further, the Facebook and Google+ groups that were created by us showed some initial greeting messages or comments but no deep, course-related interaction. Hence, our composition did not engage students in collaboration via the social media groups created for that purpose. For other online grouping participants (i.e. email and Skype) who did not answer to our post grouping survey, we cannot make the same conclusion owing to a lack of data.

However, students at least saw small descriptions of their peers in our welcome message and were partly able to see them on social media. Whether this fact, in addition to potentially unobserved interactions (e.g. via email), might have had an impact on the drop-out rate and learning performance, as well as how this relates to survey responsiveness, is analyzed in the following two subsections.

4.2 Drop-out Rate and Survey Responsiveness

We here define a 'drop-out' as any student who did not submit any quiz or assessment, and thereby did not qualify for any course score, after the group assignment.

Figure 3 shows the drop-out rate for all conditions. Unsurprisingly, all seven of the tracked flipped-classroom students stayed in the course (drop-out rate *0%*). In order to test the statistical significance of found differences in the drop-out rates, pairwise z-tests on the different conditions using a two-sided p-value were performed. For our conclusions about significance, we thus applied a Bonferroni correction to the significance level. The p-values in Section 4.2 are given in their non-Bonferroni-corrected form.

First of all, survey responsiveness plays a major role in the analysis. Among the participants of the treatment group that were not grouped, those who gave insufficient survey responses seem to be less likely to drop out than those who did not respond at all, yet this difference is not statistically significant (NoG-IA: *71.05%*, NoG-NA: *82.31%*, $p=0.07$). Further, in the control group without grouping, those who interacted with the motivational survey had a considerably lower drop-out rate than those who did not (NoG-CG-R: *62.75%*, NoG-CG-NR: *82.57%*, $p=0.001$). We can conclude that non-responsive students (with regard to a survey) are more likely to drop out than responsive students.

Hence, when analyzing the interplay between grouping condition and drop-out rate, we need to control for survey responsiveness. Since the randomly composed students did not respond to the grouping survey, we need to compare them to the students in the control group who did not respond to the motivational survey (RandCG: *77.78%*, NoG-CG-NR: *82.57%*, $p=0.26$). And since students from the algorithm composed groups responded to our grouping survey, they need to be compared with the fraction of the control group responding to the motivational survey (AlgoCG: *59.24%*, NoG-CG-R: *62.75%*, $p=0.65$). With this control for survey responsiveness, we thus find no statistically significant effects.

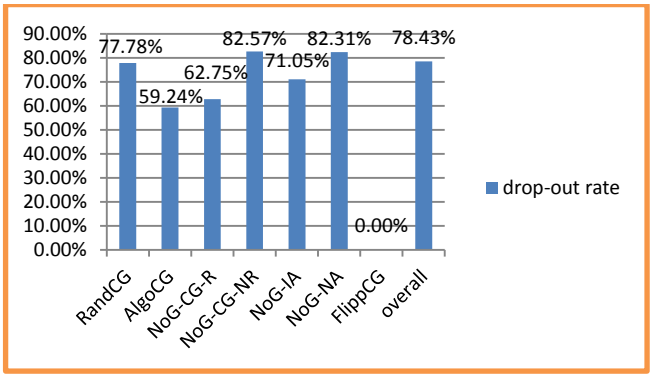


Figure 3: Bar-plot showing student's drop-out rates.

4.3 Learning Performance

In order to analyze the experiment's impact on student's learning performance, we looked at students' scores on quizzes and homework. Figure 4 visualizes average as well as minimum and maximum scores within the various experiment conditions. The flipped classroom condition outperformed all other conditions in terms of median score (FlippCG: 32, others: below 20). However, we do not find evidence for a positive impact of any condition on learning performance as measured by score. A one-way ANOVA implied no statistically significant difference between the conditions ($F(6,518)=1.284$, $p=0.265$).

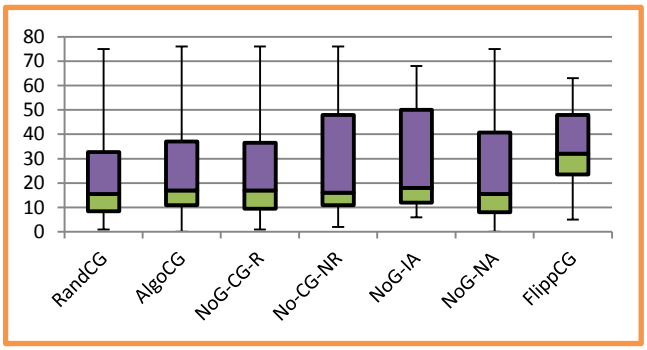


Figure 4. Boxplot showing student's scores from quizzes and homework.

5. DISCUSSION AND FUTURE WORK

In this paper, we presented a scalable and reproducible method to create small groups in online learning environments. We used minimal intervention methods and freely available collaboration tools as well as an adapted k-means clustering algorithm. Within the study, a flipped classroom approach outperformed all composed groups having no drop-out and above average learning scores. This is only partially surprising, as the flipped classroom students were in a formal education setting and most of the others were not. Further, survey responsiveness was found to be predictive of the drop-out rate. Comparing student conditions according to this insight, we found indications that composing

small learning groups in MOOCs (at least the way we did it) might not directly increase learning performance online, but could possibly decrease drop-out rates. However, these findings are limited by lack of statistical significance, self-selection biases and little observed interaction in the groups. These limitations need to be addressed within replications and extensions of this experiment.

Statistical significance: The scope of our experiment was a single but massive open online course with quite a high number of participants (1,730), which is far beyond the possibilities of a traditional classroom experiment. However, we faced low response rates and had to assign online students to rather complicated conditions, varying in size between 38 and 1,029 students (cf. Figure 1). The flipped classroom condition only had 7 students. Together, these impediments had a negative impact on the statistical power. For replication, even bigger courses should be chosen.

Self-selection: While only those who completed our grouping survey were assigned to the AlgoCG condition, we chose to compose RandCG from students who did not respond to this survey (for the sake of having enough groups in the AlgoCG condition). This self-selection problem was addressed analytically by also splitting our control group into responders and non-responders to our motivational survey. However, those interventions are not exactly equal: The email containing the motivational survey expresses the wish of the instructor and platform to get to know the students in order to adjust courses accordingly. The email containing the grouping survey, on the other hand, addresses the student's potential wish to collaborate in a group.

Group interaction: Finally, only very low actual collaboration could be observed in the Facebook and Google+ groups. How can small learning groups have an effect if nothing is going on in the groups? Some students claimed in the post grouping survey to have collaborated and it might be the case that the Facebook and Google+ groups were avoided (as an iversity team member was part of the group) and other, private, channels were preferred for collaboration.

In order to overcome the limitations within future student grouping experiments, we deduced new research hypotheses from our results.

Hypothesis 1: Using learning environments that are specifically designed for group work (including reminders, definition of learning goals, assignment of individual group roles or scheduled group meetings) will increase collaboration within small learning groups.

Hypothesis 2: Dynamic group (re-)composition using genetic or particle swarm algorithms will increase collaboration within the small learning groups, by solving the problem of drop-out in learning groups.

Hypothesis 3: Establishing small and regularly interacting sub-communities within a large online course may reduce students' drop-out rate. Just being aware of one another, even if not working together, is crucial.

6. ACKNOWLEDGMENTS

We thank Frank Hoffmann, Michael Sartor and Michael Fröba for collaborating with us and making this research possible; Jo Corrall for helpful feedback; and the China Scholarship Council (CSC) for providing a scholarship that supported this research.

7. REFERENCES

[1] COCEA, M. and MAGOULAS, G., 2010. Group Formation for Collaboration in Exploratory Learning Using Group Technology Techniques. In *Knowledge-Based and Intelligent Information and Engineering Systems*, R. SETCHI, I. JORDANOV, R. HOWLETT and L. JAIN Eds. Springer Berlin Heidelberg, 103-113.

[2] HUANG, Y.M., 2011. A systematic approach for learner group composition utilizing u-learning portfolio. *Journal of educational technology & society 14*, 3, 102-117.

[3] JORDAN, K., 2014. MOOC Completion Rates: The Data. In *Available at:http://www.katyjordan.com/MOOCproject.html*, [Accessed: 27/08/2014].

[4] OH, H., CHUNG, M.-H., and LABIANCA, G., 2004. Group Social Capital and Group Effectiveness: The Role of Informal Socializing Ties. *The Academy of Management Journal 47*, 6, 860-875.

[5] ONAH, D., SINCLAIR, J., and BOYATT, R., 2014. DROPOUT RATES OF MASSIVE OPEN ONLINE COURSES: BEHAVIOURAL PATTERNS. *EDULEARN14 Proceedings*, 5825-5834.

[6] UYS, D., 2014. How we Used the Echo Nest API for Engagement & Learning, Available at: http://info.p2pu.org/2014/01/13/how-we-used-the-echonest-api-for-engagement-learning/, [Accessed: May 2014].

[7] PAPPANO, L., 2012. The year of the mooc. In *In New York Times, November 2012*, Available at: http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html, [Accessed: October 2014].

[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. In *Proceedings of the first ACM conference on Learning @ scale conference* (Atlanta, Georgia, USA, 2014), ACM, 2567879, 197-198.

[9] ROSÉ, C.P. and SIEMENS, G., 2014. Shared task on prediction of dropout over time in massively open online courses. In *Proceedings of the 2014 Conference on Empirical Methods In Natural Language Processing Workshop on Modeling Large Scale Social Interaction In Massively Open Online Courses* (Doha, Qata, 2014), 39-41.

[10] SHIMAZOE, J. and ALDRICH, H., 2010. Group Work Can Be Gratifying: Understanding & Overcoming Resistance to Cooperative Learning. *College Teaching 58*, 2, 52-57.

[11] YANG, D., SINHA, T., ADAMSON, D., and ROSE, C.P., 2013. "Turn on, Tune in, Drop out": Anticipating student dropouts in Massive Open Online Courses. In *Proceedings of the Workshop on Data Driven Education, Advances in Neural Information Processing Systems* (2013).

Exploring Causal Mechanisms in a Randomized Effectiveness Trial of the Cognitive Tutor

Adam C Sales
Carnegie Mellon University
Pittsburgh, PA, USA
acsales@cmu.edu

John F Pane
RAND Corporation
Pittsburgh, PA, USA
jpane@rand.org

ABSTRACT

Cognitive Tutor Algebra I (CTAI), published by Carnegie Learning, Inc., is an Algebra I curriculum, including both textbook components and an automated, computer application that is designed to deliver individualized instruction to students. A recent randomized controlled effectiveness trial, found that CTAI increased students' test scores by about 0.2 standard deviations. However, the study raised a number of questions, in the form of evidence for treatment-effect-heterogeneity. The experiment generated student log-data from the computer application. This study attempts to use that data to shed light on CTAI's causal mechanisms, via principal stratification. Principal strata are categories of both treatment and control students according their potential CTAI usage; they allow researchers to estimate differences in treatment effect between usage subgroups. Importantly, randomization satisfies the principal stratification identification assumptions. We present the results of our first analyses here, following prior observational results. We find that students who encounter more than the median number of sections experience higher effects than their peers who encounter fewer, and students who need more assistance experience *lower* effects than their peers who require less.

Keywords

Causal Mechanisms, Principal Stratification, Intelligent Tutors, Bayesian Hierarchical Models

1. INTRODUCTION

The Cognitive Tutor Algebra I (CTAI) is a technology-based educational intervention that hopes to improve algebra I instruction by individualizing instruction to students needs, providing instant performance feedback, and implementing cognitive theories in mathematics education. [6]

Recently, a randomized controlled effectiveness trial, estimated the effect of a school's adoption of CTAI, under authentic conditions, on its students scores on an algebra proficiency exam. The results were reported in [4]. The study found that CTAI significantly increased test scores for 9th grade students in the second year of implementation, but was unable to detect effects in the experiment's first year, or in the 8th-grade group. These results raise a further question: by what mechanism, and for which students, does CTAI increase achievement? What usage patterns lead to higher effects? Can usage patterns explain the observed treatment effect heterogeneity?

The effectiveness trial produced extensive student usage data, as the computer program logged students' activity. In this paper, we begin use this data—in particular, usage data from the 2nd-year high school sample that apparently experienced a substantial CTAI effect—to explore the relationship between student usage and causal effects.

In doing so, we are guided by a previous study, [7] which (in one model specification) regressed post-test scores on CTAI usage variables, alongside student covariates and pre-test scores. That paper was aimed at post-test prediction, not causal inference, but it is of use in generating causal hypotheses: are there different effects for students who use CTAI for different amounts of time? Or for students who require more assistance from the program? Or for students who encounter more sections? This paper is a preliminary inquiry into these questions—more an exposition of the types of results that are possible than a full analysis—future work will delve more deeply into the data.

The data from the CTAI effectiveness study is invaluable for testing these hypotheses: due to its randomization design, we can draw causal conclusions without heroic assumptions. To do so, we will make use of the statistical framework of principal stratification, which we will describe in the following section. The next section will describe our models in detail, and results and conclusions will follow.

2. PRINCIPAL STRATIFICATION

Following [8], we conceptualize causal inference in terms of counterfactuals: comparing what students would have experienced with CTAI with what they would have experienced in its absence. In particular, if Y is the outcome of interest, in our case, post-test scores, we may define two "potential outcomes" for each subject: $Y_i(0)$ is what a subject i would score on the post-test if i's school were assigned to the control condition, and $Y_i(1)$ is what I would score if her school were assigned to treatment.

Principal stratification (PS) [2] is an approach to modeling a categorical or discrete post-treatment variable M within the potential outcomes framework. When treatment assignment Z is binary, each subject i has two potential values of M: $M_i(0)$—the value of M that would be observed under the control condition—and $M_i(1)$, what would be observed under the treatment condition. These define subgroups—principal strata—within which causal effects may be defined. In particular, a principal causal effect is

$$Y(1) - Y(0)|M(1) = m, M(0) = m' \quad (1)$$

that is, the effect of Z on Y among those subjects with particular potential outcomes for M of m and m'.

In this study, following [3], we use principal stratification to examine some hypothesized causal mechanisms of CTAI. For instance, consider the usage variable $totalTime$: the total amount of time students spend working CTAI problems. Since $totalTime$ is continuous, we begin by dichotomizing it; for the sake of simplicity, let $\mu = median(totalTime)$ and $M = \mathbb{1}_{[totalTime>\mu]}$. We can define four principal strata. The first is comprised of those students who, if assigned to CTAI, would use it for more time than μ—$M(1) = 1$—but if assigned to the control condition would use it less, $M(0) = 0$. Next, consider the group $M(1) = 0$; $M(0) = 1$, those students who use CTAI for less time because of their treatment assignments. The remaining two groups are $M(1) = 0$; $M(0) = 0$ and $M(1) = 1$; $M(0) = 1$, those students who would use CTAI less for less, or more, time than m regardless of treatment assignment. By examining differences between the average treatment effects in the four groups, we can learn how CTAI's impact varies for different usage patterns.

Randomization allows us to estimate principal effects as the average treatment minus control difference in gain scores within each estimated stratum. That is, randomization of treatment assignment leads to identification of principal effects: the effect of Z within principal strata. On the other hand, the difference in treatment effects between principal strata does not necessarily estimate a causal quantity. Randomization does not identify students' counterfactual gain scores had they been in alternative principal strata. That being said, differences in treatment effects across strata can suggest causal mechanisms.

Fortunately, the CTAI study's design substantially simplifies the PS analysis, by eliminating two of the principal strata. Students in the control group had (for the most part) no access to the CTAI program. Therefore, we can safely assume that for all students, $M(0) = 0$. This leaves two principal strata, $M(0) = 0$; $M(1) = 0$, and $M(0) = 0$; $M(1) = 1$—that is, the students who, if assigned to treatment, would use CTAI for more time than m and those who would not. Only one of the potential values of M is directly observed; in particular, $M(1)$ is unknown for subjects in the control group. Stated differently, the values $M(1)$ are missing for students in the control group, but they may be imputed because the "missingness mechanism," treatment assignment, is random, or ignorable. Therefore, randomization of treatment assignment allows us to identify members of each principal stratum, and effects of treatment within those strata.

3. MODELING STRATA AND OUTCOMES

In this preliminary study, we considered three of the usage variables previously modeled as predictors in [7]: $totalTime$, the total amount of time students spent working CTAI problems, $numSec$, the number of sections each student encountered, and $assistance$, the average sum of hints and errors per problem for each student. We ran a separate PS model for each usage variable, but all three PS models had the same form. Each PS model itself was a combination of two multilevel models. The first, fit only within the treatment group, modeled the usage variable M as a function of covariates X_t. This model was used to estimate the usage that control students would have experienced had they been assigned to treatment. The second model used the results of the first model, and a somewhat larger set of covariates X_y, to estimate the effect of random assignment to CTAI in each of the principal strata.

3.1 Usage Model

Modeling each usage variable was a four-step process: first, we calculated the variable's values from the available data; next, we transformed those values so that their observed distributions would be closer to a normal distribution; next, we modeled the transformed variables as a linear function of covariates X_t, and finally, we dichotomized the model's output, to define and estimate principal strata.

As students used CTAI, the program recorded timestamps at the beginning and end of each problem. The difference between these two is the amount of time the student spent on each problem, recorded in milliseconds. The sum of was the variable $totalTime$. The distribution of $totalTime$ was heavily skewed rightward, so we transformed it to ease the modeling process. The transformation that resulted in a distribution whose histogram appeared approximately normal was a box-cox transformation with a parameter of 0.3 [1].

Next, we modeled the transformed $totalTime$ as a function of a set of covariates X_t containing dummy variables for the state in which the school was located, the student's grade, race, sex, special education status, free or reduced-price lunch status and pretest scores, along with missingness indicators. Formally, the model was

$$totalTime_{ijk} = \alpha + X_{ti}^T\beta + \epsilon_{ijk} + \eta_{jk} + \nu_k \quad (2)$$

where α and β are, respectively, an intercept and a vector of coefficients estimated from the data, $\epsilon_{ijk} \sim N(0, \sigma_{st})$ is a student-level random error, $\eta_{jk} \sim N(0, \sigma_{tt})$ is a random effect for teacher, and $\nu \sim N(0, \sigma_{ut})$ is a random effect for school. The variance parameters σ_{st}, σ_{tt} and σ_{ut} are estimated from the data. In other words, $totalTime$ was modeled as multilevel, with students nested within teachers, nested within schools.

The transformed $totalTime$ values, or, in the case of the control sample, their predictions, gave rise to a dichotomous variable M, which took the value of 1 if $totalTime$ or its prediction is greater than its observed median of about 22 hours over the course of the year. The variable M defined two principal strata: those students with $M(1) = 1$ and those with $M(1) = 0$.

CTAI also automatically collected data on the number of

hints and errors students request or make. Following [7], we normalized hints and errors by section. Next, we averaged the normalized values by student, producing average assistance per problem, or *assistance*. We transformed *assistance* in the same was as *totalTime*. Next, we modeled *assistance* with equation (2), and dichotomized the results using their observed median, 0.076, which, due to the prior normalization, is not a whole number.

The third usage variable we considered here is *numSec*, the number of sections students encountered on CTAI. We transformed *numSec* with a natural logarithm, modeled it with equation (2), and dichotomized it with its median, 27 sections.

3.2 Outcome Model

For each dichotomized usage variable M, we fit a multilevel linear model to estimate principal effects of CTAI treatment on post-test scores. The post-test from the CTAI effectiveness study is the Algebra Proficiency Exam. It was analyzed with item-response-theory, and its reported scores have a mean of 0 and a standard deviation of 1, so regression coefficients may be interpreted as effect sizes [4]. To account for pre-test scores, while avoiding measurement-error concerns, we modeled students' gain scores, $\texttt{diff}_{ijk}$, the difference between their post-test and pre-test scores. The student-level model, then, was

$$\begin{aligned}\texttt{diff}_{ijkm} =& \alpha' + X_{yi}^T\gamma + \lambda M_{ijkm} + \tau Z_{km} \\ &+ \kappa Z_{km} M_{ijkm} + \epsilon'_{ijkm} + \eta'_{jkm} \\ &+ \nu'_{km} + \zeta_m \end{aligned} \tag{3}$$

Here α', ϵ', η', and ν' are, respectively, an intercept, and random effects for individual, teacher, and school. The apostrophes indicate that these are distinct from their analogues in equation (2). There is an additional random effect ζ for "match," accounting for the matched-pair randomization design. X_t is a vector of covariates equivalent to those in (2), with the addition of standardized test scores from the prior two years. The principal effects emerge from the coefficients τ and κ: τ is the average effect in the $M(1) = 0$ group, and $\tau + \kappa$ is the average effect in the $M(1) = 1$ group. Finally, λ is the difference in $Y(0)$ between the $M(1) = 1$ and $M(1) = 0$ groups.

Models (2) and (3) were fit simultaneously in JAGS [5], a Bayesian Gibbs sampler. To facilitate Bayesian model fitting, we provided weakly informative priors on all of the model parameters.

4. RESULTS

	Estimate	SE	95% Interval
$M(1) = 0$	0.12	0.06	(0.01,0.25)*
$M(1) = 1$	0.32	0.27	(-0.21,0.85)
Difference	0.20	0.27	(-0.32,0.74)

Table 1: Results for *totalTime*. Point estimates for effect size, standard errors and 95% credible intervals for the average treatment effects in two principal strata, denoted $M(1) = 1$ and $M(1) = 0$, as well as the difference between the two.

We present results for each of the three usage variables we considered. For each variable, we present the average treatment effect for subjects in the $M(1) = 0$ stratum—that is, students whose usage under the treatment condition was, or would be, less than the observed median—the effect for students in the $M(1) = 1$ stratum, and the difference between the two effects. For each effect, we present a point estimate, equivalent to the mean of the posterior distribution, a standard error—the standard deviation of the posterior—and a 95% credible interval, representing the 0.0275 and 0.975 quantiles of the posterior. Effects whose credible interval does not include 0 are marked with an asterisk.

Like [7], we were unable to establish that students who spend more time using CTAI gain more from its use. The relevant results are available in Table 1. We estimated an effect size of 0.12 in the low-time group $M(1) = 0$, and 0.32 in the high-time group $M(1) = 1$. However, the standard errors were too large to draw strong conclusions.

	Estimate	SE	95% Interval
$M(1) = 0$	-0.02	0.07	(-0.16,0.11)
$M(1) = 1$	0.30	0.13	(0.13,0.47)*
Difference	0.32	0.09	(0.14,0.49)*

Table 2: Results for *numSec*

On the other hand, as seen in Table 2, students who encountered a greater number of sections (or would have, had they been assigned to treatment) experienced a much larger effect than those who encountered fewer sections. The effect size for students who encounter more than the median number of sections is, with 0.95 probability, between 0.13 and 0.47—a very large effect. This is about 0.32 higher than for the students who encountered fewer sections, for whom there was no discernible effect at all.

	Estimate	SE	95% Interval
$M(1) = 0$	0.30	0.08	(0.14,0.45)*
$M(1) = 1$	0.12	0.09	(-0.07,0.30)
$M(1) = 2$	-0.08	0.08	(-0.23,0.06)
Difference (0–1)	-0.18	0.09	(-0.34,-0.00)*
Difference (1–2)	-0.20	0.09	(-0.36,-0.01)*

Table 3: Results for *assistance*

Lastly, we suspected that the relationship between assistance and CTAI effect might not be monotonic. That is, it might be that the effect of CTAI is low for students who request many hints and make many errors, but high for those with a medium amount, or vice versa. For that reason, we split the variable at the 1/3 and 2/3 quantiles, and estimated three principal effects. Our suspicion proved false, however, and the result was similar to what was reported in [7]: higher hints and errors corresponded to lower CTAI effects. For students who requested few hints and made few errors, the effect was between 0.14 and 0.45, while 95% intervals for the other two strata included 0. Ninety-five percent intervals on the difference from one strata two the next were entirely negative.

5. DISCUSSION

This work was a first look at causal modeling with usage variables from a randomized experiment of educational soft-

ware. We showed that without additional identification assumptions, researchers can use log data to form a deeper understanding of their software's effect. That being said, this work is preliminary, both because the statistical models we used may be improved, and because much more information is available in the CTAI log data.

In this paper, we focused on three hypotheses that were suggested in [7]. That paper used a linear regression model, fit using a convenience sample of CTAI users, to show that certain usage variables, among which are the total amount of time students spend solving problems, the number of sections students encounter, and the assistance the software provides them, can predict standardized test scores, even after controlling for a number of baseline covariates. With some very strong assumptions, one may interpret [7]'s results as causal: that seeing more sections, for instance, causes students to achieve higher test scores.

In our design, by contrast, the estimated treatment effects—comparisons between treatment and control students—are inherently causal due to the randomization design. The principal stratification approach allows us to reliably estimate causal effects within the strata. That said, this approach largely replicates the results from [7]. Students who spend more time working on CTAI problems seem to experience a larger effect, but this conclusion is ultimately unclear: the credible interval of the difference in effects between students who use the program for more time and those who use it for less contains 0. On the other hand, we found that students who encounter more sections do indeed experience larger effects. One reason for this result may be that the effect a CTAI user feels is particular to the skills the user practices—students who encounter a wider array of sections learn more from CTAI, and their performance on a wider array of sections of the posttest is improved. At the same time, students who required more assistance per problem—that is, asked for more hints and made more errors—experienced a smaller effect than their peers who required less assistance. This may be for a number of reasons. For instance, perhaps students who need more assistance per problem are struggling more, and have a greater need for a teacher's help. Alternatively, students who ask for a lot of hints and make a lot of mistakes may not be trying their hardest on CTAI, and for that reason may not experience the same rewards from CTAI. More research and data analysis is necessary to properly interpret these results.

Along those lines, we plan a number of future analyses. First, improved models may help us understand the relationships that this paper explores. For instance, dividing the usage variables into three or more categories may be more illuminating than the two categories we explore here. Additionally, it may be useful to match section- or unit-specific usage to appropriate items on the posttest.

Further along, we hope to discover and define interesting multivariate principal strata, perhaps as the result of a cluster analysis of the high-dimensional usage data.

Finally, after cultivating a more complete understanding of the usage patterns that lead to higher CTAI effects, we can explore treatment-effect heterogeneity. In particular, we may be able to answer why in the first year of implementation CTAI did not seem to boost test scores, but in the second year it did. Was differential usage to blame?

In the meantime, this paper uses rigorous causal methods to confirm some previous hypotheses about CTAI's causal mechanisms, and points a way forward for future work modeling usage variables in experimental designs.

6. ACKNOWLEDGMENTS

This work is supported by the United States National Science Foundation Grant #DRL-1420374 to the RAND Corporation and by the Institute of Education Sciences, U.S. Department of Education, through Grant R305B1000012 to Carnegie Mellon University. The opinions expressed are those of the authors and are not intended to represent views of the Institute or the U.S. Department of Education or the National Science Foundation. Thanks to Steve Fancsali, Steve Ritter, and Susan Berman for processing and delivering the CTAI usage data.

References

[1] G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[2] C. E. Frangakis and D. B. Rubin. Principal stratification in causal inference. *Biometrics*, 58:21–29, 2002.

[3] L. C. Page. Principal stratification as a framework for investigating mediational processes in experimental settings. *Journal of Research on Educational Effectiveness*, 5(3):215–244, 2012.

[4] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 2013.

[5] M. Plummer. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, 2003.

[6] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14(2):249–255, 2007.

[7] S. Ritter, A. Joshi, S. E. Fancsali, and T. Nixon. Predicting standardized test scores from cognitive tutor interactions. In *Proc. of the 6 th International Conf. on Educational Data Mining*, 2013.

[8] D. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology; Journal of Educational Psychology*, 66(5):688, 1974.

Confounding Carelessness? Exploring Causal Relationships Between Carelessness, Affect, Behavior, and Learning in Cognitive Tutor Algebra

Stephen E. Fancsali
Carnegie Learning, Inc.
437 Grant Street, 20th Floor
Pittsburgh, PA 15219 USA
1.888.751.8094 x219
sfancsali@carnegielearning.com

ABSTRACT

Studies have found positive correlations between affective states (e.g., confusion, boredom) and learning outcomes in educational technologies like ASSISTments and Carnegie Learning's Cognitive Tutor. The adage that "correlation does not imply causation" is especially apt in light of these observations; it seems counterintuitive that increasing student boredom or confusion (e.g., designing systems that bore or confuse students) will benefit learning. One hypothesis to explain positive correlations between boredom and learning suggests that carelessness is a "confounding" common cause of boredom and another construct linked to learning. We consider a Cognitive Tutor Algebra dataset in which boredom and confusion are positively correlated with learning. Prior causal modeling of this data suggests that various behavioral and affective features (e.g., boredom and gaming the system) share unmeasured common causes. We provide a correlational analysis and causal models of this data that situate carelessness among behaviors and affective states to determine whether (and how) carelessness plays a confounding role.

Keywords

causal models, causal discovery, structural equation modeling, carelessness, boredom, confusion, affect, gaming the system, off-task behavior, Cognitive Tutor, intelligent tutoring systems

1. INTRODUCTION

Recent research in educational data mining has led to the development of sensor-free, data-driven approaches to "detect" various behavioral and affective features from logs of learner interactions with technologies like intelligent tutoring systems (ITSs). Since such approaches to detecting phenomena like "gaming the system" [3-4], off-task behavior [1], and affective states [5] have been validated against field observations of learner behavior, a natural next step for researchers has been to use the predictions of detectors as inputs to predictive models of substantive learning outcomes in what have been called "discovery with models" approaches [6]. Such approaches have sought to answer questions about whether the tendency of learners to game the system or become bored using a system are predictive of outcomes like post-tests and standardized test scores (e.g., [11, 17]).

Our recent work [13] advocates seeking causal knowledge about learner behavior, affect, and learning, even when faced with non-experimental data, and that graphical causal models and data-driven search for their structure [22] provide an avenue for *causal* discovery with models. Findings using data from Carnegie Learning's Cognitive Tutor (CT) ITS [19] suggested that most affect and behavior variables shared unmeasured common causes.

The present work integrates detectors of carelessness into this work [13]. Carelessness was correlated with a variety of affective phenomena in an ITS with features similar to CT (e.g., [21]) and has been hypothesized to play a causal role among affective states as well (e.g., as a cause of boredom [17] or effect of boredom [8]). Other work emphasizes relationships between engagement and carelessness [9-10]. Our findings suggest a causal link between concentration and carelessness and possible causal links between confusion, gaming the system, and carelessness.

2. PRELIMINARIES

2.1 Motivation & Outline

Recent studies (e.g., [13, 17]) observe positive correlations between learning and the propensity to be in affective states like boredom and confusion, but it seems counterintuitive that increasing student boredom or confusion is likely to benefit learning (i.e., that these correlations are because of causal links). Several hypotheses have been proffered to explain such positive correlations. One hypothesis, for the ASSISTments system [14], is that learners become bored when they make careless mistakes and are required to work through step-by-step breakdowns of math problems [17]; learners with greater knowledge are more likely to be careless and bored, but since they are capable learners they will have better learning outcomes, providing a possible explanation for a positive correlation between boredom and learning.

Further, causal models of affect, behavior and learning in CT Algebra finds that boredom and gaming the system behavior are negatively correlated and suggest that they share an unmeasured (or latent) common cause (i.e., a "confounding" variable) [13]. Boredom's negative correlation with gaming the system, and gaming behavior's negative correlation with learning helped to explain the overall positive correlation of boredom and learning in that study. This same study also found a positive correlation between confusion and learning, and causal models suggested that confusion and gaming may be confounded. The hypothesis of [17] about carelessness may be appropriate for CT; incorrect responses despite knowledge will lead students to be presented more practice on skills they already know because CT will decrease its

estimate of skill mastery based on incorrect responses, and this could lead to boredom.

Recent work has focused on modeling carelessness [20] in systems like ITSs by using context-sensitive models to predict whether particular incorrect responses are likely examples of "slips" in which students answer incorrectly despite knowing a skill [2], and have explored correlations between carelessness and affective states (e.g., [21]).

We now introduce CT Algebra and detector models. We then review graphical causal models and data-driven structure search before explaining prior work and presenting novel causal models that incorporate carelessness. We conclude with discussion.

2.2 Cognitive Tutor (CT) Algebra

Carnegie Learning's CT is an ITS for mathematics used by hundreds of thousands of learners every year across the United States and internationally. CT breaks down mathematics subject areas like algebra into fine-grained skills or knowledge components (KCs), the mastery of which is used to determine learner progress through a series of topical sections that comprise broader units. Each section is comprised of multi-step problems that allow for the assessment of student progress toward mastery of fine-grained KCs.

CT assesses KC mastery using a probabilistic framework called Bayesian Knowledge Tracing (BKT) [12]. BKT assesses learner progress to mastery by assuming that a learner is either in the "unknown" state for a KC or the "known" state for a KC (i.e., KC mastery) and uses observations of practice opportunities for each KC to predict the state of a learner is at any given time. To make this prediction, BKT provides for four parameters for each KC: (1) the probability of prior knowledge or mastery of the KC, (2) the probability of a transition from the unknown to the known state at a given KC practice opportunity, (3) the probability that a learner guesses (i.e., is in the unknown state but answers correctly), and (4) the probability that the learner "slips" (i.e., has mastered a KC but provides an incorrect response).

2.3 Affect, Behavior, & Carelessness

Educational data mining researchers seek to avoid obtrusive, costly, and non-scalable sensor-based methods for measuring learner (dis-) engagement and affect with systems like ITSs by developing data-driven predictive models, frequently referred to as "detectors" that rely only on features that can be "distilled" from fine-grained log data. Detector models use machine learning methods applied to distilled features to make predictions about whether particular learner interactions with a system are likely to be instances of particular types of behavior. Detector models are validated against field observations in real classrooms. For correlational and causal modeling, we quantify levels of behavior per student by calculating the proportions of problem-solving steps deemed to be likely the result of behaviors like gaming the system or off-task behavior, which we now briefly explicate.

Gaming the system [3-4] refers to behavior in which learners attempt to make progress through content without genuinely learning or mastering appropriate skills (e.g., by incorrectly providing numbers within problem statements). A robust finding of previous efforts is that there is evidence that gaming the system is a cause of decreased learning [13]. Off-task behavior refers to learner disengagement from the learning environment and learning [1]. Recent efforts did not find evidence for a causal link between off-task behavior and learning.

Evidence also suggests that affective states play an important role in learning (e.g., [18]). Detector models similar to those for gaming the system and off-task behavior have been developed for affective states like boredom, confusion, and engaged concentration [5]. Modeling efforts for a CT Algebra dataset provided a somewhat complicated causal picture; while boredom and confusion may be *negatively* correlated with another factor that causes *decreased* learning, gaming the system, (hence positively correlated with learning), there are likely unmeasured common causes of these states and gaming the system.

Learner carelessness has been discussed as problematic in classrooms since at least the 1950s [21]. Other work identifies carelessness as a problem even among high-performing students [9-10]. Recent work on data-driven detector models [20] seeks to operationalize carelessness by focusing on the notion of "slipping," when learners answer incorrectly despite knowing a skill. In standard BKT, the parameter for slipping remains constant per KC over time; contextual models of guessing and slipping predict whether particular correct and incorrect responses are likely the result of learners guessing or slipping based on aspects of their performance [2]. The contextual slip model that predicts whether particular incorrect responses are instances of slipping is built in the same manner as other detector models. Operationalized as contextual slipping, carelessness can be quantified on a per learner basis by calculating the mean probability with which contextual slip models predict that incorrect actions are examples of slipping [21].

2.4 Graphical Causal Models & Model Search

We adopt directed acyclic graphs (DAGs) to represent causal relationships among variables we seek to model. We consider the context of linear relations among variables and multi-variate Gaussian joint probability distributions, where DAGs imply conditional independence constraints on observed joint distributions and covariance matrices. The set of DAGs consistent with a set of independence constraints, assuming that there are no unmeasured common causes of measured variables, comprise an equivalence class of graphs, represented by a graphical object called a pattern. Patterns and other equivalence classes of graphs can be inferred from data by asymptotically reliable algorithms developed (e.g., the TETRAD[1] project) over the past 20+ years.

We deploy the constraint-based PC algorithm to learn a pattern from data, making the strong assumption of no unmeasured common causes of measured variables [22]. From a pattern, we can choose a DAG member of the equivalence class to specify a linear structural equation model (SEM). Allowing for unmeasured common causes, we consider an equivalence class of graphs, represented by Partial Ancestral Graphs (PAGs), learned using the FCI algorithm [22]. FCI is similar to PC, but PAGs have a richer set of edges between two variables X and Y in a PAG [13, 22]:

- X o—o Y: (1) X is an ancestor (i.e., cause) of Y; (2) Y is a cause of X; (3) X and Y share a latent common cause; (4) either (1) & (3) or (2) & (3).
- X o→ Y: Either X is a cause of Y; X and Y share a latent common cause; or both.
- $X \leftrightarrow Y$: X and Y share a latent common cause in every member of the equivalence class represented by this PAG.

[1] freely-available at http://www.phil.cmu.edu/projects/tetrad/

- $X \rightarrow Y$: X is an ancestor/cause of Y in every member of the equivalence class represented by this PAG.

3. DATA + PRIOR WORK

Our data are logs for a sample of 102 adult, higher education learners using CT Algebra. We consider log data over roughly 337,000 learner actions in a module of five units concerning linear equations and inequalities, relatively late in the course. We also have a pre-test score (*Module Pre-Test*) and a *Final Exam* score for the entire algebra course, which is our learning outcome.

Assuming no unmeasured common causes of variables, causal models of this data [13] illuminated one possible explanation for the positive correlations between both *Boredom* and *Confusion* and *Final Exam*: both may cause decreased *Gaming the System* behavior, behavior which is found to cause decreased learning. While *Confusion* may cause decreased *Gaming the System* (e.g., *Confusion* being an affective state in which learners are unlikely to be able to "game"), there are reasons to suspect that this correlation and others arise due to confounding common causes.

Relaxing the assumption of no unmeasured common causes and allowing affect and behavior to co-occur, the FCI algorithm found a *robust* causal link between gaming and learning; all other links in the PAG causal model from prior work are at least possibly confounded. This fact and several common cause hypotheses in the literature explaining positive links between *Confusion* and *Boredom* and learning lead us to consider *Carelessness*.

4. MODELING CARELESSNESS

4.1 Correlational Analysis

Carelessness is positively correlated with both *Module Pre-Test* ($r = 0.36$, $p < .001$) and *Final Exam* ($r = 0.56$, $p < .001$), consistent with results that careless behavior is common even among high-performing math learners [9-10]. Correlations of *Carelessness* to other affective and behavioral variables are presented in Table 1. These results are largely consistent with those in previous work analyzing the relationship between *Carelessness* and affect [21].

Table 1. Pairwise correlations of *Carelessness* and other variables representing "detected" behavior and affective states (*p < .05; ***p < .001)

Variable / Construct	Pearson Correlation
Boredom	0.13
Confusion	0.48***
Engaged Concentration	0.75***
Gaming the System	-0.74***
Off-Task Behavior	-0.25*

4.2 Causal Models

Rather than attempt to specify and test "by hand" a multitude of alternative models that posit different causal roles for *Carelessness*, we adopt a search strategy. Assuming that affective states (including *Carelessness*) causally precede behavioral variables, the PC algorithm learns the DAG causal structure of the estimated linear SEM of Figure 1. This model fits the data ($\chi^2(19) = 23$; $p = .22$) [7] and is similar to that the model found in previous work under the same assumptions [13]. We focus on three elements of it.

First, *Engaged Concentration* is inferred to be a cause of *Carelessness*, consistent with the high correlation in the Scatterplot Study [21], and hypotheses due to Clements [10] about the relationship between engagement (i.e., *Engaged Concentration*) and *Carelessness*. San Pedro, et al. note the positive link between confidence and *Carelessness* found by Clements and posit that an engaged learner of only average knowledge might become overly confident in their ability and careless [16, 21]. This explanation suggests an intermediary along this causal pathway, a topic for future research.

Second, *Carelessness* is inferred to be a common cause of *Confusion* and *Gaming the System*, with increased *Carelessness* leading to increased *Confusion* and less *Gaming the System*. *Carelessness* as a common cause of these two variables is consistent with models in [13] in which an edge *Confusion* o→ *Gaming the System* indicated the possible presence of an unmeasured (i.e, confounding) common cause. The strong positive relationship between *Engaged Concentration* and the inferred cause of *Confusion*, *Carelessness*, provides a plausible explanation for the positive correlation of *Confusion* and learning, but this model does not suggest we pursue interventions that increase learner *Confusion*, though recent literature suggests that, in some contexts, *Confusion* may be beneficial for learning (e.g., [15]).

With respect to the other effect of *Carelessness* in Figure 1, *Gaming the System*, it is possible that there is a negative causal connection, as presumably gaming behavior is the result of at least a certain amount of non-careless affect and corresponding behavior, as learners must provide roughly appropriate responses to math problems if they are to, in fact, "game the system." However, it is also plausible that *Carelessness* and *Gaming the System* share a confounding common cause.

Relaxing the assumption of no unmeasured common causes and assuming only that *Module Pre-Test* precedes all affective and behavioral variables, all of which precede *Final Exam*, FCI learns the PAG causal model in Figure 2, with +/- signs to remind the reader of parameter estimates in Figure 1. Contrary to the model of Figure 1, either *Confusion* is a cause of *Carelessness*, or they share an unmeasured common cause. The direction of the link between *Confusion* and *Carelessness* is sensitive to the "ordering" of affective and behavioral variables. However, under nearly all combinations of behavioral and affective variable orderings and groupings, *Engaged Concentration* is a cause of *Carelessness*, consistent with past hypotheses [9-10] and correlational analyses [21]. While we infer that *Carelessness* and *Gaming the System* share an unmeasured common cause, relationships between variables like *Carelessness* and *Gaming the System* may be confounded, not only by other unmeasured phenomena, but by the underlying phenomenon itself since we provide only noisy measures using detector models.

5. DISCUSSION

We provide evidence for the hypothesis that concentration leads to (i.e., causes) careless mistakes, and this causal inference is robust under a variety of assumptions. Contrary to some hypotheses [8, 17], we do not find evidence for a causal link between *Carelessness* and *Boredom* in CT Algebra. However, that hypothesis of [17] was made with respect to the ASSISTments system. Future research should take on the problem of learning causal models from available observational data from systems like CT and ASSISTments to determine under what circumstances causal inferences of the sort we consider here generalize across

sub-populations using the same instructional system as well as different systems within the same (or different) domain.

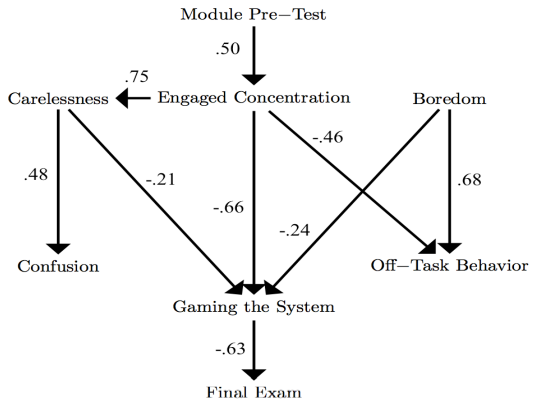


Figure 1. Estimated SEM incorporating *Carelessness*

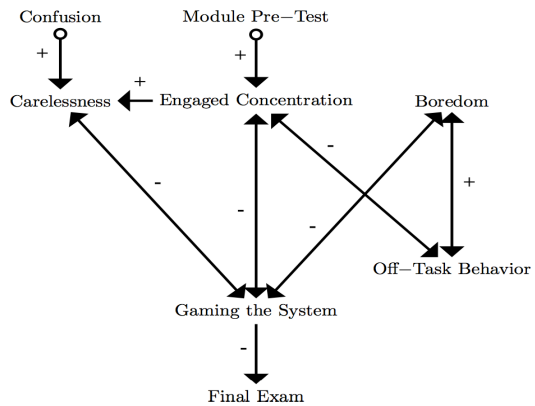


Figure 2. PAG causal model incorporating *Carelessness*

6. ACKNOWLEDGMENTS

The author gratefully acknowledges Ryan S.J.d. Baker, Susan R. Berman, Ryan Carlson, Sujith M. Gowda, Steven Ritter, and Charles Shoopak for providing code, assistance, and/or comments.

7. REFERENCES

[1] Baker, R.S.J.d. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proc. of ACM CHI 2007: Computer-Human Interaction* (San Jose, CA, April 28 – May 3, 2007). ACM, New York, 1059-1068.

[2] Baker, R.S.J.d., Corbett, A.T., Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian Knowledge Tracing. In *Proc. of ITS 2008* (Montreal, Canada, 2008). 406-415.

[3] Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R. 2008. Developing a generalizable detector of when students game the system. *User Model. User-Adap.* 18 (2008), 287-314.

[4] Baker, R.S.J.d., de Carvalho, A. M. J. A. 2008. Labeling student behavior faster and more precisely with text replays. In *Proc. of EDM 2008* (Montreal, 2008). 38-47.

[5] Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G.W., Ocumpaugh, J., Rossi, L. 2012. Towards sensor-free affect detection in Cognitive Tutor Algebra. In *Proc. of EDM 2012* (Chania, Greece, 2012). 126-133.

[6] Baker, R.S.J.d., Yacef, K. 2009. The state of educational data mining in 2009: a review and future visions. *Journal of Educational Data Mining* 1 (2009), 3-17.

[7] Bollen, K. 1989. *Structural Equations with Latent Variables*. John Wiley & Sons.

[8] Cheyne, J.A., Carriere, J.S., Smilek, D. 2006. Absent-mindedness: lapses of conscious awareness and everyday cognitive failures. *Conscious Cogn* 15 (2006), 578-592.

[9] Clements, M.A. 1980. Analyzing children's errors on written mathematical tasks. *Educational Studies in Mathematics* 11, (Feb. 1980), 1-21.

[10] Clements, M.A, 1982. Careless errors made by sixth-grade children on written mathematical tasks. *Journal for Research in Mathematics Education* 13, (Mar. 1982), 136-144.

[11] Cocea, M., Hershkovitz, A., Baker, R.S.J.d. 2009. The impact of off-task and gaming behavior on learning: immediate or aggregate? In *Proc. of AIED 2009* (Brighton, UK, 2009). 507-514.

[12] Corbett, A.T., Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap.* 4 (1995), 253-278.

[13] Fancsali, S.E. 2014. Causal discovery with models: behavior, affect, and learning in Cognitive Tutor Algebra. In *Proc. of EDM 2014* (London, UK, 2014). 28-35.

[14] Feng, M., Heffernan, N.T., Koedinger, K.R. 2009. Addressing the assessment challenge in an intelligent tutoring system that tutors as it assesses. *User Model. User-Adap.* 19 (2009), 243-266.

[15] Lehman, B., D'Mello, S.K., Graesser, A.C. 2012. Confusion and complex learning during interactions with computer learning environments. *Internet High. Educ.* 15 (2012), 184-194.

[16] Linnenbrink, E.A., Pintrich, P.R. 2003. The role of self-efficacy beliefs in student engagement and learning in the classroom. *Reading & Writing Quarterly: Overcoming Learning Difficulties* 19 (2003), 119-137.

[17] Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. 2014. Affective states and state tests: investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics* 1 (2014), 107-128.

[18] Pekrun, R., Goetz, T., Titz, W., Perry, R.P. 2002. Academic emotions in students' self-regulated learning and achievement: a program of quantitative and qualitative research. *Educ. Psychol.* 37 (2002), 91-106.

[19] Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T. 2007. Cognitive Tutor: applied research in mathematics education. *Psychon. B. Rev.* 14 (2007), 249-255.

[20] San Pedro, M.O.C.Z., Baker, R. S. J. d., Rodrigo, M. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *Proc. of AIED 2011* (Auckland, New Zealand). 304-311.

[21] San Pedro, M.O.C.Z., Baker, R.S.J.d., Rodrigo, M.T. 2014. Carelessness and affect in an intelligent tutoring system for mathematics. *Int J Artif Intell Educ* 24 (2014), 189-210.

[22] Spirtes, P., Glymour, C., Scheines, R. 2000. *Causation, Prediction, and Search*. 2nd Edition. MIT, Cambridge, MA.

Students at Risk: Detection and Remediation

Irena Koprinska, Joshua Stretton and Kalina Yacef
School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia
{irena.koprinska, joshua.stretton, kalina.yacef}@sydney.edu.au

ABSTRACT
Detecting students at risk of failing is particularly useful and desirable when it is done in a timely manner and accompanied with practical information that can help with remediation. In this paper we investigate ways to detect students at risk of failing early in the semester for timely intervention. The context of our study is a first year computer programming course. We explore whether the use of several student data sources can improve the process: submission steps and outcomes in an automatic marking system that provides instant feedback, student activity in the discussion forum Piazza and assessment marks during the semester. We built a decision tree classifier that is able to predict whether students will pass or fail their final exam with an accuracy of 87% mid semester. The obtained rules are useful and actionable for teachers and students, and can be used to drive remediation.

Keywords
Student performance prediction; classification of failing and passing students; automatic grading system; discussion board; assessment and feedback.

1. INTRODUCTION
Computer programming is an essential skill for software engineers and computer scientists, and also an increasingly required skill for graduates of many other disciplines, such as science, medicine, economics and business. Key factors in how well a person will learn programming include regular practice, as well as quick and efficient correction of mistakes and misconceptions. This means that students must be provided with tools that allow them not only to practice their programming skills but also to receive timely and useful feedback, which can be challenging, especially for large introductory computer programming courses. Lack of regular practice and sufficient feedback, often leads to students becoming uninterested or disheartened, and giving up learning to program.

Innovative technology-enhanced teaching and learning tools can help to solve this problem. In our introductory programming courses, we use a combination of an automatic marking and instant feedback system (PASTA) and a sophisticated discussion board (Piazza). These tools not only provide a semi-independent platform for students to build and test their knowledge, but also the opportunity for useful data collection and analysis, that can be used to improve teaching and learning.

In this paper we describe how data collected from these two sources, together with data from assessment marks, can be used to identify students who are at risk of failing and need more careful guidance, early enough so that remediation is possible. To illustrate this we use data from a large first year programming course. Specifically, the goal of this study is three-fold:

(i) to investigate whether students at risk of failing can be identified early enough in the semester for timely intervention, using machine learning prediction methods and information from three different sources: automatic marking system (PASTA), discussion board (Piazza) and assessment marks;

(ii) to investigate whether the information from the automatic marking system and discussion board helps improve the predictive accuracy, in comparison to just using the assessment marks;

(iii) to investigate how useful and actionable the produced rules are for remediation.

2. DATA SOURCES
An important characteristic of our study is that it triangulates data from three different sources that contain information not only about student performance, but also about student activities. Each source offers useful perspectives on student learning: progression in code writing and diagnostic (PASTA), interaction and engagement (Piazza), student performance (assessment marks).

PASTA is an automatic marking and feedback system developed in our school. It allows students to submit their solution for an assessment task online, checks this solution against public and hidden tests set by the teacher and provides immediate feedback to the student about which public tests were passed and failed. Students can then correct their mistakes and resubmit until all these public tests are passed. Feedback about the hidden tests is released when marking is completed, along with manual feedback.

The use of PASTA has resulted in better student engagement and improved learning, because of the instant feedback and multiple submissions. The PASTA data contains, for each task and student, all sequences of assessment submissions, the tests that were passed and failed (and why), the time stamps and mark obtained.

Piazza (www.piazza.com) is a mix of discussion board and wiki, allowing students and teachers to post notes, ask and answer questions individually or collaboratively. It was developed with the aim to connect students and promote classroom engagement. The Piazza data contains, for each student, the number of questions asked, answered and viewed, and the time and content of the posts.

The third data source includes all assessment marks during the semester and the final exam mark and is described in Sec.4.

3. PREVIOUS WORK
Previous work on predicting failure rate of students has been performed, normally by predicting exam grade just before the exam. Kotsiantis et al. [1] predicted final exam performance based on assignment marks throughout the semester in a distance education environment. This prediction was performed only at the end of the semester, and the attributes used would not allow for a

mid-semester prediction. They achieved an accuracy of 79% in predicting the final exam grade using an ensemble classifier.

Romero et al. [2] predicted final student marks based on Moodle usage data - the number of: quizzes passed and failed, assignments done, messages sent and read on the discussion board, and also the time spent on the assignments, quizzes and discussion board. They measured the geometric mean of the accuracies per class, which is an appropriate measure for imbalanced datasets as theirs, and achieved 67% with decisions trees. More recently, in [3] the same group investigated predicting the student grade (pass or fail) based on the student participation in a discussion forum, achieving accuracy of 75% using data collected in the middle of the semester and 90% using data collected at the end of the semester

Similar to student failure rates, student dropout rates have been studied, using a variety of assessment and non-assessment attributes. Agnihotri and Ott [4] predicted the likelihood of students dropping out of university after their first semester based on data provided such as admission information, placement tests and financial information. They were able to predict the retention of students with recall of 73% and precision 54%. Lykourentzou et al. [5] predicted dropout rate of students in an e-learning course environment, using the learning management system's extensive logs. They use machine learning techniques to achieve a 75-85% accuracy in the early sections of the course, and 97-100% accuracy in the final sections.

In this paper we extend previous work on predicting the students at risk of failing by using data from an automatic marking system and an advanced discussion board, in addition to assessment marks, from a computer programming course. We show how to define useful attributes from each data source, investigate if the student traces on the automatic marking system and discussion board help to improve predictive accuracy, and analyse how useful the prediction rules are for driving remediation.

4. CONTEXT OF THE STUDY

The study was conducted in the context of a large first year computer programming course with 223 students.

4.1 Assessment components

The six assessment components are summarised in Table 1.

Table 1. Assessment components

Homeworks	10	Weekly	Marks, Piazza
Task 1	2	Week 4	Marks, PASTA, Piazza
Task 2	6	Week 6	Marks, PASTA, Piazza
Practical test	16	Week 7	Marks, Piazza
Assignment	16	Week 12	Marks, PASTA, Piazza
Exam	50	Exam period	Marks, Piazza

The weekly homeworks were due before the computer lab and included multiple choice questions mainly requiring reading and understanding code. Their goal was to prepare students for the lab. The two tasks and assignment were programming assessments, with increasing level of difficulty, submitted via PASTA. Students were provided with skeleton code and required to complete the missing parts. The practical test involved writing code to solve five tasks with increasing difficulty levels in front of the computer. The exam, conducted at the end of the semester, was paper-based and required mainly writing code for solving problems. All assessment components were individual except for the assignment, where students had the choice of working individually or in pairs; 57% of students worked individually and 43% worked in pairs.

4.2 Predicted Variable

We predict the exam grade based on the marks of the other assessment components during the semester and the student activities on PASTA and Piazza. The two grades are defined as F (exam mark below 50, N=76), notF (exam mark of 50 and above, N=147). We chose the exam grade as a performance index because the exam: (i) is the major and most comprehensive assessment component, (ii) is conducted under strict conditions which minimises cheating, (iii) is independent of the other assessment components. The exam mark is highly correlated with the final mark (r=0.937).

4.3 Attributes

Table 2 summarises the student attributes that we defined to characterise student performance and activity.

Table 2. Attributes extracted from the three data sources

I. Assessment marks
homework_mark, task1_mark, task2_mark, prac_quiz_mark, assignment _mark (numeric) - mark (%) awarded for each assessment component
w7_homework_mark (numeric) – same as *homework_mark*, but only counting homeworks submitted before the end of week 7
II. PASTA activity – submission history
Starting and finishing times for assessments
task_start, task_finish, assignment_start, assignment_finish (numeric) – the average number of days before the due date that a student will start or finish the tasks or assignment
early_task, early_assignment (nominal, *yes/no*) - *yes* if the student starts the tasks faster than the average user; *no* otherwise
Multiple assignment submissions – improvement and consistency
marks_per_attempt_tasks, marks_per_attempt_assignments (numeric) – the average number of marks per PASTA submission of a task or assignment (including non-compiling submissions)
assignment_first_mark (numeric) - mark awarded for the student's first submission for the assignment
assignment_improvement (numeric) – the slope of the trendline of the student's assignment marks over each compiling submission; a larger number indicates rapid improvement
assignment_only_improvement (nominal, *yes/no*) - *yes* if the student's marks for compiling assignment submissions never decrease; *no* otherwise
assignment_consistency (nominal, multiple values) - goodness of fit (R^2) over each of the student's compiling submissions for the assignment, [-1, 1]; close to 1/-1 - linear increase/decrease in marks over submissions, close to 0 - random distribution of marks. Discretised as: *single* for single compiling submission, *none* for no assignment submission; *small/medium/high/very_ high* otherwise.
Pair work
pair_assignment (nominal, *yes/no*) - *yes* if the student worked in a pair for the assignment; *no* otherwise
Assignment submission statistics
single_submission (nominal, *yes/no/none*) - *yes* for one compiling assign. submission, *no* for more than one, *none* for no submission
assignment_total_submissions - total number of assignment submissions

assignment_compiling_submissions - number of compiling assignment submissions
III. Piazza activity – views, questions and answers
piazza_views, piazza_questions, piazza_answers (numeric) - number of posts viewed, questions asked or answered by the student on Piazza
piazza_activity (numeric) calculated as: (*piazza_views* + 10*(*piazza_ questions + piazza_answers)* + 5*(*piazza_posts - piazza_answers*)) / *total_posts*, where *piazza_posts* is the total number of contributions made by the student (asking or answering a question, or posting a comment), and *total_posts* is the total number of question threads on Piazza
piazza_active_viewer, piazza_active_questioner, piazza_active_answerer (nominal, *yes/no*) - *yes* if the student has an average or higher number of posts viewed, questions asked or questions answered; *no* otherwise
w7_piazza_views, w7_piazza_questions, w7_piazza_answers, w7_piazza_activity (numeric) and *w7_piazza_active_viewer, w7_piazza_active_questioner, w7_piazza_active_answerer* (nominal, *yes/no*) - same as the respective attributes without prefix *w7*, but only counting Piazza's posts up until the end of week 7

5. CAN WE PREDICT FAILING AND PASSING STUDENTS MID-SEMESTER?

We investigate whether we can predict accurately the students who will fail and pass the exam, based on the information available at two time points during the semester (and before the exam): in the middle of the semester (end of week 7) and at the end of the semester, just before the final exam (end of week 15). By the end of week 7, the students would have completed half of the homeworks, the two tasks and the practical test.

We built a Decision Tree (DT) classifier. One example in the data corresponds to one student and is described with the extracted attributes. An advantage of DTs is that the set of if-then rules they generate provides an explanation about the prediction which can be easily understood by teachers and students and directly applied.

Selecting appropriate attributes is very important for successful classification. Starting with the full set of attributes from Table 2, we used several methods for attribute subset selection [6] (manual and automatic such as correlation-based and wrapper, and combinations of them), before applying the DT algorithm. Although DTs have an inbuilt mechanism for attribute selection (only a subset of the attributes appear in the tree), their performance benefits from prior attribute subset selection. We report the best results. In all experiments, we used 10-fold stratified cross validation as an evaluation procedure.

Table 3 shows the accuracy results using data from all three sources and Figure 1 shows the generated DTs. The numbers in the brackets next to a leaf node in the trees give information about the coverage and correctness of the rule, e.g. (51/3) means that the rule covered 51 examples from the data, 3 of them we classified incorrectly and the remaining 48 were classified correctly.

Our results show that it is possible to predict the failing and passing students mid semester equally well as at the end of the semester – the two trees achieved the same accuracy, 87%. This accuracy is high enough to be useful in practical applications.

An examination of the confusion matrix shows that for the mid-semester tree the misclassifications are due to more failing students being classified as non-failing than the opposite. For the end of semester tree, there is no dominant misclassification type.

Table 3. Accuracy and number of rules using all three sources

	Marks + PASTA + Piazza
Mid sem. (week 7)	87.00 (8 rules)
End sem. (week 15)	87.00 (9 rules)

Figure 1 shows the two trees. Although equally accurate, the two DTs are different: they have different rules, using common and different attributes from the three sources. Both use *prac_quiz_mark* from assessment marks and *early_task* from PASTA but the other attributes are different, as shown below.

```
Mid semester (week 7)
prac_quiz_mark <= 81.875
|   prac_quiz_mark <= 54.375: F (51/3)
|   prac_quiz_mark > 54.375
|   |   w7_piazza_active_viewer = no
|   |   |   w7_homework_mark <= 70
|   |   |   |   early_task = no
|   |   |   |   |   task2_mark <= 70: notF (3)
|   |   |   |   |   task2_mark > 70: F (4/1)
|   |   |   |   early_task = yes: notF (2)
|   |   |   w7_homework_mark > 70: F (13/1)
|   |   w7_piazza_active_viewer = yes
|   |   |   task_finish <= 0: F (4)
|   |   |   task_finish > 0: notF (32/6)
prac_quiz_mark > 81.875: notF (114/3)

End of semester (week 15)
prac_quiz_mark <= 81.875
|   assignment_total_submissions <= 15
|   |   prac_quiz_mark <= 45: F (32)
|   |   prac_quiz_mark > 45
|   |   |   early_assignment = no
|   |   |   |   piazza_active_questioner = no
|   |   |   |   |   early_task = no: F (28/7)
|   |   |   |   |   early_task = yes: notF (3)
|   |   |   |   piazza_active_questioner = yes
|   |   |   |   |   assignment_finish <= 0: F (14/4)
|   |   |   |   |   assignment_finish > 0: notF (9/1)
|   |   |   early_assignment = yes: F (8/1)
|   assignment_total_submissions > 15
|   |   prac_quiz_mark <= 50.9375: F (2)
|   |   prac_quiz_mark > 50.9375: notF (13.0)
prac_quiz_mark > 81.875: notF (114.0/3.0)
```

Figure 1. DTs produced using all three data sources

The most important attribute in both cases is *prac_quiz_mark*, which is selected as a root of both trees and classifies correctly a large number of examples (e.g. If *prac_quiz_mark* > 81.875, then notF (114/3) in both DTs). This is expected as the practical quiz tests both theoretical and practical skills, and, similarly to the final exam, is conducted in a supervised environment, within time limits (in this case directly at the computer).

We highlight some interesting rules using attributes from PASTA and Piazza. From the mid-semester tree, the following rule shows the importance of following the discussions on Piazza, in addition to having relatively good marks on the practical quiz and homeworks:

If prac_quiz_mark ∈ (54.375, 81.875] &
w7_piazza_ active_viewer = no &
w7_homework_mark > 70
then F (13/1)

The following rule, also from the mid-semester tree, shows the importance of viewing the posts on Piazza, and also finishing the tasks earlier than on the due day, in addition to having a relatively good mark on the practical quiz:

If prac_quiz_mark ∈ (54.375, 81.875] &
w7_piazza_active_viewer = yes &
task_finish > 0
then notF (32/6)

From the end-of-semester tree, the following rule shows the importance of submitting the assignment and tasks early and asking questions on Piazza:

If prac_quiz_mark ∈ (45, 81.875] &
assignment_ total_submissions <= 15 &
early_assignment = no &
piazza_active_questioner = no & early_task = no
then F (28/7)

The rules in the two DTs generally make sense. The counter-intuitive ones (e.g. the two rules in the mid-semester tree that include *task2_mark* as their last condition and predicting F if *task2_mark* is greater than 70 and vice-versa) cover a very small number of instances (5/223 in this case) and represent coincidences in data rather than patterns.

Finally, both the mid-semester and end-of-semester trees are small (8 and 9 rules respectively), therefore easy to use by teachers.

In summary, the produced rules are compact, useful and actionable. They show the importance of the practical quiz, good practice such as starting and finishing assessments early and regularly reading the posts on the discussion board.

6. IS THE INFORMATION FROM PASTA AND PIAZZA USEFUL FOR PREDICTION?

We investigate if the information from the automatic marking system PASTA and the discussion board Piazza helps to improve the predictive accuracy, in comparison to just using the assessment marks. Table 4 shows the results when using marks only, and marks and PASTA only. The results using all three sources - marks, PASTA and Piazza - are given in Table 3.

Table 4. Accuracy and number of rules using assessment marks alone, and assessment marks and PASTA

	Marks	Marks + PASTA
Mid sem. (week 7)	84.30 (8 rules)	84.70 (13 rules)
End sem. (week 15)	82.96 (9 rules)	83.41 (14 rules)

We can see that using the assessment marks only provides a very good accuracy of 83-84%. The addition of information from the automatic grading system PASTA improves the accuracy by about 1%. Adding the information from the discussion board Piazza (Table 3) further improves the accuracy by about 3%, raising it to 87%. Hence, using information from PASTA and Piazza improves the predictive accuracy, in comparison to just using the assessment marks. However, this improvement is small in this case as the marks alone already provide high accuracy and there is a ceiling effect.

7. CONCLUSIONS

In this paper we investigate whether students at risk of failing can be identified early enough in the semester for timely intervention, using machine learning prediction methods and information from three different sources: automatic marking system, discussion board and assessment marks. We define useful attributes from each data source, to characterise student performance and activity. Using these attributes, we built a decision tree that achieved 87% accuracy in predicting whether students will pass or fail their final exam, from information available in the middle of the semester.

The produced rules are useful and actionable, and indicate the importance of starting and finishing assessments early and reading the posts on the discussion board, in addition to performing well on key assessment components. We show that using information from the automatic marking system and discussion board improves accuracy, compared to only using the assessment marks.

Our results can be used to detect students at risk of failing early in the semester and provide them with simple preventive feedback about remedial actions. Having an early flagging of students at risk also allows teachers of large classes to approach these students and provide more personalised remedial actions. At the beginning of the semester all students can also be made aware of the characteristics of the failing and passing students, to encourage better learning, good practice and improved student engagement.

An important aspect of our work is that we exploited different data sources capturing various facets of student activity during the course. This allowed the DT results to provide some concrete suggestions of remedial actions. The methodology we have followed can be applied to other contexts combining similar types of data sources. We are currently applying it to another very large course.

8. ACKNOWLEDGMENTS

This work was partially supported by the Human Centered Technology cluster at the University of Sydney.

9. REFERENCES

[1] Kotsiantis, S., Patriarcheas, K., and Xenos, M., 2010. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems 23*, 6, 529-535.

[2] Romero, C., Ventura, S., Espejo, P.G., and Hervás, C., 2008. Data Mining Algorithms to Classify Students. In *Proc. Int. Conference on Educational Data Mining (EDM)*, 8-17.

[3] Romero, C., López, M.-I., Luna, J.-M., and Ventura, S., 2013. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education 68*, 458-472.

[4] Agnihotri, L., Ott, A., 2014: Building a student at-risk model: an end-to-end perspective. In *Proc. Int. Conference on Educational Data Mining Conference (EDM)*, 209-212.

[5] Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., and Loumos, V., 2009. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education 53*, 3, 950-965.

[6] Witten, I.H., Frank, E., Hall, M.A., 2011. *Data Mining - Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Intelligent Tutor Recommender System for On-Line Educational Environments

Marian Cristian Mihăescu
University of Craiova
Department of Computer Science
Craiova, Romania
mihaescu@software.ucv.ro

Paul Ştefan Popescu
University of Craiova
Department of Computer Science
Craiova, Romania
sppopescu@gmail.com

Costel Ionaşcu
University of Craiova
Faculty of Economics and Business Administration
Craiova, Romania
icostelm@yahoo.com

ABSTRACT
This paper presents a method of using a classification procedure for retrieval of the most appropriate tutors in on-line educational environments. The main goal is assisting learners to find the most suitable colleagues that can provide them help. The retrieval is based on a user model built from experiences of previous generations of students. Performed activities represent the raw input data(i.e., the experiences), that one obtained from on-line educational environment. The goal of the developed system is to provide a list of colleagues that are willing and able to provide help. The student that is looking for a tutor will be aware of his weakness, his place among his colleagues and get some intuition regarding needed future activities that may improve effectively his knowledge level. The data processing for the retrieval mechanism is based on a classical classification engine that is custom designed for fulfilling the presented goal.

Keywords
Decision Tree, Classification Induction, Recommender System, e-Learning

1. INTRODUCTION
This paper addresses the problem of improving the knowledge level of a student that uses an online educational environment by using algorithms for indexing and retrieval mechanism. The proposed approach contains two main modules: the server side module where the indexing model is created and the client side where visualization and retrieval of a set of learners(i.e. prospective tutors) takes place. This set of colleagues represent the most appropriate options according to several predefined criteria specified by retrieval mechanism.

The first prerequisite for building a reliable recommender system is gathering high quality data in order to properly train the classifier, as main data processing unit within indexing mechanism. The activity related assets (e.g., database, log files, etc.) of the e-Learning environment are queried in order to obtain the training dataset. The assets must provide enough information such that all needed features that define students are computed and stored into the training dataset.

Once the input data for analysis is available the aim is to design a machine learning based recommender system that trains a classifier which acts as a core processing unit for the indexing mechanism. The next steps are choosing: the appropriate algorithm type (e.g., supervised, unsupervised, rule based, etc.), the algorithm itself, the features (e.g., name, meaning, type, values, etc.) and the overall setup necessary for obtaining a solution. In our case, we use supervised learning algorithms (e.g., classifier) and more exactly, decision trees [6]. The algorithm is used to classify new items, which in our educational context are represented by learners. The research issue of this paper regards designing and implementing a tutor recommender system. Addressing of this issue is accomplished by two means: properly designing a custom data analysis pipeline and building a tool that retrieves tutors in the practical context of Tesys [2] e-Learning platform.

For prototyping a general purpose classifier is used, i.e. a decision tree that is implemented in Weka [5] and this implementation is used for experiments. The key issues that are addressed regard properly setting up the general purpose classifier in a context of a practical problem in e-Learning application domain. Main ingredients for concept proof description and tool prototyping are presented in this paper along with detailed description of choices and their expected, observed and validated impact.

Once the recommender system is built, it can be used to obtain tutors for current or new learners by providing input only their computed values for the chosen features. On the client side, the learner is able to log in the application and access the tutor search utility application. The student first sees his class label in the existing model (i.e., his actual class) and then his target class which gathers the best suited tutors for him.

2. RELATED WORKS
The paper folds at boundaries between domains of machine learning and information retrieval as part of EDM and recommender systems. Educational Data Mining is an emerg-

ing discipline, concerned with developing methods for improving the relationship and interaction level between learners and professors. Since 2005 when the workshop referred to as 'Educational Data Mining' AAAI'05-EDM took place in Pittsburg, USA [4], followed by several related work-shops and the establishment of an annual international conference first held in 2008 in Montreal [1] many work has been done in this area [12] [11] .

Building recommender systems for e-Learning gathered many research efforts due to large number of practical usages within this application domain. Since e-Learning is a highly interaction domain, it is very appropriate for using Intelligent Data Analysis techniques for building various types of recommender systems. E-Learning personalization [8] represents one of the most common and general issues in e-Learning. Within this area of research issues like adapting the presentation and navigation [3], smart recommender in e-Learning [14] and various other commercial systems [9] proposes different input data, user modelling strategies or prediction techniques for reaching various business goals. Among the most used data analysis techniques there are content-based or item-based filtering, collaborative filtering, rule-based filtering, etc [10]. The general machine learning strategy of learning and predicting, information retrieval strategy of indexing and retrieval become in recommender systems for e-learning modelling and recommendation. Modelling regards thus users, content(i.e. questions, chapters, etc) and recommendation implies the existence of a implicit or explicit query. Once all these ingredients are put together in a consistent data analysis pipeline the output takes the form of a single recommended set [13].

Regarding involved technologies this paper uses Weka (Waikato Environment for Knowledge Analysis) as a popular suite of machine learning, data mining and information retrieval algorithms written in Java. The implemented algorithms are very flexible and can be used into the analyzing process of different kinds of data(from different domains). From Weka we have used J48 which is the implementation of the C4.5 [7] algorithm in Weka, a data analysis algorithm which generates a decision tree in order to classify data.

3. FRAMEWORK FOR INTELLIGENT TUTOR RETRIEVAL

The recommender module has the task of matching the query of a learner for a tutor against the existing data model. From software perspective the recommender module is a client for model builder module. Its main task is to produce results in such a way they may be intuitively displayed by the thin client application used by learner in his attempt to find a suitable tutor. Therefore the learner will see a tree-like structure due to native shape of the decision trees with actual class of the learner marked in red and with target classes marked in shades of green. In fact the green classes represent set of learners that are suitable for being tutors for learner that is querying the system.

In Fig.1 presents how the tutor recommender mechanism is designed as a data workflow. From interaction point of view students have to query the system that is integrated within Tesys-Web and after performing necessary operations on the server side they obtain the decision tree filled with

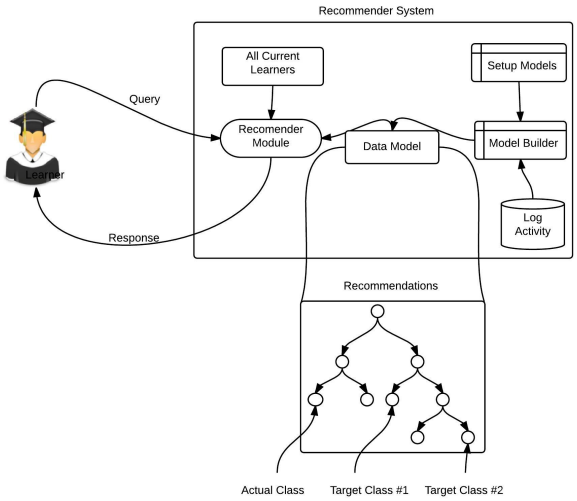


Figure 1: Activity use case diagram

prospective tutors. On the server side we have the business logic of the recommender system. Here the training dataset is built, thereafter the data model and the output as an xml file that can be displayed on the client side.

3.1 Description of Tasks within Recommender System

The main tasks performed within the recommender system regard preliminary offline data model building, setup of the recommender system, indexing currently existing learners into already created model and computing and extracting relevant tutors by applying the already setup recommendation strategy.

3.1.1 Learners Modelling Phase

We apply machine learning (i.e., decision trees) techniques to build learners profiles by using already existing implicit performed activities from usage sessions of learners that used the system in previous years. This data represents the training data and the output is represented by a set of classes (i.e. leaves in the decision tree) such that each class corresponds to a learners profile. Once sessions and corresponding activity data are delimited in such a way that all features describing a learner are processed, we can use the decision tree builder to obtain the baseline data model. Once the data model is created the currently existing learners within the e-Learning platform are placed in their corresponding leaves. At this moment currently existing learners are indexed in the decision tree data structure and are ready to be queried.

3.1.2 Tutors Recommendation Phase

The query of a learner for a tutor is regarded as a parametrized implicit query. The parameters aim tuning the retrieval mechanism such that optimal solutions are returned from solution space. The solution space is regarded as a set of classes(i.e., leaves) each class containing a set of prospective tutors. The set of classes need to fulfill one basic requirement, which is to be labeled with a "better" class label that

the one in which the querying learner resides. A total order set of classes is ergonomically computed with the first item containing learners "close" to the querying learner and the last item containing learners "further" to the querying learner. In this context parameter tuning will manage to decide a certain number of prospective tutors that are picked up from one of the target classes. Intuitively, choosing tutors from a class that is "closer" to the querying learners' class will return tutors with a profile that is better but similar. On the other hand, choosing tutors from a class that is "further" to the querying class will return tutors with the best profiles among all colleagues.

3.2 Description of the Data Analysis Process

The main concept considered in the model is the "Learner", which is also a "Tutor". Once the data model is built from the training data the current set of learners $L=\{L_1,L_2,L_3...L_n\}$ is distributed in corresponding classes according with the key feature values $f_{i,k}$. For current prototype implementation the features are not weighted since the decision tree itself provides a ranking in feature selection.

All classes of learners are considered as resources for which an *"affinity"* function needs to be defined in order to retrieve the most suitable tutors. Defining the *affinity* function needs to take into consideration several criteria such a better overall knowledge weight, specific values in communication related features(i.e., messaging activity, forum activity, etc) and demographic features. Due to its specific topology of the decision tree also ranks the leaves in classes. A normal distribution function is defined such that the lowest ranked class is assigned 0 knowledge weight and highest ranked class is assigned a value of 1 knowledge weight. All in between classes get a knowledge weight ranking between 0 an 1.

Thus the data analysis task is to identify the actual class of the learner who is querying for a tutor and to provide most suitable options from the subsequent classes in obtained ranking of the current learners. With this approach the tutor retrieval becomes a matter of properly specifying querying parameters. The proposed mechanism offers the possibility of obtaining any of the feasible solutions, somewhere between the very next learner which resides in the class with the next knowledge weight value up to the top class learners in ranking. From this perspective several parameters are defined. One manages the proximity of the class from which tutors are retrieved.

4. EXPERIMENTAL RESULTS

The main input of the server application is the activity repository. This raw data taken from the database is converted to an *.arff* file, which is used to train the classifier.

In Fig. 2 are presents the meaning of the attributes from the *.arff* file and Fig.3 presents the obtained decision tree based on the training dataset. Several functionalities were developed in order to be able to load and parse the decision trees. One of them is successor computation. Because of our dataset structure, the decision tree contains ordered leaves. This functionality is successfully used for fulfilling specific user constraints. For example, if the student does not want

userid	The identification number of the student
nrHours	The total number of hours spent on the platform
avgMark	The average mark obtained at a discipline
messagingActivity	This feature represents a discretization of the on-line involvement regarding sent and received messages. A messaging activity greater than 50% of the average number of sent and received messages means YES and smaller means NO.
noOfTests	The total number of tests taken on the platform
avgMessageLength	This feature represents a discretization of the average message length. An average length greater than 160 characters means the student sends LONG messages and smaller means SHORT messages. This value has been chosen with respect to the limitation of the SMS message.
class	The class that the user belongs to, which can be low or high. Low/high values were assigned based on the final result obtained by the student.

Figure 2: Features

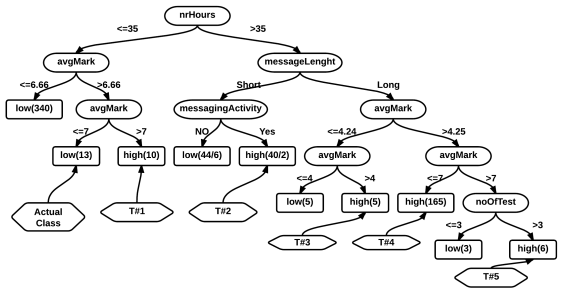


Figure 3: Tree example

to have a step by step progress, he will be able to use this feature to retrieve tutors which are i steps ahead of him.

Here is the pseudocode for the method used to locate the classified student's actual and target class. This recursive method takes two parameters: a node element containing information from the xml and a boolean variable, stating whether the parent has been marked or not. A node is marked when the student meets the requirement stated inside the node (for example if the "messageLenght" is "LONG").

```
function studentSearch(Node parent, boolean isMarked)
{
   SET nodeList to the list of child nodes of "parent"
   FOR each node in nodeList
      IF is element node THEN
         SET atr to the value of the attribute "attribute"
         IF atr is null THEN
            IF isMarked THEN
               add new atrribute to the node
            END IF
            IF "decision" attribute is "high" THEN
               SET target to the current node
            END IF
         ELSE
            SET expresie to ""
            SET belongs to false
            CASE atr IS
               "avgMark":
                  generate the expression to be evaluated
                  SET belongs to the result of the evaluation
               "nrHours":
                  generate the expression to be evaluated
```

```
            SET belongs to the result of the evaluation
            ..............................................
         END CASE
         IF belongs AND isMarked THEN
            add new atrribute to the node
            CALL studentSearch WITH current node, true
         ELSE
            CALL studentSearch WITH current node, false
         END IS
      END IF
   ELSE
      REMOVE node FROM nodeList
   END IF
END FOR
}
```

The thin client automatically computes the class of the student who is searching for a tutor. These values are used to determine the actual class of the student. The actual class of the student is marked with red and the target class is marked with green.

In Fig. 4, the inspection of the green node reveals to the student a list on colleagues that may help him as tutors. The student has a messaging system to his disposal for contacting his recommended tutors in an attempt to find answers from the right persons.

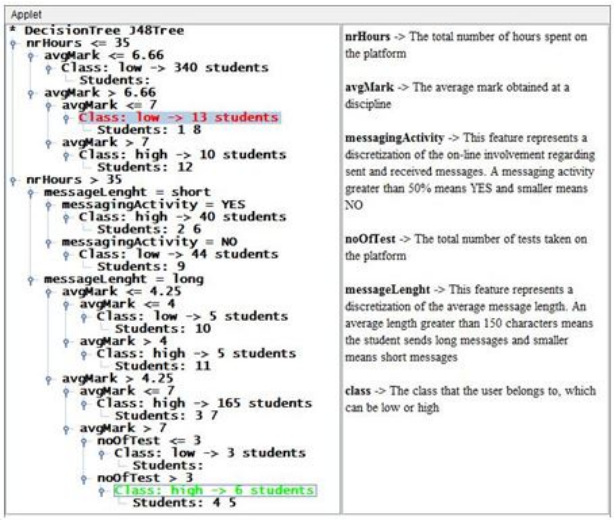


Figure 4: Prototype of the tutor retrieval system

Let us consider student S1, which has already used the e-learning platform and there is enough data logged about him, including the following values for the attributes: nrHours = 30, avgMark = 6.80, messagingActivity = No, noOfTests = 2, avgMessageLength = SHORT. After running the tool, he finds out that his actual leaf/class is the second one when parsing the tree from left to right, so he has to put some effort to catch up with his colleagues. Assuming he wants to spend the necessary time to gradually learn from his colleagues, the system can recommend him tutors belonging to the third leaf of the tree (1st level successors). He will therefore receive the contact details of the students from that leaf. After managing to improve his performances and become himself part of that leaf, he will be able to continue to the next step. If he however wants to try to learn from the best directly, the system will be able to provide him with the contacts of the best tutors available (belonging to the green leaf in Fig. 4).

5. CONCLUSIONS

The paper presents a custom approach for providing assistance to learners in on-line educational environments. The assistance regards the option of finding colleagues that may offer guidance in respect to activities that need to be performed for improving the student's knowledge level.

Each student will benefit by using this tool from the perspective of improving their knowledge. It will be easier for students to communicate with someone their own age, ask questions about the things they don't understand, and get more clarification and feedback.

6. REFERENCES

[1] T. B. J. Baker R.S.J.d.; Barnes. Educational data mining 2008 :1st international conference on educational data mining proceedings. *Educational Data Mining*, 1:20–21, 2008.

[2] M. M. C. Software architectures of applications used for enhancing on-line educational environments. *Manuscript submitted for publication.*

[3] H. Chorfi and M. Jemni. Perso : Towards an adaptative e-learning system. *Journal of Interactive Learning Research*, 15(4):433–447, 2004.

[4] B. J. Proceedings of aaai2005 workshop on educational data mining. 2005.

[5] W. I. F. E. T. L. H. M. H. G. C. S. Jo. Weka: Practical machine learning tools and techniques with java implementations. *University of Waikato, Department of Computer Science*, 1999.

[6] J.R.Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[7] J.R.Quinlan. C4.5: Programs for machine learning. *San Mateo, CA*, 1993.

[8] M. Khribi M. K.; Jemni and O. Nasraoui. Recommender system for predicting student performance. *Educational Technology & Society*, 12(4):30–42, 2009.

[9] B. Mobasher. Data mining for web personalization. *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer-Verlag, Berlin-Heidelberg, 2006.

[10] O. Nasraoui. World wide web personalization. *Encyclopedia of Data Mining and Data Warehousing*, 2005.

[11] C. Romero and S. Ventura. Data mining in e-learning (advances in management information). *WIT Pr Computational Mechanics*, 2006.

[12] B. R. S. and Y. K. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining (JEDM)*, 1:3–17, 2009.

[13] O. N. Z. Z. E. Saka. Web recommender system implementations in multiple flavors: Fast and (care) free for all. *Proceedings of the ACM-SIGIR Open Source Information Retrieval*, 2006.

[14] O. Zaiane. Building a recommender agent for e-learning systems. *Proc. of the 7th International Conference on Computers in Education,*, 3-6:55–59, 2002.

Discovering the Pedagogical Resources that Assist Students in Answering Questions Correctly – A Machine Learning Approach

Giora Alexandron
Massachusetts Institute of Technology
giora@mit.edu

Qian Zhou
Tsinghua University
zhouqian@gmail.com

David Pritchard
Massachusetts Institute of Technology
dpritch@mit.edu

ABSTRACT

This paper describes preliminary results from a study in which we apply machine learning (ML) algorithms to the data from the introductory physics MOOC 8.MReV to discover which of the instructional resources are most beneficial for students. First, we mine the logs to build a dataset representing, for each question, the resources seen prior to each answer to this question; Second, we apply Support Vector Machines (SVMs) to these datasets to identify questions on which the resources were particularly helpful. Then, we use logistic regression to identify these resources and quantify their *assistance value*, defined as the increase in the odds of answering this question correctly after seeing the resource. The assistance value can be used to recommend resources to students that will help them learn more quickly. In addition, knowing the assistance value of the resources can guide efforts to improve these resources. Furthermore the order of presentation of the various topics can be optimized by first presenting those whose resources help on later topics. Thus, the contribution of this work is in two directions. The first is Personalized and Adaptive Learning, and the second is Pedagogical Design.

Keywords

Adaptive Learning, Pedagogical Design Optimization, MOOCs

1. INTRODUCTION

A central question in online courses, as in education in general, is how to design measurably more effective pedagogy. Since online courses, and specifically MOOCs, offer "full course" environments and produce log files that can be analyzed by computational tools, it is only natural that such tools would be used to optimize online pedagogy. While most pedagogic design in online education is based on 'best practices' and subjective opinions, e.g. [4], we concur with Koedinger et al. [3] that optimizing the design of instructional resources is an area in which ML and educational data mining (EDM) techniques can add a lot of value.

We propose a machine-learning, data-driven method that yields various kinds of analytics that can be used by course designers to improve their courses. Specifically, our work concentrates on computing the assistance value of instructional resources. Seaton et. al. [6] showed that the resources used for homework and exam problems differed dramatically, but did not evaluate the *effectiveness* of the selected resources. Our method aims at discovering exactly this – the contribution of particular instructional resources (e.g., page 121 in the e-text) for solving specific questions. From this, various other measures can be derived, such as which resources are generally useful, which questions do not have good supporting resources, etc.

Our longer term vision is that this can be used to augment educational resources with meta-data describing their contribution to various tasks, in line with Mccala's 'Ecological Approach' [5]. This approach suggests using ML and EDM to automatically infer the educational value of on-line resources in order to combine them to achieve educational goals. Inspired by this, Champaign and Cohen [1] presented an algorithm for sequencing educational resources based on their educational value for a specific knowledge unit. Our work suggests means for computing these values, which their algorithm takes as an input.

In the context of personalization, a lot of work has been done in predicting performance and sequencing questions, for example the interesting algorithm of Segal et al. [7]. Our preliminary results show that considering the particular educational resources that students used can also improve the prediction of their performance. This is especially relevant in MOOCs, since the students are free to choose their path through the course, and can attempt a question without going over the pedagogical resources that are important for solving it.

Our approach is based on a two-step method for computing the assistance value of instructional resources. The first step aims at identifying questions that have strong connection to the course resources. The strength of the connection between a question and its resources is operationalized as the difference between the accuracy of a prediction model that considers resources seen prior to attempting the question *and* previous performance, and the accuracy of a model that considers *only* previous performance. On such questions we conduct a second step, aiming at identifying *which* are the contributing resources and quantifying their value. The results have two immediate payoffs. One is optimizing the course design. The other is content recommendation.

The rest of this paper is organized as follows. Section 2 describes our method in detail. Section 3 presents preliminary results obtained from running the method on the Introductory Physics MOOC 8.MRev. Section 4 discusses limitations, and Section 5 presents directions for future work.

2. OUR APPROACH

This section is organized as follows. First, we define the notion of assistance value and what we consider as resources. Second, we give a high-level description of the process for calculating the assistance values. Third, we describe in more details the steps – knowledge representation, data mining, and the ML algorithms.

2.1 Resources and Assistance values

The assistance value Rq is a measure of how much a particular pedagogical resource R (say, a video explaining gravity) contributes to solving question q. It is defined as the increase in the odds that a student seeing R will solve q correctly.

The resources considered in this study are either html pages containing textual explanations, instructional videos, or questions.

2.2 High-Level Description

The process for discovering the assistance values consists of the following steps:

I. Prepare a list of the pedagogical resources from the course structure files.

II. Mine the raw data (students' logs) to create a dataset representing the resources that the students interacted with before attempting the questions.

III. Identify questions in which the resources have a significant contribution to students' success. To achieve this, we compare, for each question, the predictive power of a SVM model that considers the resources seen before attempting this question to a baseline SVM model that considers only the aggregated performance on questions attempted before this question.

IV. For each question identified in step III, discover which resources have the highest assistance value. To achieve this, we use a logistic regression with the resources as independent variables and success/failure as the dependent variable. Then, the exponents of the coefficients are interpreted as the assistance value of each resource.

2.3 Data Mining and Knowledge Representation

As first step, we build, per question, a dataset representing the resources that the student interacted with before each attempt to each question. More specifically, we use a binary feature space, with each feature representing whether a resource was seen or not. Each attempt makes an example, with '1's for the resources seen before answering, and success/failure as the binary tag of this example. Since some of the questions allow multiple attempts, a student might contribute more than one answer to a question.

We note that we chose to start with the simplest representation, and operationalized 'interacting with a resource' as a two-state condition – seen or not. We deliberately decided to use a representation that does not preserve information such as the order in which the resources were seen, the amount of time spent on each resource, and other relevant aspects of the interaction between a student and a resource, as encoding them has an exponential effect on the size of the feature space.

Performance as an additional feature. Student's ability is an important factor when it comes to predicting performance. Thus, we add it as a feature to the model. Student's ability was operationalized as percentage of success on previous attempts.

Preparing the Data. The data mining algorithm, implemented in Python, works as follows*:* For each time-sorted student log file, the algorithm scans the log while maintaining, per student, a list of the resources seen so far and an ability parameter. Each time a resource is accessed, it is added to the list (unless it is already there). Each time a question is attempted, the algorithm adds to the dataset of this question a new vector with the resources seen, the ability parameter, and a tag indicating whether this attempt was successful or not. Then the algorithm updates the ability parameter.

Exploring Various Models. To achieve the best results, we consider various models, which differ on the 'length of their memory', namely, how many resources they keep in the list. For example, a model with memory_length=5 considers only the last 5 resources seen before each attempt. Thus, for each question we actually build several datasets, one per memory_length value. The values that are considered are 1/2/3/5/10/1000. A dataset with memory_length=0 is also prepared, for benchmarking (see next section). This dataset does not 'remember' resources, only student's aggregated performance (student's ability) before attempting the question. We denote the dataset of length *j* for question *q* with *Dqj* (and omit *q* when referring to this dataset for all the questions).

The rationale underlying testing various options is mainly that we assume that some questions might require many resources, while for others, a 'long memory' might include a lot of irrelevant data.

2.4 Using SVM as a Filtering Scheme

To find questions for which the instructional resources used are significant, we train and test (using a standard 10-fold cross-validation) for each question *q* a SVM model on each of the datasets *Dqj*, for *j* = 0/1/2/3/5/10/1000 (we denote the SVM model trained on dataset *j* of question *q* with *Mqj*, and omit *q* in case we refer to this model in general). The baseline described in the previous subsection is *Mq0*. Model accuracy is measured as the average accuracy of the 10-fold cross-validation and denoted *accuracy(M)*. We then compute the relative improvement that each of the models *Mqj*, *j* = 1/2/3/5/10/1000, give over Mq0, and pick the model that gives the highest *relative improvement*, defined as $\frac{accuracy(Mj)-accuracy(M0)}{1-accuracy(M0)}$. We consider questions on which the best model gives more than 10% relative improvement as questions with strong connection to the course resources.

2.5 Using Logistic Regression to Compute Assistance Values

As described above, the role of the Logistic Regression is to identify the resources with highest assistance value for each question. This step is conducted as follows. For each problem found by the SVM to have a strong connection with the resources, we train a logistic regression on the dataset that produces the best SVM model. For example, if for a specific question *q* the most accurate model was *Mqj*, we train a logistic regression on *Dqj* (in case several SVM models give the same performance, we follow Ockham's Razor rule and take the lowest j).

The result is that per question *q*, we have a logistic model that predicts the probability of answering *q* correctly as a function of the resources seen and the ability. As described above, the coefficient attached to each feature quantifies the contribution of this feature to the final outcome, with the *p value* representing the level of confidence.

The coefficient attached to each feature is interpreted as the *assistance value* of the resource that this feature represents, and we consider only those with high confidence (defined as *p value* < 0.05).

We note that an alternative approach was to use one method both for the prediction and for quantifying the value of the resources. This approach was tried with logistic regression and with Decision Trees, which are easily interpretable machine-learning methods. However, the prediction accuracy gained by these methods was relatively low, comparing to the accuracy achieved by SVM, which on the other hand, is a much less interpretable model. Thus, we separate the process into two phases, one aims at prediction and built on SVM, and one aims at quantifying the assistance values and built on logistic regression.

In Section 5 we discuss various ways in which the prediction models and the assistance values can be used for pedagogic design and recommendation.

3. CASE STUDY – INTRODUCTORY PHYSICS MOOC 8.MReV

Context. We applied the above method on the data obtained from the 2014 instance of the introductory physics MOOC 8.MReV given by the third author and his team through the edX platform. The course attracted about 13500 students. Gender distribution was 83% males, 17% females. Education distribution was 37.7% secondary or less, 34.5% College Degree, and 24.9% Advanced Degree. Geographic distribution includes the US (27% of participants), India (18%), UK (3.6%), Brazil (2.8%), and others (total of 152 countries). (All numbers are based on self-reports.) The course lasted for 14 weeks, with content divided between 12 mandatory units and two optional ones. From the course structure file we extracted 1362 pedagogic resources (1020 problems, 273 pages, 69 videos).

Data Mining. We considered 1308 questions for which there were more than 100 student attempts. (For problems that contain several graded sections, we consider each of them as a question. Thus this number is bigger than that in the previous paragraph.) We used the logs of all the students who attempted these questions rather than restricting to those students who exceeded a particular benchmark of participation. As described in Subsection 2.3, for each question we created 7 datasets, each representing a different 'memory length'.

SVMs and choosing the questions. On the next step, we trained a SVM model on each of the datasets, i = 0/1/2/3/5/10/1000 as described in Subsection 2.4, using *R*'s libsvm [2]. This yields seven SVM models for each question, each tagged with its accuracy level. The results show that in overall, models *M1...10* performed better than *M0*, which was always at least good as majority-class prediction. This was evaluated using a paired one-side t-test that tested the hypothesis that the accuracy of *Mi* over all questions is higher than the accuracy of *M0* on all the questions, for i=1,2,3,5,10. For *M1000*, the null hypothesis was not rejected, so we cannot say that in general this model behaved better than performance-based prediction. We believe that the main explanation for this is that considering resources used long before the question at hand was even opened introduces a lot of noise into the data, reducing the weight of the proximate resources. This is exemplified in Figure 1, which shows, for 5 typical questions, the relative improvement that models 'remembering' i=1,2,3,5,10,1000 previous resources give relative to remembering only aggregated performance of each student.

Next step was to choose, per question, the best model. Figure 2 shows, per question, the relative improvement of the best model compared with the accuracy of *M0* on this question. We took relative improvement > 10% as the cut-off for defining questions with strong relation to the pedagogic resources (the line is marked in the figure). In total, of 337 questions were above this threshold.

Logistic Regression. For each of the questions identified by the previous step, we trained a logistic regression on the data that produce the best SVM model, using standard packages in *R*. For example, if for question q the best SVM was *Mi*, we trained a logistic regression for q on *Dqi*. For each question, we sorted the coefficients with p *value* <0.05 in decreasing order. This yields the assistance values. Table 1 shows an example of the two most significant resources found for a question from homework 12, which deals with gravity and orbits. According to the model, the two most significant factors that correlate with answering this question correctly are seeing the html page Angular_Momentum_of_Orbits, which explains content related to this question, and student's performance on previous questions.

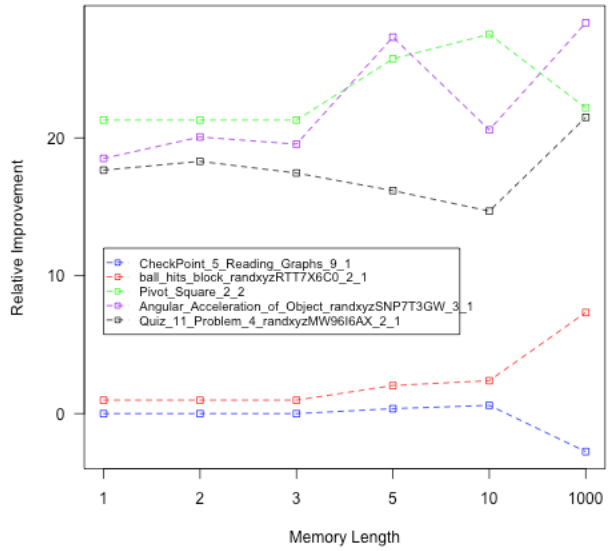


Figure 1. Relative improvement vs different memory length.

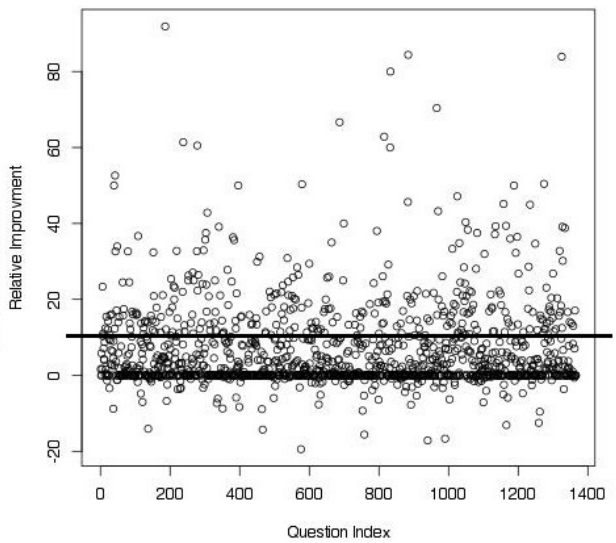


Figure 2. Relative improvement.

Table 1. Example of two most significant predictors

Question: : Homework 12, Gravity and Orbits				
Resource name	**Estimate**	**Std..Error**	**t.value**	**Pr...t..**
Angular_Momentum_of_Orbits	0.73	0.36	2.03	0.042
performance	0.62	0.2	3.05	0.003

Validation. In order to evaluate the meaningfulness of the results, we executed an expert validation protocol aimed at measuring the *precision* of the algorithm. In our case, *precision* is defined as the fraction of retrieved resources that are relevant. We gave to one of the course designers a list of 10 questions, each with 3-5 resources found to have assistance value. The course designer was asked to mark whether each resource is irrelevant/slightly-relevant/highly-relevant to the question. Weights were 0/0.5/1, respectively. In total, the precision on this sample, according to the expert, was

42.5%. We did not measure *recall*, which is the fraction of relevant resources that are retrieved, and is typically used in conjunction with *precision*, since the number of relevant resources for a specific question is unknown, and some of them can be interchangeable. Due to lack of space, we omit a detailed analysis that was done with the expert on the results given for a specific question.

4. LIMITATIONS OF THE MODEL

Our model has limitations in several areas. Due to lack of space we present them very briefly.

Cognitive. Currently the model makes simplistic assumptions on the nature of knowledge acquisition and retention. For example, it does not give any weight to the *time* spent on the resource, the *time since* seeing the resource (knowledge can be forgotten), order of resources is not considered, and it is assumed that the relation between the resources is additive (we used SVM with a linear kernel).

Model. Another limitation is that some of the independent variables in our model are collinear (i.e., A is a resource of B; A and B are resources of C). One effect on Logistic Regression is that the ability to infer the value of specific coefficients is reduced. A possible remedy is discussed in Subsection 5.2.

Data. As typically happens in real world examples, our data is skewed. For example, many of the participants already know the material (i.e., Physics teachers taking the course for professional development), so the resources they see have low effect on their ability. This adds a lot of noise to the data. Also, the ratio of examples-to-features is about 1:1, far from optimal.

5. FUTURE WORK

In this paper we described a method for computing the assistance value of pedagogic resources, presented preliminary results, and discussed limitations. Below we present directions for future work, which include further evaluation of the use of the various applications of this method, and removing limitations.

5.1 Using the Assistance Values

Finding the assistance value of resources will be useful for Pedagogical Design and for constructing Recommender engines.

5.1.1 Pedagogical Design Optimization

The assistance value can be used to address several interesting issues:

What types of resources are most effective: resources that have significant assistance value for a number of questions tell us what learning to emphasize. We can also determine the characteristics of resources that are most helpful - e.g. types (videos vs. e-text) or topics (momentum vs. energy).

Questions that lack good resources: If questions lack resources that help students to solve them this might indicate that the designer should add or improve (or possibly move closer to that question) the resources that ought to help.

Identifying redundant/bad instructional resources: If a particular resource is of little assistance for all questions, it is probably a distraction from good instruction (or covers a topic not assessed by any question).

Location of resources: Good resources that are located far from the question that they support may help students learn foundational skills.

5.1.2 Recommender Systems

In the future assistance values can be used for constructing an online resource-recommendation engine. Before a student attempts a question, the engine could use the logistic model to predict the probability that a student will get it correctly. In case this is low, a list of resources can be provided, recommended based on their assistance value, with simple metadata about each (e.g. whether e-text, a worked example, a video… as well as the median time students spent on it). This would allow the student to select the type of resource they prefer. Furthermore it would enable us to obtain much more data on the effective resources so we could determine which were best for students with different overall abilities and even possibly with different learning preferences.

5.2 Removing Limitations

Logistic regression is used both for its interpretability – to get the assistance values, and for its probabilistic classification – to predict the probability that a student will answer a question correctly. If this probability is low, we can recommend the resource with the highest assistance value that was not seen yet. One direction that we investigate is to separate this between two models – one for interpretability and another for probabilistic classification. This will allow considering other models, such as probabilistic SVMs. We note that for recommendation only, a strong probabilistic classifier is enough, and knowing the assistance values explicitly is not necessary. The process for finding the best recommendation is simple. For each unseen resource *r*, the engine will run the classifier on a vector consisting of the resources seen so far + *r*, and will recommend the resource that leads to the highest probability.

6. ACKNOWLEDGMENTS

This work is supported by a Google Faculty Award and MIT. We would like to thank Christopher Chudzicki, Zhongzhou Chen, Sait Gokalp, and Youn-Jeng Choi for useful comments, and to edX for accessing the data.

7. REFERENCES

[1] J. Champaign and R. Cohen. Ecological content sequencing: from simulated students to an effective user study, 2013.

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines, 2011.

[3] K. R. Koedinger, E. Brunskill, R. S. J. de Baker, E. A. McLaughlin, and J. C. Stamper. New potentials for data-driven intelligent tutoring system development and optimization, 2013.

[4] T. L. Leacock and J. C. Nesbit. A framework for evaluating the quality of multimedia learning resources, 2007.

[5] G. McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners, 2004.

[6] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? 2014.

[7] A. Segal, Z. Katzir, K. Gal, G. Shani, and B. Shapira. Edurank: A collaborative filtering approach to personalization in e-learning, 2014.

Using Topic Segmentation Models for the Automatic Organisation of MOOCs Resources

Ghada Alharbi
Department of Computer Science
Sheffield University
Sheffield, S1 4DP, UK
galharbi1@sheffield.ac.uk

Thomas Hain
Department of Computer Science
Sheffield University
Sheffield, S1 4DP, UK
t.hain@sheffield.ac.uk

ABSTRACT

As online courses such as MOOCs become increasingly popular, there has been a dramatic increase for the demand for methods to facilitate this type of organisation. While resources for new courses are often freely available, they are generally not suitably organised into easily manageable units. In this paper, we investigate how state-of-the-art topic segmentation models can be utilised to automatically transform unstructured text into coherent sections, which are suitable for MOOCs content browsing. The suitability of this method with regards to course organisation is confirmed through experiments with a lecture corpus, configured explicitly according to MOOCs settings. Experimental results demonstrate the reliability and scalability of this approach over various academic disciplines. The findings also show that the topic segmentation model which used discourse cues displayed the best results overall.

1. INTRODUCTION

In recent years, Massive Open Online Courses (MOOCs) have been in the spotlight of the media, education professionals, entrepreneurs and technologically aware members of society. As a result, leading universities have been convinced to run their courses online, by establishing open learning platforms, as seen with MIT Open Course Ware (OCW)[1] and Open Yale Courses (OYC)[2].

The majority of these learning platforms organise their resources in line with a pedagogical model, which will allow easy online browsing and accessing [23]. On the other hand, organising these resources takes a great amount of time and is platform dependent, and a large percentage of these platforms have varying formats and structures of the pedagogical model they are based on [23]. In order to decrease the above efforts, unstructured text can be automatically split into coherent sections, which are thus more suitable for online browsing. As these sections include the content of the learning units, an automatic pedagogical annotation model can be employed to organise these units into introductions, descriptions, explanations, examples and other pedagogically significant notions, as examined by [14]. Even though the use of automatic pedagogical annotation models appears suitable, a number of MOOCs sources are structured in line with both the pedagogical and topical approaches. An example of this would be Figure 1(a), the physics lecture from OYC, which displays both ways of structuring. The first and fifth sections depict the pedagogical elements, while the remainder includes the topic segments. This can also be seen in the economics lecture in Figure 1(b).

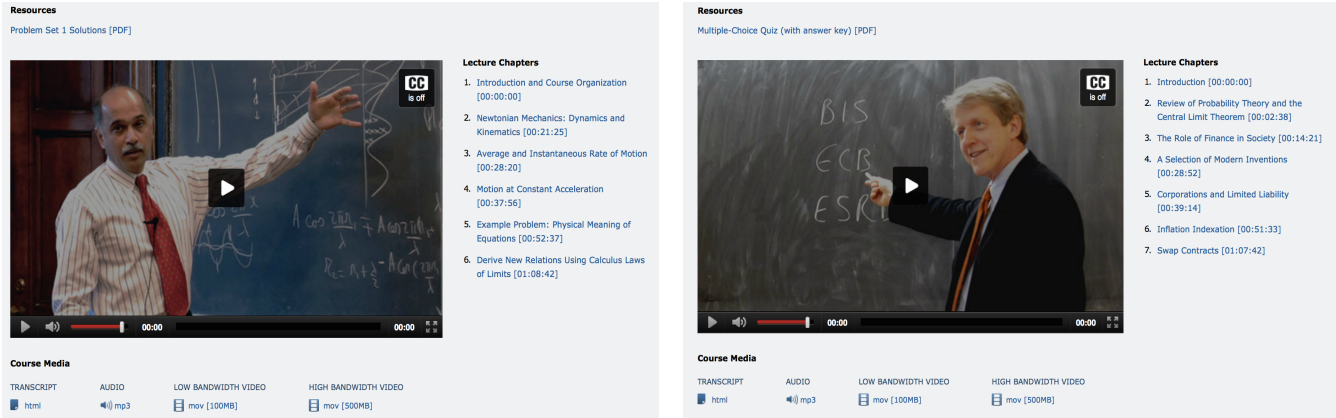
This paper will examine the use of state-of-the-art topic segmentation models to structure lecture resources into cohesive segments, making them suitable for MOOCs content browsing. To evaluate the segmenting applications in the proposed scenario, a test corpus was established using two different disciplines, which were physics and economics, derived from the OYC platform [25, 21]. The topic segmentation models employed in this research include similarity-based models, as seen in [3, 16, 11], language model-based, such as [7, 28] and topic model-based, as seen with [6, 22]. The key strengths of this methodology are its discipline and platform neutrality, which are highlighted in the results of this study. Furthermore, the impact of lexical and discourse cues were examined as features of the segmentation model.

It can be seen from the outcomes that the topic segmentation model which used discourse cues, together with lexical features, showed superior results for the two disciplines. This is due to the fact that discourse cues are often employed to signal the lecturer's aim of the discourse, which means that their learning units are represented more effectively [9]. Despite this, further analysis is required, since the current topic segmentation models hypothesise that discourse cues occur only at the start of an utterance, as seen in [18, 11, 7]. However, other studies have noted that discourse cues can occur at any point in an utterance, and they are a small part of a larger linguistic expression of a writer or speaker [27, 5].

2. BACKGROUND

A number of studies have shown how an automatic pedagogical annotation can be applied to organise lectures resources [14, 24]. However, instead of introducing new aspects such as pedagogical concepts, this paper examined the

[1] http://ocw.mit.edu/index.htm
[2] http://oyc.yale.edu



(a) (b)

Figure 1: Examples of the Interface used for browsing (a) physics and (b) economics lectures in OYC [21, 25].

wider applicability of topic segmentation models for structuring MOOCs content into cohesive units suitable for browsing. In turn, this section concentrates on the work of topic segmentation models, and specifically unsupervised topic segmentation, for either written or spoken language. There has been extensive research on unsupervised segmentation of text, based on lexical cohesion, but certain studies tried to involve other elements, such as discourse or visual cues [7, 8]. This paper will focus mostly on how lexical cohesion is modeled either as similarity-based, language model-based or topic model-based.

TextTiling [12] is considered the first similarity-based model to calculate the cosine similarity between two adjacent blocks of words based purely on word frequency. C99 [3] is based on divisive clustering with a matrix-ranking scheme, while LCSEG modeled lexical chain repetitions of a given lexical term, throughout a fixed-length window of sentences and then chose segmentation points at the local maxima of the cohesion function [11]. MCS [16] optimised normalised minimum-cut criteria, centred on a variation of the cosine similarity between sentences.

An early language model-based algorithm, UI, has been proposed by [28], who tried to find segmentations with compact language models. Furthermore, [7] employs a generative Bayesian model BSEG for topic segmentation. The algorithm computes the maximum likelihood estimates by looking at the entire sequence of sentences, at specific topic boundaries. Also, the model utilises the initial of the potential boundary utterances as discourse cues for the unsupervised model, which is an extension of the work by [11], who automatically identified discourse cues using true labeled boundaries in a supervised fashion.

Latent Dirichlet Allocation (LDA) is a generative model which uses latent structures to model the underlying similarities among observations and it is widely adopted in text analysis to model the shared topics among documents [2]. Topic model-based segmentation was initially interpreted by [26] and built upon by [17]. The most recent LDA based segmenter is TopicTiling [22], which undertakes linear topic segmentation with a pre-trained LDA topic model and estimates the similarity between segments to evaluate text coherence, based on a topic vector representation with cosine similarity. Only the most common topic ID is given to every word in a sentence through Gibbs sampling, in order to maintain efficiency. [6] have shown a hierarchical Bayesian model, which makes use of both Bayesian segmentation and structured topic modelling STM. Superior performance over various models, in both written and spoken texts [6], has been seen with this model. Likewise, the segmentation method of PLDA [20] samples segment boundaries, but also jointly samples a topic model.

The applications of topic segmentation models range from information retrieval to topic tracking [13], summarisation [14] and segmentation of multi-party conversations [11, 20].

3. METHODS

3.1 Data Preparation

Under the Creative-Common license, freely accessible lectures on the OYC website are used as data sources. Expert speakers conducted the lectures, and appear as high quality video and audio data, transcripts, subtitles and lecture segmentation on the course website, as part of MOOCs's initiative. Examples of this segmentation in physics and economics lectures are illustrated in Figure 1. High-level structure distinguishes the lecture as shown in the segmentation. These labelled segments boundaries used as the reference dataset to evaluate the models performance. From these data sources, the two distinct disciplines of physics and economics were selected to establish a new dataset. During the preparation of this study, the total sum of lectures was 47, made up of 24 physics lectures and 23 economics lectures. The average number of annotated segments for the physics lectures was 6, whereas it was 7.1 for the economics lectures. Table 1 shows the new dataset's relevant statistics.

3.2 Segmentation Models

The performance of six competitive models from the literature was compared, with regards to organising MOOCs text content: C99 [4]; UI [28]; LCSEG [11]; MCS [16]; BSEG [7]; STM [6]. All models are explained in Section 2. The publicly available executable given by the authors was employed in all cases, except for LCSEG[3].

[3]This software needs a copyright license from `http://www.cs.columbia.edu/nlp/tools.cgi#LCseg`

	#Lect	#Segments Per Lect	#Total Segments	#Total Words	#Sentences
Physics	24	6	144	260k	18k
Economics	23	7.1	172	212k	15k
Overall	47	6.5	316	472k	33k

Table 1: Lecture Corpus Statistics.

	Physics		Economics	
Text Segmenter	P_k	WD	P_k	WD
C99	0.429	0.433	0.419	0.426
UI	0.426	0.442	0.425	0.435
LCSEG	0.387	0.394	0.356	0.388
MCS	0.439	0.446	0.378	0.383
BSEG	0.364	0.385	0.313	0.334
BSEG+DC	**0.359**	**0.379**	**0.309**	**0.328**
STM	0.372	0.396	0.311	0.330

Table 2: Results of the comparison between segmentation models: WD denotes WindowDiff. Both metrics are penalties, so lower scores indicate better performance.

The paper's specified parameter values [11] were used in the case of LCSEG. MCS needs parameter settings to be tuned on a development set. The corpus of this study does not incorporate development sets, and as a result the tuning was undertaken with the configuration given by the author on the lecture transcript corpus [16]. On the other hand, C99 and UI do not need parameter tuning and can be used without any modification [4, 28]. BSEG also do not need any parameter tuning, but priors are re-estimated, as noted in the paper [7]. The STM model 10 randomly initialised Gibbs chains were used, where every chain ran for $30,000$ iterations, with $25,000$ for burn-in. Following this, 200 samples used the discount parameter $a = 0.2$, and $\lambda_0 = \lambda_1 = 0.1$ and the Dirichlet prior is $\alpha = 0.2$ and $\gamma = 0.01$. In all experiments, the number of segments is assumed to have been given beforehand.

3.3 Evaluation Metrics

All experiments are evaluated with regards to the widely utilised P_k [1] and WindowDiff (WD) [19] metrics. Both metrics run a window throughout a document, and evaluate if the sentences on the edges of the window were suitably segmented with regards to one another. WD is stricter because it needs the number of intervening segments between the two sentences to be exactly the same in both the hypothesised and reference segmentations, whereas P_k only checks if the two sentences are in the same segment. P_k and WD are penalties, so lower values show superior performance. [10] has provided the evaluation source code that was being used.

4. RESULTS AND DISCUSSION

The different performances of the six segmenters using P_k and WD values are shown in table 2. Overall, superior results across the two disciplines were seen in the BSEG model, especially with discourse cues, and the gain and fails of each model across the two disciplines were described. It should be highlighted that these models show better performance using P_k and produce less improvement on the WD metric. This is explained in Section 3.

Notably, the output of the MCS model, which produces segmentation as a graph cut problem, for the physics lectures yields 0.439 P_k, which is worse off compared to more straightforward similarity-based models, such as the C99 and LCSEG. Other models, such as UI, which do not specifically depend on pairwise similarity analysis, have better performance ($P_k = 0.426$) in physics lectures, when compared to MCS. UI calculates a better segmentation performance by estimating alterations to the language model predictions through various partitions, as described in Section 2. On the other hand, economics lectures differed, as MCS had superior performance ($P_k = 0.378$) compared to both C99 and UI, which yielded $P_k = 0.419$ and $P_k = 0.425$ respectively. This is due to the difference in distributional properties of the physics lectures, which were not coherent in their thematic shifts and thus caused a level of distributional differences.

A further note from Table 2 with regards to the LCSEG model was that it had superior performance on P_k metric for both disciplines ($P_k = 0.387$ in physics and $P_k = 0.356$ in economics), compared to all other models used with the exception of BSEG and STM. STM achieved favourable performance, especially in economics lectures, and attained results close to the BSEG model in physics lectures. This can be attributed once again to the lack of coherence in physics lectures, which results in smooth distributional variations. A substantial and consistent increase is seen through the use of BSEG+DC for all lecture subjects. This can be justified from the existence of discourse cues, as depicted in the results of $P_k = 0.359$ in physics and $P_k = 0.309$ in economics. As spoken language is more impulsive and not as planned as written language, the speaker must inform the listener of any alterations to topic content, through the introduction of subtle cues, and references to prior topics during topical transitions [9].

A further analysis study of discourse cues was undertaken, using the labelled topic boundaries. For every word in the lecture corpus, the number of its occurrences near any topic boundary (with a window size of 5 seconds on either side of the target boundary, inclusive) are counted, and set against those further away. The findings were utilised in the undertaking of the χ^2 significance test. The chi-square test allows the calculation of the significance of the near-against distinct-statistics by comparing with the overall statistics, where the null hypothesis is assumed. The word with an χ^2 value in opposition to the hypothesis under 0.01-level confidence (the rejection criterion is $\chi^2 \geq 6.635$) were chosen. Table 4 shows discourse cues sorted by chi-squared value, where bold denotes the common cues of both disciplines. The corpus was manually examined to find these automatically selected discourse cues, and it was discovered that these cues establish linguistic expressions, as in the study by [27] on summarisation task. An example of this is the cue "*topic*", which is part of one expression, such as "*The topic of this lecture is*" or a very different expression, like "*Let's move to another topic*". These expressions can obviously show the function and the purpose of the discourse, and thus show the pedagogical element of this segment. However, current topic

	Near	Distant	Near	Distant
today	14	53	29	92
Other	15254	244776	17838	193820

Table 3: $\chi^2 = 24.73$ in Physics and 35.82 in Economics.

Physics		Economics	
DC	χ^2	DC	χ^2
topic	110.25	**talk**	140.07
last	105.78	about	126.15
okay	51.89	wanted	97.61
now	37.88	**lecture**	67.97
next	32.05	let	66.97
today	24.73	**we**	65.46
alright	22.52	move	59.17
lecure	18.84	**today**	35.83
we	11.40	conclude	29.07
talk	8.68	start	21.58

Table 4: Automatically selected discourse cues (DC), sorted by chi-squared χ^2 value at the level of $p < 0.01$. Boldface indicates that these cues are common across disciplines.

segmentation models do not account for these expressions, possibly because of the fact that these models lack conversational analysis. Additional research is required to examine this aspect, including the induction of these expressions in the segmentation model and the possibility of using an automatic method to identify and extract these expressions, such as in the study by [15] on the extraction of expressions from student essays.

5. CONCLUSION AND FUTURE WORKS

The application of topic segmentation models for the automatic organisation of MOOCs resources has been presented above. The manual analysis of these resources shows that their structure is centred on both pedagogical and topical aspects, and so a new corpus has been established based on this scenario, through two different domains. The study employs the different features of the topic segmentation models in order to compare the results. The outcomes show that the topic segmentation model which utilised linguistic cues (e.g. *today*, *okay*) had the highest results. An important element for future research is the automatic detection and extraction of linguistic expressions, which are used to show various purposes and functions in discourse, in order to be able to involve them in the topic segmentation model. It can be hypothesised that this type of model would have superior performance in the representation of MOOCs learning units.

6. REFERENCES

[1] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Mach. Learn.*, 34(1-3):177–210, Feb. 1999.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[3] F. Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of NAACL'00*, pages 26–33, 2000.

[4] F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. In *In Proceedings of EMNLP'01*, pages 109–117, 2001.

[5] R. Correia, N. Mamede, J. Baptista, and M. Eskenazi. Using the crowd to annotate metadiscursive acts. *In Proceedings 10th Joint ISO-ACL SIGSEM*, page 102, 2014.

[6] L. Du, W. L. Buntine, and M. Johnson. Topic segmentation with a structured topic model. In *HLT-NAACL*, pages 190–200, 2013.

[7] J. Eisenstein and R. Barzilay. Bayesian unsupervised topic segmentation. *Proceedings of EMNLP'08*, page 334, 2008.

[8] J. Eisenstein, R. Barzilay, and R. Davis. Gestural cohesion for topic segmentation. In *Proceedings of ACL-08: HLT*, pages 852–860, 2008.

[9] J. Flowerdew and L. Miller. The teaching of academic listening comprehension and the question of authenticity, 1997.

[10] C. Fournier. Evaluating Text Segmentation using Boundary Edit Distance. In *Proceedings of ACL'13*, 2013.

[11] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. Discourse segmentation of multi-party conversation. In *Proceedings of ACL'03*, pages 562–569, 2003.

[12] M. A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, Mar. 1997.

[13] X. Huang, F. Peng, D. Schuurmans, N. Cercone, and S. E. Robertson. Applying machine learning to text segmentation for information retrieval. *Information Retrieval*, 6(3-4):333–362, 2003.

[14] N. Kokhlikyan, A. Waibel, Y. Zhang, and J. Y. Zhang. Measuring the structural importance through rhetorical structure index. In *HLT-NAACL*, 2013.

[15] N. Madnani, M. Heilman, J. Tetreault, and M. Chodorow. Identifying high-level organizational elements in argumentative discourse. In *Proceedings of NAACL'12: HLT*, pages 20–28, 2012.

[16] I. Malioutov and R. Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings ACL '06*, pages 25–32, 2006.

[17] H. Misra, F. Yvon, J. M. Jose, and O. Cappe. Text segmentation via topic modeling: An analytical study. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1553–1556, 2009.

[18] R. J. Passonneau and D. J. Litman. Discourse segmentation by human and automated means. *Comput. Linguist.*, 23(1):103–139, Mar. 1997.

[19] L. Pevzner and M. A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36, Mar. 2002.

[20] M. Purver, T. L. Griffiths, K. P. Körding, and J. B. Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of COLING-ACL'06*, pages 17–24, 2006.

[21] S. Ramamurti. Fundamental of physics 1 (yale university: Open yale courses). `http://oyc.yale.edu/physics/phys-200#overview`, 2006. (Accessed December 20, 2014), License: Creative Commons BY-NC-SA.

[22] M. Riedl and C. Biemann. Topictiling: A text segmentation algorithm based on lda. In *Proceedings of ACL'12 Student Research Workshop*, pages 37–42, 2012.

[23] O. Rodriguez. The concept of openness behind c and x-moocs (massive open online courses). *Open Praxis*, 5(1):67–73, 2013.

[24] K. Sathiyamurthy and T. V. Geetha. Automatic organization and generation of presentation slides for e-learning. *Int. J. Distance Educ. Technol.*, 10(3):35–52, July 2012.

[25] R. J. Shiller. Financial markets (yale university: Open yale courses). `http://oyc.yale.edu/economics/econ-252-11`, 2011. (Accessed December 22, 2014), License: Creative Commons BY-NC-SA.

[26] Q. Sun, R. Li, D. Luo, and X. Wu. Text segmentation with lda-based fisher kernel. In *Proceedings of ACL'08: Short Papers*, pages 269–272, 2008.

[27] S. Teufel and M. Moens. Summarizing scientific articles - experiments with relevance and rhetorical status. *Computational Linguistics*, 28:2002, 2002.

[28] M. Utiyama and H. Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of ACL'01*, pages 499–506, 2001.

How High School, College, and Online Students Differentially Engage with an Interactive Digital Textbook

Jeremy Warner[1], John Doorenbos[2], Bradley N. Miller[2], Philip J. Guo[1]

[1] University of Rochester, Rochester, New York, USA
[2] Luther College, Decorah, Iowa, USA

ABSTRACT

Digital textbooks have been growing popular as a lower-cost and more interactive alternative to paper books. Despite the recent rise in adoption, little is known about how people use these resources. Prior studies have investigated student perceptions of digital textbooks in the classroom via interviews and surveys but have not quantified actual usage patterns. We present, to our knowledge, the first large-scale quantitative study of digital textbook usage. We mined 6.8 million log events from over 43,000 people interacting with *How To Think Like a Computer Scientist*, one of the most widely-used Web-based textbooks for learning computer programming. We compared engagement patterns among three populations: high school students, college students, and online website viewers. We discovered that people made extensive use of interactive components such as executing code and answering multiple-choice questions, engaged for longer when taking high school or college courses, and frequently viewed textbook sections out of order.

Keywords

Digital textbooks; student engagement; server log data mining

Categories and Subject Descriptors

H.5.1. [**Information Interfaces and Presentation (e.g. HCI)**]: Multimedia Information Systems

1. INTRODUCTION

Digital textbooks have grown popular in the past decade as more students gain access to laptop computers, tablet devices, and broadband Internet. Some of their claimed benefits over paper textbooks include lower cost, lighter physical weight, full-text search, electronic note-taking, and better accessibility for sight-impaired students via text-to-speech [4]. As the costs of paper textbooks continue to rise, university professors are adopting digital alternatives to save money for their students [13]. Governments are pushing for widespread adoption of digital textbooks at the K-12 level as well. For instance, in his 2011 State of the Union address, U.S. President Barack Obama challenged all K-12 schools to adopt digital textbooks by 2016, and the FCC Chairman and Secretary of Education followed up with a plan to implement this vision [12]. The publishing industry has responded to recent events by converting many of their paper textbooks into digital formats. By some estimates, digital textbook sales will be a $1.5 billion business and account for over 25% of all new textbook sales by 2016 [11]. In parallel, universities [1], non-profits, and independent volunteers [8] are developing freely-available digital textbooks.

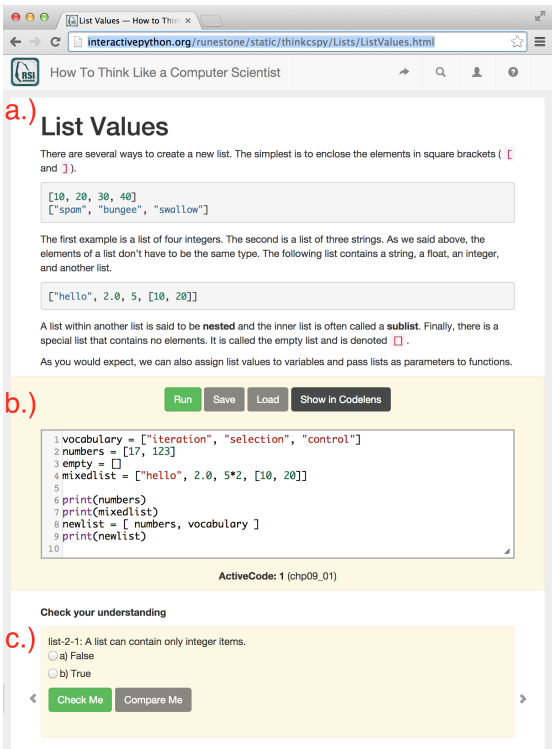


Figure 1: *How To Think Like a Computer Scientist* [8] is a Web-based interactive digital textbook for learning computer programming. A user can: a.) read text, b.) edit and run Python code to see outputs, and c.) answer multiple-choice questions.

Aside from classroom use, online digital textbooks are a form of educational technology similar to MOOCs. Anyone with a computer and Internet connection can learn topics ranging from computer programming [8, 10] to math using digital textbooks. In recent years, many researchers have studied how students use MOOCs [3, 6], but to our knowledge, there has never been an analogous large-scale study of digital textbook usage. Given the growing prominence of digital textbooks, it is important to understand how stu-

dents use them in a variety of educational settings, and how that could inform the design of the next generation of digital textbooks.

This paper contributes, to our knowledge, the first large-scale study of how students use an interactive digital textbook. We studied *How To Think Like a Computer Scientist* [8], a Web-based digital textbook for learning computer programming (Figure 1). We analyzed two years of server logs containing 6.8 million events from 43,416 students. This data is far larger, more diverse, more precise, and finer-grained than prior digital textbook studies that relied on questionnaires sent on university campuses [2, 9, 13].

Specifically, we quantified how students navigated through the textbook and engaged with interactive components such as live code and multiple-choice questions. We segmented students into three populations: those taking a high school course, a college course, and those visiting the public textbook website. These comprise the three main populations of textbook readers. We investigated three sets of research questions: 1.) How much does each population engage with interactive components of the textbook? 2.) When do people in each population access the textbook, and for how long do they persist before quitting? 3.) How do readers navigate non-linearly and skip around when accessing textbook contents?

The first generation of digital textbooks were simply paper books converted into electronic formats such as PDF. The current generation features interactive topic-specific widgets (Figure 1) but does not take advantage of the scale afforded by tens of thousands of online readers. This study is one step toward providing data to inform the design of the next generation of digital textbooks, which can leverage such data to assist students, instructors, and book authors.

2. RELATED WORK

Researchers have studied student attitudes toward digital textbooks in the classroom, with mixed findings. Questionnaire studies of 446 students in the University of Cape Town in South Africa [13] and of 5,000 business school students across 127 U.K. universities [9] found high self-reported enthusiasm for adopting digital textbooks. In contrast, a survey of 662 students across five California State University campuses found that only 1/3 were satisfied with digital textbooks and only 1/2 felt they were easy to use [2]. Prior studies were all done on non-interactive digital textbooks, comparing them to nearly-identical paper counterparts. And they all relied on questionnaires and exam results but did not analyze log data on actual textbook usage. To our knowledge, we are the first to study an interactive digital textbook in-the-wild in a large-scale online setting. Our sample contains 43,416 students from around the world, which is one to two orders of magnitude more students than prior studies.

3. METHODOLOGY

We studied usage patterns of *How To Think Like a Computer Scientist* [8], a widely-used Web-based digital textbook for learning introductory computer programming. This textbook is viewable online for free at `http://interactivepython.org/`. Figure 1 shows how it intersperses textual content, snippets of editable and runnable Python code, and multiple-choice questions. This digital textbook shares similarities with computer programming MOOCs. Both feature multiple-choice questions and runnable Python code as interactive components. However, unlike a MOOC, the main pedagogical modality here is text rather than video. Also, registration is not mandatory. Readers can register with a free account to save their code and track personal analytics, but this is an open resource that anyone can access on the Web. Finally, there is no notion of a fixed course schedule with, say, weekly releases of new materials like there is in some MOOCs. All textbook materials are always present, which supports self-paced learning.

We mined the server logs from June 2012 to 2014, fetching 6,834,244 events from 43,416 students. Each event has the following fields:

- **Timestamp** – server time in the U.S. Central Time Zone
- **Student type** – High School, College, Open (public website)
- **Student ID** – either a registered username or an IP address
- **Event type** – Page load, Run code, Code error, Viz interaction (Python code visualization), or Multiple-choice attempt
- **Textbook location** – the chapter and sub-chapter to which this event belongs (e.g., chapter 5, sub-chapter 3).

Event types: The *Event type* field has one of the following values:

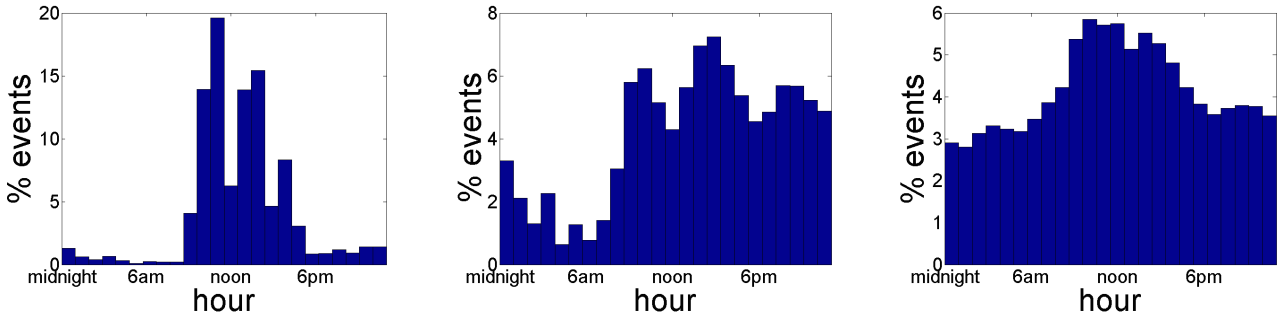
- **Page load** – Load a webpage, which displays the content for a specific sub-chapter of the textbook
- **Run code** – Press the "Run" button to run a piece of Python code, and the code executes successfully (Figure 1b)
- **Code error** – Press the "Run" button to run a piece of Python code, but the code has a syntax or runtime error
- **Viz interaction** – Interact with a Python code visualization widget by taking one step forward or backward in the embedded visual single-step debugger tool [5]
- **Multiple-choice attempt** – Attempt to answer a multiple-choice question within a webpage (Figure 1c)

Non-Linear Navigation: We define a ***backjump*** as any consecutive pair of events for one student where the first occurred in chapter n and the second in chapter m, where $n > m$. A ***sub-backjump*** is either a regular backjump, or a pair of events in the same chapter that went from sub-chapter n to sub-chapter m, where $n > m$. We define skip and sub-skip similarly. A ***skip*** is any consecutive pair of events where a student jumped from chapter n to chapter m, where $m > n+1$. Note that we use $n+1$ because simply going to the next chapter is ordinary sequential navigation, not a skip. A ***sub-skip*** is either a regular skip, or a pair of events in the same chapter that went from sub-chapter n to sub-chapter m, where $m > n+1$. The intuition behind these metrics is that if a student navigated through the textbook in a perfectly sequential fashion, starting with chapter 1, sub-chapter 1, and ending with the final sub-chapter of chapter 15, then they would have zero backjumps or skips. Thus, backjumps and skips indicate non-linear navigation.

4. FINDINGS

4.1 Engagement with Interactive Components

Most students actively engaged with the interactive components rather than just passively reading. Figure 2 shows that page loads accounted for only around 10% of total events. If students had simply been using this textbook as a static reference, then *all* events would have been page loads. By far the most common event type was attempting to run Python code. *Run code* and *Code error* events comprise around three quarters of total events. Recall that pieces of Python code are embedded throughout the textbook (Figure 1b.). Some are complete working examples that can be run verbatim without triggering errors, while others are incomplete snippets that students must complete as an exercise. For all three populations, attempting multiple-choice problems and interacting with code visualizations were about as common as page loads, which again indicates that students did not just passively read the book.



(a) High School (b) College (c) Open

Figure 3: Distributions of events throughout the day, recorded as server time in the U.S. Central Time Zone.

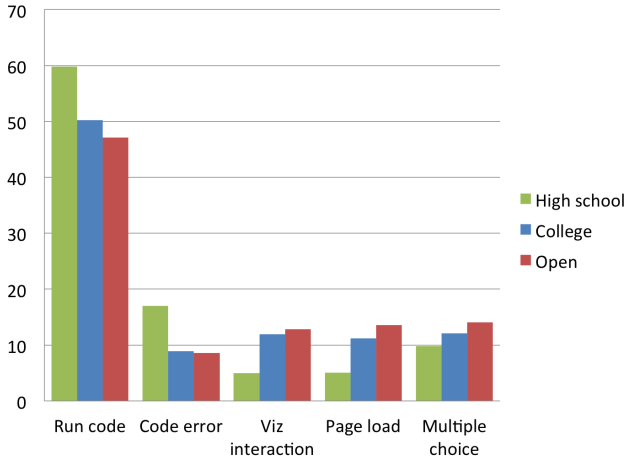


Figure 2: Percentages of total events by type.

4.2 Writing, Running, and Debugging Code

Figure 2 shows that high school students ran the most code, with ∼10% more *Run code* events and twice as many *Code error* events than college and open. Also, for high school students, 22% of total code run attempts resulted in an error, versus only 15% for college and open. High school students made, on average, 112 errors per student, versus 35 errors per student for college and 12 for open.

One interpretation is that high school students made more errors because they were less experienced at coding, but we do not have the data to support this claim. Since this is an introductory textbook, presumably the college and open students also did not have much prior coding experience. A more likely interpretation is that the high school students used this textbook in a more structured and instructor-guided manner than college and open. We have anecdotal evidence from high school teachers who sent emails to the textbook creators requesting technical support that many intended to use this strictly within their classrooms. A typical use case is a teacher directing students to spend the class period reading through a sub-chapter and attempting to do all of the code-related exercises. The teacher would then walk around the classroom and help students debug their faulty code. Thus, high school students ran more code and persisted in debugging, fixing their errors, and re-running possibly because an instructor was present in the classroom.

In contrast, college and open students are usually less supervised. College instructors typically assign readings from a textbook but do not monitor students as closely as high school teachers do. Since running code and attempting multiple-choice problems are ungraded formative exercises, students can work on them at their leisure. Open students might be self-directed learners with little to no supervision. Thus, they make fewer code errors (12 per student) not necessarily because they are better at coding, but simply because they might give up after seeing an error and not persist in fixing it.

4.3 Activity Levels by Time of Day

Visualizing activity levels by time of day confirms that high school students mostly use this textbook in class during school hours, while college and open students use it throughout the day. Figure 3 shows the distribution of event times. The majority of high school activity occurs between school hours of 9am to 4pm, with a sharp dip at noontime for lunch. This pattern indicates in-class usage, supervised by a teacher. In contrast, college activity occurs evenly throughout most waking hours from 8am to midnight.

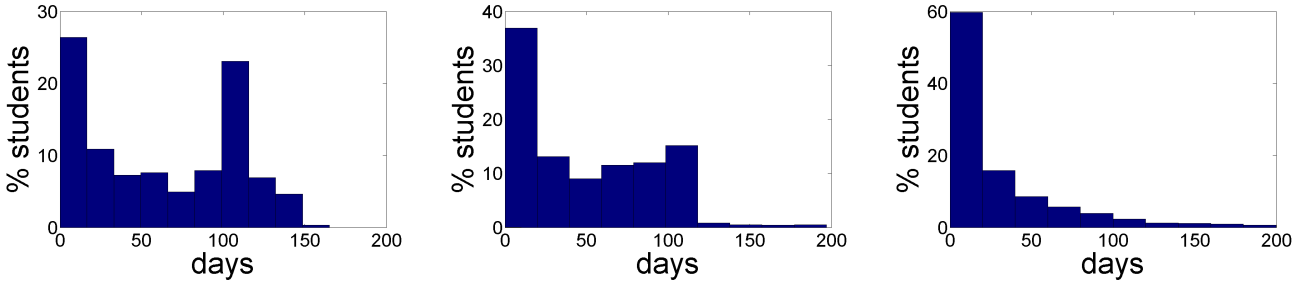
Note that the event timestamp is the server's time (U.S. Central Time Zone), so it does not take the student's local time zone into account. However, by geolocating IP addresses of high school and college students, we found that the majority with a geolocatable IP were from the U.S. and Canada (89% of high school and 94% of college students), so the true time for those students lies within a few hours of the U.S. Central Time Zone.

Whereas high school and college students came mostly from the U.S. and Canada, the open student population was much more international. Only 57% of open students were from the U.S. or Canada, and many came from countries such as Australia, New Zealand, the U.K., and India. Unsurprisingly, those are all English-speaking countries, since this textbook is in English. The presence of many international students explains the relatively even levels of activity throughout the day and night in Figure 3c, although there is still a spike during mid-day in the U.S. and Canada.

4.4 Engagement Duration

For how long does each student engage with the textbook before quitting? We quantified engagement duration by calculating the difference between the first and last event times for each student. Figure 4 plots the distributions for all three student types. High school and college students engaged for up to a semester (∼ 105 days) because they used the textbook as part of a course. The high school spike at around 105 days is much more pronounced than the college one, which could be a result of greater teacher supervision.

In contrast, the open population engagement drops off sharply in a long-tail-like distribution, which mirrors the high initial dropout



(a) High School (mean=62, median=62) (b) College (mean=48, median=39) (c) Open (mean=28, median=13)

Figure 4: Distributions of how many days each student was actively engaging with the textbook, split by student type.

Student Type	Backjumps		Sub-Backjumps	
	mean	median	mean	median
High School	41.0	2	58.4	11
College	13.2	2	21.8	4
Open	3.8	0	6.1	1
	Skips		**Sub-Skips**	
	mean	median	mean	median
High School	38.5	4	67.3	13
College	13.1	1	27.4	7
Open	4.3	1	9.2	3

Table 1: Non-linear navigation statistics for all student types.

rates in MOOCs [3, 7]. Half of the open students used the textbook for less than two weeks. However, unlike many MOOCs, which incrementally release new course materials on a weekly basis, all of the material in this textbook is always available. Thus, it is possible for self-directed learners in the open population to engage for a week or two, learn what they want, and then leave. Thus, semester-long engagement is simply an artifact of formal course schedules.

4.5 Non-Linear Navigation

How frequently did students jump backward to earlier textbook locations or skip forward to latter ones out of sequence? Table 1 summarizes the levels of backjump and skip activity by student type. For all four measures we defined (backjump, sub-backjump, skip, sub-skip), high school students exhibited the most non-linear navigation, followed by college, then open. Even controlling for differing levels of activity per student, high school students perform twice the number of backjumps and skips as college and open students. For instance, 6.2% of all high school events involved backjumps, versus only 3.4% of college and 2.7% of open events.

Non-linear navigation indicates engagement, since it takes more active effort to jump around rather than following the default sequential ordering of the textbook by simply clicking the “Next page" link at the bottom of each page. One explanation for the high numbers of backjumps and skips for high school students is that they are using the textbook in the classroom, so their teacher can proactively direct them to other parts of the textbook as they are trying to solve coding problems. Without other people present in-person to guide or direct one’s learning, it is easier to default back to the more passive style of reading through the textbook in a linear way.

Another interpretation is that high school and college students navigate non-linearly to review materials when studying for exams. A related study of non-linear navigation in MOOCs showed that students often backjumped from exam pages back to earlier lecture pages [6]. In contrast, open students might be self-studying without taking a graded course, so they do not need to review as much.

5. REFERENCES

[1] Open SUNY Textbooks – `http://opensuny.org/`.

[2] Baek, E.-O., and Monaghan, J. Journey to textbook affordability: An investigation of students’ use of eTextbooks at multiple campuses. *The International Review of Research in Open and Distance Learning 14*, 3 (2013).

[3] Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., and Seaton, D. T. Studying learning in the worldwide classroom: Research into edX’s first MOOC. *Research and Practice in Assessment 8* (Summer 2013).

[4] Courduff, J. Digital Textbooks and Students with Special Needs – `http://goo.gl/JGTaA9`, Accessed: Oct, 2014.

[5] Guo, P. J. Online Python Tutor: Embeddable Web-based Program Visualization for CS Education. SIGCSE ’13, ACM (2013), 579–584.

[6] Guo, P. J., and Reinecke, K. Demographic differences in how students navigate through MOOCs. L@S ’14, ACM (New York, NY, USA, 2014), 21–30.

[7] Kizilcec, R. F., Piech, C., and Schneider, E. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. LAK ’13 (2013), 170–179.

[8] Miller, B. N., and Ranum, D. L. Beyond PDF and ePub: Toward an interactive textbook. In *Proceedings of ITiCSE* (2012), 150–155.

[9] Nicholas, D., Rowlands, I., and Jamali, H. R. E-textbook use, information seeking behaviour and its impact: Case study business and management. *Journal of Information Science 36*, 2 (Apr. 2010), 263–280.

[10] Pritchard, D., and Vasiga, T. CS Circles: An In-browser Python Course for Beginners. SIGCSE ’13, ACM (New York, NY, USA, 2013), 591–596.

[11] Reynolds, R. Trends influencing the growth of digital textbooks in us higher education. *Publishing Research Quarterly 27*, 2 (2011), 178–187.

[12] Usdan, J., and Gottheimer, J. FCC Chairman: Digital Textbooks to All Students in Five Years. `http://goo.gl/VJ9NA0`. Accessed: Oct, 2014.

[13] van Heerden, M., Ophoff, J., and Van Belle, J.-P. Are university students ready to dump their textbooks? A survey on student attitudes towards e-readers and tablet computers. *Int’l Jour. Cyber Ethics in Education 2*, 3 (2012), 15–44.

Modeling Exercise Relationships in E-Learning: A Unified Approach

Haw-Shiuan Chang, Hwai-Jung Hsu, Kuan-Ta Chen
Institute of Information Science, Academia Sinica, Taipei, Taiwan
{ken77921, hjhsu, swc}@iis.sinica.edu.tw

ABSTRACT

In an e-learning system, relationships between a large amount of exercises are complex and multi-dimensional; measuring the relationships and arranging curriculums accordingly used to be time consuming and costly tasks which require either enormous log collection or large-scale human annotations. Moreover, accurately quantifying the relationships is difficult because there are too many factors which affect our measurement based on the data, such as the ability of exercise takers and the subject bias of annotators. To overcome these challenges, we propose a unified model that extracts information from both human annotations and usage log using regression analysis. The proposed model is applied to quantify the *similarity*, *difficulty*, and *prerequisite* relationships between every two exercises in a curriculum. As a case study, we collaborate with Junyi Academy, a popular e-learning platform similar to Khan Academy, and infer the pairwise relationships of 370 exercises in its mathematics curriculum. We show that the model can predict exercise relationships as well as an expert does with human annotations of a few sample exercise pairs (2% in our experiments). We expect the introduction of the proposed unified model can improve the relationships among exercises and learning pathways of students in other e-learning platforms.

Keywords

Exercise relationships, Prerequisite, Curriculum, Human annotations, Regression Analysis, Khan Academy

1. INTRODUCTION

Estimating relationships between items has a wide range of applications in educational data mining (EDM). For example, curriculum arrangement [2, 5] and adaptive testing [6, 9] are often based on the estimations of difficulty and prerequisite relationships between courses, knowledge components, or exercises. Furthermore, estimating the similarity and prerequisite relationships between exercises can improve the quality of knowledge components [12, 13] and student modeling [3, 1, 4]. In this paper, we focus on studying the relationships of exercises (i.e., complete question units), which can facilitate personalized education in the future.

Meanwhile, in large and dynamic e-learning websites, manually organizing the growing number of exercises becomes more and more difficult. For instance, Junyi Academy[1], an e-learning platform in Taiwan similar to Khan Academy[2]. Junyi Academy provides over 300 interactive exercises for its mathematics curriculum, which is visualized by the knowledge tree as shown in Figure 1. We can see that there have

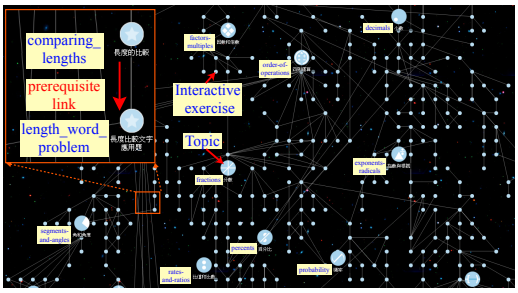


Figure 1: Part of the knowledge map on Junyi Academy. To visualize the prerequisite structure, the knowledge tree is laid out in a 2D plane called knowledge map.

been many complex prerequisite links in the knowledge tree, so it is very time consuming to manually validate how appropriate the prerequisite links are and whether there are better ways to arrange the links of the exercises. Moreover, the instructors need to consider hundreds of exercise candidates when determining the prerequisites for a new exercise.

Based on exercise taking log, researchers discover the relationships through item response theory (IRT) [10], inferring Bayesian model of students [3, 12, 1, 4], factor analysis [8], association rule learning [5], assuming a known Q-matrix [13], or assuming students would perform better after they have taken prerequisite or similar exercises [12, 11, 15], etc. Most of the aforementioned data-driven methods develop a specific learning algorithm for estimating a specific relationship between exercises. The learning algorithms usually require a large amount of log data so as to simultaneously infer all latent factors affecting our observation in data, such as relationships of exercises and capability of every student over time. However, data in some e-learning platforms might not be sufficient to accurately profile various behaviors of every student. As a result, the estimation of relationships between exercises might be misleading in a new system with only a small amount of usage log [16, 10].

On the other hand, the collected data are often noisy [1] and have different statistical characteristics in different systems, which might violate the assumptions made by a data-driven model. For example, many e-learning websites, such as Khan Academy and Junyi Academy, allow learners to browse any exercise without actually answering them. In fact, around 70% of the first answers are correct for the first problem of each mathematical exercise on Junyi Academy, which shows that learners tend to skip exercises they cannot answer. The freedom of selecting exercises would degrade the performances of purely data-driven approaches on more difficult exercises with less responses [16], and also cause challenges to identify the difficulty and prerequisite

[1] Junyi Academy (`http://www.junyiacademy.org/`) is established in 2012 on the basis of the open-source code released by Khan Academy.
[2] `https://www.khanacademy.org/`

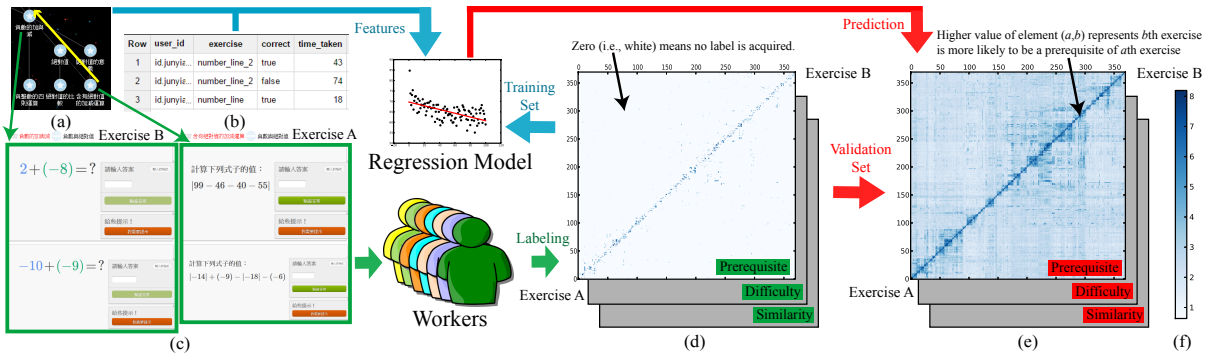


Figure 2: The proposed work flow. (a) A screen shot of the local knowledge map, (b) examples of usage log, (c) a example of exercise pairs, (d) the sparse similarity matrices labeled by workers, (e) the dense similarity matrices predicted by a regression model, and (f) the color code of (d) and (e).

relationships between exercises (See details in Sec. 2.3).

To solve the challenges, we advocate a hybrid method which integrates the power of crowdsourcing and machine learning as [14] did for finding prerequisite relationships among documents. As illustrated in Figure 2, we first quantify the *similarity*, *difficulty*, and *prerequisite* relationships of mathematical exercise pairs using crowd wisdom. Then, we characterize each exercise pair by various types of features extracted from the user practice log and website contents. Given labels and features, a regression model can be trained to predict relationships of every exercise pair. Finally, collected labels can be used to quantitatively evaluate both the prediction of machines and humans. Our experiments show that predictions generated by the proposed models are closer to the crowd consensus (i.e., average opinions of workers) than most of individuals' ratings.

2. RELATIONSHIP DISCOVERY

2.1 Label Collection

As previously discussed, the exercise relationships are hard to define objectively from usage log. Recently, Wauters et al. [16] pointed out that as more annotators judge difficulty of each exercise, their average score converges to a more steady value, which is highly correlated with the difficulty inferred by IRT model. Therefore, if we collect more subjective labels with high quality, their average responses are more representative (i.e., more likely to be agreed by most learners and instructors) and less sensitive to subject bias.

To collect high-quality labels from wide range of people, we divide the task of comparing exercise relationships into several questionnaires and apply several quality control methods. The method includes mathematical ability qualification, malicious workers detection by checking the elapsed time and the variances of their responses in each questionnaire, and outlier filtering using crowd consensus as [7] did.

At each section of questionnaires, we consecutively compare an exercise A with 1–7 other exercises which might be related to A. Note that potentially related exercises are paired according to student modeling and knowledge tree in Figure 1, and the order of comparisons is randomly determined. An example of comparison could be seen in Figure 2(c).

Any target relationship of exercise pairs could be quantified by a specific question. In this work, we ask the workers to choose the 1–9 score for the following questions, which query about *similarity*, *difficulty*, and *prerequisite* relationships of each exercise pair (A and B), respectively.

- How similar is the knowledge required for answering these two exercises?
- How much more difficult is exercise B compared to exercise A, where a higher score means B is more difficult than A and a score of 5 indicates that they have the same difficulty?
- After students learned to correctly answer exercise B, how appropriate is utilizing exercise A to deepen the students' knowledge on the topic step by step?

2.2 Feature Extraction

To automatically predict the relationships, we extract the usage log from Oct. 2012 to July 2014 on Junyi Academy, which contains over 10 million answering records from over 100 thousand users. When describing relationships between exercise A and exercise B, we extract the potentially helpful features from usage log and cluster them into 6 categories:

(i) ***Student Modeling (4 features)*** is extracted based on the practice history of each student. To be more specific, the student is modeled by applying random forest regressor to predict his/her accuracy on every exercise which has not been done by the student. Then, we compute original and normalized feature importance of log data in B for predicting students' accuracy in answering A, and the corresponding importance of A for the prediction of B.

(ii) ***Answering Time Duration (6 features)*** includes the difference between the average answering time duration of A and that of B (i.e., $(time\ for\ A)-(time\ for\ B)$), the logarithm difference of their average answering time duration (i.e., $\log(time\ for\ A)-\log(time\ for\ B)$), the difference and the logarithm difference of their answering time duration on the average of users' correct answers, and on the average of users' first correct answers of the exercises.

(iii) ***#Problems Taken in Exercises (4 features)*** (# means *the number of*) includes the difference and the logarithm difference between #total problems taken in A and B, the difference and the logarithm difference of #problems which are answered correctly in A and B.

(iv) ***Answering Accuracy (6 features)*** includes the difference and the logarithm difference between accuracy of A and that of B on the average of users' first, last, and all answers in the exercises, where the accuracy is defined by $\frac{\#correct\ answers}{\#total\ answers}$. Note that we only count the first answer of each learner in the same problem.

(v) ***#User Taking Exercises (3 features)*** includes the difference and the logarithm difference between #users taking A and that of B, and the Euclidean distance between #users vectors of A and that of B. The ith element in the #users vector of A records the #users who have done exercise i correctly before A.

(vi) ***User Answering Orders (6 features)*** include #users who practice A before B (denoted as $\#U[A \rightarrow B]$), #users who do B before A ($\#U[B \rightarrow A]$), $\frac{\#U[A\rightarrow B]}{\#U[A\rightarrow B]+\#U[B\rightarrow A]}$, #correct answers for A before answering B ($\#C[A \rightarrow B]$), the corresponding #answers for B before A ($\#C[B \rightarrow A]$), and $\frac{\#C[A\rightarrow B]}{\#C[A\rightarrow B]+\#C[B\rightarrow A]}$.

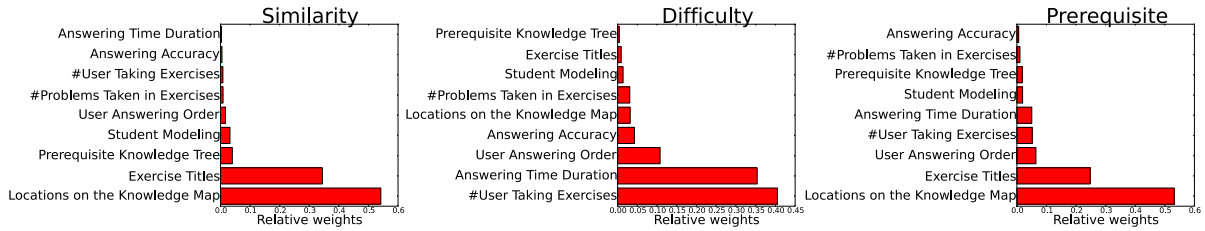


Figure 3: The feature importance for predicting relationships on Junyi Academy. The red bar of each category means the summation of all the feature importance in the category, and symbol # represents *the number of*.

As pointed out in [6, 5, 10, 8, 13], different types of tags on exercises or courses labeled by experts are useful information for determining their relationships. Therefore, we additionally extract exercise-related information from website contents on Junyi Academy, which can be grouped into following 3 categories:
(i) ***Prerequisite Knowledge Tree (5 features)*** includes whether B is a parent of A in the knowledge tree (i.e., the directed acyclic graph), whether B is a sibling of A, distance between A and B in the directed acyclic graph, and the corresponding distances after reversing and removing the direction of every edge in the graph.
(ii) ***Locations on the Knowledge Map (3 features)*** include Euclidean distance between A and B on the knowledge map, and coordinate difference between A and B on x-axis and y-axis in the knowledge map (e.g., the length and the coordinate vector of the yellow arrow in Figure 2(a)).
(iii) ***Exercise Titles (3 features)*** include edit distances of Chinese and English titles between A and B, and summation of the minimal edit distances among English words in their titles.

2.3 Relationship Prediction

Given the features and relationship labels, we formulate the relationship prediction task as a regression analysis. In Sec. 3, we use the collected labels to experiment on the effects of using different regression algorithms. To know the effectiveness of our 40 dimension features, we show the importance of feature categories which are determined by random forest regressor in Figure 3.

Compared with *Answering Accuracy*, *#User Taking Exercises* is a much better type of features for predicting the difficulty difference of exercises, because learners tend to skip exercises they cannot answer as we mentioned in Sec. 1. For the similarity and prerequisite relationships, the *Locations on the Knowledge Map* are the strongest type of features for the tasks, while the *Prerequisite Knowledge Tree* surprisingly has relatively low feature importance. An explanation for the observation is that instructors usually maintain similar exercises in close distance on the knowledge map, which are often good prerequisite candidates for each other. However, when they manually assign the prerequisite links in the knowledge tree, the graph needs to be kept sparse to ensure the clarity and simplicity of its layout.

Figure 3 also illustrates that the information contained in the *Exercise Titles* is much more correlated with the prerequisite relationships on Junyi Academy than features based on *Student Modeling* and *Answering Accuracy*, of which the analysis is extensively studied by many previous works such as [3, 12, 1]. Therefore, it would be interesting to investigate whether the observation is still valid in other platforms which probably have different rules of naming titles or of recommending exercises to learners.

3. EXPERIMENTS

Our proposed method is evaluated in the exercise system of Junyi Academy. To prevent scarce usage log skewing the statistical distribution of our features, we exclude the exercises which are answered by less than 100 users. The remaining 370 exercises of interest are randomly divided into two sets: the training set containing 240 exercises, and the testing set with 130 exercises. On average, each exercise of interest in training set is paired with 4.7 other exercises where around 10% of exercises are randomly selected, and each one in testing set is paired with 6.3 other exercises where the percentage of randomly selected exercises reaches around 30% to verify our generalization capability.

To evaluate how good humans and machines perform, one of metrics we adopt is relative squared error (RSE), which is defined as $\frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$, where $\hat{y}_i$ and y_i are our prediction and the ground truth for a relationship of exercise pair i, respectively, and $\bar{y}$ is the mean of y_i over all i. In addition, we transform every score of exercise relationships into its rank, and compare the similarity between the ranks from the predicted scores and the ranks from the ground truth scores. Then, we evaluate the predicted rank by Spearman's ρ and Kendall τ rank correlation coefficients.

3.1 Performance of Workers

After excluding malicious and unqualified workers, we hire 3 teachers, 8 online workers, and 43 people to work in the lab. All workers in the lab are at least graduated from senior high school, and most of them have a college degree. Each exercise pair in the training set are labeled 6.6 times on average by total 51 normal workers, and teachers are asked to score all the exercise pairs in the testing set. For the interest of the consistency between judgements from crowd consensus (i.e., the average scores from all workers) and that from experts, we also ask 2 among 3 teachers to label every pair in the training set. The total costs of collecting above labels are around 1,000 USD.

Manually quantifying the relationships between mathematical exercises is a demanding cognitive task, which requires a certain level of skills in abstract reasoning. Using the average of ratings from all workers (including teachers) as our ground truth, we first evaluate the performances of recruited workers and whether teachers (i.e., experts) perform better in the tasks. The results in the training set are presented in Table 1. Note that smaller RSE and larger rank coefficients indicate better performances. From Table 1, it is clear that the performance of workers (including experts) measured by RSE is significantly lower than the ones measured by rank coefficients compared with the performances of machines. The results illustrate that workers' annotations often contain systematic subject bias (i.e., workers tend to rate every query higher or lower than most of other people), so averaging scores rated by multiple workers is an effective way to improve the labeling quality for the task.

Table 1: Performance comparisons of different methods in the training set of Junyi Academy using cross validation, and best performances among regressors are highlighted in bold font.

Methods			Similarity			Difficulty			Prerequisite		
			RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ
Humans	An Normal Worker	Range	0.193–1.124	0.188–0.854	0.208–0.750	0.492–3.235	0.063–0.820	0.050–0.747	0.316–2.381	0.000–0.813	-0.007–0.725
		Mean	0.574	0.598	0.524	1.096	0.516	0.439	0.986	0.458	0.387
	A Teacher	Range	0.493–0.543	0.648–0.718	0.560–0.625	0.619–0.741	0.625–0.634	0.539–0.540	0.858–1.054	0.571–0.684	0.504–0.594
		Mean	0.518	0.683	0.593	0.680	0.630	0.539	0.956	0.638	0.549
Regressors	Linear Regression		0.370	0.658	0.567	0.470	0.593	0.504	0.424	**0.624**	**0.541**
	nu-SVR		0.349	**0.683**	**0.594**	0.483	**0.611**	**0.526**	0.402	0.611	0.520
	Random Forest Regression		0.320	0.662	0.575	0.493	0.576	0.493	0.376	0.608	0.516
	GBR		**0.288**	0.680	0.590	**0.453**	0.610	0.521	**0.346**	0.600	0.514
Features (GBR)	w/o KT and KM		0.311	0.681	0.589	0.474	0.626	0.532	0.378	0.607	0.515
	w/o KT, KM, and ET		0.433	0.607	0.521	0.472	0.642	0.546	0.472	0.567	0.478
	w/ SM, AA, UN, and PT		0.548	0.516	0.438	0.502	0.610	0.516	0.595	0.463	0.377
	w/ SM and AA		0.598	0.524	0.446	0.632	0.486	0.400	0.666	0.417	0.346
	w/ KT		0.674	0.463	0.418	0.869	0.448	0.382	0.717	0.360	0.318

Table 2: Performance comparisons of different methods in the testing set of Junyi Academy. Note that the meaning of all abbreviations is the same as Table 1.

Methods			Similarity			Difficulty			Prerequisite		
			RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ	RSE	Spearman's ρ	Kendall's τ
Humans	A Teacher	Range	0.200–0.300	0.764–0.848	0.656–0.757	0.398–0.474	0.732–0.791	0.629–0.696	0.322–0.467	0.696–0.764	0.583–0.665
		Mean	0.235	0.814	0.719	0.427	0.762	0.661	0.406	0.721	0.617
Regressors	GBR		0.269	0.786	0.678	0.553	0.580	0.476	0.311	0.771	0.660

3.2 Prediction Accuracy

For the training set, we evaluate our prediction by 5-fold cross validation, and Table 1 compares the resulting outputs generated by different regression models and different subsets of features. The table summarizes the results of five regression algorithms including linear regression, nu support vector regression (nu-SVR), random forest regression, and gradient boosting regression (GBR). Compared with teachers' ratings in the training set, our approach can generate competitive performances measured by rank coefficients while having lower RSE, especially for more complex regressors such as the random forest or gradient boosting algorithms. This means that after being trained by collected labels, machines could predict exercise relationships closer to crowd consensus than most of the individuals. Note that to make the comparison fair, we round all of the scores predicted by machines into integers between 1–9.

In Table 1, we also provide control experiments on different types of features using gradient boosting regression. There might not be the knowledge tree (KT) and the knowledge map (KM) in other interactive learning environments, so we first present the performance without related categories of features. The results show that removing KT and KM can still produces competitive performances, but the performance would decrease by a margin if we further remove more features such as *Exercise Titles* (ET), *User Answering Orders*, *Answering Time Duration*, *User Numbers* (UN), and *#Problems Taken in Exercises* (PT), *Student Modeling* (SM), and *Answering Accuracy* (AA).

In order to verify our generalization ability across different types of annotators, we train the regression models on the training set (mostly labeled by normal workers) and evaluate their performance on testing set (labeled by teachers). As shown in Table 2, the performances of regression models are still very promising. Note that the exercise pairs in the testing set are only rated by 3 teachers whose labels have larger impact on ground truth, so the real performances of experts might be worse than this estimation.

4. CONCLUSIONS

The relationships of exercises are important for curriculum arrangement of e-learning platforms. In this work, we demonstrate that the relationships can be quantified by subjective labeling and predicted by regression models. The experiments on Junyi Academy show that the quality of predicted relationships are competitive against teachers' labels.

References

[1] E. Brunskill. Estimating prerequisite structure from noisy data. In *EDM*, 2011.

[2] E. Brunskill and S. J. Russell. RAPID: A reachable anytime planner for imprecisely-sensed domains. In *UAI*, 2010.

[3] C. Carmona, E. MillÃąn, J.-L. P. de-la Cruz, M. Trella, and R. Conejo. Introducing prerequisite relations in a multi-layered bayesian student model. In *User Modeling*, 2005.

[4] M. C. Desmarais. Performance comparison of item-to-item skills models with the IRT single latent trait model. In *UMAP*, 2011.

[5] T.-C. Hsieh and T.-I. Wang. A mining-based approach on discovering courses pattern for constructing suitable learning path. *Expert Syst. Appl.*, 2010.

[6] C. Koutsojannis, G. Beligiannis, I. Hatzilygeroudis, and C. Papavlasopoulos. Using a hybrid AI approach for exercise difficulty level adaptation. *IJCEELL*, 2007.

[7] B. Lakshminarayanan and Y. W. Teh. Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. *CoRR*, 2013.

[8] A. S. Lan, C. Studer, A. E. Waters, and R. G. Baraniuk. Tag-aware ordinal sparse factor analysis for learning and content analytics. In *EDM*, 2013.

[9] D. Lynch and C. P. Howlin. Real world usage of an adaptive testing algorithm to uncover latent knowledge. In *ICERI*, 2014.

[10] M. L. Nguyen, S. C. Hui, and A. C. M. Fong. Content-based collaborative filtering for question difficulty calibration. In *PRICAI*, 2012.

[11] Z. A. Pardos and N. T. Heffernan. Determining the significance of item order in randomized problem sets. In *EDM*, 2009.

[12] P. I. Pavlik, H. Cen, L. Wu, and K. R. Koedinger. Using item-type performance covariance to improve the skill model of an existing tutor. In *EDM*, 2008.

[13] R. Scheines, E. Silver, and I. Goldin. Discovering prerequisite relationships among knowledge components. In *EDM*, 2014.

[14] P. P. Talukdar and W. W. Cohen. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 2012.

[15] A. Vuong, T. Nixon, and B. Towle. A method for finding prerequisites within a curriculum. In *EDM*, 2011.

[16] K. Wauters, P. Desmet, and W. van den Noortgate. Acquiring item difficulty estimates: a collaborative effort of data and judgment. In *EDM*, 2011.

Using knowledge components for collaborative filtering in adaptive tutoring systems

Peter Halkier Nicolajsen
University of Copenhagen
srb816@alumni.ku.dk

Barbara Plank
University of Copenhagen
bplank@cst.dk

ABSTRACT

In adaptive tutoring systems, accurately assessing the ability of a student is central to prescribing the tasks that best facilitate learning. For the 2010 KDD Cup challenge a data set of logs from the Cognitive Tutor system was made available, and contestants were asked to predict the correctness of a student's attempt to answer questions. A successful approach included a collaborative filtering system which predicted student performance on the basis of the performance of similar students. In this paper, we present an extension of this approach. Rather than finding similar students on the basis of their performance on specific questions, we based our similarity measure on the performance on questions that require the same "knowledge components" (or skills). This approach increases the amount of users with whom it is possible to compare performance, which in turn increases the likelihood of finding similar students. The experiments using our question type-based distance measure yield promising results.

Keywords

Adaptive tutoring systems, collaborative filtering, distance measure

1. INTRODUCTION

Education is getting more expensive, a reason for this is that the spread of technology-based improvements in productivity have been very limited compared to other industries. If technological advances allow the same amount of labor to be more productive, that production will become less expensive. Education is a sector where the amount of *output* (i.e., students taught) per hour of teacher labor has been relatively constant. This means that relative to sectors with more technology-based productivity gains–most sectors–education becomes more expensive. In economics this is referred to as *Baumol's cost disease* [3].

Assessment is an element of teaching that is amongst the most labor intensive and thus calls most for technological advancement. In order to give appropriate feedback, it is necessary for a teacher to have an accurate, and up to date picture of the ability of the students. Assessing students requires attention, which naturally limits the number of students that can be effectively supervised. If assessments could be made more efficient, more of the teacher's time could be spent giving appropriate feedback. An educational technology that is based on this idea is the adaptive tutoring system (ATS). An ATS is a platform that delivers educational materials (e.g. lectures, problems etc.) while assessing the student and—as the student uses the system—adapting the material to best suit each student. One instance of an adaptive tutoring system is Carnegie Learning's Cognitive Tutor. This system is based on the ACT-R model of cognition [1, 2]. Logs of interactions with this system for Algebra courses were made available in the 2010 KDD Cup [7], where the task was to predict student performance based on logs of previous interactions. If we use performance as a proxy for ability, a more accurate performance prediction corresponds to a better ability assessment.

In this paper, we propose to extend the work of Töscher & Jahrer [9] (referred to as TJ). Part of their solution was a k-nearest neighbor system that predicted scores based on a weighted average of the 41 most similar students. Here, we propose using a different distance measure, by looking at the students with highly correlated performance scores on similar problems, rather than on identical problems.

2. ADAPTIVE TUTORING SYSTEMS

The Cognitive Tutor is an adaptive tutoring system that provides practice for different subjects. The system assigns specific problems for the user to rehearse on. The student's performance on these problems then allows the system to suggest the appropriate level of additional problems. Figure 1 shows an example screenshot from the system.

The Cognitive Tutor has been developed on the basis of the ACT-R model of cognition [1, 2]. There are two elements of ACT-R that are particularly relevant to learning. The first element is the idea that all complex knowledge is the combination of smaller, discrete, pieces of knowledge, so-called *knowledge components* (KCs). The second element is that a student improves a KC by rehearsing it often and in different contexts. When using the Cognitive Tutor, a student will acquire some complex knowledge by incrementally rehearsing each of the required KCs. Any subject for which

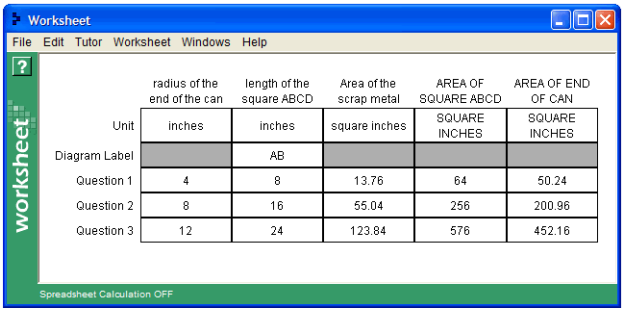


Figure 1: Screen shot from Cognitive Tutor. Each field is one step, while each column consists of three steps that share one knowledge component.

a Cognitive Tutor is implemented, must first be subject to a decomposition analysis, where subject experts identify all of the required KCs, and arrange them in a hierarchy based on the order in which they must be learned. The idea is that, once the subject has been mapped, the student will then be assigned problems that rehearse KCs appropriate to the current level of the student. To assess the student's level, the system keeps track of whether or not the student is able to consistently and correctly, solve problems associated with each KC, in different contexts. For example, being consistently able to solve $4 + 3$ is not the same thing as being consistently able to solve single digit addition. In the course of a session, the system will thus assign several different problems, that require the same KCs.

3. COLLABORATIVE FILTERING FOR ATS

Next, we outline the collaborative filtering approach that was part of [9] (ranking 3rd in the competition) and our proposed extension.

3.1 Töscher and Jahrer

Töscher and Jahrer [9] adopted a collaborative filtering solution, used in the field of recommender systems (e.g., Netflix challenge), and adapted it to the KDD cup challenge. Conceptually the challenges have similarities. The task in the Netflix Prize competition was to recommend movies based on ratings that different users would give different movies, based on the other movies they had rated. The KDD Cup task also required assigning values to different items (steps) for different users based on their previous data. Given these similarities, they proposed a user-based collaborative filtering approach based on the k-nearest neighbor algorithm with correlation shrinkage, described next.

The k-nearest neighbor algorithm found the 41 most similar students for each student, based on how correlated their results were on the basis of the steps they had in common. The prediction is then made on the basis of this group of similar students by using a weighted average (see details below). The stronger a neighbor correlated with the student, the more weight was given to their contribution to the prediction. If the correlation with the whole group was not very strong, the prediction would be corrected toward the student's own mean score. Despite creating the groups based only on correlations in the correct first attempt rate for the different steps, this classifier reached good performance.

The distance measure TJ used was Pearson correlation. Because there was a lot of variation in how many steps each pair of students had in common, the correlation value was transformed to reflect the support for each correlation, giving higher value to correlations based on more common steps. For the sake of consistency, we will use the same terminology as TJ in the algorithm description. They use the terms *students* and *items* to describe the main elements of the model. The items here are the step names. The students s are in the set $\mathbb{S}$, while the steps i are in the set $\mathbb{I}$. The variable to be predicted is whether a student s answered correctly on the first attempt at a step i, is called c_{is}, while the predicted value for this is $\hat{c}_{is}$.

To find the most similar students, the Pearson correlations are calculated between all pairs of students for the steps that both students s_1 and s_2 have answered. The set of steps for s_1 is $\mathbb{I}s_1$, so the set of common steps is $\mathbb{I}s_1 s_2 = \mathbb{I}s_1 \cap \mathbb{I}s_2$. Then, the Pearson correlation ρ between s_1 and s_2 is given by:

$$\rho_{s_1 s_2} = \frac{\frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i \in \mathbb{I}_{s_1 s_2}} (c_{s_1 i} - \mu_{s_1})(c_{s_2 i} - \mu_{s_2})}{\sqrt{\frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i \in \mathbb{I}_{s_1 s_2}} (c_{s_1 i} - \mu_{s_1})^2}\sqrt{\frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i \in \mathbb{I}_{s_1 s_2}} (c_{s_2 i} - \mu_{s_2})^2}}$$

where
$\mu_{s_1} = \frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i \in \mathbb{I}_{s_1 s_2}} c_{s_1 i}$ and $\mu_{s_2} = \frac{1}{|\mathbb{I}_{s_1 s_2}|}\sum_{i \in \mathbb{I}_{s_1 s_2}} c_{s_2 i}$.
To account for the large variability in the number of common steps, they perform a shrinkage transformation that adjusts the correlation by scaling it to the number of common steps $|\mathbb{I}_{s1s2}|$, this transformation of the correlations is calculated as:

$$\bar{\rho} = \frac{|\mathbb{I}_{s_1 s_2}| \cdot \rho_{s_1 s_2}}{|\mathbb{I}_{s_1 s_2}| + \alpha}$$

They set the meta parameter α to a value of 12.9. In the KDD Cup paper [9] they do not describe how they obtain α, but in the Netflix Prize competition paper [8]–where they use an identical shrinkage transformation–they explain that they used a random search method in which they iterate through parameter values selected from a normal distribution, until they find the value that minimizes error. This method is also used to find the other meta-parameters K (set to 41), β (set to 1.5), δ (set to 6.2) and γ (set to -1.9). We here use the same parameters throughout the paper, and leave parameter optimization for future work.
Finally, another transformation is performed on the correlations, in order to minimize the error. The transformation uses the sigmoid function[1]: $\sigma(x) = \frac{1}{1+e^{(-x)}}$ The sigmoid function is then applied to the correlations according to:

$$\tilde{\rho}_{s_1 s_2} = \sigma(\delta \cdot \bar{\rho}_{s_1 s_2} + \gamma)$$

To calculate a predicted score, the scores of the 41 most similar students (k=41) are averaged for the relevant step. Each student's average for the step is then weighted by how strong the correlation is.

$$\tilde{c}_{is} = \frac{\sum_{\tilde{s} \in \mathbb{S}_i(s;K)} \tilde{\rho}_{s\tilde{s}} c_{i\tilde{s}}}{\sum \tilde{s} \in \mathbb{S}_i(s;K) \mid \tilde{\rho}_{s\tilde{s}} \mid}$$

where $\mathbb{S}_i(s;K)$ is the set of nearest neighbor.
The last element of the algorithm is a final correction of the

[1]Note that the original paper [9] contains a typo, describing the sigmoid function as $\sigma(x) = \frac{1}{1-e^{(-x)}}$

prediction towards the mean score μ_s of student s. This is also necessary in case there is not enough support among the neighbors to make a prediction. The β term ensures that the summed correlation to the neighbors is strong enough that the prediction can be based on it, if the correlation is 0, the prediction will simply be the average score for the student.

3.2 Extension of the approach

The system described in the following is a replication and extension of the k-nearest neighbor model described above. In contrast to the TJ model, we here propose to find similarities based on knowledge components rather than just steps. This idea can be seen as abstracting from concrete question instances to basic concepts of knowledge.

The distance measure used was the correlation between students on correct answer rates for steps sharing the same knowledge component, rather than the same step name. The fact that KCs each represent several step names, means that on average, each pair of students will have more steps on which to be compared. Referring to Figure 1, this would correspond to comparing performance on steps in the same column, rather than on identical steps. In the internal training set the average number of common steps between any pair of students is 40.66, while the average number of common KCs is 52.20. Using this distance measure can be advantageous both by expanding the number of other students with which it is possible to test correlation, and by providing a broader base of problems from which to predict a score.

The procedure for finding the neighbors is the same as above, only with the compared items being different. They are now KC names rather than step names. So the knowledge components KC are in the set $\mathbb{KC}$. The predicted value for the to be predicted CFA (cf. section 4.1) then becomes: $\hat{c}_{KCs}$. The Pearson correlations are again calculated between all pairs of students, this time for the steps that have KCs that both students s_1 and s_2 encounter. The set of KCs for s_1 is $\mathbb{KC}s_1$, so the set of common steps is $\mathbb{KC}s_1s_2 = \mathbb{KC}s_1 \cap \mathbb{KC}s_2$ The Pearson correlation ρ between s_1 and s_2 is given by:

$$\rho_{s_1 s_2} = \frac{\frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{i \in \mathbb{KC}_{s_1 s_2}} (c_{s_1 KC} - \mu_{s_1})(c_{s_2 KC} - \mu_{s_2})}{\sqrt{\frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{KC \in \mathbb{KC}_{s_1 s_2}} (c_{s_1 KC} - \mu_{s_1})^2}\sqrt{\frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{i \in \mathbb{KC}_{s_1 s_2}} (c_{s_2 KC} - \mu_{s_2})^2}}$$

where
$\mu_{s_1} = \frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{KC \in \mathbb{KC}_{s_1 s_2}} c_{s_1 KC}$ and
$\mu_{s_2} = \frac{1}{|\mathbb{KC}_{s_1 s_2}|}\sum_{KC \in \mathbb{KC}_{s_1 s_2}} c_{s_2 KC}$. The shrinkage transformation is also changed to reflect the number of steps with common KCs:

$$\bar{\rho} = \frac{\mid \mathbb{KC}_{s_1 s_2} \mid \cdot \rho_{s_1 s_2}}{\mid \mathbb{KC}_{s_1 s_2} \mid + \alpha}$$

The correlations again undergo the same sigmoid transformation as in the case of the stepwise algorithm:

$$\tilde{\rho}_{s_1 s_2} = \sigma(\delta \cdot \bar{\rho}_{s_1 s_2} + \gamma)$$

The calculation of the predicted score is altered to use the all of the steps of the most similar students that share a KC with the to be predicted score, again weighted by each neighbor $\tilde{s}$'s correlation to s:

$$\tilde{c}_{KCs} = \frac{\sum_{\tilde{s} \in \mathbb{S}_{KC}(s;K)} \tilde{\rho}_{s\tilde{s}} c_{KC\tilde{s}}}{\sum \tilde{s} \in \mathbb{S}_{KC}(s;K) \mid \tilde{\rho}_{s\tilde{s}} \mid}$$

where $\mathbb{S}_{KC}(s;K)$ is the set of nearest neighbors.

4. EXPERIMENTS

4.1 KDD Cup 2010

In 2010 a large amount of log files from the Cognitive Tutor system for algebra was made available for the KDD Cup competition held in conjunction with a data mining conference. These logs contained data on interactions for more than 3,000 students over the course of a school year. Every entry was an interaction of a student with the system. For each student there was an an average of 2700 interactions.

Student ID	Problem Hierarchy	Problem Name	Problem View	Step Name	Knowledge Components	Correct First Attempt
stu_de2777346	Unit CTA1_01, Section CT	L1FB12	1	R1C1		1
stu_de2777346	Unit CTA1_01, Section CT	L1FB12	1	R1C2		0
stu_de2777346	Unit CTA1_01, Section CT	L1FB12	1	R2C1		1
stu_de2777346	Unit CTA1_01, Section CT	L1FB12	1	R2C2	Identifying units	1
stu_de2777346	Unit CTA1_01, Section CT	L1FB12	1	R3C1	Define Variable	1
stu_de2777346	Unit CTA1_01, Section CT	L1FB12	1	R3C2	Write expressior	1
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	MoreOrFew	Compare units	0
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	Conversionl	Enter unit conve	1
stu_de2777346	Unit UNIT-CONVERSIONS	UNITCONVEF	1	SelectFracti	Select form of or	1

Figure 2: Structure of data, from [5].

The information provided in the data set (see excerpt in Figure 2) included unique identifiers for the student and the interaction, identifiers for the task, information on the success of the student on this interaction, as well as time-stamp information. Every interaction was also marked with an indicator for whether the user solved the step correctly at the first attempt (CFA). The task of the competition was then to predict the "correct first attempt" value of each student for each step, on the basis of the data describing the previous interactions with the system. The step on which the CFA was to be predicted was always drawn from an interaction occurring later than the interactions in the data set.

Since the official test set is not available, we follow standard data splitting practices [10, 4]. In the same way that the organizers had created their test set by taking the last instance of each problem, we created an internal test set by separating out the last two instances of each step within the training set to create an internal test set roughly one tenth the size of the training set. As a result of this split, any step name that occurred fewer than three times was discarded. Ultimately, that left an internal data-set of 6,596,059 training instances, with 662,074 instances in the test set. This internal set then contains 13,128 distinct steps. This also meant that some students with very few lines were discarded, which left 3,079 students.

Due to time constraints it was only possible to test the predictions on a sample of 50 students. Results for the baseline systems are reported on these same 50 students, which means that they are tested on 11,888 rows in the test set (note, results are similar to the entire data set, cf. Section 4.3). The k-nearest neighbor systems still find the 41 most similar students among all 3,079, just like the average based baselines are still calculated from all 3,079 students.

4.2 Evaluation Method

We here use the same evaluation measure as in the KDD cup, i.e., root mean squared error: $RMSE = \sqrt{\frac{\sum(\tilde{c}-c)^2}{n}}$ where $\tilde{c}$ is the predicted score, c is the actual score, and n is the number of predictions.

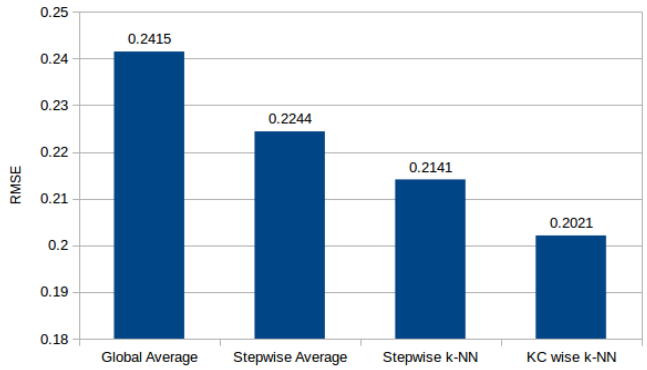


Figure 3: RMSE scores on the 50 student sample from the internal test set.

4.3 Results

Global average baseline. The first, and most basic baseline simply predicts the same score for every problem, 0.8494. The prediction is the average rate of correct first attempts, from the whole training set: $\tilde{c} = \frac{\sum c_{train}}{n}$
For first 50 students in the test set (11,888 predictions) this gave a score of: $RMSE = 0.2415$. For comparison, if we consider the entire test sets (662,074 rows), this system gave a score of $RMSE = 0.2394$.

Stepwise average baseline. The second baseline was already a clear improvement. This system distinguishes between stepnames, and uses the average score for the step in the training set to predict: $\tilde{c_i} = \frac{\sum c_{train_i}}{n_i}$
For first 50 students in the test set this gave a score of: $RMSE = 0.2244$ ($RMSE = 0.2255$ on the full set).

k-nearest neighbor (stepwise) baseline. The third baseline system is the replication of TJ's nearest neighbor system, which makes predictions by taking a weighted average of the scores on the predicted steps for the 41 students with the most similar results in the training set (cf. Section 3.1) This baseline gave further improvement on the second baseline. For the first fifty students in the development set this gave a score of: $RMSE = 0.2141$.

Knowledge component-based k-nearest neighbor system. Our expanded version of the k-nearest neighbor system also predicts on the basis of a weighted average of the scores for the 41 most similar students, but measures proximity on common steps with the same KCs rather than on common steps with the same names. For the first fifty students in the development set this gave a score of: $RMSE = 0.2021$. The results are visualized in Figure 3.

5. RELATED WORK

The 2010 KDD cup received submissions based on a large variety of approaches, many of the highest scoring system being ensemble methods such as [10] (ranking first). Another approach which also accounts for differences between students and problems combines HMMs with bagged decision trees, ranking fourth [6].

6. CONCLUSIONS

We propose to use the performance on similar steps instead of performance on identical steps as a novel distance measure in a collaborative filtering approach to ATS. So far, we only evaluated it on a reduced but reasonably large sample (11,888), but we hypothesize that the prediction error would remain low on the full set, particularly with optimization of hyper-parameters. One potential argument against using KCs is that an expert is needed to decompose the subject material and annotate the KCs. In order to provide learning material, it is necessary have a overview of what the material consists of and the order in which the different elements should be prescribed to best facilitate learning. It would be interesting to automatically learn such a structure, as in fact exploiting latent content is important for improved prediction [4]. However, the aim of this paper is to gauge whether exploiting KC information is sensible, and our preliminary results show that KCs are a potentially valuable source of information. They provide an opportunity to leverage the higher-level structure of the material to gain information about the learning process.

7. REFERENCES

[1] J. Anderson and C. Schunn. *Implications of the ACT-R learning theory: No magic bullets. Advances in instructional psychology.* Educational design and cognitive science, 2000.

[2] J. R. Anderson. ACT: A simple theory of complex cognition. *American Psychologist*, 51(4):355, 1996.

[3] W. J. Baumol. *The next twenty-five years of public choice*, chapter Health care, education and the cost disease: a looming crisis for public choice, pages 17–28. Springer Netherlands, 1993.

[4] S. Cetintas, L. Si, Y. P. Xin, and R. Tzur. Probabilistic latent class models for predicting student performance. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2013.

[5] T.-N. Nguyen. *Predicting Student Performance in an Intelligent Tutoring System.* PhD thesis, University of Hildesheim, 2011.

[6] Z. A. Pardos and N. T. Heffernan. Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research W & CP.*, 2010.

[7] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra I 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge., (2010). Find it at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

[8] A. Töscher and M. Jahrer. The bigchaos solution to the netflix prize 2008. *Netflix Prize Report*, 2008.

[9] A. Töscher and M. Jahrer. Collaborative filtering applied to educational data mining. *Proceedings of the KDD Cup 2010 Workshop*, 2010.

[10] H.-F. Yu, H.-Y. Lo, and et al. Feature engineering and classifier ensemble for kdd cup 2010. *Proceedings of the KDD Cup 2010 Workshop*, Feature engineering and classifier ensemble for KDD cup 2010.

Exploring the influence of ICT in online students through data mining tools

Javier Bravo
javier.bravo@udima.es

Sonia Janeth Romero
soniajaneth.romero@udima.es

María Luna
maria.luna@udima.es

Sonia Pamplona
sonia.pamplona@udima.es

Madrid Open University, UDIMA
Carretera de la Coruña 38.500. Vía de servicio 15.
28400, Collado Villalba, Madrid (Spain)

ABSTRACT

The aim of the present work is to evaluate differences according to age in digital competence, usages, and attitude towards ICT in a sample of 1231 online students of a distance university. To fulfill this goal, hypothesis testing, correlation analysis, and data mining techniques were performed on the basis of a 72-item survey. Results showed no strong differences between extreme groups of age. Besides, some interesting correlations between variables and additional information through association rules were found. This study led to better knowledge of online students in order to improve teaching and learning processes.

Keywords

Association rules, ICT attitude, ICT usages, distance education, online education, correlation, Mann-Whitney test, digital competence.

1. INTRODUCTION

During the last decades the proportion of higher education students taking at least one online course has outstandingly increased [1]. A research line developed in the field of e-learning in higher education focuses on the students' access, competences, actions and attitudes towards digital tools and devices and on how those variables are related to learning and well-being. As it is known, distance education, fostered by ICT, increases the variety of learners attending higher studies, creating new challenges for educators and institutions [2]. Specifically there are recent studies on whether the students' age is an important variable. In contrast to the concept of "digital natives" [10] , several studies find no evidences for strong discontinuity of young people on the use and attitudes about digital technology [6] [8]. Nevertheless differences related to age have been found, such us a deeper approach to studying of older students and less time spent using ICT (although this last difference is more noticeable at face-to-face universities than at the distance ones) [5,6].

In this paper we try to address this problem by developing the following objectives: (1) Compare digital competence, uses, and attitude towards ICT between young and students over 50; (2) Analyze relationships between variables in young and students over 50; (3) Obtain additional information about relationships between variables and group of ages by using data mining tools.

This paper is organized as follows. The next section presents a brief selection of related works. The method is described in section three. In section four the results are exhibited. Finally, the section five concludes the paper with discussion and plans for future work.

2. RELATED WORKS

Recently, some research studies were proposed to address the usage of data mining techniques in education especially in association rule mining.

Fattah et al. presented an association rule discovery model to investigate and analyze a university admission system database [3]. The model discovered the relation between students' data and their application status in the university system. The information discovered was very important to the admissions office in the analyzed university because it showed how to filter the applicants with respect to their record in high school.

García et al. described a data mining tool that uses association rule mining and collaborative filtering in order to make recommendations to instructors about how to improve e-learning courses [4]. This tool enables the sharing and scoring of rules discovered by other teachers in similar courses. The work showed and explained some examples of rules discovered in an adaptive web-based course.

Romero et al. explored the extraction of rare association rules when gathering student usage data from a Moodle system [11]. They showed how some specific algorithms, such as Apriori-Inverse and Apriori-Rare, are better at discovering rare-association rules than other non-specific algorithms, such as Apriori-Frequent and Apriori-Infrequent. Finally, they showed how the rules discovered by rare association rule mining algorithms can help the instructor to detect infrequent student behavior/activities in an e-learning environment such as Moodle.

Merceron and Yacef gave an interpretation of two measures of interest through association rules: cosine and added value [9]. In addition, they presented a case study that depicts a standard situation: a LMS that provides additional resources for students as a complement to the face-to-face teaching context. An important conclusion of this work is that common LMS are far from being data mining friendly. Thus, LMS should be enhanced with a special module with good facilities for exploring data.

Kumar and Chadha [7] used association rules mining in discovering the factors that affect assessment in Haryana University (India). They analyzed data for some courses taught in order to measure the students' performance based on factors such as instructor behavior, curriculum design, time schedule and students' interests.

3. METHOD

3.1 Participants, variables and instruments

A total of 1231 students participated voluntarily (with informed consent) in this study, 600 females and 631 males. They were all

students recruited from Madrid Open University in Spain. 63.44% of the sample were studying a Bachelor's degree and 36.56% were Master's students . 40.76% of the students worked in ICT related areas and 57.66% had completed undergraduate studies previously. All the participants were between 18 and 69 years old (mean= 36.01, SD=9.59) and 110 are older than 50 (age 50+ group).

A survey of 72 items was designed to test students' self-reported ICT abilities, uses and attitudes. This survey is divided into four parts: demographical data and academic performance, actions with digital devices (computers, Smartphones, and other digital devices), frequencies of use of ICT tools (digital devices, communications, Moodle, file managing, and other tools) and attitude towards ICT in the process of learning.

(1) Demographical data and academic performance. Students were asked about: age (integer), gender (1=female, 2=male), grade (from 0 to 10), study area (58 values), first enrollment in UDIMA (from 2009-10 to 2014-15), previous degrees, if they work in fields related to ICT and average grade on academic record (retrieved using student identity).

(2) Actions with digital devices. It is composed of 20 items distributed on three scales: Actions with Other Digital Devices (AODD-4 items), Actions with Computers (AC-8 items) and Actions with Smartphones and Tablets (AST-8 items). The format used is 4-point Likert type, from 1 (I cannot do it) to 4 (I can do it and explain it to others). Descriptive results on this block of the instrument are: AODD (min=4; max=16; mean=15.02; SD=1.75); AC (min=10; max=32; mean=28.65; SD=3.96); AST (min=11; max=32; mean=29.04; SD=3.76).

(3) Frequencies of use of ICT tools. It is composed of 25 items distributed on five parts: a) Other Digital Devices (FODD-5 items), b) Communications (FC-5 items), c) Moodle (FM-7 items) d) File Management (FFM-3 items), and e) Other Tools (FOT-4 items). The format used is 4-point Likert type, measuring frequency of use, from 1 (I do not use/do not know) to 4 (Very often). Descriptive results on this block of the instrument are: FDD (min=5; max=20; mean=14.97; SD=3.01); FC (min=7; max=24; mean=17; SD=3.69); FM (min=7; max=28; mean=19.48; SD=4.23), FFM (min=3; max=12; mean=6.32; SD=2.33); FOT (min=4; max=16; mean=6.69; SD=2.46).

(4) Attitude towards ICT in the learning process. It is composed of 24 items distributed on three scales: affective, cognitive and behavioral. The format used is 5-point Likert type from 1 (totally disagree) to 5 (totally agree). Two items on each dimension are inversely rated. The higher test score indicates greater favorable attitude towards the incorporation of ICT in the learning process. Descriptive results shows: min=29; max=120; mean=97.79; SD=13.49. Cronbach Alpha (considering all 24 items of attitude) was 0.89 indicating the high reliability of the test.

3.2 Data analysis

Data analysis included hypothesis testing, correlation and data mining analysis. These are detailed below.

3.2.1 Hypothesis testing and correlation

a) Hypothesis testing. Wilcoxon and Mann-Whitney tests were made to test the hypothesis about differences between extreme ages (24- and 50+).

b) Correlation. Pearson correlation matrix between continuous variables was made in order to evaluate possible associations.

3.2.2 Data Mining Techniques

To complement and provide additional information we used four data mining techniques: OneR, Decision trees (J48), Naïve Bayes and association rules. In this stage it is important to perform the preprocessing phase [12].

a) Preprocessing phase. It is important to note that in this analysis we utilized the whole data. In addition, the items of "Frequencies of use ICT tools" were grouped in four nominal variables: FODD (it contains the sum of five items of "Other Digital Devices"), FC (sum of five items of "Communications"), FM (sum of seven items of "Moodle"), FFM (sum of three items of "File Management"), FOT (sum of 4 items of Other tools). In the same way, the items of "Attitude towards ICT in the learning process" were grouped in the ictAttitude variable. Classification techniques works better with nominal variables. Therefore, age and ictAttitude were discretized to ictAttitude3groups and age4groups respectively. The ictAttitude variable is a continuous variable ranged from 29 to 120. We discretized this variable in three nominal values according to its 33 percentile, 66 percentile, and 99 percentile. Regarding to age variable, it ranged from 18 to 69. This variable was discretized in four values according to its four quartiles. This new variable is called ictAttitude3groups.

b) Selection of variables. In this phase we only use 14 variables: codDegree, gender, firstEnrollment, gradeRound, AODD, AC, AST, FODD, FC, FM, FFM, FOT, age4groups, and ictAttitude3groups. In addition, we utilized the WraperSubsetEval method provided by Weka [13]. This metaselection method selects the most appropriate variables for a data mining technique. This method receives two variables: the selection method and the search method. Since OneR, J48, and Naïve Bayes are classification techniques we indicated to this method to use J48 for selection method (selection mode: 10-fold cross-validation). Also, BestFirst forward method was used for searching method. As a result 8 variables were selected: gender, AODD, AC, FM, FFM, FOT, age4groups, and ictAttitude3groups.

c) Application of techniques. In this phase we utilized three classification techniques and association rules. The classification techniques utilized were: OneR, J48, and Naïve Bayes. We utilized for these techniques the 8 variables listed above (class variable: ictAttitude3groups). In order to select the most appropriate technique we calculated the accuracy (number of correctly classify instances) of each technique. As the size of data was enough to apply the split method, which divides the sample in two parts: training and testing data, we utilized it instead of the cross-validation method. Moreover, we utilized 80% of data for training and 20% for testing. It is well known that if the data size is large enough both methods of dividing the data should give similar accuracies. Concerning association rules we utilized the Apriori algorithm with confidence=0.9 and minimum support=0.1.

4. RESULTS

4.1 Hypothesis testing and correlations

a) Hypothesis testing. First, assumption evaluation was made in order to decide statistical techniques to compare students of extreme ages: above 50 years (50+), and below 24 years (24-). Levene's test shows a lack of homoscedasticity between groups in the variables related with actions: L=20.068, (1, 125), p<.001 for

AODD; L=19.177, (1, 125), p<.001 for AC and L=58.028, (1, 125), p<.001for FDD. The Shapiro-Wilk test shows a lack of normality in all the variables except FC. Due to failure to meet the assumptions we decided to use nonparametric statistic (Mann-Whitney test) to compare both groups. As it can be seen in Table 1, significant differences between both groups (50+ and 24-) were found on actions with other digital devices, actions with Smartphones and tablets, and frequency of use of communication tools. The sum of ranks (see Table 1) indicated high scores on AODD, AST and FC in the group of age 24-.

b) Correlation. The Pearson correlation matrix between continuous variables was calculated for both groups. In the group of age 50+ all the action variables correlated significantly and positively with frequency variables and attitude towards ICT: the lowest positive and significantly correlation was r=.111(AC-FC) and the highest r=.724 (AC-AODD), this result was expected because the more digital competence self-perceived, the more frequent use of ICT tools and the more positive attitude towards ICT use in the learning process. On the other hand, AODD, AST and FC correlated negatively with age (r=-.148, p<.01; r=-.208, p<.01 and r=-.069, p<.05 respectively) indicating as the age increased, the self-perceived competence in actions with digital devices and Smartphones decreased and the use of communication tools was less frequent. The average grade correlated positively with self-perceived competence in actions with computers (r=.136, p<.01). And finally, the number of years studying through distance learning correlated with AC (r=.087, p<.01), FDD (r=.068, p<.05), FM (r=.083, p<.01), attitude (r=.128, p<.01) and age (r=.278, p<.01) indicating that the more years of experience studying online, the higher self-perceived competence of actions with computers and higher frequency of use of digital devices, Moodle, and better attitude toward ICT.

In the group of young students we found less significant correlations, as the number of years studying online did not correlate with any variable, the average grade only correlate positively with AC (r=.197, p<.01), the age correlated inversely with AST (r=-.226, p<.05) and FDD (r=-.307, p<.01). The attitude toward ICT correlated positively with all the actions (r=.205, p<.01 for AODD; r=.221, p<.05 for AC and r=.305, p<.01 for AST) and also with FM (r=.300, p<.01). Finally, the scales of frequencies correlated positively with each other but not with the actions scales.

Table 1. Mann-Whitney U, Wilcoxon W, Z and significance

	ADD	AST	FC
U	5098.500	3429.500	5528.500
W	10454.500	8785.500	10781.500
Z	-4.082	-6.895	-2.438
Sig.	.000	.000	.015
Age	Rank sum	Rank sum	Rank sum
24-	18225.5	17170.5	19894.5
50+	10454.5	11509.5	8785.5

4.2 Data mining techniques

4.2.1 Classification techniques

The accuracies of applying the OneR, J48 and Naïve Bayes technique are as follows: 41.9%, 43.8%, and 42.9%, respectively. For example, J48 classifies correctly in 43.8% of the instances. It is clear that neither of the three techniques has an accuracy greater than 44%. As none of these techniques obtained reliable results, we applied the association rules technique.

4.2.2 Association rules

This trial consisted of using 14 variables with the Apriori algorithm. It is important to highlight that the Apriori algorithm only works with nominal variables. Therefore, the grade variable was removed from the data. The parameters for this algorithm were: minimum support=0.1; confidence=0.9; number of rules=20; instances=1231; attributes=(codDegree, gender, firstEnrollment, AODD, AC, AST, FODD, FC, FM, FFM, FOT, age4groups, ictAttitude3groups).

The result of applying this algorithm is presented in Table 2. It is shown that the Apriori algorithm selected 16 rules. Thus, it is shown that AR6 indicates that students with the best score in actions with computers and best ICT attitude will have the best score in actions with other digital devices. The AR7 and AR14 rules indicate that both genders will have the best score in "actions with other digital devices" if they have the best score in "actions with computers". Both association rules are redundant, since this fact is indicated in AR11. The AR9 contains the variable age. It indicates that the students between second and third quartile of age with the best score in "actions with computers" are experts managing other digital devices.

Regarding the other association rules interesting relations between several variables were found. The first rule relates actions with computers, actions with Smartphones, and actions with other digital devices. Thus, it means that students with the best ICT attitude who demonstrate a high level of actions with computers and Smartphones, will have a high level of actions with other digital devices. The AR2 rule shows a similar relation, but only for male students. The AR3 is informed about a general relation between actions with computers, actions with Smartphones, and actions with other digital devices. A value of 32 indicates a high level of actions with computers and Smartphones, and a value of 16 is also a high level of actions with other digital devices. Thus, a student with a high level of actions with computers and Smartphones, he/she will have a high level of actions with other digital devices. Interesting information is revealed in the AR4 rule, since it relates the first enrollment, action with computers, Smartphones, and with other digital devices. Thus, the students of the 2014-15 year show a high level of action with computers and Smartphones, and also with other digital devices. This association rule is similar to the AR10 and AR15, but with less information. The rules AR5 and AR6 show that students with a high ICT attitude, and a high value in actions with computers or Smartphones, will have a high level in actions with other digital devices. Finally, the AR8 and AR16 rules relate the gender, action with computers and with other digital devices. Consequently, female and male students report the same abilities in actions with Smartphones and other digital devices.

5. CONCLUSIONS

The present study gathered a large sample composed of 1231 online students in a distance university with a range of age from 18 to 69 years. Our results agree to a great extent with other related studies [5][6]. In fact, we did not find enough evidence of strong differences among extreme groups of age, although results showed slight differences in variables related with the frequency

of use and perceived competence with Smartphones and communication tools.

Another interesting conclusion is that attitude towards ICT did not correlate inversely with age, on the contrary, students aged 50+ exhibited positive attitudes towards the implementation of ICT for the learning process. These conclusions lead to better knowledge about students attending online higher education. Therefore, these results should provide improvements in the methodology of the e-Learning courses and foster the utilization of communication tools (less utilized by 50+ students).

This work also showed that data mining techniques can provide complementary information to traditional analysis methods. Although classification techniques did not provide reliable results, since its accuracy was less than 44%, the association rules technique provided deeper information. In fact, the Apriori algorithm obtained 16 association rules. These association rules showed relationships between the following variables: actions with computers, Smartphones and other digital devices, gender, ITC attitude, and first enrollment in UDIMA. This information was not provided by the hypothesis testing, therefore, we have demonstrated that association rules are appropriate to analyze these data.

For future work it will be appropriate to analyze other parameters of the Apriori algorithm that could provide rules with more information. For instance, to test and evaluate other selection methods based on Lift or Leverage is an interesting future line of research [9].

Table 2. Best rules of the Apriori algorithm

Rule		Cov.	Conf.
AR1	AC=32 AST=32 ictAttitude3groups=3 ==> AODD=16	137	1
AR2	gender=2 AC=32 AST=32 ==> AODD=16	221	0.99
AR3	AC=32 AST=32 ==> AODD=16	307	0.99
AR4	firstEnrollment=2014-15 AC=32 AST=32 ==> AODD=16	134	0.99
AR5	AST=32 ictAttitude3groups=3 ==> AODD=16	167	0.98
AR6	AC=32 ictAttitude3groups=3 ==> AODD=16	187	0.97
AR7	gender=1 AC=32 ==> AODD=16	143	0.97
AR8	gender=2 AST=32 ==> AODD=16	264	0.97
AR9	AC=32 age4groups=3 ==> AODD=16	123	0.96
AR10	firstEnrollment=2014-15 AC=32 ==> AODD=16	191	0.96
AR11	AC=32 ==> AODD=16	426	0.96
AR12	AST=32 ==> AODD=16	390	0.95
AR13	gender=2 firstEnrollment=2014-15 AC=32 ==> AODD=16	126	0.95
AR14	gender=2 AC=32 ==> AODD=16	283	0.95
AR15	firstEnrollment=2014-15 AST=32 ==> AODD=16	182	0.93
AR16	gender=1 AST=32 ==> AODD=16	126	0.91

6. ACKNOWLEDGMENTS

This work has been funded by European Commission Lifelong Learning Programme through the AGE50+ project (hash code: 1B8E6E156B13780A).

7. REFERENCES

[1] Allen, I. E., & Seaman, J. 2013. *Changing Course: Ten Years of Tracking Online Education in the United States*. Sloan Consortium. PO Box 1238, Newburyport, MA 01950, USA.

[2] Altbach, P. G., Reisberg, L., & Rumbley, L. E. 2009. Trends in global higher education: Tracking an academic revolution. UNESCO.

[3] Fattah Mashat, A., M.Fouad, M., S. Yu, P. and F. Gharib, T. 2013. Discovery of Association Rules from University Admission System Data. *International Journal of Modern Education and Computer Science* 5,4 (May 2013), 1–7.

[4] García, E., Romero, C., Ventura, S. and de Castro, C. 2011. A collaborative educational association rule mining tool. The Internet and Higher Education 14, 2 (March 2011), 77–88.

[5] Hosein, A., Ramanau, R., & Jones, C. 2010. Learning and living technologies: a longitudinal study of first year students' frequency and competence in the use of ICT. *Learning, media and technology* 35, 4 (2010), 403-418.

[6] Jelfs, A., & Richardson, J. T. 2013. The use of digital technologies across the adult life span in distance education. *British Journal of Educational Technology* 44, 2 (2013), 338-351.

[7] Kumar, V. and Chadha, A. 2012. Mining association rules in student's assessment data. *International Journal of Computer Science Issues* 9, 5 (September 2012), 211–216.

[8] Lai, K. W., & Hong, K. S. 2014. Technology use and learning characteristics of students in higher education: Do generational differences exist?. *British Journal of Educational Technology.* doi:10.1111/bjet.12161

[9] Merceron, A. and Yacef, K. 2008. Interestingness measures for association rules in educational data. *Proceedings of the 1st International Conference on Educational Data Mining* (Montreal, Quebec, Canada, June 20-21, 2008). 57–66.

[10] Prensky, M. 2001. Digital natives, digital immigrants part 1. *On the horizon, 9*, 5 (2001), 1-6.

[11] Romero, C. and Romero, J. 2010. Mining rare association rules from e-learning data. *Proceedings of the 3rd International Conference on Educational Data Mining* (Pittsburgh, PA, USA, June 11-13, 2010). 171–180.

[12] Vialardi, C., Bravo, J., & Ortigosa, A. 2008. Improving AEH Courses through Log Analysis. *Journal of Universal Computer Science* 14,17 (Sept. 2008), 2777-2798.

[13] Witten, I.H., Frank, E., & Hall, M.A. 2011. Data Mining: practical machine learning tools and techniques. Morgan Kaufmann Publishers. Burlington, MA 01803, USA.

Understanding Revision Planning in Peer-Reviewed Writing

Alok Baikadi
University of Pittsburgh
3939 O'Hara St
Pittsburgh, PA 15213
baikadi@pitt.edu

Christian Schunn
University of Pittsburgh
3939 O'Hara St
Pittsburgh, PA 15213
schunn@pitt.edu

Kevin Ashley
University of Pittsburgh
3939 O'Hara St
Pittsburgh, PA 15213
ashley@pitt.edu

ABSTRACT
Revision is a core writing skill that presents challenges to both novice and expert writers. Within the context of peer review, peer feedback has the potential to provide rich guidance for revision, especially when making content-level changes. However, authors must review and evaluate each piece of feedback for meaningful critiques that can be applied to further drafts. In this work, we analyzed several factors that influenced students' decisions to fix or ignore comments they received. We found that feedback on content dimensions, as well as critical remarks by both the reviewers, and by the authors regarding papers they reviewed, were correlated with the amount of revisions made between drafts.

Keywords
peer review, revision, writing instruction

1. INTRODUCTION
Revision has long been seen as one of the cornerstones of effective writing [6]. Practicing revision has been shown to not only improve the produced writing, but also help on first drafts of future writings [9]. One of the discriminators between expert and novice writers is how they approach revision. While both groups often make many surface-level edits, such as spelling, grammar, and stylistic revisions [4, 14, 17, 2], expert writers often make a higher proportion of content-level edits than do novices [3].

By using a peer-review approach, students were able to employ more strategic revision strategies given peer feedback [10], make fewer surface-level changes [15], and add more details in their writing [12], especially when peers provide justification for their feedback [7]. Once feedback is received, it is not always implemented in future drafts [5, 2]. Sometimes students indicate an intention to implement meaningful changes but do not follow through with the intent [4]. Checklists [16] and revision memos [1] have been used to focus students' revisions on important aspects of their writing. Within peer-review, it has not been clear how often students forget about the feedback received during revision, rather than make a choice to disregard the feedback. An accurate model of revision behavior could allow a teacher or intelligent system to intervene for students who require additional support. Diagnostic information could also be presented to the teacher as to what kinds of comments are being made, how they are being received, and what sorts of revisions to expect in future drafts. An effective model could also be used to provide hints to students, about how their feedback may be received as reviewers and which comments provide meaningful feedback for revision as an author.

In this work, we investigated this decision within a web-based peer-review application. We present a revision planning application designed to scaffold the process of evaluating feedback received in the peer-review process. We analyzed their responses within the system in order to better understand why some comments may be addressed while others are ignored. Critical comments about the content of the paper, rather than the surface aspects, were more likely to be included in their revision plan, and were more highly correlated with changes in the second draft.

2. REVISION PLANNING CORPUS
Web-based, computer-supported peer review has been shown to be an effective tool for improving students' writing skills. Students still need support, however, in organizing the reviews they receive and planning how to revise their own papers. This paper describes a revision environment that helps students to cluster and prioritize reviewers' suggestions, develop a plan for revision their papers, and make note of lessons learned about writing for future use. We report here about students' experiences in using the tool in an undergraduate Cognitive Psychology course.

2.1 SWoRD Peer Review
Scaffolded Writing and Rewriting in the Disciplines (SWoRD) is a web-based reciprocal peer review system. Over the past 12 years, it has been used by over thirty-five thousand students across grade levels and across a variety of academic disciplines. The peer review process within SWoRD takes place in three phases: An Authoring phase, a Review phase, and a Revision phase. In the first phase, students submit a response to an instructor-provided writing prompt. Students may either enter text into the web interface, or upload a pre-existing document in order to submit their assignments. During the Review phase, students are presented

with the grading rubric and comment prompts the instructor has provided along with the submitted document. The student reads the document and provides written feedback for each evaluative dimension, as well as numerical scores on a seven-point rating scale. In the final phase, students receive the feedback and scores generated by their peers. The process is then repeated for the second draft.

2.2 Revision Planning

During the course of peer review, students have the opportunity to learn from both giving and receiving feedback. During the review process, students are asked to critically evaluate a peer's submission on the same rubric with which their own writing will be judged. While reviewing, students may notice aspects of their peers' submissions that they can incorporate into their own work. Many revisions occurred when the student both recognized it in a peer's work, as well as received feedback on the same topic from their peers [13].

To support this process, the Revision Planning system has two components. The Lessons Learned page, shown in Figure 1, is available to the student during the reviewing process. It encourages them to make observations on the papers they are reading, and how that may be applied to their own document. They are able to identify the observation as a good idea that they'd like to consider for their revisions, or a problem that they would like to avoid.

The Revision Planner, shown in Figure 2, allows students to consider how they would address each comment they receive from their peers. For each comment, they can elect to ignore it or fix it. If they choose to fix it, they can then assign a priority and make notes on what the revision will be. If they choose to ignore it, they can select a reason from a drop-down menu, or add text to explain why it is being ignored. Both the Revision Planner and the Lessons Learned are visible during revision. The system can also generate checklist that the students can use during their revisions.

2.3 Data Collection

The data were collected from 75 college students in an introductory Cognitive Psychology course, all of whom had completed a required writing seminar prior to enrollment. The students were asked to write a 1,000 word article imitating a newspaper style that connects topics discussed in class with their everyday lives. The rubric included several dimensions regarding the communicativeness of the article, such as its interestingness, word choice, and quality of writing, and several about the course content, such as the relevance and accuracy of the concepts introduced in the course. Of the 75 students, 60 completed the Revision Plan, and 44 completed the Lessons Learned. A second draft was submitted, and subjected to the same peer review process, without additional revision planning support.

Each student was asked to review four peer submissions during the revision phase. In addition, students were allowed to perform bonus reviewing for extra credit. For each review (n=297), we collected 10 numerical scores, which were separated among the five evaluation dimensions. Students were required to write at least one textual comment for each dimension, though they could provide up to five different textual comments for a single dimension. For each textual comment the student received (n=1822), we recorded the decision to "Fix" or "Ignore" the comment, a discretized reason for marking the comment as "Ignore" when provided, as well as the text of the intended revision and priority.

3. REVISION PLANNING BEHAVIOR

Using the data described above, we investigated four main research questions: (1) what factors influenced the students' decision to fix or ignore a comment that they received, (2) what were the reasons that students gave for ignoring a comment, (3) how is the process of revision planning within Anonymous related to the amount of revisions between the first and second drafts, and (4) how are the observations made on the Lessons Learned page related to the amount of revisions between the first and second drafts.

3.1 Fix and Ignore Decisions

For each comment, we calculated a score given by the reviewer by averaging all scores for the comment's dimension. If there were multiple comments within the same dimension, they received the same score. The score serves as a proxy for how critical a comment is. A dimension type (content or communication) was derived by grouping the three communication-related dimensions together, and grouping the other two dimensions as content. Prior work [17] has indicated that content feedback is more likely to result in content revisions. The length of the comment was computed in number of characters, following the intuition that longer comments are more likely to contain useful feedback.

On average, students elected to mark only 44% of their comments as "Fix" (sd=0.21). We performed a logistic regression analysis, shown in Table 1, to determine which factors influenced the decision to fix or ignore a comment. All three factors were shown to have a significant main effect, and there was a marginally significant interaction between the score and the dimension type.

Table 1: Logistic Regression for Fix Decisions

Variable	Coefficient	z-Value	p-Value
Score	0.74	-5.61	< 0.001
Content Dimension	3.71	238	0.018
Comment Length	1.00	9.73	< 0.001
Score x Dimension	0.85	-1.74	0.082

On average, students elected to fix approximately 40% of their comments in the Communication dimensions, compared to 48% of their comments in the Content dimensions. Comments marked as "Fix" were on average longer (mean=283) than those marked as "Ignore" (mean = 188). Figure 3 shows the proportion of comments fixed by score and the type of dimension.

3.2 Ignore Reasons

There were seven categories of reasons students could select when they ignored a comment: no critique was given, the student disagreed with the comment, the comment was already mentioned elsewhere, the comment is only praise, the comment is only a summary, the comment was confusing, and other. Figure 4 shows the distribution of categories that were provided if any was given. Since the "Summary",

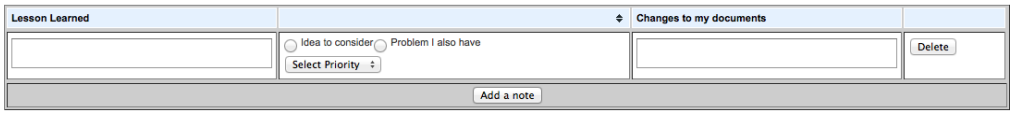


Figure 1: Lessons Learned Page

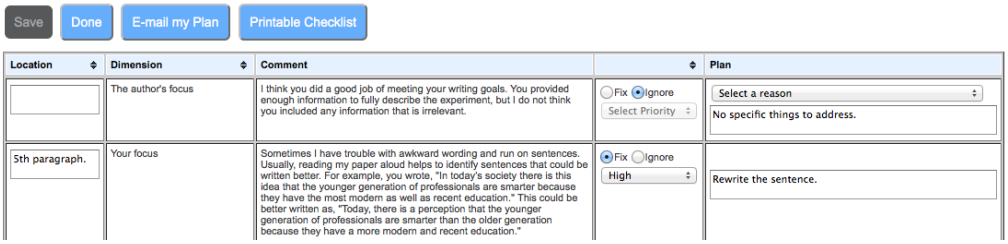


Figure 2: Revision Planning interface

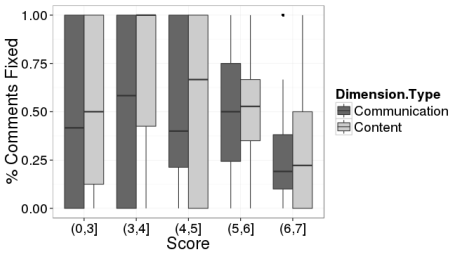


Figure 3: Percent of Comments marked as Fix by Score and Dimension Type

"Confusing", and "Other" categories occurred relatively infrequently, we omitted them from further analyses.

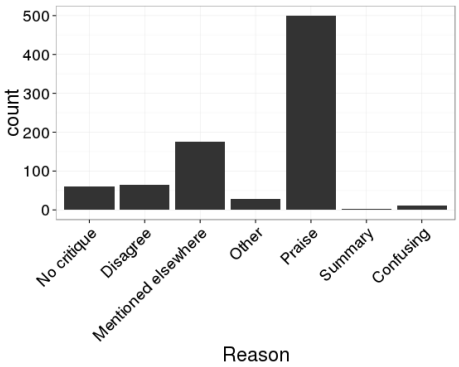


Figure 4: Distribution of Ignore Reasons

Table 2 shows the results of a multinomial logistic regression analysis, relative to the "Praise" category, to determine which factors influenced the category. There was a significant effect of the score for distinguishing all categories. In addition, there was a significant effect of dimension type for the "Mentioned Elsewhere" category, and a significant effect for the comment length on both the "Disagree" category, and the "Mentioned Elsewhere" category.

Table 2: Logistic Regression for Ignore Reasons

Reason	Content	Score	Comment Length
No Critique	0.62	0.66 **	0.99
Disagree	1.38	0.35 ***	1.01 ***
Elsewhere	0.46 *	0.32 ***	1.01 ***

3.3 Revision Planning and Revision

In order to measure changes in the drafts, all submissions were first converted to a plain text format. Both drafts were then segmented using the Stanford Parser [11] and compared using CompareSuite, a software package for analyzing text documents. Edits were compared at the sentence level by calculating how many sentences were added, deleted, or modified [8]. These numbers were then compared against the number of sentences in the first draft to calculate the amount of change between drafts. There was a weak correlation (r=0.20) between the proportion of comments labeled as "Fix", and the amount changed. However, there was a moderate relationship with the proportion of Content comments labeled as "Fix" (r=0.37), while there was no relationship (r=0.10) with the proportion of Communication comments labeled as "Fix".

3.4 Lessons Learned and Revision

For students who completed the lessons learned (n=44), we also investigated how the different types of observations effected the revisions. Students made an average of 2.8 (sd=1.96) observations (See Figure 5). Pearson correlations showed that neither the number of good observations (r=-0.14) nor the total number of observations (r=-0.039) was correlated with the amount of revisions. However, the number of critical observations was moderately correlated with the amount of revision (r=0.31).

4. CONCLUSIONS AND FUTURE WORK

In this work, we analyzed several factors that influenced students' decisions to fix or ignore a comment they received. The content dimensions offered the most insight into the revision behavior of the students. Content comments were more likely to be marked as a comment to fix, and when they were fixed were more highly correlated with the amount of

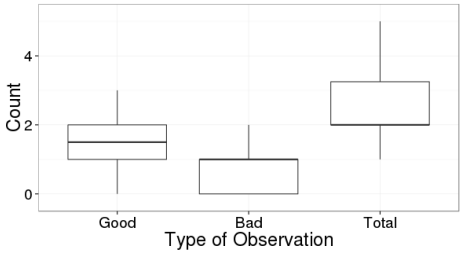


Figure 5: Distribution of Lessons Learned

revision done between drafts. While we did not analyze the comments made by the students, the students were specifically instructed to give feedback on the breadth and accuracy of the domain content in the content dimensions. In addition, lower scoring comments were more likely to be marked as fix or marked as "Mentioned Elsewhere", especially in the content dimensions. This latter selection indicates that the students intended to fix these issues, but had recognized them either through their own experience or through other comments, and were therefore more willing to ignore the specific feedback in those comments. Comments that were highly scored were more likely to be praise or otherwise lack critique. Relatively few comments were ignored because the students disagreed with the feedback received, and those tended to be at the extremes of the scores. The fact that few comments were ignored due to a disagreement with the critique, and the fact that critical observations made from other peers' submissions were more highly correlated with the amount of revision between drafts suggests that students benefit more from critical analysis of the papers they have both read and written.

One of the discriminating features between novice and expert writers is how they approach revision, particularly in terms of how often they revise for deeper meaning. While our results show correlations to the amount of revision done, further analysis will need to be done regarding the quality of the revisions. While comment length was surprisingly informative, it is an extremely shallow measure of the comment text. There are also many other factors that could inform the students' decisions on how to approach the comments they get, such as the helpfulness rating, and the relative strength of the writing skills between the author and reviewer. In terms of student revision process, a more fine-grained analysis of whether students fixed the comments they said they would, could be instrumental in supporting the effectiveness of the scaffolding mechanisms. It was also somewhat surprising that critical observations of peers' papers in the Lessons Learned were also correlated with more revision. One question raised by this observation is whether students learn more from giving critical feedback of peers' work than they do from giving positive feedback.

5. ACKNOWLEDGMENTS

This work is funded by the Institute of Education Sciences, under grant R305A120370.

6. REFERENCES

[1] Bardine, B. A., and Fulton, A. Analyzing the Benefits of Revision Memos during the Writing and Revision Process. *The Clearing House, 81* (4). 149–154.

[2] Calvo, R. A., Aditomo, A., Southavilay, V., and Yacef, K. The use of text and process mining techniques to study the impact of feedback on students' writing processes. *International Conference of the Learning Sciences*, 1 (2012), 416–423.

[3] Fitzgerald, J. Research on Revision in Writing. *Review of Educational Research, 57* (4). 481–506.

[4] Fitzgerald, J., and Markham, L. R. Teaching children about revision in writing. *Cognition Instruct, 4* (1). 3–24.

[5] Fitzgerald, J., and Stamm, C. Effects of Group Conferences on First Graders' Revision in Writing. *Writ Commun, 7* (1). 96–135.

[6] Flower, L., and Hayes, J. A cognitive process theory of writing. *Coll Compos Commun, 32* (4). 365–387.

[7] Gielen, S., Peeters, E., Dochy, F., Onghena, P., and Struyven, K. Improving the effectiveness of peer feedback for learning. *Lear Instr, 20* (4). 304–315.

[8] Hashemi, H. B., and Schunn, C. D. A Tool for Summarizing Students' Changes across Drafts. In *Intelligent Tutoring Systems* (Honolulu, HI, 2014), Springer International Publishing, pp. 678–682.

[9] Hillocks, G. J. The Interaction of Instruction, Teacher Comment, and Revision in Teaching the Composing Process. *Res Teach Engl, 16* (3). 261–278.

[10] Keen, J. Strategic revisions in the writing of Year 7 students in the UK. *The Curriculum Journal, 21* (3). 255–280.

[11] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the Association for Computational Linguistics: Systems Demonstrations* (2014), pp. 55–60.

[12] Morris Kindzierski, C. M. "I Like It the Way It Is!": Peer-Revision Writing Strategies for Students With Emotional and Behavioral Disorders. *Preventing School Failure: Alternative Education for Children and Youth, 54* (1). 51–59.

[13] Patchen, M. *Peer Review of Writing: Learning From Revision Using Peer Feedback and Reviewing Peers' Text.* Doctoral dissertation, University of Pittsburgh, 2011.

[14] Patthey-chavez, G. G., Matsumura, L. C., and Valdés, R. Investigating the process approach middle to writing instruction in urban schools. *Journal of Adolescent & Adult Literacy, 47* (6). 462–476.

[15] Peterson, S. Peer Response and Students' Revisions of their Narrative Writing. *Educational Studies in language and Literature, 3*. 239–272.

[16] Smede, S. D. Interior Design: Revision as Focus. *Engl J, 90* (1). 117–121.

[17] Yagelski, R. P. The role of classroom context in the revision strategies of student writers. *Res Teach of Engl, 29*, (2). 216–238.

Convergent Validity of a Student Model: Recent-Performance Factors Analysis

Ilya Goldin
Center for Digital Data, Analytics, and Adaptive Learning
Pearson
ilya.goldin@pearson.com

April Galyardt
University of Georgia
110 Carlton St.
Athens, GA
galyardt@uga.edu

ABSTRACT

Models of student performance can incorporate a skill decomposition that lists the skills that each activity requires. A good model must be sensitive to improvements in skill decomposition. We validate the Recent-Performance Factors Analysis model of student performance by checking its sensitivity to the skill decomposition. We use a dataset from a tutoring system where the skill model has been improved by the Learning Factors Analysis algorithm for skill model refinement and by expert validation. We find that R-PFA reflects improvements in the skill model, providing evidence of convergent validity of R-PFA. We argue that R-PFA may be sensible as a predictive model in Learning Factors Analysis because of its convergent validity and because the R predictor of R-PFA represents mastery-aligned learning curves.

1. INTRODUCTION

Predictive models of student performance often incorporate a skill model. For example, the Additive Factors Model [3] embeds a Q-matrix [11, 1] to relate prior practice on a skill to subsequent practice on the same skill. Bayesian Knowledge Tracing [4] similarly uses a skill model in that all BKT parameters are specific to a skill.

A skill model annotates instructional activities in terms of the skills that the activities require. This tagging can be wrong, or at least suboptimal, degrading instruction in several ways. For instance, if the tagging fails to distinguish two skills, it will treat all assessments of the two separate skills as assessments of one combined skill. In fact, because a student may have differential mastery of the two skills, the combined assessment may cause a tutoring system to call for extraneous practice for one skill, and insufficient practice for another. It follows that the refinement of a skill tagging of activities can advance instruction and assessment.

When a predictive model of student performance incorporates a skill model, we can validate the performance model by seeing if it is sensitive to changes in the skill model. A learning curve represents the "power relationship between the error rate of performance and the amount of practice" [3], plotting average error across students at every practice opportunity. If the curve treats a whole curriculum as one skill, its slope will be flat, because there will be both drops and spikes in the error rates as students learn one part of the curriculum after another. If we plot separate curves for distinct skills, their slopes will not be flat, corresponding to error rates dropping as students learn. This is the intuition for the Learning Factors Analysis algorithm [3], which searches the space of possible refinements to a skill model.

Prior study of representations of recent student performance, including box and exponential kernels with a range of bandwidths, produced the Recent-Performance Factors Analysis (R-PFA) model [6, 5]. In the recency representations with the highest predictive accuracy, the weight given to the each observation decreased with the age of the observation, placing $\sim 50\%$ of weight on the last 2 attempts, and $\sim 80\%$ on the last 5. This optimal weighting was consistent across real data and a variety of simulated student behaviors.

The current work validates R-PFA by checking whether its fit to data is improved by sensible changes to the skill tagging in a dataset. The following section describes a dataset and its multiple skill models, and presents R-PFA and several comparison models. The subsequent section reports that R-PFA and the other models are all sensitive to improved skill tagging, but R-PFA has the highest predictive accuracy among the models. Finally, we discuss how R-PFA may be interpreted as representing mastery-aligned learning curves [8], and R-PFA may fit within the Learning Factors Analysis algorithm for skill model refinement.

2. METHODS

We evaluate R-PFA on a dataset in which the skill tagging has been well-studied and revised [7], originating from Cognitive Tutor Geometry by Carnegie Learning [10, 2]. This tests R-PFA in two ways; first, how will R-PFA perform in terms of predictive accuracy? Second, does R-PFA agree with prior refinement of the skill model in this dataset [7]?

This Geometry dataset has three skill models that vary in how they treat "forward" and "backward" computations of area of geometric figures [7]. The original tagging (called Merged) separates area computation by geometric shape (square, circle, etc.), but merges together forward and backward computation. The Circle-Square tagging has separate

skills for the forward and backward computations for circles and squares. The Distinct tagging has separate forward and backward skills for each of many shapes. The geometry data set contains 38,426 unique actions by 82 students. The total number of skills in each tagging is 56 in Merged, 58 in Circle-Square, and 66 in Distinct.

We compare R-PFA to baseline models Item Response Theory 1PL, Additive Factors Model [3] and Performance Factors Analysis [9] (Eqs. 1-4). All student and skill intercepts and slopes are "random", that is, drawn from a common distribution. Treating skill parameters as random "borrows strength" for their estimation by proposing that infrequently practiced skills ought to have similar parameters as skills for which more data are available. Notation: j indexes skills, i indexes students, t indexes practice opportunities. T_{ijt} is the count of prior practice, S_{ijt} is the count of prior successes, and F_{ijt} is the count of prior failures.

$$\text{IRT 1PL} \quad \theta_i + \beta_j \tag{1}$$

$$\text{AFM} \quad \theta_i + \beta_j + \gamma_j T_{ijt} \tag{2}$$

$$\text{PFA} \quad \theta_i + \beta_j + \alpha_j S_{ijt} + \rho_j F_{ijt} \tag{3}$$

$$\text{R-PFA} \quad \theta_i + \beta_j + \delta_j R_{ijt} + \rho_j F_{ijt} \tag{4}$$

R_{ijt} is the proportion of recent successes in R-PFA (Eq. 5):

$$\text{exponential kernel } R_{ijt} = \frac{\sum_{p=-2}^{t-1} d^{(t-p)} X_{ijp}}{\sum_{p=-2}^{t-1} d^{(t-p)}} \tag{5}$$

3. RESULTS AND DISCUSSION

3.1 Predictive Accuracy

We compare predictive model accuracy in terms of AIC, a metric that rewards models for predictive accuracy and penalizes them for using excessive parameters. AIC is comparable to cross-validation with a prediction error loss function, but is more appropriate for sparse datasets, such as when only a handful of students may practice a skill [6].

Table 1: Predictive accuracy (lower AIC is better).

	Skill tagging		
Model	Merged	Cir-Sq	Distinct
IRT 1PL	21652	21538	21523
AFM	21373	21252	21272
PFA	21326	21197	21211
exp R-PFA r(0.7), f(0.1)	21142	20969	21003
exp R-PFA "best" from search	21134	20949	20977
"best" decay rates: R, F	0.7, 0.3	0.5, 0.3	0.4, 0.3

For all 3 skill taggings, R-PFA has higher predictive accuracy than the other models, with PFA, AFM, and Item Response Theory 1PL following in that order (Table 1). IRT 1PL has the lowest predictive accuracy, likely because it does not reflect learning over time. At the best-performing R and F decay weights from prior work (0.7 and 0.1, respectively), the number of parameters in PFA and R-PFA is exactly the same, and R-PFA's advantage in AIC over PFA is due to increased predictive accuracy.

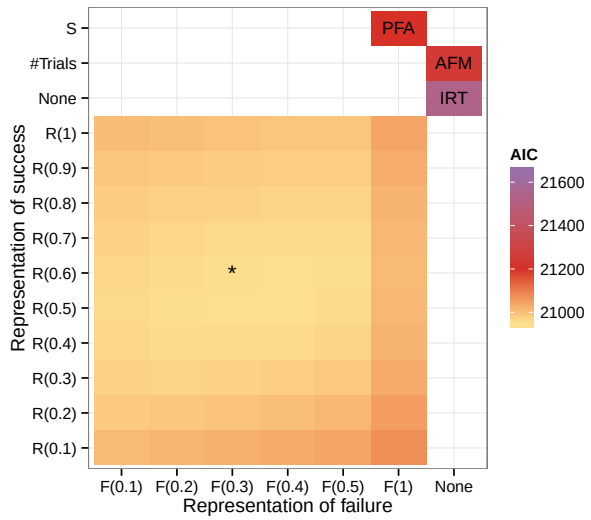


Figure 1: AIC for all 63 models on Circle-Square tagging. * denotes the best overall model.

Searching over decay rates shows that R-PFA is robust to a range of rates (Fig. 1). Even though the strictly lowest AIC uses decay rates that differ from prior work, this effect is smaller (26 points on Distinct, Table 1) than the effect of using R-PFA over other models or of improving the skill tagging, and R-PFA's performance degrades gracefully. Tuning decay rates separately for skill models has only a marginal benefit, and may confound skill model comparison.

We compare the learning curves of the 4 performance models (Fig. 2 and 3), omitting practice opportunities with fewer than 5 students. The red curves show the empirical percent correct at each opportunity, with a binomial 95% Bayesian credible interval that uses a Jeffreys prior. For example, at the 1st opportunity for circle-area backward, the mean is 45% correct, with CI (21%, 41%). The intervals make no adjustment for multiple comparisons (at each practice opportunity), so they are overly narrow, but remain useful for comparing model predictions to student performance.

The model fit curves (black) show the 2.5th and 97.5th quantiles of the model predictions. A model should predict that some students have a lower probability of a correct answer than the population percent correct, and other students, respectively, have a higher probability. If a model fits the data well, the black model curves should be centered over the empirical red curves, but should have wider bars on early attempts where there are many students in the sample.

R-PFA consistently tracks the empirical learning curve more closely than the alternative models for all 6 skills, but most clearly in circle-area backward and square-area backward (Fig. 2). Consider AFM and R-PFA predictions on circle-area backward opportunity 1: AFM predicts that 60% of students will respond correctly, when only 45% do; in fact 95% of model predictions for AFM are above the empirical percent correct. AFM produces many false positives on this early opportunity. For R-PFA, the model predictions are

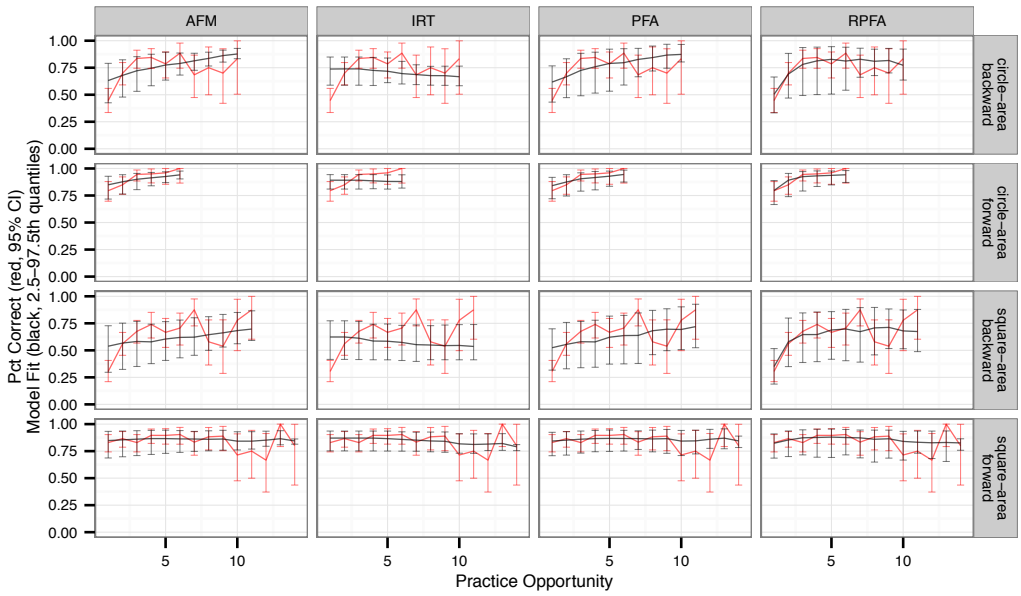


Figure 2: Empirical learning curves (red) and model fits (black) for newly split skills tagged in Cir-Sq.

centered over the empirical percent correct, and producing fewer false positives. On opportunity 4, AFM predictions are too low. AFM underestimates the amount of learning that has occurred, while R-PFA predictions track the empirical percent correct. Moreover, the R-PFA predictions range from below 0.5 to above 0.9, indicating that R-PFA is able to distinguish students who have learned the skill from those who need more practice.

3.2 Sensitivity to Skill Tagging

All models except IRT 1PL (which has the worst AIC) replicate the ranking of the three skill taggings [7]. The Cir-Sq tagging provides the best balance of predictive accuracy and data fit, compared to the Distinct tagging (which may be more granular than necessary to describe this dataset), and the Merged tagging (not sufficiently granular). While both the tagging and R-PFA are merely imperfect models, the replication provides convergent evidence for the validity of both. Skill model refinement need not improve predictive accuracy, but if it does and if the refinement makes sense in terms of instruction and cognition, that provides some evidence that the change represents an aspect of learning that is reflected in student performance.

R-PFA with the Merged tagging has a lower AIC score than any other model with the Cir-Sq tagging. Even though the Cir-Sq split is sensible and R-PFA benefits from it, R-PFA is more robust to the absence of such a split than other models. This shows in R-PFA's fit to the learning curve of circle-area (Fig. 3). AFM's predictions do not reflect the performance drop on opportunities 11 and later, but R-PFA does. This decrease motivated splitting circle-area into forward and backward skills, as in Cir-Sq [7], but R-PFA hews to the curve even without the split.

3.3 R-PFA Disaggregates Learning Curves

R-PFA effectively disaggregates the learning curves of individual students. Traditional learning curves are aligned at the first practice opportunity. Mastery-aligned curves [8] are aligned in terms of the opportunity at which students first achieve mastery. Traditional curves may conceal learning, such as if students differ in their relevant skill knowledge before their first observed practice opportunity, or if a skill model conflates two distinct skills [8]. The proportion of recent successes R by itself is a decay-weighted moving average that represents (in a non-parametric, non-model based way) the probability of mastery. R reflects the mastery-aligned curve in a predictive model, analogous to how total practice T represents the traditional learning curve in AFM.

The slope of R in R-PFA requires a different interpretation than the slope of T. A history of practice where recent success is positively associated with subsequent success (and recent failure is positively associated with subsequent failure) will have a positive slope, i.e., a positive effect on predicting the outcome. Practice relatively far in the past, whether successful or not, will have comparatively little effect on the prediction. (With the decay rate $d = 0.7$, practice older than about 5 opportunities has little effect on the prediction [6].)

One case in which the direction of the slopes of R and T may differ is in the case of a "blip" [4], i.e., when two skills follow each other in one curve, and the success rate drops in the middle of the curve, corresponding to the beginning of practice on a second skill (circle-area in Fig. 3). The slope of T ought to be flat in such a circumstance, which has been taken to mean that the skill may require a split. The slope of R will be positive, representing the fact that there is learning along the first disaggregated curve, and then along the second disaggregated curve. In fact, slopes of circle-area

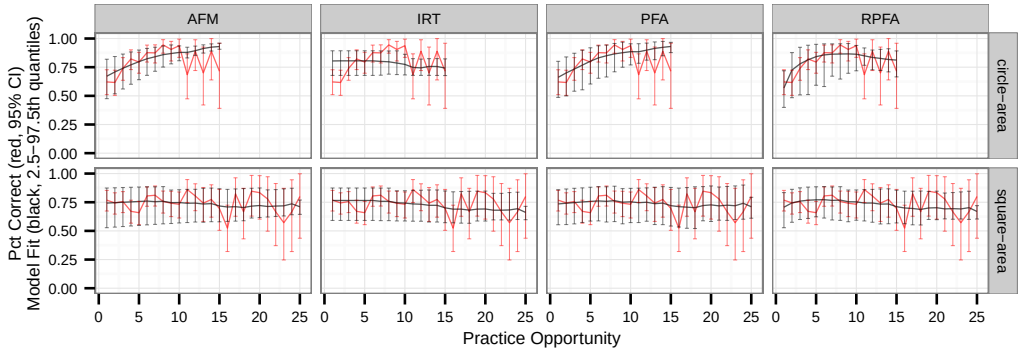


Figure 3: Learning curves (red) and model fits (black) for skills tagged in Merged, but later split in Cir-Sq.

are positive according to both AFM and R-PFA, and slopes of square-area are flat according to both AFM and R-PFA, suggesting that the slope of T is not an ideal heuristic for choosing a skill for a split.

An alternative heuristic is that when the slope of R is negative or flat, that implies that even disaggregated, mastery-aligned learning curves are a poor representation of the skill at hand. This suggests issues with the tagging of problems for this skill. This is a reasonable opportunity to invite experts to investigate "difficulty factors" for this skill, and to use LFA to apply these factors.

4. CONCLUSIONS

This investigation validates the R-PFA model of student performance in predictive accuracy on a real-world dataset. It provides convergent validity evidence for R-PFA by showing that it is sensitive to changes in a well-documented skill tagging, and yet robust to noise in a skill model. Given that no skill model is perfect, a predictive model that is accurate even in the face of such noise could be an asset to adaptive learning technologies.

The skill tagging refinement algorithm LFA [3], which incorporates AFM, may benefit by switching to R-PFA. LFA uses AFM in two ways: as a component in A* search, and as an interpretable learning curve slope. R-PFA may be a better component in A* search, because it is a more accurate model that is still sensitive to skill model changes, and because it reflects a mastery-aligned curve rather than an aggregate curve. The interpretation of the slope parameter is different, but sensible.

5. ACKNOWLEDGMENTS

We thank Ran Liu for thoughtful comments and assistance.

6. REFERENCES

[1] T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence Educational Data Mining Workshop*, 2005.

[2] M. Bernacki and S. Ritter. Motivation for learning HS Geometry 2012 (geo-pa). Dataset 748 in DataShop. Retrieved from `https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=748`, 2014.

[3] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of 8th International Conference on Intelligent Tutoring Systems*, pages 164–175, 2006.

[4] 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, 1995.

[5] A. Galyardt and I. Goldin. Recent-Performance Factors Analysis. In J. Stamper, Z. Pardos, M. Mavrikis, and B. McLaren, editors, *Proceedings of 7th International Conference on Educational Data Mining*, pages 411–412, 2014. (Poster paper).

[6] A. Galyardt and I. M. Goldin. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining*, Accepted.

[7] R. Liu, K. Koedinger, and E. McLaughlin. Interpreting model discovery and testing generalization to a new dataset. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.

[8] R. C. Murray, S. Ritter, T. Nixon, and al. Revealing the learning in learning curves. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Proceedings of 16th International Conference on Artificial Intelligence in Education*, pages 473–482, 2013.

[9] P. I. Pavlik, H. Cen, and K. Koedinger. Performance Factors Analysis–a new alternative to Knowledge Tracing. In *Proceedings of 14th International Conference on Artificial Intelligence in Education*, pages 531–538. IOS Press, 2009.

[10] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. T. Corbett. The Cognitive Tutor: Applied research in mathematics education. *Psychonomics Bulletin & Review*, 14(2):249–255, 2007.

[11] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.