Posters

Modeling Network Dynamics of MOOC Discussion Interactions at Scale

Jingjing Zhang
Beijing Normal University
Beijing, China
jingjing.zhang@bnu.edu.cn

Maxim Skryabin
Stepik
Saint Petersburg, Russia
ms@stepik.org

ABSTRACT
This paper attempts to model network dynamics of MOOC discussion interactions. It contributes to providing alternatives to conducting null hypothesis significance testing in educational studies. Using data collected from two successive psychology MOOCs in 2014 and 2015, the probabilistic longitudinal network analysis was performed by employing stochastic actor-based models with statistical accuracy. Understanding the mechanisms that drive the dynamics of discussions shed light on the design of a self-generated and learner-supported learning environment to meet the challenges of accommodating a massive and global student body.

1. Author Keywords

interactions; SIENA; probabilistic longitudinal network analysis; network dynamics; peer-supported learning.

2. ACM Classification Keywords

I.6.4 Simulation and Modelling: Model Validation and Analysis.

INTRODUCTION

Understanding learning at scale is a challenging task. As stated earlier, particular concerns are the extremely high rates of attrition and the pattern of steeply unequal participation in MOOCs. Using traditional educational methods fail to link the observed behavioral patterns within a network to the underlying the effects of network structure and the role of the participants that may explain why these patterns emerge. This study is an empirical investigation of the network dynamics of MOOC discussions, and attempts to make a contribution to providing alternatives to conducting null hypothesis significance testing in educational studies. Understanding the mechanisms that drive the dynamics of discussions shed light on the design of a self-generated and learner-supported learning environment to meet the challenges of accommodating a massive and global student body.

Using data collected from two successive psychology MOOCs in 2014 and 2015 and applying probabilistic longitudinal network analysis, this study seeks to rigorously measure the dynamic mechanisms that drive discussion change over time. The probabilistic analysis was performed by employing stochastic actor-based models with statistical accuracy.

METHODS

The probabilistic longitudinal network analysis was performed by employing stochastic actor-based models defined and evaluated with the program Simulation Investigation for Empirical Network Analysis. Four hypotheses are proposed to test the network dynamics of MOOC discussions.

Hypothesis 1 (H1): There is a tendency towards reciprocation in studied discussion networks (i→j and j→i). (Dyadic Level)

Hypothesis 2 (H2): There is a tendency towards transitivity (i.e. increasing transitivity and reducing distance between actors; i→j, j→k and i→k). (Triadic Level)

Hypothesis 3 (H3): There is a tendency towards the increasing volume of interactions between learners themselves.

Hypothesis 4 (H4): There is a tendency towards preferential attachment within the studied networks.

PRELIMINARY RESULTS

Descriptive statistics of the discussion network

In 2014 MOOC, 1915 participants posted 5251 messages in total, of which 217 are threads, 5034 are replies and comments, while in 2015 psychology MOOC, 962 threads were provided, and 3097 are replies and comments.

In 2014 Psychology MOOC, there are topics initiated by TAs to collect feedbacks for individual sections and to answer content-related Q&A for each section. As shown in Figure 1, the number of the postings falling into the discussing categories initiated by TAs is relatively larger than the number of the same topics which are initiated by learners themselves. The category "content-related Q&A initiated by TAs for individual sessions" seems to attract a good number of replies and comments over time. Interestingly, as shown in Figure 1, the discussions of exercises share a similar quantitative pattern of content-related discussions; while the enquiries about the logistics of the course follow a similar pattern of technical discussions in both two offerings of psychology MOOCs. In

2015 Psychology MOOC, technical problems occurred during the mid-examination, showing as a peak in Figure 1.

Network Dynamics

Table 2 and 3 present the results of SIENA estimation. As shown in Table 2 and 3, the results of Model 0 (network effects: reciprocity; transitivity) indicate a tendency for participants to create mutual relationships at both dyadic and triadic levels, which leads to cohesiveness in the studied networks. This confirms that hypothesis H1 and H2 are accepted. The exceptional case is the transitivity effect identified in the category of "feedback" (i.e. general feedbacks to instructors and TAs initiated by learners), where there is no tendency for participants to create mutual relationship at triadic levels. This deserves a detailed examination in the future analysis. Interestingly, under the topic categories of "feedback" and "TA about" (i.e. enquiries about the logistics of the course initiated by TAs), when same role is used as a control variable, the transitivity effect is significant with a negative coefficient. Compared to discussions in other categories, it is less likely to create cohesive subgroups when learners provide feedbacks to the course and enquiries about course logistics.

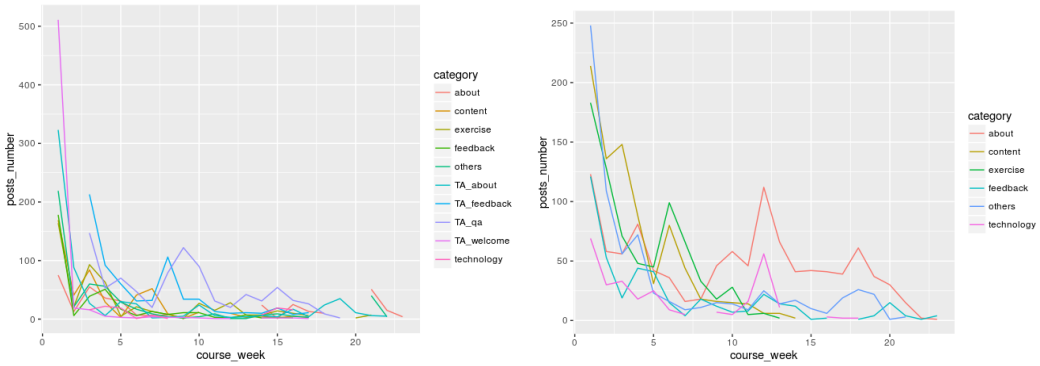


Figure 1. The number of postings within different discussion topics over time (2014 left & 2015 right).

In both courses, same role is a significant covariate effect with a negative coefficient. Thus, H3 (Model 1: reciprocity; transitivity; same role) is rejected, indicating that there is no tendency towards an increasing volume of interactions between learners.

H4 (Model 2: reciprocity; transitivity; Activity of alter) states that there is a tendency towards preferential attachment within the studied networks. The preferential attachment effect is not consistent among discussions of different topics. In most discussions, there is a tendency for participants who are actively involved in forum discussions in the early stages to become even more engaged over time. Nevertheless, when discussing exercises in 2014 Psychology MOOC, there is no preferential attachment effect, which deserves a future examination.

Category	Model 0	Model 1	Model 2
about	3.49* (0.37) 1.06* (0.29)	3.27* (0.40) 0.82* (0.32) -2.25* (0.24)	3.89* (0.35) 1.32* (0.31) -0.51 (0.34)
content	4.53* (0.32) 0.76* (0.23)	4.48* (0.32) 0.84* (0.23) -1.56* (0.28)	4.27* (0.29) 0.63* (0.23) 0.20* (0.09)
exercise	4.27* (0.35) 0.35 (0.34)	4.17* (0.34) 0.33 (0.36) -2.34* (0.35)	5.44* (0.46) 0.97 (0.43) -1.26* (0.60)
feedback	3.36* (0.50) -0.92* (0.41)	3.33* (0.50) -0.96* (0.40) -1.16* (0.44)	4.03* (0.78) -0.38 (0.74) -0.89 (0.93)
technology	3.26* (0.61) 0.03 (0.57)	3.02* (0.67) -0.15 (0.59) -2.18* (0.30)	3.63* (0.54) 0.36 (0.64)
			-0.53 (0.52)
TA about	5.13* (0.33) 0.16 (0.12)	3.67* (1.02) -0.41* (0.13) -5.91* (0.10)	4.71* (0.78) -0.81* (0.15) 0.20* (0.01)
TA feedback	3.30* (0.33) 0.87* (0.18)	0.64 (0.35) 0.08 (0.09) -4.98* (0.11)	0.67 (0.44) 0.28 (0.16) 0.12* (0.004)
TA Q&A	1.56* (0.44) 1.35* (0.17)	0.37 (0.48) 0.50 (0.08) -4.05* (0.10)	0.42 (0.46) 0.89* (0.17) 0.19* (0.01)

Table 1. Estimation results of network effects with standard errors in parentheses (2014 Psychology)

Category	Model 0	Model 1	Model 2
about	3.63* (0.21) 1.61* (0.17)	3.39* (0.23) 1.21* (0.17) -3.02* (0.19)	3.08* (0.22) 1.02* (0.20) 0.16* (0.01)
content	4.23* (0.23) 1.37* (0.20)	4.26* (0.22) 1.35* (0.20) -1.39* (0.49)	3.89* (0.26) 0.71* (0.21) 0.09* (0.01)
exercise	3.46* (0.27) 1.23* (0.23)	3.52* (0.26) 1.23* (0.24) -0.02 (1.18)	3.28* (0.25) 1.04* (0.23) 0.11* (0.04)
feedback	3.68* (0.37) 1.03* (0.33)	3.50* (0.38) 1.01* (0.34) -2.69* (0.28)	3.33* (0.35) 0.82* (0.37) 0.28* (0.10)

Table 2. Estimation results of network effects with standard errors in parentheses (2015 Psychology)

Studying MOOC Completion at Scale Using the MOOC Replication Framework

Juan Miguel L. Andres
Ryan S. Baker
University of Pennsylvania
Philadelphia, PA 19104
+1 (877) 736-6473
andresju@gse.upenn.edu,
rybaker@upenn.edu

George Siemens
Catherine A. Spann
University of Texas Arlington
Arlington, TX 76019
+1 (817) 272-2011
gsiemens@gmail.com,
caspann17@gmail.com

Dragan Gašević
University of Edinburgh
Edinburgh EH89YL, UK
+44 (131) 650-1000
dragan.gasevic@ed.ac.uk

Scott Crossley
Georgia State University
Atlanta, GA 30303
+1 (404) 413-5000
sacrossley@gmail.com

ABSTRACT

Research on learner behaviors and course completion within Massive Open Online Courses (MOOCs) has been mostly confined to single courses, making the findings difficult to generalize across different data sets and to assess which contexts and types of courses these findings apply to. This paper reports on the development of the MOOC Replication Framework (MORF), a framework that facilitates the replication of previously published findings across multiple data sets and the seamless integration of new findings as new research is conducted or new hypotheses are generated. MORF enables larger-scale analysis of MOOC research questions than previously feasible, and enables researchers around the world to conduct analyses on huge multi-MOOC data sets without having to negotiate access to data.

Keywords

MOOC, MORF, replication, meta-analysis.

1. INTRODUCTION

Massive Open Online Courses (MOOCs) have created new opportunities to study learning at scale, with millions of users registered, thousands of courses offered, and billions of student-platform interactions [1]. Both the popularity of MOOCs among students [2] and their benefits to those who complete them [3] suggest that MOOCs present a new, easily scalable, and easily accessible opportunity for learning. A major criticism of MOOC platforms, however, is their frequently high attrition rates [4], with only 10% or fewer learners completing many popular MOOC courses [1, 5]. As such, a majority of research on MOOCs in the past 3 years has been geared towards increasing student completion. Researchers have investigated features of individual courses, universities, platforms, and students [2] as possible explanations of why students complete or fail to complete.

A majority of this research, however, has been limited to single courses, often taught by the researchers themselves, which is due in most part to the lack of access to other data. In order to increase access to data and make analysis easier, researchers at UC Berkley developed an open-source repository and analytics tool for MOOC data [6]. Their tool allows for the implementation of several analytic models, facilitating the re-use and replication of an analysis in a new MOOC.

Running analyses on single data sets, however, still limits the generalizability of findings, and leads to inconsistency between published reports [7]. In the context of MOOCs, for example, one study investigated the possibility of predicting course completion based on forum posting behavior in a 3D graphics course [8]. They found that starting threads more frequently than average was predictive of completion. Another study investigating the relationship between forum posting behaviors, confusion, and completion in two courses on Algebra and Microeconomics found the opposite to be true; participants that started threads more frequently were *less* likely to complete [9].

The current limited scope of much of the current research within MOOCs has led to several contradictory findings of this nature, duplicating the "crisis of replication" seen in the social psychology community [10]. The ability to determine which findings generalize across MOOCs, and what contexts findings stabilize, will lead to knowledge that can more effectively drive the design of MOOCs and enhance practical outcomes for learners.

2. MORF: GOALS AND ARCHITECTURE

To address this limitation, we have developed MORF, the **MO**OC **R**eplication **F**ramework, a framework for investigating research questions in MOOCs within data from multiple MOOC data sets. Our goal is to determine which relationships (particularly, previously published findings) hold across different courses and iterations of those courses, and which findings are unique to specific kinds of courses and/or kinds of participants. In our first report of MORF [11], we discussed the MORF architecture and attempted to replicate 21 published findings in the context of a single MOOC.

MORF represents findings as production rules, a simple formalism previously used in work to develop human-understandable computational theory in psychology and education [14]. This approach allows findings to be represented in a fashion that human researchers and practitioners can easily understand, but which can be parametrically adapted to different contexts, where slightly different variations of the same findings may hold.

The production rule system was built using Jess, an expert system programming language [15]. All findings were programmed into if-else production rules following the format, "If a student who is <attribute> does <operator>, then <outcome>." Attributes are pieces of information about a student, such as whether a student reports a certain goal on a pre-course questionnaire. Operators are actions a student does within the MOOC. Outcomes are, in the case

of the current study, whether or not the student in question completed the MOOC (but could represent other outcomes, such as watching more than half of the videos). Not all production rules need to have both attributes and operators. For example, production rules that look at time spent in specific course pages may have only operators (e.g., spending more time in the forums than the average student) and outcomes (i.e., whether or not the participant completed the MOOC).

Each production rule returns two counts: 1) the confidence [16], or the number of participants who fit the rule, i.e., meets both the if and the then statements, and 2) the conviction [17], the production rule's counterfactual, i.e., the number of participants who match the rule's then statement but not the rule's if statement. For example, in the production rule, "If a student posts more frequently to the discussion forum than the average student, then they are more likely to complete the MOOC," the two counts returned are the number of participants that posted more than the average student and completed the MOOC, and the number of participants who posted less than the average, *but still* completed the MOOC. As a result, for each MOOC, a confidence and a conviction for each production rule can be generated.

A chi-square test of independence can then be calculated comparing each confidence to each conviction. The chi-square test can determine whether the two values are significantly different from each other, and in doing so, determine whether the production rule or its counterfactual significantly generalized to the data set. Odds ratio and risk ratio effect sizes per production rule are also calculated. Stouffer's [18] Z-score method can be used in order to combine the results per finding across multiple MOOC data sets, to obtain a single statistical significance.

Currently, 40 MOOC data sets and 21 production rules related to pre-course survey responses, time spent in course pages, forum posting behaviors, forum post linguistic features, and completion are incorporated in the framework.

3. FUTURE WORK

First, we plan to expand the current set of variables being modeled in MORF, both in terms of predictor (independent) variables and outcome (dependent) variables. This will enable us to replicate a broader range of published findings. Our first efforts do not yet include findings involving data from performance on assignments or behavior during video-watching, two essential activities in MOOCs.

Second, we intend to add to MORF a characterization of the features of the MOOCs themselves, towards studying whether some findings fail to replicate in specific MOOCs due to the differences in design, domain, or audience between MOOCs. Understanding how the features of a MOOC itself can explain differences in which results replicate may help us to explain some of the contradictory findings previously reported in single-MOOC research. Doing so will help us to understand which findings apply in which contexts, towards understanding how the different design of different MOOCs drive differences in the factors associated with student success.

4. REFERENCES

[1] Jordan, K. (2014). Initial trends in enrolment and completion of massive open online courses. *The International Review of Research in Open and Distributed Learning*, *15*(1).

[2] Adamopoulos, P. (2013). What makes a great MOOC? An interdisciplinary analysis of student retention in online courses.

[3] Zhenghao, C., Alcorn, B., Christensen, G., Eriksson, N., Koller, D., & Emanuel, E. (2015). Who's Benefiting from MOOCs, and Why. *Harvard Business Review*

[4] Clow, D. (2013). MOOCs and the funnel of participation. In Proceedings of the Third International Conference on Learning Analytics and Knowledge (pp. 185-189). ACM

[5] Yang, D., Sinha, T., Adamson, D., & Rose, C. P. (2013). Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In Proceedings of the 2013 NIPS Data-driven education Workshop (Vol. 11, p. 14)

[6] Pardos, Z. A., & Kao, K. (2015, March). moocRP: An open-source analytics platform. In *Proceedings of the Second (2015) ACM conference on learning@ scale* (pp. 103-110). ACM.

[7] Łukasz, K., Sharma, K., Shirvani Boroujeni, M., & Dillenbourg, P. (2016). On generalizability of MOOC models. In *Proceedings of the 9th International Conference on Educational Data Mining* (No. EPFL-CONF-223613, pp. 406-411).

[8] Andersson, U., Arvemo, T., & Gellerstedt, M. (2016). How well can completion of online courses be predicted using binary logistic regression?. In *IRIS39-The 39th Information Systems Research Conference in Scandinavia, Ljungskile, Sweden, 7-10 August 2016*.

[9] Yang, D., Wen, M., Howley, I., Kraut, R., & Rose, C. (2015). Exploring the effect of confusion in discussion forums of massive open online courses. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale (pp. 121-130). ACM.

[10] Makel, M. C., & Plucker, J. A. (2014). Facts are more important than novelty replication in the education sciences. *Educational Researcher*, 0013189X14545513.

[11] Andres, J.M.L., Baker, R.S., Siemens, G., Gašević, D., & Spann, C.A. (in press). Replicating 21 Findings on Student Success in Online Learning. *Technology, Instruction, Cognition, & Learning*.

[12] Schmidt, F. L., & Hunter, J. E. (2014). *Methods of meta-analysis: Correcting error and bias in research findings*. Sage publications.

[13] 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. *Handbook of educational data mining*, *43*.

[14] Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, *12*(4), 439-462.

[15] Friedman-Hill, E. (2002). Jess, the expert system shell for the java platform.*USA: Distributed Computing Systems*.

[16] Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Associations between Sets of Items in Massive Databases. In *Proceedings of the ACM-SIGMOD Int'l Conference on Management of Data* (pp. 207-216).

[17] Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997, June). Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record* (Vol. 26, No. 2, pp. 255-264). ACM.

[18] Stouffer, S.A., Suchman, E.A., DeVinney, L.C., Star, S.A. & Williams, R.M. Jr. (1949). *The American Soldier, Vol. 1: Adjustment during Army Life*. Princeton University Press, Princeton.

Clustering Students in ASSISTments: Exploring System- and School-Level Traits to Advance Personalization

Seth Adjei, Korinn Ostrow, Erik Erickson, Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
{saadjei, ksostrow, eerickson, nth}@wpi.edu

ABSTRACT
Few attempts have been made to create student models that cluster student and school level traits as a means to design personalized learning interventions. In the present work, data from ASSISTments was enriched with publicly available school level data and K-Means clustering was employed. Results revealed the importance of school locale, measures of district wealth, and system interaction patterns as potential foci for personalization. Clusters were then applied to a test set of held out data and cluster assignments were used to help predict end-of-year standardized mathematics test scores. Findings suggest that while cluster interpretations were not generalizable to held out data, clustering was generally helpful in predicting standardized test scores.

Keywords
K-Means Clustering, Student-System Interactions, School Level Characteristics, Standardized Tests, Ensembled Prediction Model.

1. INTRODUCTION
The focus of research using vast educational data often lends itself to the development of learner models, or various sophisticated predictive models that help to pinpoint when and how learning occurs on a personalized level. Popular approaches include Bayesian Networks (i.e., Bayesian Knowledge Tracing) [3], Performance Factors Analysis [6], and Neural Networks (i.e., Deep Learning) [4]. However, it is valuable to ask if simpler models built to leverage student, school, and district level data can be useful in establishing learner profiles.

The use of clustering to group similar students within various types of online learning environments has typically been a successful endeavor [1, 2, 7, 8]. The present work seeks to balance the complexity of working with high volumes of educational data and building simple predictive learner models through clustering by answering the following research questions:

1. Are there distinct types of learners within ASSISTments [5] that can be identified by clustering student, school, and district level characteristics and measures of student/system interaction?
2. What student types are defined via cluster interpretation? Do interpretations generalize to unseen data?
3. Can clusters help predict significant differences in end-of-year test scores?

2. METHODOLOGY
The present work assessed log files from students in the state of Maine working in ASSISTments [5], an online learning system focused on middle school mathematics, during the 2014-2015 academic year. This data was extended by merging additional school and district level data from the Common Core of Data supported by the NCES and IES (https://nces.ed.gov/ccd/). Students' scores on the standardized, end-of-year TerraNova mathematics test were also included in the dataset.

For each student, the dataset contained averages for the following student/system interaction features: problem count, time spent on problems, percent correct across assignments, hints used per problem, number of problems per assignment for which hints were used, and assignment completion rate. Additionally, each student's data included continuous measures retrieved from the NCES/IES data (i.e., the percentage of students in the school eligible for free or reduced lunch) as well as one-hot encoded forms of categorical features like school locale. The cleaned dataset represented 1,557 unique students from 21 schools, with 171,983 unique student/assignment pairs stemming from 35,127 assignments. Each observation or row represented the overall performance and characteristics of a single student and their school or district. De-identified data is available at tiny.cc/EDM2017Clustering for further reference.

The modeling approach used in the present work was adapted from that in [1]. An initial 70% of the data was randomly selected to form the training set. The training set was used for initial K-Means clustering and cluster interpretation. The K-Means algorithm was sourced from R's statistics package, implementing Euclidean distance as the default distance measure. The remaining 30% of the data was used to form the test set. The test set was used to build models predicting TerraNova scores. First, predictions were made to assign students in the test set to a cluster. Following student assignment, clusters were reinterpreted to verify whether trained interpretations generalized to unseen data. Cluster membership was then used to help predict TerraNova scores alongside student-system interaction features using cluster-specific stepwise linear regressions. These regression models were then ensembled and measures of model accuracy were compared to a traditional approach where K = 1.

3. TRAINING
In order to determine the optimal value for K, 10-fold cross validation was implemented on the training set to build scree plots. To determine the most appropriate value from this set, the mean and median of optimal K values across folds were considered (M = 4.1, Med. = 4). As such, four clusters were forced using K-Means on the training data. The four resulting clusters were characteristic of unique types of students, ultimately labeled as "proficient," "struggling," "learning," and "gaming." Graphics and additional information on cluster characteristics are available at tiny.cc/EDM2017Clustering for further reference.

Table 1. Coefficients, Standard Errors, and Model Statistics per cluster on test set data when K=1 and K=4.

	K = 1		*K = 4*							
	1 (n = 442)		*1 (n=127)*		*2 (n=160)*		*3 (n=124)*		*4 (n=31)*	
IVs	*b*	*SE*	*b*	*SE*	*b*	*SE*	*b*	*SE*	*b*	*SE*
Intercept	631.94***	20.37	712.95***	51.78	504.41***	30.36	567.63***	34.66	*680.14****	*63.13*
Percent Correct	110.66***	22.76	81.95	61.70	268.30***	33.92	131.02***	35.16	18.73	68.74
Ave. Time	-0.08**	0.03	-0.10	0.07	0.01	0.04	0.09	0.06	-0.09	0.09
Completed	0.35	12.13	-63.05	39.89	8.47	15.55	22.10	18.68	-18.80	34.25
Total Hints	1.73	2.69	7.01	6.08	8.00*	3.66	-38.84***	8.02	-52.73*	19.80
Hint Instances	-0.11	3.53	-9.34	11.25	-4.13	4.13	49.75***	9.68	71.09**	24.23
Model Stats										
F (DF)	17.55*** (5, 436)		1.30 (5, 121)		22.87*** (5, 154)		8.18*** (5, 118)		2.00 (5, 25)	
R^2 (Adj. R^2)	0.168 (0.158)		0.051 (0.012)		0.426 (0.408)		0.257 (0.226)		0.286 (0.143)	

4. TESTING & MODEL EVALUATION

Using the remaining 30% of the data that had been held out from the training set, student, school, and district level features (excluding TerraNova test score) were used to predict student assignment to one of the four clusters developed in training. Following student assignment, clusters were interpreted to verify whether initial cluster labels generalized to this unseen data. Cluster characteristics varied for the test set, suggesting that cluster interpretations did not generalize. Graphics and additional information on cluster characteristics are available at tiny.cc/EDM2017Clustering for further reference.

Cluster membership was then used to help predict TerraNova scores alongside student/system interaction features using cluster-specific stepwise linear regressions. Following the ensembling approach used in [7], separate regression models were built for each cluster before being ensembled to form a prediction model. Cluster models helped to depict the relative importance of student/system interaction features in the prediction of TerraNova scores for each value of K, as shown in Table 1. Variability in feature significance was observed across clusters. An alternative prediction model was constructed using the full dataset (essentially, K=1) in order to compare the accuracy of ensembled cluster models to an unclustered baseline. Table 1 presents unstandardized beta coefficients, standard errors, significance values, and overall model statistics across clusters and values of K, and reveals that cluster assignment was sometimes significant in predicting TerraNova scores.

In terms of prediction model accuracy, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were both lowest when K=4 (23.27 and 30.32, respectively, compared to 25.88 and 33.44 when K=1). Additionally, the difference between MAE and RMSE was lower when K=4 (7.05 compared to 7.56), suggesting that the variance in individual prediction errors decreases as K increases. Variance explained, as measured by R^2, was also higher when K=4, suggesting that the ensembled model was a stronger option than grouping all data together into a single cluster.

5. DISCUSSION

Results of our clustering exploration revealed that there are distinct types of learners within ASSISTments that can be identified by using K-Means to cluster student, school, and district level characteristics and measures of student/system interaction. Results suggested that clusters contained identifiably different patterns of student behavior. However, applying these clusters to a test set revealed that cluster interpretations did not generalize well to held out data. The results of subsequent linear regression models suggested that if clustering could be reliably linked to student features, the approach could potentially be used to help drive personalization within the ASSISTments platform.

Limitations of this work include being bound by the hierarchical nature of the data, assumptions inherent to K-Means analysis, and the potential for artificial inflation of model accuracy due to regression to the mean. As it stands, clustering does not necessarily fail as a method of personalization. Understanding the features that are important to each cluster, as well as the overall accuracy of ensembled cluster models and how such accuracy differs with varying values of K, could help to guide the design of learning interventions specific to particular students. However, the reliability of the approach may be extremely sensitive to the quantity and quality of available data, making clustering a difficult approach for personalized learning.

6. ACKNOWLEDGMENTS

Thanks to NSF (1440753, 1252297, 1109483, 1316736 & 1031398), U.S. D.O.E. (IES R305A120125 & R305C100024 and GAANN), ONR, and the Gates Foundation.

7. REFERENCES

[1] Amershi, S. & Conati, C. 2007. Unsupervised and supervised machine learning in user modeling for intelligent learning environments. Proc 12th Int Conf on Int UI. ACM, 72-81.

[2] Bouchet, F., Harley, J.M., Trevors, G.J., & Azevedo, R. 2013. Clustering and Profiling Students According to their Interactions with an Intelligent Tutoring System Fostering Self-Regulated Learning. *JEDM*. 5(1): 104-146.

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

[4] Deng, L. & Yu, D. 2014. Deep Learning: Methods and Applications. *Found and Trends in Sig Proc.* 7(3-4): 1–199.

[5] Heffernan, N. & Heffernan, C. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *Int. J AIED*. 24(4): 470-497.

[6] Pavlik, P.I., Cen, H., & Koedinger, K.R. 2009. Performance Factors Analysis: A New Alternative to Knowledge Tracing. AIED. 531-538.

[7] Trivedi S., Pardos Z.A., & Heffernan N.T. 2011. Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions. Proc 15th Int Conf on AIED. 377-384.

[8] Zakrzewska, D. 2008. Using Clustering Technique for Students' Grouping in Intelligent E-Learning Systems. In A. Holzinger (Ed.): USAB 2008, LNCS 5298, 403–410.

Application of the Dynamic Time Warping Distance for the Student Drop-out Prediction on Time Series Data

Alexander Askinadze
Institute of Computer Science
Heinrich Heine University Düsseldorf
askinadze@cs.uni-duesseldorf.de

Stefan Conrad
Institute of Computer Science
Heinrich Heine University Düsseldorf
conrad@cs.uni-duesseldorf.de

ABSTRACT

It is reported by different universities that over 40% of students do not complete their studies within 6 years. Especially in technical courses, the drop-out rate is already very high at the beginning. Therefore an automatic drop-out prediction is useful for a monitoring system. Since the study progress data can be sorted by time, we show how they can be transformed into a multivariate time series. Then we examine the dynamic time warping (DTW) distance in conjunction with the k-nn classifier and show how DTW can be used as an SVM kernel for drop-out prediction on the time-series data. With this approach, we are able to recognize about 67% of the drop outs from the course of study after the first semester and about 60% after the second semester.

1. INTRODUCTION

The number of drop out is a big problem for many universities. Over 40% of students do not complete their studies within 6 years [1]. Especially in technical courses, the number of drop outs in the first semesters is high. So in [5] it is reported that in the Electrical Engineering course the drop-out rate of beginners is about 40%. Human monitoring is used to solve this problem [5]. With a large number of students, this can lead to a huge manual effort, so that a machine-made pre-selection could facilitate the work of a human decision-maker. Most students fail in the first semesters, which requires an early prediction. The quality of the available data is very important for automatic drop-out prediction. However, due to data protection laws, often little data are available for use. The data is often restricted to only a small amount of private data and the study progress data, so that only examinations, their corresponding grades, and the number of attempts per semester are given. Because of the dearth of data, it is important to obtain as much semantics as possible from the data, such as temporal aspects. The study progress data can be viewed as a multivariate time series. In this paper, we will investigate methods that can perform drop-out predictions on time-series data.

2. RELATED WORK

Many studies have been published on student drop-out prediction like [1], [5], [6]. The data mining methods used include SVM, decision trees, k-nn, and neural networks. Studies were also made in the field of time series analysis. In [6] the authors investigated time series clustering to identify at-risk online students. Several studies, for example [7], have used DTW for time series clustering to identify distinct activity patterns among students. The results of the individual publications are difficult to compare with each other because the data used and the goals are very different. While some seek to prevent drop outs from a study subject, others seek to prevent drop outs from the whole study. We also use DTW, but not for clustering, but as distance for classifiers.

3. METHOD

If only the data of the study progress are available per semester, as much semantic information as possible must be collected from the data. Assuming that the study progress of a student S consists of n semesters $s = \{sem_1, ..., sem_n\}$, a function $\Phi : s \to T^S_{n,m}$ with $\Phi(s) = \Phi(\{sem_1, ..., sem_n\}) = \{\phi(sem_1)^\top, ..., \phi(sem_n)^\top\} = \{s_{1 \le k \le n} = [s_{1,k}, ..., s_{m,k}] \in \mathbb{R}^m\} = T^S_{n,m}$ which transforms the ordered set s into a multivariate time series is needed.

In each semester the students have the possibility to take q courses. The results of each course can be expressed by a number p of properties such as the final score or the number of trials. All information of a semester can thus be represented in a vector of size $m = q \times p$. If, for example, the 3 properties were: 1) achieved grade (numeric), 2) passed (binary), and 3) number of attempts (numeric), and in a certain semester, a student had taken the first and last course from the list of all possible courses then the resulting vector for a semester could look like the one shown below.

$$\phi(sem_i) = [\underbrace{5\ \ 1\ \ 1}_{course\ 1}\ \ \underbrace{0\ \ 0\ \ 0}_{...}\ \ \underbrace{1\ \ 0\ \ 2}_{course\ q}]$$

Thus, we can represent a student as a temporal sequence of his completed semesters. To compare these two sequences, we need a distance for multivariate time series. A well-researched distance for time series is the d_{DTW} distance.

Dynamic Time Warping (DTW) [3] is an algorithm from the domain of time series. It is generally defined for univariate time series and can be used to calculate a distance of the two time series $a = (a_1, ..., a_n), a_i \in \mathbb{R}$ and $b = (b_1, ..., b_m), b_j \in \mathbb{R}$ with different length. To extend the DTW distance for multivariate time series, various methods have been proposed in the literature like DTW_D [8]. DTW_D is calculated just as in the one-dimensional case, except that the pairwise distance $d(a_i, b_j)$ is calculated with the Euclidean distance.

The drop-out prediction is a binary problem. One of the most popular binary classifiers is the *support vector machine* (SVM) [4] because it can separate linear separable sets optimally from each other. If the training dataset is not linearly

separable, a kernel trick is used to solve the problem. An often used kernel is the Gaussian kernel. In [2], an adaptation of the Gaussian kernel to the Gaussian DTW (GDTW) kernel was made for sequential data. The GDTW kernel K_{GDTW} can be defined by $K_{GDTW}(x, y) = e^{-\gamma d_{DTW}(x,y)}$.

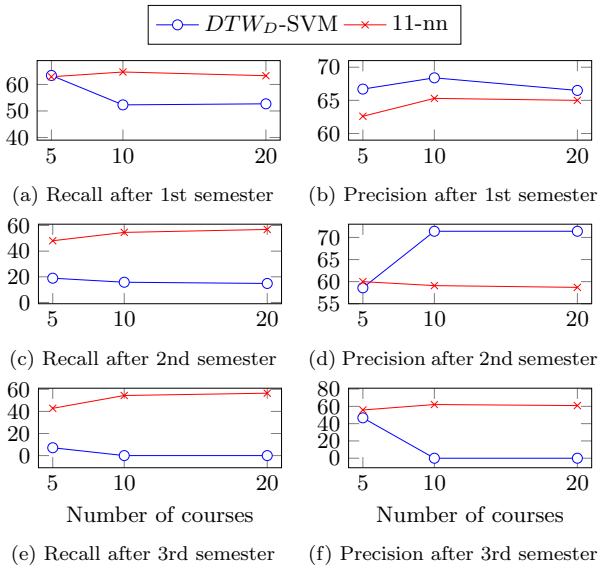
4. EVALUATION

We have a data set with 704 students of which 310 did not successfully complete their studies within 10 semesters. For each student the following information per semester is available: idCourse, number of attempts, examination status (passed, failed), recognized exam (true, false), reached grade, and semester. We use recall and precision as evaluation measures. The evaluation is performed 3 times for all parameters with a 10-fold cross-validation for the two approaches DTW_D-SVM and DTW_D-k-nn (ordinary k-nn classifier that uses the DTW_D distance). It is examined per semester how good the prediction is at the end of the semester. For example, the students who have studied at least 2 semesters are considered for the training and prediction of the drop out after the second semester. The length of the resulting multivariate time series vectors depends strongly on the number of courses used. Therefore, we will examine the influence of the number of courses used to create the vectors. In the dataset there are more than 100 courses. Because most students of our dataset drop out after a few examinations, we sort all courses according to the number of students who have enrolled in them. Then the 5, 10 and 20 courses with the highest enrollment will be used for further study. After the first investigation, we have found that the k-nn parameter $k = 11$ is comparatively well suited and is therefore used for the evaluation.

We first consider the prediction after the first semester. The recall and precision results are shown in Figure 1. In the second semester, 609 students are still active, of whom 215 will be leaving in the future. After the last examination of the second semester, almost 60% of these students can be recognized with 11-nn. The precision of 11-nn is also about 60%. DTW_D-SVM achieves a 10% higher precision when using more than 10 courses to create the multivariate time series vectors. However, the recall value of DTW_D-SVM is significantly smaller. At the end of the 3rd semester, the limits of DTW_D-SVM are recognizable. Both the recall and the precision values are smaller for 5 courses, and decrease to 0 for more used courses. 11-nn remains stable and provides similar results as after the second semester. In the third semester, 542 students are still active, of whom 143 will be leaving. 11-nn can recognize about 84 of these 143 students.

5. CONCLUSION

We have shown how a study progress can be transformed into a multivariate time series. Then we demonstrated that the DTW_D distance can be used within an SVM kernel to make an SVM usable for student time series data. We compared the DTW_D-SVM with the 11-nn classifier, which also uses the DTW_D distance on a dataset with 704 students and found that the k-nn classifier is better suited to achieve higher recall values in the drop-out prediction. The DTW_D-SVM is only suitable until the second semester and provides better precision results. In the later semesters, the values become worse due to most of the students in the first semester fail because of a few specific courses. For the students from the technical courses, it is usually the first mathematics courses. In the later semesters, the reasons cannot be stated so easily. Generally this approach is only for the prediction and not to determine the reasons. In future work we want additionally determine the reasons for drop outs.



(a) Recall after 1st semester (b) Precision after 1st semester
(c) Recall after 2nd semester (d) Precision after 2nd semester
(e) Recall after 3rd semester (f) Precision after 3rd semester

Figure 1: Recall and Precision results

6. REFERENCES

[1] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West. Predicting student dropout in higher education. *arXiv preprint arXiv:1606.06364*, 2016.

[2] C. Bahlmann, B. Haasdonk, and H. Burkhardt. Online handwriting recognition with support vector machines-a kernel approach. In *Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on*, pages 49–54. IEEE, 2002.

[3] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] G. Dekker, M. Pechenizkiy, and J. Vleeshouwers. Predicting students drop out: A case study. In *Educational Data Mining 2009*, 2009.

[6] J.-L. Hung, M. Wang, S. Wang, M. Abdelrasoul, W. He, et al. Identifying at-risk students for early interventions? a time-series clustering approach. *IEEE Transactions on Emerging Topics in Computing*, 2015.

[7] E. Młynarska, D. Greene, and P. Cunningham. Time series clustering of moodle activity data. In *24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16), University College Dublin, Dublin, Ireland, 20-21 September 2016*, 2016.

[8] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh. Generalizing dtw to the multi-dimensional case requires an adaptive approach. *Data Mining and Knowledge Discovery*, 31(1):1–31, 2017.

Student Use of Scaffolded Inquiry Simulations in Middle School Science

Elizabeth McBride
University of California
Berkeley
bethmcbride@berkeley.edu

Jonathan Vitale
University of California
Berkeley
jonvitale@berkeley.edu

Marcia Linn
University of California
Berkeley
mclinn@berkeley.edu

ABSTRACT
Interactive simulations can help students make sense of complex phenomena in which multiple variables are at play. To succeed, these simulations benefit from scaffolds that guide students to keep track of their investigations and reach meaningful insights. In this research, we designed an interactive simulation of a solar oven design and explored how students utilized the simulation during learning and how scaffolds functioned to alter the learning experience. We used a table for recording trials and guiding questions to scaffold students' interactions with the simulation. We employed data mining techniques to analyze student interactions for use of the control of variables strategy and other approaches. We found that the control of variables strategy may not be as beneficial for learning as an exploratory strategy.

Keywords
Interactive Simulations, Science Education, Inquiry, Log Data

1. INTRODUCTION
Simulations can be powerful tools for allowing students to engage in inquiry, especially in science disciplines. To succeed, these simulations generally benefit from scaffolds that guide students to keep track of their investigations and reach meaningful insights [6]. In this study, we examine guiding questions and recording of trials in a table as scaffolds. We use a simulation of a solar oven that allows students to investigate the multiple variables at play in energy transformation and gives representation to invisible phenomena.

We used the knowledge integration framework to create the curriculum about solar ovens, because the framework focuses on building coherent understanding [4]. This framework offers instructional design principles to enhance connections between design decisions and scientific principles. The knowledge integration framework has proven useful for design of instruction featuring dynamic visualizations [8] and engineering design [1, 6].

Various scaffolding methods are often used with interactive simulations. Often, these scaffolds are implicit, or built into the system with the simulation [7]. For example, guiding questions are used with inquiry simulations to direct students' attention toward certain features of simulations [2]. Other tools, like concept maps and note-taking spaces can also assist students in making sense of inquiry simulations [3].

Using log files from student interactions with the curriculum and output from the automatically generated tables (simulation scaffolding), we use feature engineering to identify how students use the model and whether these uses have an impact on learning.

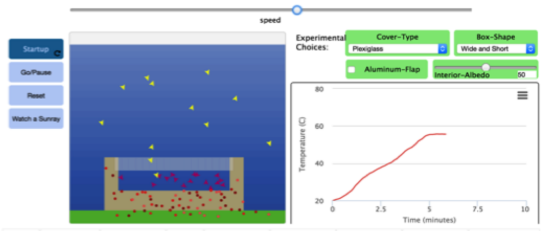
2. CURRICULUM
This research focuses on a curriculum about solar ovens that is run using the Web-based Inquiry Science Environment (WISE). During this curriculum, students design, build, and test a solar oven. Students use an interactive computer simulation to test the different materials in their oven during the design process.

This curriculum takes between 10-15 hours, and students complete the project in groups of 2 or 3. Students also complete individual pretests and posttests.

2.1 Interactive Computer Simulation
The scaffolds we developed for the interactive simulation are twofold; short response style questions direct students to investigate capabilities and limitations of the simulation and an automatically generated table helps students to keep track of trials they have run. The table includes information about all of the settings used in that trial, as well as the results of the trial at certain time points.

3. DATA
This data comes from 635 students across three schools and five teachers. These students formed 255 teams. After dropping students who did not complete significant portions of the curriculum, there were 558 students and 246 groups or partial groups remaining.

4. DESCRIPTIVE STATISTICS
Of the 246 groups who participated in the curriculum, 216 (87.80%) of the students used the computer model to pro-



#	Box Shape	Cover Type	Flap?	Int Albedo	Temp at 0m	Temp at 1m	Temp at 5m	Temp at 10m
1	Wide and S	None	No	50	20	24	30.2	31.5
2	Skinny and	None	No	50	20	26.2	29.7	31.1
3	Wide and S	Plexiglass	No	50	20	25.6	55.4	53.4

Figure 1: The interactive simulation used by students to test solar ovens and visualize energy transformation; below the table simulation is output from the automatically generated table

duce at least one row of data during the first design iteration. We consider each row of data produced to be a trial. As seen in figure 2, many groups do not use the simulation scaffolds at all and produce zero rows in the automatically generated table. Still more students produce only 1 row in the table, which may mean they are confirming their ideas for a solar oven that they have already discussed and planned prior to using the simulation and without any evidence outside of their intuitions.

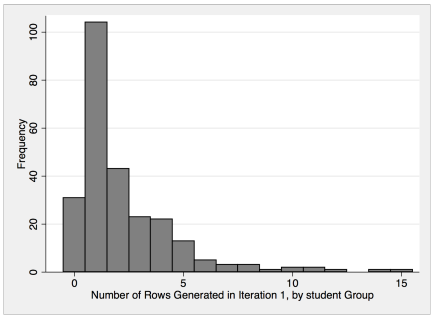


Figure 2: Histogram depicting the frequency of the number of trials run by a group of students during the first iteration of using the simulation (Mean: 2.27)

5. CONTROLLING VARIABLES

We define a control of variables strategy as changing a single variable at a time. We use feature engineering to develop a variable, *COV Trials*, that represents the number of trials a student ran using the control of variables strategy. Overall, 137 (55.69%) of the 246 groups employed a control of variables strategy. There were 216 groups that used the table scaffolds to generate at least one row of data. Of the groups that generated at least two rows in the table (115), 103 of them (89.56%) employed a control of variables strategy.

6. EFFECT ON LEARNING

Using pretest and posttest scores we aimed to understand the effect of actions with the simulation on learning. We found that the number of rows generated during the simulation was a significant predictor of learning (b = 0.10, t(546) = 2.68, $p < 0.01$). However, simply employing a control of variables strategy was not a significant predictor of learning. There were also two short response scaffolding questions. We generated a variable based on the number of questions students answered (0, 1, or 2). This was predictive of learning (b = 0.10, t(546) = 2.56, $p = 0.011$).

Overall, evidence suggests that students should be encouraged to experiment with the model and guided to produce at least two rows of data in the table to improve learning outcomes and use the short response questions. Perhaps changing more than one variable at a time in this type of environment indicates that students are spending more time thinking about possible outcomes.

7. LIMITATIONS

While we have found simulations to be beneficial for student learning in previous work [5], it is important to note that not all student learning is due to interactions with the simulation. While there is likely some difference between students who generated one row versus those who generated two or more rows, it is difficult to understand the differences between using a control of variables strategy and generating multiple rows of data in the table.

8. REFERENCES

[1] J. Chiu, P. Malcolm, D. Hecht, C. DeJaegher, E. Pan, M. Bradley, and M. Burghardt. Wisengineering: Supporting precollege engineering design and mathematical understanding. *Computers & Education*, 67:142–155, 2013.

[2] C. Hmelo and R. Day. Contextualized questioning to scaffold learning from simulations. *Computers & Education*, 32(2):151–164, 1999.

[3] Y. Kali and M. Linn. Technology-enhanced support strategies for inquiry learning. *Handbook of research on educational communications and technology*, pages 145–161, 2008.

[4] M. Linn and B. Eylon. *Science learning and instruction: Taking advantage of technology to promote knowledge integration.* Routledge, 2011.

[5] E. McBride, J. Vitale, L. Applebaum, and M. Linn. Use of interactive computer models to promote integration of science concepts through the engineering design process. In *Proceedings of the 12th International Conference of the Learning Sciences*, Singapore, Singapore, June 2016 2016.

[6] K. McElhaney and M. Linn. Investigations of a complex, realistic task: Intentional, unsystematic, and exhaustive experimenters. *Journal of Research in Science Teaching*, 48(7):745–770, 2011.

[7] N. Podolefsky, E. Moore, and K. Perkins. Implicit scaffolding in interactive simulations: Design strategies to support multiple educational goals. *Chemistry Education Research and Practice*, 14(3):257–268, 2013.

[8] K. Ryoo and M. Linn. Can dynamic visualizations improve middle school students' understanding of energy in photosynthesis? *Journal of Research in Science Teaching*, 49(2):218–243, 2012.

Modeling Dormitory Occupancy Using Markov Chains

David D. Pokrajac
Delaware State University
1200 N DuPont Hwy
Dover, DE 19901
+1-302-857-7614
dpokrajac@desu.edu

Kimberley Sudler
Delaware State University
1200 N DuPont Hwy
Dover, DE 19901
+1-302-857-7036
krsudler@desu.edu

Diana Yankovich
Delaware State University
1200 N DuPont Hwy
Dover, DE 19901
+1-302-857-6308
dyankovich@desu.edu

Teresa Hardee
Delaware State University
1200 N DuPont Hwy
Dover, DE 19901
+1-302-857-7837
thardee@desu.edu

ABSTRACT
We introduce a Markov chain based model that quantifies university dormitory occupancy as a function of parameters related to university housing policies, students' success and academic progress, and customer satisfaction/dorm availability. The model provides sensitivity of university housing occupancy on change of the parameters. We demonstrated functionality of the model on several case scenarios from a public university.

Keywords
Modeling, dormitory occupancy, university housing, Markov chains, sensitivity, students' success, Banner.

1. INTRODUCTION
In this study, we introduce a housing occupancy model based on Markov chains [e.g., 1]. The model determines relationship between the number of students in dormitories, number of students in incoming class and probabilities quantifying students' retention, advancement between ranks (freshmen, sophomores, etc.), customer satisfaction and availability of housing. The model provides an opportunity for what-if analysis and assessment of change in housing occupancy due to variation of model parameters. The values of model parameters are learned from a transactional database.

We provide a case study based on three years data from Delaware State University, a public comprehensive historically black college/university in Delaware and demonstrate quantitative change of housing occupancy as results of possible changes in housing policy, housing demand and retention. The proposed technique is applicable to universities offering predominantly undergraduate programs and can be easily adapted for universities with substantial graduate programs and participation of international students.

2. METHODOLOGY
2.1 Problem
We consider a university offering undergraduate programs. The students at the university may be of in-state or out-of-state domicile (in-state students are the students whose residence is in the same state as the university). During the course of study, out-of-state students may convert to in-state or vice versa. A new student at the university can be enrolled as a new freshman (NF) or a new transfer (NT). For a student retained at the university, a rank depends on the cumulative number of credits (earned at the university + transferred). The ranks satisfy partial order. Thus, a NF or NT, if retained, may continue as returning freshmen (RF), sophomore (SO), junior (JR) or senior (SR). Retained RF may continue as RF or progress into SO, JR or SR. Retained SO may continue as SO, or progress as JR or SR. Each student in a particular year can be a dorm resident. If retained, a student may change dorm residency status, i.e., a dorm non-resident may become dorm resident or vice versa.

Our goal is to determine the relationship between various parameters characterizing students' population and academic progress and the total number of dorm residents in a particular year.

2.2 Markov Chain Model
We model the considered problem with a time-homogeneous Markov chain [1]. A student at the university can be described by a state $s_{(i,j,k)}$ determined by an ordered triple of indices i, j, and k indicating domicile, rank and dorm residence: $i \in \{InState, OutOfState\}$, $j \in \{NF, NT, RF, SO, JR, SR\}$ and $k \in \{DormResident, NotDormResident\}$. The starting states correspond to $i \in \{InState, OutOfState\}$, $j \in \{NF, NT\}$, $k \in \{DormResident, NotDormResident\}$. The total number of non-absorbing states is 24. In addition, a student can graduate or leave the university, corresponding to an absorbing state, denoted with s_a. The transition between states $s_{(i,j,k)}$ and $s_{(i',j',k')}$ is uniquely determined by transition probability that, under the assumption of time homogeneity is denoted by $p_{(i,j,k),(i',j',k')}$. In addition, the model includes transition probabilities $p_{(i,j,k),a}$ from states $s_{(i,j,k)}$ to the absorbing state.

2.3 Model Implementation

To operationalize the model, we introduce the following assumptions and simplifications:

1) Students can transition only from out-of-state to in-state status;

2) For in-state students who continue to stay in dorms, the transition probability can be expressed as product of probabilities that a student is retained, that a student advanced from rank *j* to *j'* and the probability that a student stayed in dorm;

3) For out-of-state students who continue to stay in dorms as out-of-state, the transition probability is expressed as a product of probabilities that a student is retained, that a student does not change out-of-state status, that a student advanced from rank *j* to *j'* and the probability that a student stayed in dorm;

4) For out-of-state students who continue to stay in dorms as in-state, the transition probability is expressed as a product of probabilities that a student is retained, that a student changes out-of-state status to instate, that a student advanced from rank *j* to *j'* and the probability that a student stayed in dorm;

5) We compute probabilities that a dorm resident with domicile *i'* and rank *j'* was a dorm resident in the previous year.

2.4 Model Sensitivity

After the parameter values are estimated, the sensitivity s_l of the number of students in dorms on a particular parameter π_l can be determined as: $s(\pi_l) = \frac{\Delta N^y}{\Delta \pi_l}$, where ΔN^y is change of number of students in dorms, due to change $\Delta\pi_l = \pi_l^{new} - \pi_l$ of a parameter. Subsequently, the influence of change of particular model parameters on the model output—the number of students in dorms can be linearized such that: $\Delta N^y = \sum_l s(\pi_l)\Delta\pi_l$.

3. RESULTS

3.1 Data Set

We estimated the model discussed in Section 2 on data from Delaware State University (DSU), a historically black college/university (HBCU) located in Dover, DE, USA. DSU utilizes Banner® Version 8 (Ellucian, Fairwax, VA, USA) as a higher education enterprise resource planning (ERP) system. The dataset contained the total of 13,709 records from years 2013/14—2015/16. Each record had the values of attributes: StudentID, Year, Rank, DormResidence, Domicile. StudentID is a unique identifier of a student and together with Year comprise the primary key of the extracted table.

3.2 What-if Analyses

In this section we analyze realistic cases for changes of some of the model parameters and their influence on the change of number of students in dormitories.

Case 1. Due to policy change, *all* new freshmen and new transfers are expected to stay at university housing *regardless* whether they are in-state or out-of-state. We can easily obtain the increase of the number of students in dormitories of ΔN^y=467.

Case 2. Due to implementation of initiatives to address needs of incoming and returning freshmen, the retentions of in-dorm new and returning freshmen increase to 80%. This leads to the increase of ΔN^y=175 students in dorms.

Case 3. Owing to improvement of dorm facilities, the demand for dorm housing for upper rank students increases. This can, thus, be considered as a result of increased customer satisfaction. As a consequence, this leads to the increase of ΔN^y=83.

4. DISCUSSION

The proposed model makes it possible to account for retention that is frequently a key performance indicator related to university strategic plans and one of common quantitative measures of students' success. Further, the model involves parameters related to academic progress of students. Also, we can indirectly model housing satisfaction and availability. The model makes it possible to consider in-state and out-of-state students separately, as the two groups of students that may have different demography, socio-economical conditions and academic success. Also, it is possible to evaluate the relationship between the size of the incoming class (new freshmen and transfers) and the housing occupancy.

The model considers only two categories of students: in-state and out-of-state students. For universities with substantial numbers of international students, they can be added as an additional category and treated similarly as out-of-state students. The model assumes that in-state students cannot become out-of-state. However, the assumption can be relaxed by introducing a non-zero probability that in-state students of rank *j* become out-of-state.The assumptions 2—4 (probability independencies) may be contingent on university policies (distribution of students within dorms and on-campus housing allocation across student classes/ranks). Hence, they should be validated prior to the application of the proposed models at another institution of higher education. The current model assumes that the students who leave the university without graduating do not come on a later date. In reality, some students may leave the university temporarily and return ("stop-outs"). Note that we utilized point estimates, hence the accuracy of parameter estimates (e.g., standard deviation) has not been addressed. Future work will include the development of interval estimates for model parameters as well as an application of validation techniques (e.g., leave-one-out cross-validation) to more strictly justify predictive ability of the model.

5. CONCLUSION

We proposed a Markov chain-based model of university housing occupancy and demonstrated it in a case study of a public university. We have shown that the proposed model can be useful in quantifying what-if scenarios related to changes in housing policy, retention and customer satisfaction. The model is developed for a university offering primarily undergraduate programs. It can be extended to graduate program offering institutions, with a challenge that graduate (especially PhD) programs are typically less structured (as evidenced in lack of ranks corresponding to sophomores, juniors, seniors in undergraduate programs). We demonstrated the use of a model with parameters estimated from data readily available on an industry-standard ERP system (Banner). As such, the model can be easily deployed at an institution of higher education that utilizes this or similar technology.

6. ACKNOWLEDGMENTS

This work has been supported through a grant from the Bill and Melinda Gates foundation.

7. REFERENCE

[1] Grinstead, C.M. 1997. *Introduction to Probability*, 2nd edn. American Mathematical Society, Providence, RI.

Improving Models of Peer Grading in SPOC *

Yong Han, Wenjun Wu, Xuan Zhou
State Key Laboratory of Software Development Environment,
School of Computer Science, Beihang University, China
{hanyong, wwj, zhouxuan}@nlsde.buaa.edu.cn

ABSTRACT

Peer-grading is commonly used to allow students to work as graders to evaluate their peer's open-ended assignments in MOOC courses. As a variant of MOOCs, SPOC (Small Private online course) adopt the peer-grading method to grade a number of student submissions. We propose a new ability-aware peer-grading model for SPOC courses by introducing prior knowledge level of each student grader as their grading ability in the process of calculating grading score.

1. INTRODUCTION

Small Private online course (SPOC) is a version of MOOCs used locally with on-campus students. It often has the relatively smaller number of students than a MOOCs course. SPOC students may come from the same classroom and know each other. Previous research efforts on peer-grading suggest that there is great disparity between the observed scores presented by student graders and the true scores (the instructor-given scores). Therefore, it is a major challenge on how to correctly aggregate peer assessment results to generate a fair score for every homework submission.

To solve the problem, we propose a group of new peer-grading models by considering the student mastery of knowledge level as a major factor for estimating final scores. Throughout the paper, we call the mastery of knowledge level as the students' grading ability. Based on every student's learning behavior and quiz-answering outcomes, we design a two-stage individualized knowledge tracing model to accurately assess their grading ability. Moreover, we introduce the new peer-grading models by integrating every student's grading ability into the factor of reliability. Experimental results in our SPOC course verify the effectiveness of our new models.

2. RELATED WORK

Many research efforts have been made to investigate the factors that can affect the grader bias and reliability.

*The accompanying appendix at: http://admire.nlsde.buaa.edu.cn/paper/2017-3.pdf

Goldin et al. [1] used the Bayesian models for peer grading in the setting of traditional classrooms. They explored the major factors including grader bias, and the rubric biases in their models. Walsh introduced a new algorithm named by PeerRank[4] based on the assumptions that the ability of student graders can be measured by the grades they received in the process of peer grading. Our models are inspired from the previous research work done in [3, 2]. We introduce the grading ability of students in their models and develop an individualized knowledge tracing model to estimate such ability.

3. DATASETS

The data sets in our experiments were collected in a SPOC course named by "The Experiment of Computer Network" that is hosted on our MOOC platform. The course is designed to teach both 4th grade CS undergraduate and the first-year graduate students about basic knowledge and skills on designing networking plans and configuring networking devices at the multiple levels of link protocol, TCP/IP protocol and network applications.

The course comprises of 10 chapters, each of which has 8-14 problems as homework assignment for students. The course also includes two open-ended assignments in graduate courses and three open-ended assignments in undergraduate courses. Preliminary statistical analysis of the dataset reveals that most peer-graded score tend to be higher than instructor-given scores for the same submissions.

4. PROBABILISTIC MODELS OF PEER GRADING IN SPOC

In this paper, we first establish a two-stage model to assess student mastery level of each knowledge skill, which can be used for estimating the graders' reliability. And then, we present three probabilistic graph models for peer grading by extending the models PG4 and PG5 of [3].

4.1 Individualized Knowledge-Tracing model for Ability Estimation

At the first stage, we extract interpretive quantities to predict the probability that a student has mastered the knowledge of that certain chapter in which the logistic regression method is used to fit these features and predict the engagement level of every student[5]. At the second stage, our work adopts the knowledge tracing model and ameliorates it by combining the prediction results obtained in the first stage. The sequence of the exercises in each unit is modeled by H-

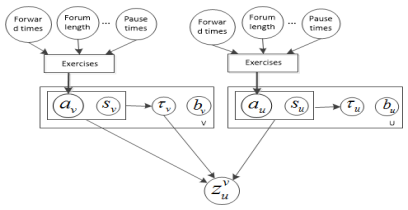


Figure 1: The relationship of the factors used in our models.

MM named as PPS (the Prior Per Student Model). We refer to the results that the HMM generated as a_v, which denotes the ability of graders prior to the peer-grading tasks. We train the model of HMM by using a_v as the initial element of the sequence and then introduce it and the true score as the parameters to model the reliability of a grader by a distribution of Gamma or Gaussian.
Our Experiments show that our estimated ability has relevance with the true score and can be used to estimate the grader reliability. Thus it is reasonable to use grader ability to estimate the reliability.

4.2 Peer-Grading models

We represent a_v as the prior distribution of estimating every grader's mastery of preparatory knowledge, τ_v as the reliability of the student grader v, b_v as the bias of the student grader v, s_u as the true score of a submission, and z_u^v as observed score for the submission.
Model PG6

$$\tau_v \sim \mathcal{G}(a_v, \beta_v)$$
$$b_v \sim \mathcal{N}(0, 1/\eta)$$
$$s_u \sim \mathcal{N}(\mu_0, 1/\gamma_0)$$
$$z_u^v \sim \mathcal{N}(s_u + b_v, 1/\tau_v)$$

We refer to our first model as PG6: the reliability variable τ_v follows the Gamma distribution with a_v as the shape parameter instead of the true score in PG4 in [2] and utilize the student's performance on multiple-choice exercises to estimate his reliability in the process of peer-grading tasks.
Based on Model PG6, we introduce the Model PG7 by remodeling the reliability variable τ_v ($\tau_v \sim \mathcal{N}(a_v, \beta_v)$) with the Gaussian distribution instead of the Gamma distribution. The mean value of the Gaussian distribution in PG7 is still a_v. We also make further extension on Model PG7 by adding the true score s_v with the a_v to calculate the mean of the reliability variable τ_v ($\tau_v \sim \mathcal{N}(\theta_1 a_v + \theta_2 s_v, 1/\beta_v)$) and introduce the parameter λ to re-model the observed variable z_u^v ($z_u^v \sim \mathcal{N}(s_u + b_v, \lambda/\tau_v)$). This extended model is named as Model PG8.
In the above three models (PG6-PG8), we assume the overall bias random variable b_v follows the Gaussian distribution with the mean value at zero. The true score s_u follows the Gaussian distribution with the mean value at μ_0. Moreover, the hyper-parameters β_0, η_0, μ_0, γ_0, θ_1, θ_2, λ are the priors. For the observed scores z_u^v in the PG8, the parameter λ is similar to β_0 in PG6 and PG7, whose function is to scale the variance of its Gaussian.

4.3 Inference and evaluation

The details of the model inference procedures for PG6, PG7 and PG8 are described in the appendix. Our experiments are all based on Gibbs sampling. At the beginning of the Gibbs sampling process, the values of these parameters β_0, η_0, μ_0, γ_0 and λ are initialized to empirical values. We run our experiments by running for 400 iterations with the first 50 burn-in samples eliminated.

5. EXPERIMENTAL RESULTS

We compare our models PG6-PG8 with the baseline model based on simple median value, the models of PG1-PG3 proposed in [3], and the models of PG4-PG5 defined in [2]. The evaluation metric is the root-mean-score-error (RMSE), which is computed as the deviation between the estimated score and the true score assigned by the course staff.
Compared to PG1-3 and PG4-5, our models PG6 and PG7 demonstrate the same level of RMSE in most cases. The model PG8 has more obvious improvement than PG6-7, achieving the lowest RMSE. Therefore, it confirms that PG8 demonstrates the best performance among all the models on average. By combining the grader ability and the true score, the model PG8 is the best approach among all the models for estimating the peer-grading scores in SPOC courses.

6. CONCLUSIONS

In this paper, we first introduce a two-stage individualized knowledge tracing model to estimate each grader's level of knowledge mastery as their grading ability. And then, we propose three new probability graph models by introducing the grading ability as the major parameter for the latent variable of grader reliability. The experiments based on the dataset of our SPOC course demonstrate that our models can be effectively applied to aggregate the peer grades in SPOC courses.

7. ACKNOWLEDGMENTS

This work was supported by grant from State Key Laboratory of Software Development Environment of Beihang university of China (Funding No. SKLSDE-2015ZX-03) and NSFC (Grant No. 61532004).

8. REFERENCES

[1] Ilya M Goldin. Accounting for peer reviewer bias with bayesian models. In *Proceedings of the Workshop on Intelligent Support for Learning Groups at the 11th International Conference on Intelligent Tutoring Systems*, 2012.
[2] Fei Mi and Dit-Yan Yeung. Probabilistic graphical models for boosting cardinal and ordinal peer grading in moocs. In *AAAI*, pages 454–460, 2015.
[3] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. *arXiv preprint arXiv:1307.2579*, 2013.
[4] Toby Walsh. The peerrank method for peer assessment. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, pages 909–914. IOS Press, 2014.
[5] Hsiang-Fu Yu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, et al. Feature engineering and classifier ensemble for kdd cup 2010. In *KDD Cup*, 2010.

Personalized Feedback for Open-Response Mathematical Questions using Long Short-Term Memory Networks

Joshua J. Michalenko
Rice University
jjm7@rice.edu

Andrew S. Lan
Princeton University
andrew.lan@princeton.edu

Richard G. Baraniuk
Rice University
richb@rice.edu

ABSTRACT

In this paper, we explore the problem of automatic grading and feedback generation for open-response mathematical questions. We resort to the long short-term memory (LSTM) network to learn the simple task of polynomial factorization and use the trained network for grading and feedback. We use Wolfram Alpha to synthetically generate a training dataset that consists of step-by-step responses to polynomial factorization questions to train the LSTM network. Preliminary results validate the efficacy of LSTMs in learning to factor low-order polynomials; we also demonstrate how to leverage the trained network for automatic grading and personalized feedback generation.

Keywords

Automatic grading, Feedback generation, Long short-term memory networks, Mathematical expressions

1. INTRODUCTION

In spite of tremendous advances in technology for education, learning today largely remains a "one-size-fits-all" approach. Personalized learning is the manifestation of *differentiation*, the idea that all students access content and develop mastery differently. The personalized learning experience necessitates a scalable approach since the number of students is much larger than the number of teachers. Many recent advances focus on using machine learning algorithms to analyze student data, but mostly resort to limited utility multiple-choice questions for grading a feedback [5].

The mathematical language processing (MLP) framework proposed in [4] is the first automatic grading and feedback generation tool for open-response mathematical questions. MLP is capable of automatically grading a large number of student responses requiring minimal human effort, but lacks an effective feedback mechanism because it not capable of truly understanding mathematics, and is therefore unable to provide informative feedback. A series of recent tools based on recurrent neural networks (RNNs) [3] have found great success in various NLP tasks (e.g., machine translation, image captioning, etc.) and predicting the output of simple computer code [7]. Natural language processing for the purposes of grading and feedback has also made substantial progress in several restricted domains including essay evaluation and mathematical proof verification [2, 6]. These successes inspires us to use RNNs to analyze responses to mathematical questions due to their sequential, step-by-step format and their algorithmic nature. They support our belief that LSTMs have the ability to learn simple mathematical operations such as factoring polynomials from data and providing relevant feedback.

1.1 Contributions

In this paper, we apply the LSTM network [3], a type of RNN, to try to understand simple mathematics for automatic grading and feedback generation for open-response mathematical questions. In particular, we study the simple problem of *polynomial factorization* due to the fact that responses to polynomial factorization questions are typically short and require only simple mathematical operations. We first generate a synthetic dataset using the Wolfram Alpha API consisting of responses (step-by-step solutions with mathematical expressions and text explaining the mathematical operations performed) to polynomial factorization questions. We then train multiple LSTM networks on the dataset and evaluate their performance on factoring previously unseen polynomials. Preliminary results show that the trained character level networks can factor previously unseen polynomials up to the second order with sufficient accuracy, after training on enough examples. More importantly, we showcase how the trained networks have the potential for automatic grading and feedback generation for open-response mathematical questions.

We emphasize that our proposed method has the capability to go beyond Wolfram Alpha. First, the ability of the trained LSTM networks to generalize to previously unseen examples enables *transfer* between domains, i.e., these networks have the capability of learning a rule in a certain context and apply it in another context. This property enables a LSTM network to build on its own knowledge as more and more training data becomes available, which is a much more scalable approach than the rules-based Wolfram Alpha system, which requires new rules to be manually coded for every new domain.

2. EXPERIMENTS

Experimental setup. We generate factorable polynomials that are subsequently used by the Wolfram Alpha API to produce responses on how to fully factor these polynomials. The responses include step-by-step solutions that consist of a series of mathematical expressions that end up in a fully-factored final form, together with concise text describing the mathematical operations involved. The data generation process is limited to polynomials with a single variable, co-

	Character Level % Error			Expression Level % Error		
# units	1 Layer	2 Layer	3 Layer	1 Layer	2 Layer	3 Layer
50	31.11	20.98	20.40	87.93	80.76	78.28
200	11.79	10.68	10.12	68.55	59.39	56.80
512	12.94	**8.21**	10.32	42.38	39.94	**38.95**

Table 1: Character and expression level misclassification errors on the test set. Performance of the best models are highlighted in bold.

efficients that are less than 10 and up to the third order. We construct a training dataset including 200,000 responses to various factoring questions this way. A test dataset is constructed with 20 first, 20 second, and 20 third order polynomials to be factored. We emphasize that, while for the simple task of polynomial factorization, Wolfram Alpha is able to generate the correct response, our aim is to develop a method that can generalize to more complicated mathematical operations that are too complicated for a rules-based system like Wolfram Alpha to cover. We train our LSTM networks to operate on a character-by-character level, i.e., use each character in a response as input and output data at each time instant. We train 9 different LSTM networks with varying number of hidden units ($N \in \{50, 200, 512\}$) and layers (1, 2, and 3). We use 95% of the generated training dataset for training and 5% as the validation dataset; We train the LSTM networks for a total of 50-150 epochs or terminate the training process early if the validation error shows minimal change across 10 epochs. In order to achieve faster training, we apply the curriculum learning approach [1], i.e., we start by training the LSTM networks on factorizations of first order polynomials until the validation error cannot be further reduced, and then proceed to train on responses factoring second order polynomials and beyond.

Results and discussion. We evaluate the performance of our trained LSTM networks on factoring previously unseen polynomials using two metrics. The first metric computes the character-level misclassification error rate by comparing every character in the correct factorization to the maximum-likelihood predicted character by the trained LSTM network. The second metric computes the expression-level misclassification error rate by comparing every full mathematical expression in the correct factorization to the full predicted expression by the trained LSTM network; a successful classification means that the entire expression is correctly predicted.

Experimental results for all 9 LSTM networks on both metrics are shown in Table 1. In general, LSTM networks with more hidden units and layers achieves lower misclassification error rates. We note that the expression-level misclassification rate is much higher (the best model achieves an error rate of 38.95%) than the character-level misclassification rate (the best model achieves an error rate of 8.21%). This observation is not surprising since correctly predicting the entire expression is much more difficult than successfully predicting a character. Moreover, we observe that the best model achieves error rates of 0% and 15%, respectively, on factoring first and second order polynomials but a 100% error rate on third order polynomials. This result is due to the fact that factoring third order polynomials is hard since it requires first factoring out a second order polynomial as an intermediate step.

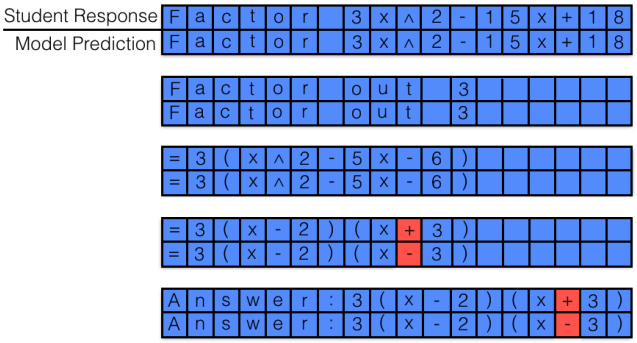


Figure 1: Illustration of how to use of a trained LSTM network to detect when a student's response deviates from the correct response.

Using trained LSTM networks for grading and feedback. We now illustrate how the trained LSTM networks can be used for automatic grading and feedback generation. Figure 1 shows a typical use case with an actual student response and a direct comparison to the maximum-likelihood character the trained LSTM network predicts given the previous characters as input. For automatic grading, we can calculate the predictive likelihood of every character in a student's response using a trained LSTM network. We can then assign a grade to a response by its total predictive likelihood; since our LSTM networks are trained on correct responses, a correct response will have a higher predictive likelihood than an incorrect one. For personalized feedback generation, we can automatically alert a student that they might have made an error if the predictive likelihood of the next input character is lower than a certain threshold. In Figure 1, such an error is shown in red where the student response contains a character that the trained LSTM network predicts as highly unlikely. Using these predictive probabilities, we can also automatically provide hints to a student about the most likely next expression in case they get stuck.

3. REFERENCES

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proc. 26th Intl. Conf. Mach. Learn.*, pages 41–48, June 2009.

[2] M. Cramer, B. Fisseni, P. Koepke, and D. Kühlwein. The naproche project controlled natural language proof checking of mathematical texts. In *Cont. Nat. Lang.*, pages 170–186, 2009.

[3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[4] A. S. Lan, D. Vats, A. E. Waters, and R. G. Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proc. 2nd ACM Conf. Learn. at Scale*, pages 167–176, Mar. 2015.

[5] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *J. Mach. Learn. Res.*, 15:1959–2008, June 2014.

[6] A. Naumowicz and A. Kornilowicz. A brief overview of mizar. In *In Proc. 22nd Intl. Conf Theorem Proving in Higher Order Logics*, pages 67–72, 2009.

[7] W. Zaremba and I. Sutskever. Learning To Execute. *arXiv preprint arXiv:1410.4615*, pages 1–25, Feb. 2015.

Intelligent Composition of Test Papers based on MOOC Learning Data*

Lin Ma
Department of Computer Science and Technology, Tsinghua University
Beijing, China 100084
ml16@mails.tsinghua.edu.cn

Yuchun Ma
Department of Computer Science and Technology, Tsinghua University
Beijing, China 100084
myc@mail.tsinghua.edu.cn

ABSTRACT

In recent years, most of the studies related to MOOC are mainly about prediction and data analysis, while how to evaluate the learning performance is still based on the experience of teachers. Especially, how to compose a proper exam paper is still a tedious work. In this paper, we use genetic algorithm to compose test papers with the support of MOOC learning data considering various constraints and objectives. The experimental results based on a MOOC course show that the mean absolute error of prediction model is roughly around 12 points on 100 points scale and we can successfully achieve the intelligent composition of test papers with various objectives optimized.

Keywords

MOOC(Massive Open Online Course); Machine Learning; Performance Prediction; Genetic Algorithm; Automatic Composition of Test Paper

1. INTRODUCTION

In this paper, we focus on how to evaluate MOOC learners' learning performance. Traditional written test's high dependence on the teacher and neglect of the learners make it ineffective in the MOOC learning environment. So in this paper, we provide a novel approach that the final exam papers could be automatically composed with the support of MOOC learning data considering various constraints and objectives. In our approach, different machine learning techniques are employed to construct a prediction model of learning performance based on MOOC learning data. With the prediction model of the learning performance, an intelligent composition approach is proposed with various objectives and constraints considered.

2. RELATED WORK

From 2012 to now, more and more people start to study MOOC, such as [2, 1]. Common algorithms of automatically generating test papers mainly include stochastic selection with approximate matching[6], backtracking and genetic algorithm[4, 5].

3. MODEL AND OVERALL FRAMEWORK

3.1 Model

Figure 1 shows the whole process of using MOOC learning data to intelligently auto-generate test paper. The input is MOOC learners' learning data, and the output is a test paper. Here we use the scores of usual quiz and homeworks as learning data, and use the score of final exam to represent learning performance. The whole process is composed of two important phases, performance prediction and test paper's composition. In the first phase, we use machine learning techniques to train the performance prediction model. And in the second phase, we use genetic algorithm to generate test paper.

3.2 Classified Performance Prediction Model for Different Levels of Learners

Performance prediction is a very common and simple regression problem. However, if model is constructed simply for all learners, the prediction results are always not very satisfactory because of the complexity and diversity of learners. Intuitively, we know that students with different learning levels will have different learning patterns [2]. Therefore, the features which are useful and contribute to the prediction results are obviously different for different levels of learners. Hence, the performance prediction of massive learners should be based on the level of learners, rater than treating them as a whole. Different levels of learners should have different prediction model.

3.3 Intelligent Composition of Test Papers Based on Genetic Algorithm

The goal of this section is to generate a test paper that meets all constraints as much as possible. The constraints include total score, difficulty, question types and knowledge points. We need to format all constraints to a argument matrix as the input of the composition of test papers[6]. For question types and knowledge points, it can be obtained by multiplying distribution matrix by total scores. For difficulty, most of the statistical analysis show that a good test has a normal distribution of scores, so we can generate it according to the

*This paper is supported by Online Education Fund of Quan Tong Education (2016ZD304).

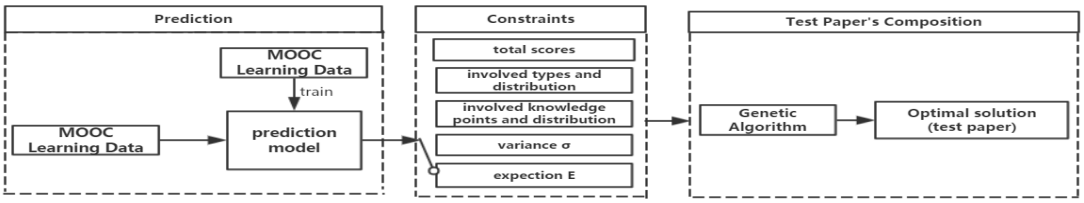


Figure 1: The model framework of intelligent composition of test paper based on MOOC learning data

Table 1: Prediction Error of Machine Learning Algorithms

Model	M5rules	SMOreg	LWR	LR	BP
Overall	21.103	21.423	21.657	21.132	34.006
Classified	12.069	12.82	11.127	13.026	15.058

expected scores E and variance σ. The expected score is exactly our predicted results in the last phase. The proportion of a certain difficulty level can be derived from the proportion of students in the corresponding scores. For instance, the proportion of "easy" level is equal to the proportion of students in scores 80-100 if there are a total of 5 levels. The design of the genetic algorithm can be obtained from [4] and [6].

4. EXPERIMENTAL RESULTS

4.1 Data Description

Our data comes from *Combinatorial Mathematics*, a math class opened for graduates majored in computer science and technology, Tsinghua University. It has been opened in both EdX and xuetangX. We can get a total of 35 features, including 25 quiz scores, 8 homework scores and 1 final exam score. And the feature need to be predicted is final exam score since we use it to represent learner's learning performance.

4.2 Prediction Experiment and Results

This experiment is a comparative experiment of the classified prediction model and the overall prediction model. We adopt machine learning algorithms used in [3]. In classified model, we divided the learners into two groups according to their academic performance, passing the exam as a group and the rest as a group. The final prediction results are shown in table 1. Note that here we adopt mean absolute error as our prediction error and all of the scores appearing in this paper are converted to percentile scores. From the results, we find classified model for different levels of learners can greatly reduce the prediction error by around 10 points.

4.3 The Composition of Test Paper Based on MOOC Learning Data

This experiment is conducted to verify the performance of the composition algorithm. In this experiment, we first randomly select n testers from 17 testers. And then generating a test paper according to the average performance of all selected testers to test them. From the experimental results shown in table 2, we find that predicted scores(performance)

Table 2: Examination Results

number of testers	predicted scores (performance)	real exam scores
17	77.59	75.08
16	79.75	71.37
13	73.91	69.23
12	75.94	61.14
6	82.48	69.63

are very close to their real exam scores and the error decreases as the number of testers increases, which indicates that our model is effective for evaluation of a group of MOOC learners' learning performance.

5. CONCLUSION

The general idea of this paper is automatically generating personalized papers under the guidance of MOOC learners' usual performance, so as to guide their further study. But there are still many details need to be further refined, such as prediction accuracy, efficiency of the composition algorithm, and so on. Therefore, it's just a first step in integrating machine learning, MOOCs, and test development. Our future work will continue to focus on these details to make it better.

6. REFERENCES

[1] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 2013.

[2] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton. Studying learning in the worldwide classroom: Research into edx's first mooc. *Research & Practice in Assessment*, 8, 2013.

[3] S. B. Kotsiantis and P. E. Pintelas. Predicting students marks in hellenic open university. In *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, pages 664–668. IEEE, 2005.

[4] Y. Ou-Yang and H.-F. Luo. Design of personalized test paper generating system of educational telenet based on genetic algorithm. In *Computer Science & Education, 2009. ICCSE'09. 4th International Conference on*, pages 170–173. IEEE, 2009.

[5] Y. Qing. Research on auto-generating test paper based on genetic algorithm. *JOURNAL OF JINAN UNIVERSITY(SCIENCE AND TECHNOLOGY)*, 18(3):228–231, 2004.

[6] G. M. Wang Yuying, Hou Shuang. Algorithm for automatic test paper generation. *JOURNAL OF HARBIN INSTITUTE OF TECHNOLOGY*, 35(3):342–346, 2003.

Toward Replicable Predictive Model Evaluation in MOOCs

Josh Gardner, Christopher Brooks
School of Information
University of Michigan
{jpgard, brooksch}@umich.edu

ABSTRACT

In this paper, we present and apply a procedure for evaluating predictive models in MOOCs. First, we expand upon a procedure to statistically test hypotheses about model performance which goes beyond the state-of-the-practice in the community and covers the full scope of predictive model-building in MOOCs. Second, we apply this method to a series of algorithms and feature sets derived from a large and diverse sample of MOOCs ($N = 31$), concluding that several models built with simple clickstream-based feature extraction methods outperform those built from forum- and assignment-based feature extraction methods.

1. INTRODUCTION AND RELATED WORK

Building predictive models of student success has emerged as a core task in the fields of learning analytics and educational data mining.[1] The process of building such models in MOOCs involves at least three key stages: (1) extracting structured data and informative features from raw platform data (clickstream server logs, database tables, etc.); (2) selecting algorithms and models; and (3) tuning hyperparameters. Together, these stages profoundly influence the performance of predictive models. We identify at least two methodological gaps in current educational data mining research as it relates to this task: (1) current research typically isolates these steps, e.g., evaluating different approaches to feature extraction or algorithm selection separately without considering their relation to each other; and (2) procedures for rigorous and reproducible statistical inference about the relative performance of these models, and accounting for the many model specifications considered in the course of an experiment, are often not followed.

Previous predictive modeling research in MOOCs has evaluated features derived from clickstreams, discussion fora, assignments, and surveys, among other sources. In addition, this research has applied a variety of algorithms to such data for dropout prediction, including linear and logistic regression, support vector machines, tree-based methods, ensemble methods, neural networks, and deep learning. However, a literature survey by the authors indicated that accepted statistical practices for evaluating these models are often neglected by such research[2] In particular, more than half of surveyed research did not utilize any statistical testing for evaluating model performance, despite obtaining estimates directly on the training set through cross-validation for multiple models. These methods are susceptible to spurious results and low replicability due to multiple comparisons, biased performance estimates, and random variation from resampling schemes [3, 4, 7, 11]. Recent research has provided evidence that some MOOC research may not be replicable when applied to new or different courses [1]; at the very least, this highlights the importance of adopting reproducible and statistically valid methods for model evaluation in MOOCs [8]. An extensive literature exists on statistically reliable methods for model evaluation [4, 6, 11].

2. METHODOLOGY

We implement a testing and inference procedure from [3] for selecting the best of $k > 2$ models across $N > 1$ datasets (in this experiment, a *model* is a *feature set-algorithm-hyperparameter combination*), which consists of two steps. First, a Friedman test is used to test the null hypothesis that the performance of all models is equivalent [5]. The Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{1}$$

where R_i^j is the rank of the jth of k algorithms on N datasets and the statistic is χ_{k-1}^2 distributed, is compared to a critical value at the selected significance level ($\alpha = 0.05$ in this experiment). If H_0 is rejected, then we proceed to the second stage, the post-hoc Nemenyi test, where

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{2}$$

is used to determine whether the performance between any two classifiers is significantly different, where q_α is based on the Studentized range statistic divided by $\sqrt{2}$.

This two-stage procedure allows us to conduct comparisons across multiple models and datasets to draw inferences about

[1]The current work evaluates models of student dropout in MOOCs, but this methodology applies to any supervised predictive modeling task.

[2]This survey reviewed the 2014-2016 International Society for Educational Data Mining (EDM) and the International Learning Analytics and Knowledge (LAK) conference proceedings, and included research which attempted to predict completion or performance using behavioral or academic features with features derived from MOOC platform data; a full survey is forthcoming in a future work.

whether true performance differences exist, accounting for the number of comparisons k and datasets N. Unlike using simple average cross-validated training performance, this procedure uses statistical testing to evaluate whether the observed difference is statistically significant or may be merely spurious, based on the available data. In applying this method to a *feature set + algorithm + hyperparameter* combination, we can (1) evaluate feature extraction as a testable modeling component; (2) capture and evaluate the synergy between feature extraction, algorithm, and hyperparameters; and (3) draw inferences which fully account for the number of comparisons across all of these elements. [3]

3. EXPERIMENT AND RESULTS

As an illustrative example, we compare a series of models using three feature sets and two predictive algorithms on a set of 31 offerings of 5 unique courses offered by the University of Michigan on Coursera, with 298,909 total learners. From the raw clickstream files and database tables, we extracted a series of features intended to replicate (with some additions) features shown to be effective dropout predictors, with each utilizing information from a different raw data source: *clickstream* [10], *assignment* [9], and *forum* features [1].

We train two classifiers – standard classification trees and adaptive boosted trees – on various combinations of the three feature sets, performing no hyperparameter tuning (to limit the number of comparisons, k). Figure 1 presents the results of our analysis.

Results from dropout prediction after course week 2 are shown in Figure 1, but our findings were consistent across all four weeks examined. We find that models utilizing clickstream features consistently outperform those using forum and quiz features. This difference was statistically significant for all model configurations tested. Changing the classification algorithm had little effect on the performance of quiz- and forum-featured models, which were statistically indistinguishable from each other in every week evaluated. When the clickstream features are combined with forum and quiz features to form a "full" model, this model achieves better performance than the clickstream features alone, but this improvement is never statistically significant over the best clickstream-only model. This suggests that the forum and quiz features contain useful structure which may require powerful, flexible classification algorithms to capture. Our conclusion – that the highest-performing model is statistically indistinguishable from other models in this analysis – stands in contrast to the practice of much of the prior research surveyed, which often concludes that the best average performance is the "best" model; this is intended to serve as an example for inferential language in future research.

4. FUTURE RESEARCH

Future research should utilize this or other methods for statistically evaluating performance comparisons of predictive models. In particular, it should explore Bayesian methods for model evaluation, which allow the direct estimation of probabilities of hypotheses, avoid concerns about multiple comparisons, and have other additional advantages [2].

[3]There are clear advantages to adopting this specific procedure over other testing approaches such as ANOVA, or other nonparametric approaches; see §3.2.1 of [3] for detailed discussion of these benefits.

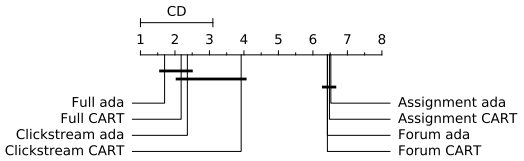


Figure 1: Critical Difference (CD) diagram of week 2 dropout prediction models. Models are plotted by average rank, with bold CD lines indicating statistically indistinguishable models (at $\alpha = 0.05$). We reject H_0 of equivalent performance for models not connected by CD lines. These results show a statistically significant performance gap between clickstream features and assignment or forum features.

5. REFERENCES

[1] J. M. L. Andres, R. S. Baker, G. Siemens, D. GAŠEVIĆ, and C. A. Spann. Replicating 21 findings on student success in online learning.

[2] A. Benavoli, G. Corani, J. Demsar, and M. Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. 14 June 2016.

[3] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7(Jan):1–30, 2006.

[4] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923, 15 Sept. 1998.

[5] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.*, 11(1):86–92, 1940.

[6] S. Garcia and F. Herrera. An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. *J. Mach. Learn. Res.*, 9(Dec):2677–2694, 2008.

[7] C. Nadeau and Y. Bengio. Inference for the generalization error. *Mach. Learn.*, 52(3):239–281, 2003.

[8] F. van der Sluis, T. van der Zee, and J. Ginn. Learning about learning at scale: Methodological challenges and recommendations. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pages 131–140, New York, NY, USA, 2017. ACM.

[9] K. Veeramachaneni, U.-M. O'Reilly, and C. Taylor. Towards feature engineering at scale for data from massive open online courses. 20 July 2014.

[10] W. Xing, X. Chen, J. Stein, and M. Marcinkowski. Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. *Comput. Human Behav.*, 58:119–129, 2016.

[11] O. T. Yildiz, E. Alpaydin, and Senior Member. Ordering and finding the best of k>2 supervised learning algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3), 2006.

Modeling the Zone of Proximal Development with a Computational Approach

Irene-Angelica Chounta, Bruce M. McLaren
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh PA, 15213, USA
{ichounta,bmclaren}@cs.cmu.edu

Patricia Albacete, Pamela Jordan, Sandra Katz
Learning Research and Development Center
University of Pittsburgh
Pittsburgh PA, 15260, USA
{palbacet, pjordan, katz}@pitt.edu

ABSTRACT

In this paper, we propose a computational approach to modeling the Zone of Proximal Development of students who learn using a natural-language tutoring system for physics. We employ a student model to predict students' performance based on their prior knowledge and activity when using a dialogue tutor to practice with conceptual, reflection questions about high-school level physics. Furthermore, we introduce the concept of the "Grey Area", the area in which the student model cannot predict with acceptable accuracy whether a student has mastered the knowledge components or skills present in a particular step.

Keywords

Natural-language tutoring systems, intelligent tutoring systems, student modeling, zone of proximal development

1. INTRODUCTION

Intelligent Tutoring Systems (ITSs) support students in grasping concepts, applying them during problem-solving activities, addressing misconceptions and in general improving students' proficiency in science, math and other areas [6]. ITS researchers have been studying the use of simulated tutorial dialogues that aim to engage students in reflective discussions about scientific concepts [4]. However, to a large extent, these systems lack the ability to gauge students' level of mastery over the curriculum that the tutoring system was designed to support. This is also challenging for human tutors, who do gauge the level of knowledge and understanding of their tutees to some degree, although they are poor at diagnosing the causes of student errors [3]. We argue that in order to provide meaningful instruction and scaffolding to students, a tutoring system should appropriately adapt the learning material with respect to both content and presentation. A way to achieve this is to dynamically assess students' knowledge state and needs. Human tutors use their assessment of student ability to adapt the level of discussion to the student's "zone of proximal development" (ZPD)—that is, "*the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers*" [7]. Deriving ways to identify and formally describe the ZPD is an important step towards understanding the mechanisms that drive learning and development, gaining insights about learners' needs, and providing appropriate pedagogical interventions [2]. Following the practice of human tutors, we propose a computational approach to model the ZPD of students who carry out learning activities using a dialogue-based intelligent tutoring system. We employ a student model to assess students' changing knowledge as they engage in a dialogue with the system. Based on the model's predictions, we define the concept of the "Grey Area", a probabilistic region in which the model's predictive accuracy is low. We argue that this region can be used to indicate whether a student is in the ZPD. Our research hypothesis is that we can use the outcome of the student model (i.e., the fitted probabilities that predict students' performance) to model students' ZPD. To the best of our knowledge, this is a novel approach to modeling the ZPD. Even though we focus on dialogue-based tutoring systems, we expect that our approach can be generalized and extended to other kinds of ITSs.

2. METHODOLOGY

In this study, we used data collected during three previous studies with the Rimac system to train a student model and frame the proposed approach. Rimac is a web-based natural-language tutoring system that engages students in conceptual discussions after they solve quantitative physics problems [5]. Rimac's dialogues present a directed line of reasoning (DLR) where knowledge components (KCs) relate to tutor question/student response pairings. To model students' knowledge we used an Additive Factor Model (AFM) [1]. The model predicts the probability of a student completing a step correctly as a linear function of student parameters, knowledge components and learning parameters. AFM takes into account the frequency of prior practice and exposure to skills but not the correctness of responses. The dataset consists of training sessions of 291 students over a period of 4 years (2011-2015). Students worked on physics problems that explore motion laws and address 88 knowledge components (KCs). The dataset contains in total 15,644 student responses that were classified as correct or incorrect using the AFM student model.

Our research hypothesis is that we can use the fitted probabilities, as predicted by the student model, to model the ZPD. The core rationale is that if the student model cannot predict with high accuracy whether a student will answer a tutor's question correctly, then it might be the case that the student is in the ZPD. The student model provides predictions at the step level: each step consists of one question/answer exchange from the tutorial dialogue. A step may involve one or more KCs. The classification threshold (i.e., the cutoff determining whether a response is

classified as correct or incorrect) is 0.5 and it was validated by the ROC curve for the binary classifier. We expect that the closer the prediction is to the classification threshold, the higher the uncertainty of the model and thus, the higher the prediction error. Based on our hypothesis, this window of uncertainty can be used to approximately model the student's zone of proximal development. We refer to this window as the "Grey Area".

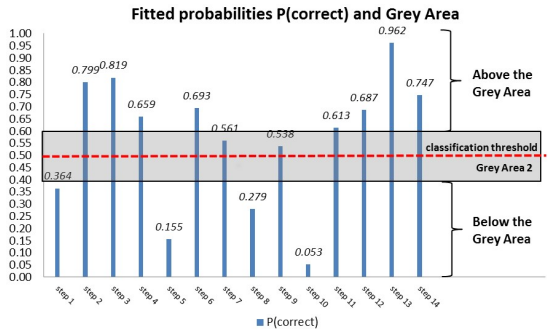


Figure 1. The Grey Area concept with respect to the fitted probabilities as predicted by the student model for a random student and for the various steps of a learning activity. Here we depict the example of a symmetrical Grey Area extending on both sides of the classification threshold.

The concept of the Grey Area is depicted in Figure 1. The space "Above the Grey Area" denotes the area where the student is predicted to answer correctly and consequently may indicate the area above the ZPD; that is, the area in which the student is able to carry out a task without any assistance. Accordingly, the space "Below the Grey Area" denotes the area where the student is predicted to answer incorrectly and consequently may indicate the area below the ZPD; that is, the area in which the student is not able to carry out the task either with or without assistance. In this paper, we model the grey area symmetrically around the classification threshold for simplicity and because the binary classifier was set to 0.5. However, the symmetry of the Grey Area is something that could change depending on the classification threshold and the learning objectives. Furthermore, we do not propose a specific size for the Grey Area. We believe that the decision about the appropriate size (or shape) of the Grey Area is not only a modeling issue but mainly a pedagogical one since it relies on the importance of the concepts taught, the teaching strategy and the learning objectives.

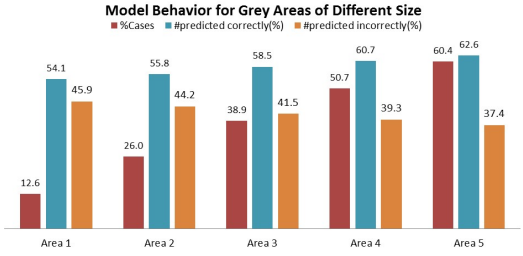


Figure 2. Model behavior (total number of predicted cases, cases predicted correctly and cases predicted incorrectly) within five grey areas of different sizes. The areas are ordered from the most narrow (Area 1) to the widest (Area 5).

Figure 2 presents an analysis of the cases that are contained in the Grey Area. In this preliminary analysis, we examined five Grey Areas of different size. On one hand, choosing a narrow grey area to model the ZPD would limit the number of cases we scaffold since fewer cases would fall within the area. On the other hand, choosing a wide grey area would affect the accuracy; that is, some cases that could be predicted correctly would be falsely labeled as "grey". However this work does not aim to define the appropriate size for the Grey Area but rather to study how the model's behavior may change for areas of different size.

3. DISCUSSION

In this paper, we present a computational approach that aims to model the Zone of Proximal Development in ITSs. To that end, we introduce the concept of the "Grey Area". Our proposal is that if the model cannot predict the state of a student's knowledge, it may be that the student is in the ZPD. We envision that the contribution of the proposed approach, besides its novelty (to the best of our knowledge there is no quantified operationalization of the ZPD) will be in defining and perhaps revising instructional methods to be implemented by ITSs. Choosing the "next step" is a prominent issue in the case of dialogue-based intelligent tutors. Not only should the task be appropriate with respect to the background knowledge of the student, but it should also be presented in an appropriate manner so that the student will not be overwhelmed and discouraged. To address this issue, we need an assessment of the knowledge state of each student and insight into the appropriate level of support the student needs to achieve the learning goals. This is described by the notion of ZPD. It is evident that if we can model the ZPD then we can adapt our instructional strategy accordingly. A limitation of our work is that we have not yet been able to conduct a rigorous evaluation of our approach; however, plans to validate our modeling methods are being developed. Our immediate plan is to carry out extensive studies to explore the proposed approach to modeling the ZPD further, as well as to better understand the strengths and limitations of using a student model to guide students through adaptive lines of reasoning.

4. REFERENCES

[1] Cen, H., Koedinger, K., and Junker, B. 2008. Comparing two IRT models for conjunctive skills. In *International Conference on Intelligent Tutoring Systems*, 796–798.

[2] Chaiklin, S. 2003. The zone of proximal development in Vygotsky's analysis of learning and instruction. *Vygotsky's educational theory in cultural context.* 1: 39–64.

[3] Chi, M.T., Siler, S.A., and Jeong, H. 2004. Can tutors monitor students' understanding accurately? *Cognition Instruct*. 22, 3: 363–387.

[4] Di Eugenio, B., Glass, M., and Trolio, M.J. 2002. The DIAG experiments: Natural language generation for intelligent tutoring systems. In *INLG02, The Third International Natural Language Generation Conference*, 120–127.

[5] Katz, S., and Albacete, P.L. 2013. A tutoring system that simulates the highly interactive nature of human tutoring. *J. Educ. Psychol.* 105, 4: 1126.

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

[7] Vygotsky, L. 1978. Interaction between learning and development. *Readings on the development of children.* 23, 3: 34–41.

A Prediction and Early Alert Model Using Learning Management System Data and Grounded in Learning Science Theory

Wonjoon Hong
University of Nevada, Las Vegas
4505 S Maryland Pkwy Las Vegas, NV 89154, USA
+001(702)895-3253
hongw1@unlv.nevada.edu

Matthew L. Bernacki
University of Nevada, Las Vegas
4505 S Maryland Pkwy Las Vegas, NV 89154, USA
+001(702)895-4013
matt.bernacki@unlv.edu

ABSTRACT

Students experience considerable challenge in STEM coursework and many struggle to earn the grades needed to move forward in their majors. Interventions informed by prediction models can support learners to ensure successful completion of STEM courses and entry into the STEM workforce. In order to accurately target intervention efforts, we developed a prediction model based on log data generated by student use of content hosted on a learning management system (LMS; Blackboard Learn) course site in the first weeks of the course. The prediction model employed a forward selection logistic regression algorithm (with 10-fold cross validation) trained on four semesters of data, and provided instructors the opportunity to message students and provide learning support before the first major exam, potentially intervening before onset of poor performance. The best fitting model was used to identify students unlikely to obtain the required grade (B or better) in the course. Among 106 students predicted to perform poorly, 63 received a message from the instructor's account that referenced an upcoming exam and linked students to supportive materials. Messaged students who accessed learning supports outperformed non-messaged but eligible students (n = 43) on each of five subsequent exams throughout the semester (ds = .64 - .88). Fifty-eight percent earned a B or better, compared to 25% of non-messaged peers predicted to earn a C or worse. This study affirms that data-driven early alert messages can provide targeted support and boost achievement in challenging STEM courses.

Keywords

Learning management system, Prediction modeling, Early warning system, STEM learning, learning sciences

1. INTRODUCTION

Learning management system (LMS) have become a central tool in higher education. Logs of learning events can be combined with achievement data in order to identify (un)productive patterns of events and predict the achievement of future students based on their behavioral match to prior students who achieved certain levels of performance [1].

2. METHODS

The university LMS, Blackboard Learn, captures and records student use of materials hosted on course sites. Student activity and achievement data (N=510) from 4 semesters of an undergraduate calculus course taught by two instructors (identical content, assessments) from fall 2014 to spring 2016 informed prediction modeling (Table 1).

Table 1. Training and testing data

Section	Training set	Testing set
Instructor A	Fa 2014 & Sp 2015 (n=167)	Fa 2015 (n=96)
Instructor B	Fa 2014 & 2015 (n=161)	Sp 2016 (n=86)
Both	Instructor A (Fa 2014 & Sp 2015) Instructor B (Fa 2014 & 2015) (n=328)	Instructor A (Fall 2015) Instructor B (Spring 2016) (n=182)

Developing the prediction model went through two main phases, training and testing process. In the training phase, logistic regression with forward selection was used to build the prediction model, and the problem of overfitting was examined through 10-fold cross-validation. In the testing phase, the most accurate prediction model developed in the training phase was applied to the testing data set to assess potential overfitting and ensure generalizability to future students' data [2].

Based on the Kappa (κ) and recall, the best 3-week prediction model developed through the training and testing phases was then applied to data from fall 2016 Calculus students to identify students in need of an early alert message that provides learning support.

In order to investigate the effect of messaging identified students, those identified as likely to perform poorly by the prediction model were randomly divided into two groups, a "Message" group who would receive a message that focused attention on an upcoming exam and some useful learning resources (Figure 1) and a "No Message" group who would not.

Hi [Name]!
Our first course exam is coming up on Friday…
1. The first is a one-page summary of advice from students who have completed the course with an excellent grade in the past.…
2. A set of learning modules called "The Science of Learning to Learn." These modules describe learning strategies you can use with our course materials…

Figure 1. Message to students

3. RESULTS

Among three models, the prediction model based on Instructor B's students produced the best Kappa ($\kappa = 0.26$) and recall (73%) values. The model accurately identified ≥ 7 in 10 students who would ultimately earn less than 80% of points (i.e., a C or Worse). We thus moved forward to the testing phase using the Instructor B model (Table 2) and for the prediction and messaging phase.

Table 2. Prediction models in the training and testing phase

	True: Predicted				K	Accuracy (%)	Precision (%)	Recall (%)
	1;1	1;0	0;1	0;0				
Training set								
Instructor A (Fall 2014 & Sp 2015)	79	12	46	27	.25	65	87	37
Instructor B (Fa 2014 & 2015)	**39**	**36**	**23**	**63**	**.26**	**63**	**52**	**73**
Both	97	69	63	96	.19	59	59	60
Testing set								
Instructor A (Fa 2015)	16	25	9	46	.24	65	65	84
Instructor B (Sp 2016)	19	23	11	33	.20	61	59	75
Both	35	48	20	79	.21	63	62	80

In the testing phase, attributes and their weights achieved from the training phase were applied to the testing data to examine risk of overfitting. The prediction model resulted in the Kappa value of .20 or more for all testing sets. In addition, values of recall were 84, 75, and 80 respectively, all of which were greater than result in the training phase. We thus retain the Instructor B model for the prediction and messaging phase.

Upon sending the message four days prior to the first exam, student access of recommended resources and performance on exams were tracked throughout the remainder of the semester. For all exams throughout the semester, the students in treatment group (i.e., Message & Access) performed better than those without any treatment (No Message, No Access; $p < .05$). In addition, effect sizes for all exams were more than "medium" ($d > .5$) (Table 3).

Table 4. Contingency Table

		Predicted C or Worse		Total
		Messaged	Control	
True	B or Better	11 (58%)	7 (25%)	18
	C or Worse	8 (42%)	21 (75%)	29
Total		19	28	47

Table 4 shows the proportion of students who performed better than (i.e., B or Better) vs. as projected (i.e., C or Worse). A Chi-square analysis indicated that a significantly greater proportion of students (58%) in the Message and Access group earned a final grade of B or better, $\chi^2 (47) = 5.18$, $p = .02$. Only 25% of students predicted to earn a C or worse outperformed their prediction in the No Message, No Access control group.

4. DISCUSSION

In this study, those who received a brief email message from a course instructor and accessed a learning resource outperformed non-messaged students on all exams. Results thus indicate that data-driven interventions can be provided relatively early in the semester – six weeks earlier than the typical data-driven indicator of poor future outcome: a week 9 response to midterm grades. The >200-word message required only a minute or two of a typical student's time, and a visit to the advice page – the common material accessed – required only slightly more time investment from messaged students (~900 words).

The benefits of receiving a message and accessing the resources it recommends were substantial: 12% on all exams, or a full letter grade. Surprisingly, few students heeded the early alert as intended; 30% of messaged students accessed supportive materials, confirming that obtaining students' attention is a clear challenge to realization of the benefits messaging can provide. Messaging efforts thus clearly require improvement. We must also consider how to provide more adaptive message contents based on students' likelihoods of poor performance, or different supports based on the maladaptive practices summarized by features present in students' prediction models. More specific feedback about the kinds of learning behaviors that require adjustment may further increase messages' effects.

5. ACKNOWLEDGMENTS

This project was supported by National Science Foundation Award number #1420491 and Office of Information Technology.

6. REFERENCES

[1] Arnold, K. E., & Pistilli, M. D. 2012. Course signals at Purdue: Using learning analytics to increase student success. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (Apr. 2012). ACM, New York, NY, 267-270.

[2] Hämäläinen, W. and Vinni, M. 2010. Classifiers for educational data mining. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizky, and R. Baker, Eds. Data Mining and Knowledge Discovery Series. CRC Press, Boca Raton, FL, 57-7.

Table 3. Result of t-test of scores for all exams

	No Message & No Access			Message & Access			t	df	Sig.	Mean difference	Cohen's d
	N	Mean	SD	N	Mean	SD					
Exam 1	24	77.0	11.0	17	85.5	8.3	2.701	39	0.010	8.51	0.877
Exam 2	23	73.7	19.0	17	85.7	10.4	2.349	38	0.024	12.01	0.783
Exam 3	22	59.5	14.8	18	71.5	22.2	2.047	38	0.048	12.00	0.637
Exam 4	22	58.9	15.9	19	71.0	20.3	2.136	39	0.039	12.09	0.663
Final	22	55.7	23.8	19	70.9	23.6	2.043	39	0.048	15.17	0.640

Cluster Analysis of Real Time Location Data - An Application of Gaussian Mixture Models

Alvaro Ortiz-Vazquez
EdLab
Teachers College Columbia University
New York, New York USA
ao2444@columbia.edu

Xiang Liu
EdLab
Teachers College Columbia University
New York, New York USA
xl2438@tc.columbia.edu

Ching-Fu Lan
EdLab
Teachers College Columbia University
New York, New York USA
cl2483@tc.columbia.edu

Hui Soo Chae
EdLab
Teachers College Columbia University
New York, New York USA
hsc2001@tc.columbia.edu

Gary Natriello
EdLab
Teachers College Columbia University
New York, New York USA
gjn6@tc.columbia.edu

ABSTRACT
Clustering analysis in the context of education is important for determining the effectiveness of group activities especially when participants freely rotate between groups such as in a gallery exhibit or other informal learning space or set-up. In this paper, we cover a method of applying Gaussian Mixture Models to two-dimensional data. We further describe the analysis procedure, and the success of implementing this analysis using simulated data and real data. Finally, we discuss some educational applications as well as future directions for this research.

Keywords
Gaussian Mixture Models, MCMC, Gibbs Sampling, Real-Time Location System, Informal Learning Spaces, Learning Analytics, Dynamic Mixture Model

1. INTRODUCTION

Real-time locating systems have become increasingly popular and are predicted to be more widely adopted in informal learning institutions such as libraries, museums, and after school spaces in the next few years [2] [4]. Location intelligence and contextually relevant information can inform dynamically customized information and meaningful learning analytics for both learners and educators based on visitors and/or learners' location [3]. Such data are especially useful to understand social interactions in informal learning events. Therefore, it is essential for researchers to develop data mining methods to more efficiently and effectively explore real-time location data of learners.

Gaussian Mixture Models (GMM) are very useful for analyzing two-dimensional data which may be clustered into groups such as that collected by a real-time locating system in an informal learning space. To estimate the parameters of the GMM we employ a Markov Chain Monte Carlo method of Gibbs sampling [1] whose stationary state is the posterior distribution of the mixture model. This method applied to a frozen snapshot of the two-dimensional real-time location tracking data allows us to gain information about the groups, such as group membership, group location, and internal group dispersion, based only on the tag position data. Other algorithms such as k-means clustering may similarly cluster two-dimensional data but are non-parametric whereas Gibbs sampling is parametric.

2. DATA ANALYSIS

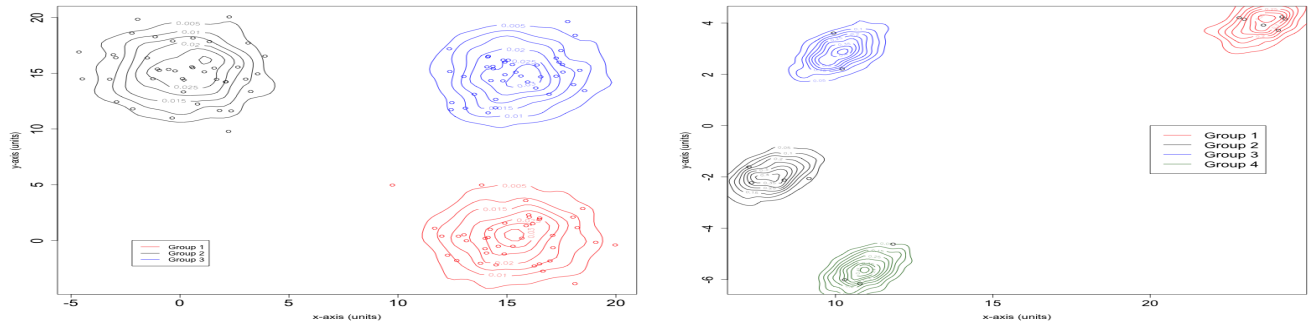
2.1 Simulations

To test the Gibbs sampling process and our R code we have drawn a set of location data points from bivariate normal distributions centered around three different centers ($\boldsymbol{\mu}_1 = (15, 15)$, $\boldsymbol{\mu}_2 = (15, 0)$,$\boldsymbol{\mu}_3 = (0, 15)$) with a common covariance. We observed the latent parameters of our Gibbs sampler reaching a stationary state in less than 100 iterations. In Figure 1a we generate estimated points using the estimated group centers and covariance and perform kernal density estimates to generate the coverage contours plotted over the original generated data. The percentage of estimated points outside the contours is marked on the contour lines. In this case we see that for 120 data points, a small number of the data lie outside of the 99.5% percent coverage contours. We can also verify the results by comparing the generating values for the centers and covariance with the estimated values.

2.2 Applications on Real Data

The real data were collected at an Edlab meeting at an innovative learning space: the Smith Learning Theater at Teachers College Columbia University. The Smith Learning Theater features technologies such as the Quuppa ^{TM} real-time locating system, installed to return measurable results and provide feedback to organizers and facilitators. In this meeting, 15 EdLab members wore Quuppa real-time locating tags and freely explored four stations of augmented/virtual reality apps in order to provide reviews for a national edtech competition. Applying the Gibbs Sampling method over the real data we again obvserved convergence within just 100 iterations. Again the coverage contours are drawn onto the plot of the positions in Figure 1b. In this analysis we did not have previous knowledge on the station device locations likely to be correlated with the group centers. However, we can still verify the success of the algorithm by noting that the data points are largely within the ninety-five percent coverage region. As such, our method returns accurate group information even with a small dataset.

3. DISCUSSION



(a) (b)

Figure 1: Kernal Density Estimate Contour Plots Over Simulation Data (a) and Real Data (b)

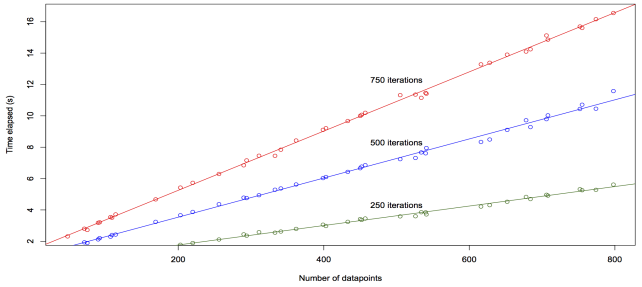


Figure 2: Linear Correlation Between Computational Time and the Number of Data Points

3.1 Educational Research and Applications

Our method has the limitation that the expected number of groups must be specified prior to performing the Gibbs sampling. This quantity can be available for events where group work takes place, or participants move around through different stations. In such an event our analysis can be implemented repeatedly over a series of consecutive discrete snapshots covering a period of time. By observing the group membership at each snapshot, the educator can determine information about who moved together as a group, or who moved mostly independently. Common group membership can be denoted in an adjacency matrix for the tags where the value for each index (i, j) is the number of snapshots in which two locating tags y_i, y_j shared the same group assignment. This approach has the potential to provide information about whether the learning space or activity was better suited for group learning or independent learning and the preferences of each participant to remain with the same group of people or move about with different people. In other events where group work may be taking place one can easily determine the amount of cross-group collaboration during a period of time by again looking at the cumulative group assignment data.

3.1.1 *Feasibility Analysis*

The implementation of the Gibbs Sampling algorithm takes linear $\mathcal{O}(N)$ time where N is the number of position data points in a single snapshot. We can generate N position data points and record the time elapsed for M iterations and visualize the linear relationship in Figure 2. Given an hour long event with 500 participants, covered by 360 snapshots, the linear model suggests that one could perform 250 iterations of the sampler over every snapshot in under twenty minutes. As such implementation of our method is feasable for most educational contexts.

3.2 Future Work

While our model is useful to see the group information within a snapshot of real-time location data, we believe that more important data will arise from extending our current mixture model to a Dynamic Mixture Model (DMM) [5]. In such a DMM, the group distribution of each snapshot would be dependent on the previous one. According to Wei et al. (2007) the assumption that two consecutive snapshots are dependent can allow us to analyze important patterns that would otherwise be missed in discrete snapshot analysis. By incorporating the temporal component, we expect to more accurately model transitions between groups. The application of our method is especially valuable in informal learning spaces as many learning events in these spaces encourage free exploration and group interactions, and evaluating learners' engagement and social group dynamics is challenging using other traditional research methods.

References

[1] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE transactions on pattern analysis and machine intelligence*, 6(6):721–41, jun 1984.

[2] B. Herr-Stephenson, D. Rhoten, D. Perkel, C. Sims, A. Balsamo, M. Klosterman, and S. S. Bautista. *Digital Media and Technology in Afterschool Programs , Libraries , and Museums*. 2011.

[3] K. Jaebker and G. Bowman. Context is king: Using indoor-location technology for new visitor experiences | MW2015: Museums and the Web 2015, 2017.

[4] L. Johnson, S. Adams Becker, M. Cummins, V. Estrada, A. Freeman, and C. Hall. *Horizon Report: 2016 Higher Education Edition*. The New Media Consortium, Austin, Texas, museum edi edition, 2016.

[5] X. Wei, J. Sun, and X. Wang. Dynamic mixture models for multiple time series. *Proceedings of the 20th international joint conference on Artifical intelligence*, (Dmm):2909–2914, 2007.

An LDA Topic Model and Social Network Analysis of a School Blogging Platform

Xiaoting Kuang
EdLab
Dept. of Human Development
Teachers College
Columbia University
xk2120@columbia.edu

Hui Soo Chae
EdLab
Teachers College
Columbia University
hsc2001@columbia.edu

Brian Hughes
EdLab
Teachers College
Columbia University
bsh2001@columbia.edu

Gary Natriello
EdLab
Teachers College
Columbia University
gjn6@columbia.edu

ABSTRACT
Pressible is a school blogging and content management system developed by EdLab at Teachers College Columbia University. In this paper, social network analysis and natural language processing with Latent Dirichlet Allocation topic model approaches were utilized to gain insights into Pressible, to explore four developmental stages of a college-wide social network and their associations with blog content. The results showed that professors who developed courses became the most influential persons in the network. Students extended the online discussion topics beyond the scope of course topic set by professors.

Keywords
SNA, NLP, Topic Model, LDA

1. INTRODUCTION
EdLab adapted the Wordpress Content management systems (CMS) framework and developed Pressible for the Teachers College (TC) community in 2008. It was designed for fast content delivery, minimization of users' time spent managing technology, and developing connections between users (Zhou, 2013). From the perspective of social constructivist theory, people communicate, contribute and acquire knowledge through social engagement and discussion of topics (Vygotsky, 1978). People also gain knowledge online via connecting information (Siemens, 2004). Massive Open Online Courses (MOOCs) provide more opportunities for people to study for personal intellectual growth (Kizilcec et al., 2017). Social factors from online discussion forums (Rose, et al., 2014) and engaging in higher order thinking behaviors enhanced learning in MOOCs (Wang, et al., 2016). Higher Education utilizes academic blogging to facilitate social networking, self-directed learning, and collaboration. Simulation studies on the blogosphere indicate that improved management facilities on course blogs positively affect the density and connectedness in learning networks (Wild & Sigurðarson, 2011). This study utilized social network analysis (SNA) to investigate human-human interaction and the development of social connections on this blogging platform. Next, Latent Dirichlet Allocation (LDA) topic model method was applied to understand human-information interaction during different developmental stages of Pressible. This study provides an exploratory examination of four developmental stages of an online learning community in a school blogging system.

2. METHODOLOGY
2.1. Participants and Data Collection
The data were collected from the entire Pressible database and contained 3598 users and 594 sites, with 50422 posts in total. The specific aim of this study was to explore the social network and its association with content creation. Only the interactions between registered IDs were counted as valid connections. After the reconstruction of the database for SNA, there were 172 blogs with data on a total of 11146 connections and 429 interactive users.

2.2. Social Network Analysis
SNA is a method to analyze the connections, relationships, and interactions between individuals and communities in the collaborative social network, expressed as the node and edge diagrams (Wild, 2016; Slater et al., 2017). In this study, R package igraph (Csardi & Nepusz, 2006) constructs, modifies and calculates the social networks. Density measures the proportion of contacts observed between pairs of nodes in the network; Eigen centrality measures the importance of a node's network by weighting its top connecting nodes' indegree and outdegree centrality (Daniel, et al., 2010).

2.3. Latent Dirichlet Allocation Topic Modeling
To analyze the content of comments and posts in the blogs, LDA topic modeling was utilized to discover and infer the general topics by scanning the words and their distribution probabilities within documents (Blei, et al., 2003). The R package *tm* was used to construct the corpus for text mining. The *tm* package removes spaces, stop words, numbers, spaces, and punctuation, converting the words to lower case and roots to construct a term-document matrix, which allows analysis of individual words in the corpus (Feinerer & Hornik, 2015; Lang, 2017). The R packages *topicmodels and tidytext* were utilized to calculate the term frequency, construct the inverse document matrix, remove the uncommon terms, find the most common words for individual topics and group the documents by generated topics (Grün & Hornik, 2011; Lang, 2017; Silge & Robinson, 2017).

3. RESULTS AND DISCUSSIONS
3.1. Social Network Development
Descriptive statistics analysis on yearly data was conducted to show the general social network activity in Pressible by developmental stages (Tables 1). The results indicate that this blogging system shifted from a development stage (beginning to 2010 Summer), to a stable growth stage (2010 Fall to 2012 Summer), a rapid growth stage (2012 Fall to 2015 Summer), into a decline stage (2015 Fall until now). The active member numbers increased from the development stage to rapid growth stage and decreased in the decline stage. Their engagement rates as average connection numbers increased from development to the rapid growth stage, which also dropped at the decline stage. Therefore, the number of active members and their engagement rate determine the growth of this online social learning community. The density of the social network among active members decreased while the network was growing from 2011 to 2015 (Fig. 1), indicating that the network became decentralized as more active members joined. Most of the participants were students. They became less active in interactions on Pressible after graduation. New students joined the social network and formed new social centers. Thereby, the global social density decreased because of the dynamic student community (Fig. 1). As more professors built their courses on Pressible, more active students joined this online learning community for discussions and made meaningful connections. Recruiting more professors to take

advantages of Pressible for its online course creation features is a key to maintaining the rapid growth of this social network.

Table 1. Descriptive Statistics by Developmental Stage

Stage	Ave. conn. (/year)	Ave. active IDs (/year)	Ave. conn. per IDs (/year)	Top popular topic of the stage
Development	215.5	36.5	5.9	video game
Stable Growth	1333	89.5	14.9	teach and learn
Rapid Growth	2021	118.7	17.0	think and know
Decline	993	104	9.6	music performance

3.2. Most Influential Members and Topic Interaction Analysis by Developmental Stage

To determine the optimal number of topics of the whole Pressible database, the perplexity values of models were calculated. The LDA topic training model was constructed based on 10000 documents with the range of 2 to 50 topic numbers. The other 1146 documents are used to test the model with the calculation of perplexity and entropy. Based on the perplexity of testing data, 30 is the optimal topic number for this dataset.

During the developmental stage, the library staff was the most active members in the network. Their online discussions focused on the topics: "video game, education", indicating that library staff was using Pressible as a communication tool to share thoughts and discuss education media.

During the stable growth stage, a TC professor (ID: 1490) from the music education program built his courses on Pressible for three years (2011 to 2013), and he continuously received the highest eigen centrality score for three years. During the stable growth stage, the popular topics became focused on education. People who talked about "think and know" were also interested in "video game" at this stage.

During the rapid growth stage, the professor with ID 3132 brought new students into this blogging system though his courses *Creativity & Problem Solving in Music Education*. It was a course extended from the materials developed by the professor with ID 1490, with the same topic "read" and high-frequency words "music, read" for most of the posts. This was the pedagogy course to meet the New York State and national teacher preparation standards.

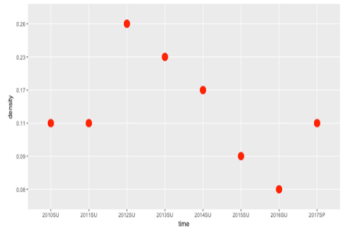


Figure 1. Social Network Density by Year.

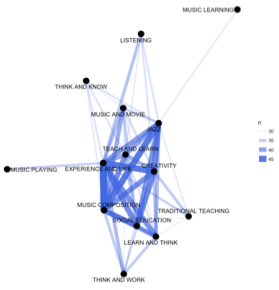


Figure 2. Topic Co-occurrence frequency in the rapid growth stage

Individuals' topic co-occurrence indicated a robust network in the rapid growth stage (Fig. 2). People talked about the topics of "creativity", "music composition", "Jazz", "social education", "learn and think", "experience and life" and "teach and learn" at high co-occurrence frequencies (above 30). In the decline stage, the topic co-occurrence network dropped in topic connection intensity which might be due to less active members in the overall network (Table 1). This finding indicated that more active members encouraged online discussions with more diverse topics. In course blogs, students extended discussion topics to the perspectives that they care about: "music learning, music playing, social education, creativity and experience and life", beyond the scope of the professor's set topic "read".

4. IMPLICATIONS

This study identifies and explores four developmental stages of the social network: development, stable growth, rapid growth, and decline. The SNA and topic model analysis results imply that the influential people will bring new communities into the social network by sharing the content of the hottest topics. Deliberately recruiting more influential people into the social network would accelerate its transition from the stable growth stage to the rapid growth stage.

5. REFERENCES

[1] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3(Jan), 993-1022

[2] Csardi, G., & Nepusz, T. (2006) The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006.

[3] Daniel, M., Messing, S., Nowak, M., & Westwood, S. J. (2010) Social Network Analysis Labs in R. Stanford University

[4] Feinerer, I., & Hornik, K. (2015). tm: Text Mining Package. R package version 0.6-2.

[5] Grün, B., & Hornik, K. (2011). "topicmodels: An R Package for Fitting Topic Models." Journal of Statistical Software, 40(13), pp. 1-30

[6] Kizilcec, R. F., Pérez-Sanagustín, M., & Maldonado, J. J. (2017). Self-regulated learning strategies predict learner behavior and goal attainment in Massive Open Online Courses. *Computers & Education*, *104*, 18-33.

[7] Lang, C. (2017) HUDK 4051: Learning Analytics: Process and Theory. Columbia University. New York

[8] Rosé, C. P., Carlson, R., Yang, D., Wen, M., Resnick, L., Goldman, P., & Sherer, J. (2014). Social factors that contribute to attrition in MOOCs. In Proceedings of the first ACM conference on Learning (pp. 197-198). ACM

[9] Siemens, G. (2004). Connectivism: A learning theory for the digital age. elearnspace. Retrieved December 12, 2007, CHI '00. ACM, New York, NY, 526-531

[10] Silge, J., and Robinson, D. (2017) "Text Mining with R: A Tidy Approach" O'Reilly Media

[11] Slater, S., Joksimovic, S., Kovanovic, V., Baker, R., & Gasevic, D. (2017) Tools for Educational Data Mining: A Review. Journal of Educational and Behavioral Statistics. 2017, Vol. 42, No. 1 p85-106

[12] Vygotsky, L. (1978). Mind and society: The development of higher psychological processes. Cambridge, MA: Harvard University Press.

[13] Wang, X., Wen, M., & Rosé, C. P. (2016, April). Towards triggering higher-order thinking behaviors in MOOCs. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (pp. 398-407). ACM

[14] Wild, F., & Sigurðarson, S. E. (2011). Simulating learning networks in a higher education blogosphere–at scale. In European Conference on Technology Enhanced Learning (pp. 412-423). Springer Berlin Heidelberg

[15] Wild, F. (2016). Learning analytics in R with SNA, LSA, and MPIA. Springer.

[16] Zhou Z. (2013) Connecting Teacher Bloggers: Unleashing the Educational Power of Wordpress

Supporting the Encouragement of Forum Participation

Aashna Garg
Stanford University
aashna94@stanford.edu

Andreas Paepcke
Stanford University
paepcke@cs.stanford.edu

1. INTRODUCTION

Since 2011 many courses in universities have used the Piazza forum facility for their internal courses. As [4] and others have shown, forum participation can be beneficial for all students. Are there potential intervention points in time during an academic period when encouragements (e.g. [6]) might strengthen forum use?

We analyze the forum contribution rates during forty offerings of 12 college courses, reaching back to 2011. By 'offering' we denote a course taught during a particular quarter. Multiple offerings of the same course afforded us longitudinal observations. We looked for significant leaps in forum contribution, hypothesizing that those times are important intervention points.

From a social graph model of forum activity we computed weighted out degrees and page ranks for each student. The out degrees reflect the number of posts a student contributes. We performed change point analyses through bootstrap procedures over the CUSUM data of the post contributions through each quarter. We thereby identified significant week by week changes in the rate at which the top ten percent of forum contributors post messages. We hypothesize that such change points might be appropriate encouragement opportunities, and we find that sudden rate shifts do occur along, sometimes in regular patterns, primarily in science and engineering courses. We propose and demonstrate the use of control charts to monitor forum traffic, and show how the historic data can be used to provide personalized encouragement messages.

2. ANALYSIS

Our question was "when would be good times during a course for encouraging students who lag behind in forum contributions?" We hypothesize that times when the top 10% of contributors speed up, other students might be encouraged to do the same.

The results indicate that week six is particularly likely to experience changes in posting rates. The studied university operates on a quarters schedule, we can hypothesize that the traffic is related to midterms. Week eight might be related to final projects coming due not too far out. Note, however, that the number of offerings (shown in parentheses with the course names) differ across courses. Thus 20% for urban studies means only one of five offerings experienced a significant posting acceleration. The only courses with an

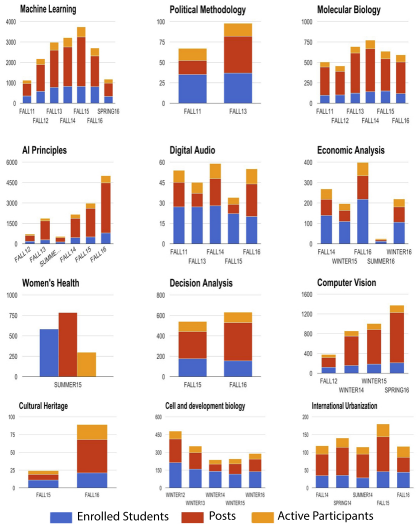


Figure 1: Number of students enrolled in the course, number of students active in the forum, and number postings.

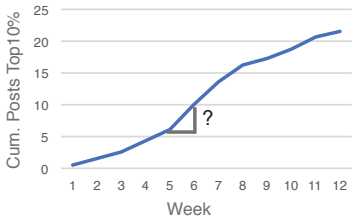


Figure 2: Cumulative number of postings by top 10% contributors throughout an academic quarter (machine learning class).

appreciable number of change points among their offerings are engineering courses. The humanities and social sciences, while using forum facilities, have not included the forum as a central discussion hub. The numbers of students attending these courses are also smaller than the science/engineering classes.

The forum change point computations we outlined above require data from all weeks of an offering to be available. In the presence of historic data this requirement is not a problem. But what could an instructor do to discover unusual posting frequencies while the offering is running?

2.1 Control Charts for Forum Alerts

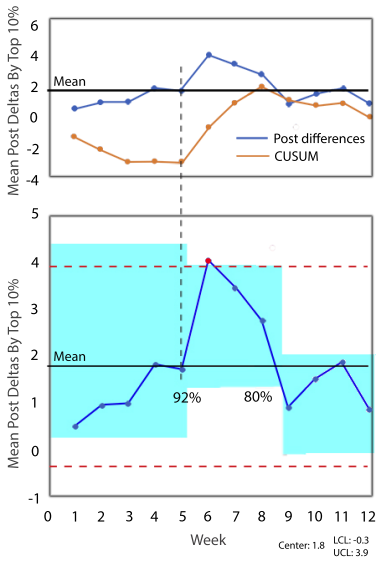


Figure 3: Change points in forum posting time series (machine learning class).

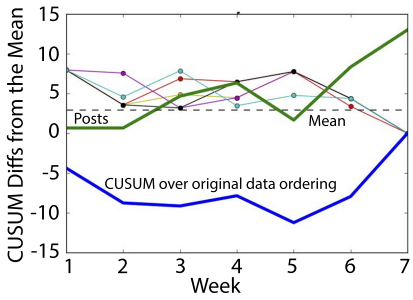


Figure 4: CUSUMs from permuted weekly post contribution rates hover around the mean, green line indicating the number of posts.

Figure 3's lower chart hints at a possibility we have not yet discussed. The horizontal dotted lines are *process limits*, a term from process control practice [3]. The limits bound the values within which a process is expected to vary. For industrial processes the variations might lie around a known optimal operating level, such as a temperature. When no such level is known from the domain, a mean can be used. Our process limits denote $2 * \sigma$ distance from the mean.

In the context of instructional forum use it would be possible to detect points, such as the one above the upper control limit in Figure 3. Such change to above-normal might indicate confusion among the students, or the discovery of an exciting topic. Either way, the instructor's attention might be warranted, as might be pointing passive students to the increased activity.

2.2 Personalized, Quantitative Encouragement

The data framework we discussed could also be deployed to provide personalized encouragement for passive students. Rather than admonishing students for past passivity, a forward-looking nudge could be provided. Figure 5 illustrates this option. The dot represents one student and their contribution as of week six: two postings. Two of many possible options are shown in the Figure for catching up to the top-

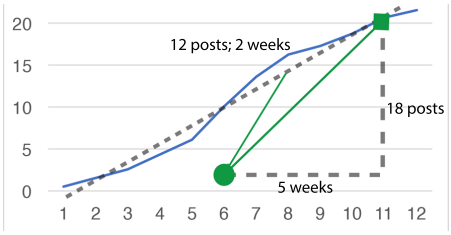


Figure 5: Personalized, quantitative nudges: catching up with the top-10% within five weeks, or two weeks.

10% contributors. These, and other options are derived by drawing a line from the student's position to an intersection with the regression line. That line is known from past course offerings. The larger the slope of the connecting line, the more aggressive the plan for catching up. For example, the long line would catch the student up within five weeks, assuming a weekly rate of $(20-2)/5 \approx 4$ messages per week.

Alternatively, the shorter line would call for six messages per week, to catch up within two weeks. The square at the end of the long option is intended to represent a slider that the student could run along the regression line to make a plan. The number of required weekly messages would be updated continuously as the student operates the slider. While this sketch is not the intended user interface, it illustrates the ideas of using past and current forum data to provide (*i*) forward-looking encouragement in place of recrimination, and (*ii*) to empower the student by personalizing the message, and providing a tool for planning. Studies are needed to determine whether postings in response to even such personalized messages prove beneficial.

For discussions on the impact of forum participation, see [6, 2, 5], and [1].

3. CONCLUSION AND FUTURE WORK

The most important next step is to test whether existing change points truly are effective encouragement moments. We also plan to use page rank measurements to see TA work sharing patterns.

4. REFERENCES

[1] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, and J. Leskovec. Engaging with massive online courses. *CoRR*, abs/1403.3100, 2014.

[2] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong. Learning about social learning in moocs: From statistical analysis to generative model. *CoRR*, abs/1312.2159, 2013.

[3] Department of Commerce. Nist/sematech e-handbook of statistical methods. Electronic book, 2012.

[4] J. Huang, A. Dasgupta, A. Ghosh, J. Manning, and M. Sanders. Superposter behavior in mooc forums. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 117–126, New York, NY, USA, 2014. ACM.

[5] D. Yang, M. Wen, and C. Rose. Peer influence on attrition in massive open online courses, 2014.

[6] M. Yeomans and J. Reich. Planning prompts increase and forecast course completion in massive open online courses. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 464–473, New York, NY, USA, 2017. ACM.

Untangling The Program Name Versus The Curriculum: An Investigation of Titles and Curriculum Content

R. Wes Crues
University of Illinois
Dept. of Educational Psychology
1310 South Sixth Street
Champaign, Illinois
crues2@illinois.edu

ABSTRACT

This investigation focuses on the relationship between skills taught during business programs and whether the skills taught relate to the title of the program, as deemed by subject-matter experts. We hone-in on formal degree and non-degree programs in small business education, entrepreneurship education, or a blend of these two to determine if the *name* of the program is related to the *skills taught* in said program. We use a collection of excerpts from college catalogs, which are all descriptions of the formal academic programs. We then use $k-$means clustering to group program descriptions into interpretable clusters. We discuss the findings from the cluster analysis.

Keywords

text mining, clustering, higher education, business education

1. INTRODUCTION

Major academic disciplines are typically collections of finer-grained specialties; for example, a computer science department might consist of experts in human-computer interaction, artificial intelligence, algorithm design, among others. Colleges likely have departments with similar names, but we want to understand if similarly named degree programs at different universities equip students with similar skills. To discern whether or not this task is tractable, we used a collection of program descriptions from college catalogs about programs claiming to teach students entrepreneurship, small business, or a blend between these two curriculum areas. These definitions are used throughout:

- A *program description* is at least one, but often composes a few paragraphs, which delineates skills taught in programs, and might provide some learning goals and a listing of courses;
- *Entrepreneurship* is defined as " trying to identify opportunities and putting useful ideas into practice" [1] (p. 6);
- and, *small business management* is "the ongoing process of owning and operating an established business" [3] (p. 28).

Our study explores whether we can use text clustering to identify a clear distinction between these two areas of business education, determine if there are differences between two-year and four-year programs, and whether there are differences between degree and non-degree programs.

2. METHOD

A research team manually assembled a collection of 697 program descriptions from college catalogs for institutions located in the United States. Research assistants went to college websites and manually extracted text from published college catalogs online. The initial list of programs was derived from the 2013 Integrated Postsecondary Education Data System (IPEDS) maintained by the United States Department of Education. After filtering institutions which did not have any business programs, a random sample of programs arrived at the collection used.

Program descriptions spanned programs focusing in entrepreneurship, small business management, or a blend of the two. Additional program descriptions were collected which were considered special focus programs; these were programs which teach a specific skill set on operating a business (examples include funeral home management to hair weaving and braiding entrepreneur). We also considered formal degree (e.g., associates and bachelor degrees) or non-degree programs (e.g, certificates or specializations), and whether the home institution is public or private, for-profit or not-for-profit, and whether the institution is a 2-year, 4-year, or 4-year and beyond institution [5]. Table 1 presents the distribution of program labels and whether the program is a degree or non-degree program.

Table 1: Distribution of Program Descriptions

Program Label	N	Degree/Non-Degree
Entrepreneurship	444	247/197
Small Business & E-ship	82	42/40
Small Business	79	20/59
Special Focus	92	34/58

2.1 Preprocessing Program Descriptions

Program descriptions were transformed into raw text format, tokenized into unigrams, except for a few words. A few bigrams and trigrams were specified using knowledge from a domain-expert, for example, business plan(s), social entrepreneurship, home based business, and venture capital. Punctation, numbers, and top words were removed using the pre-defined English stop word list in the "tm" package in R [2]. We used stemmed words by using the Porter stemming algorithm [6]. We used binary indicators to determine whether a term was present in each program description when constructing the document-term matrix [4].

2.2 Corpus Statistics

Our initial document-term matrix contained 7799 unique terms with a sparsity of 99%. We removed very frequent terms deemed to have no substantive value by a domain expert. Due to the nature of the corpus (i.e., program descriptions), words such as catalog, college, semester, requirements, and introduction, among others, were excluded. Eventually, we used the "removeSparseTerms" function in the "tm" package in R [2], which resulted in a document-term matrix with 16 unique terms, however, still 70% sparse.

2.3 Program Description Clustering

We utilized $k-$means because this clustering technique was favored in prior studies [7]. We experimented with various numbers of centroids, and after discussions with domain experts, we determined $k = 10$ was an optimal solution. The domain expert believed this solution provided an interpretable and reasonable grouping of programs. Specifically, the distribution of whether the program was an entrepreneurship, small business, a blend of these, or a special focus program, coupled with their expectations of distribution of formal degree programs versus non-degree programs. More than ten centroids resulted in clusters containing less than five documents, while less than ten resulted in a solution which did not provide what domain experts believed to be the most interpretable.

3. RESULTS

Five of the clusters exhibited a focus on teaching entrepreneurship in the context of having an idea, creating a start-up, with the intention of scaling the business into a large enterprise. Within these clusters, two clusters had words indicating programs might teach entrepreneurship to equip students to solve global problems and health concerns. Words indicating entrepreneurship might be taught to professionals in fields besides business (i.e., law and engineering) appeared in one cluster. One cluster appeared to teach general business skills, without a clear focus on entrepreneurship or small business. Another cluster contained special focus programs, which seek to prepare students for a specialized, technical career, such as a travel agent or carpenter. Two clusters contained small business programs, where one focused on keenly on running ones' own business, while the other included this while teaching students to innovative. One cluster contained very detailed program descriptions from one institution.

4. DISCUSSION & CONCLUSIONS

We found the definition of entrepreneurship which pertains to creating and expanding new enterprise appeared to be almost exclusively in four-year colleges, especially research universities. In contrast, small business management and operating a small business were taught almost exclusively at two-year colleges. A few of the two-year colleges also had many specialized programs in applied fields, such as the cosmetology; these types of programs were nearly exclusive to two-year colleges. Another element of entrepreneurship is creativity and innovation. These skills, specifically innovation, seemed to be taught primarily in the four-year sector. The programs that considered themselves a blend tend to focus more on small businesses than entrepreneurship. We found innovation and these skills to be taught more in degree. On the other hand, skills related to managing a small business were in non-degree programs.

From our findings about entrepreneurship and small business education, we generally found labels of programs match the skills one would expect to learn given the name of the program. However, one cluster in our analyses did not indicate skills in the targeted areas were being specifically taught. A limitation of our study is program descriptions vary in length and detail, which might be problematic for clustering. Our further work plans to consider whether skills taught have changed over time; for example, are skills being taught today the same skills taught a decade ago?

5. ACKNOWLEDGMENTS

The author would like to acknowledge two domain experts, Dr. Cindy Kehoe and the late Dr. Paul Magelli for their expertise in entrepreneurship education. Their advice about contextual meaning of results was invaluable in interpreting these analyses. The author would also like to acknowledge the Ewing Marion Kauffman foundation, which funded this work through a grant to inventory and interpret entrepreneurship education in higher education in the United States.

6. REFERENCES

[1] B. R. Barringer and R. D. Ireland. *Entrepreneurship: Successfully launching new ventures*. Pearson, Upper Saddle River, New Jersey, fourth edition, 2012.

[2] I. Feinerer, K. Hornik, and D. Meyer. Text mining infrastructure in r. *Journal of Statistical Software*, 25(5):1–54, 2008.

[3] T. S. Hatten. *Small business management: Entrepreneurship and beyond*. Houghton Mifflin Company, Boston, Massachusetts, fourth edition, 2009.

[4] P. Howland and H. Park. Cluster-preserving dimension reduction methods for efficient classification of text data. In M. W. Berry, editor, *Survey of Text Mining*, pages 3–24. Springer Science+Business Media, 2004.

[5] National Center for Education Statistics. *IPEDS Glossary*, 2017.

[6] M. F. Porter. An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 40(3):211–218, 2006.

[7] M. Steinbach, G. Karypis, V. Kumar, et al. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, volume 400, pages 525–526. Boston, 2000.

Emerging Patterns in Student's Learning Attributes through Text Mining

Kejkaew Thanasuan
Learning Institute
King Mongkut's University of Technology, Thonburi[1]
(+662) 470-8395
kejkaew.tha@kmutt.ac.th

Warasinee Chaisangmongkon
Institute of Field Robotics
King Mongkut's University of Technology, Thonburi[1]
(+662) 470-9716
warasinee.cha@kmutt.ac.th

Chanikarn Wongviriyawong
Institute of Field Robotics
King Mongkut's University of Technology, Thonburi[1]
(+662) 470-9717
chanikarn@fibo.kmutt.ac.th

ABSTRACT
Text mining has been used in various fields including education. Using unsupervised sentiment analysis combined with a clustering algorithm, we discovered 2 emerging clusters of learning characteristics (traditional (T) and experiential (E)), and correlations among learning attitudes such as motivation, peer relationship and positive attitude. We found a positive correlation between social learning and peer relationship ($p<0.005$), but negative between social learning and negative attitude ($p<0.05$) in E. Social learning was positively correlated with positive attitude ($p<0.001$) in T.

Keywords
Text mining, clustering algorithm, sentiment analysis, motivation, engagement

1. INTRODUCTION
Studies have shown that attitudes are related to motivation, engagement and outcome in learning. When learners have positive attitude, they would spend more time engaging in learning [5, 9]. Difference in students with positive attitude and motivation in e-learning settings was observed [6]. Students with boredom have poorer learning outcome than those with frustration [1]. Hence, sentiment analysis could be used to harness learning attitudes.

Recently, machine learning methods in natural language processing have become prevalent, while there are many training datasets for supervised learning algorithms. However, the task of opinion mining without such dataset can be a challenge. We combined one symbolic technique for an unsupervised machine learning with clustering algorithm to discover emerging patterns among texts written in *Thai* that could reflect student's learning attitudes. Our findings demonstrated how such approach could be useful in exploring and understanding relationships among learning attitudes.

2. METHODS
2.1. Data Acquisition
Our subjects were 83 freshman undergraduate students (M:F = 62:21) (average age = 17.2) in Robotics and Automation Engineering, at King Mongkut's University of Technology Thonburi. They consented to participate in the study. This data set was collected while students were taking same classes. Students wrote in *Thai* about what they learned each week for all 14 weeks.

2.2. Data Analyses
We used an open source Lexitron dictionary (NECTEC, 2006) as word database in *Thai* and an open source algorithm Lexto (NECTEC, 1994) to tokenize texts into longest words possible. We had 383 entries. On average, each entry had 124.3 words.

Word frequency was calculated for each student as the ratio of the number of times each unique word appeared in any learning journal and the total number of words appeared. Irrelevant words (prepositions, conjunctions, and generic verbs and nouns) or words that appeared less than 20 times in all entries were filtered out. Negation and irrealis phenomena, out-of-topic sentences, or irony and sarcasm were not treated in our analysis. We performed several clustering algorithms on the distance matrix with various initial conditions and different number of clusters (2, 3, or 4) to determine if any pattern of word clusters could emerge.

Among frequently-used words, instructors chose words that represented these six attitudes: 1) positive relationship with others (Peer relationship), 2) desire to improve oneself (Motivation), 3) positive emotion (Positive attitude), 4) negative emotion (Negative attitude), 5) engagement in learning on one's own (Solitary learning), and 6) engagement in learning that involves others (Social learning). The associated words were also evaluated by another group of students to indicate levels of congruity of each attitude[2]. The results are shown in Table 1. We calculated a student's attitude score to be the sum of percentage of word frequency for each word associated with each of the 6 attitudes. Pearson correlation coefficient and p-value of the correlation were computed between any two attitudes. Correlation analyses were performed independently for each cluster.

3. RESULTS AND DISCUSSION
We found that 2 clusters emerged, yielding the most consistent set of words. The first cluster contained words such as take exams, read books, problem sets, formula, lessons, math, writing, calculus, physics, language, etc. The second contained words such as human being, people, work, see, team, fun, talk, play, like,

[1] King Mongkut's University of Technology, Thonburi's address: 126 Pracha Uthit Rd, Bang Mot, Thung Khru, Bangkok 10140 Thailand

[2] The data were collected from 28 native Thai speakers (average age = 20.18). They were asked to rate how each pair of words and an attitude was meaningfully or semantically related (e.g. Peer Relationship vs. Group) in a 5-point Likert scale.

group, together, etc. The first cluster was labelled T for traditional and the second, E for experiential. Although initial conditions and clustering algorithms were varied, these two clusters emerged.

Table 1. Words Associated with 6 attitudes and their rating[3] (mean score and standard deviation in parentheses)

Attitude	Associated Words	Rating
Peer Relationship	group, talk, help, together, team, help each other, we, everyone, etc.	3.87 (0.48)
Motivation	improve, practice, better, goals, development, improvement, etc.	3.76 (0.4)
Positive Attitude	fun, enjoy, like, happy, funny, good, excited, etc.	3.64 (0.37)
Negative Attitude	stressed, confused, sleepy, slow, difficult, do not understand, etc.	3.01 (0.45)
Solitary Learning	exams, formula, scores, books, grades, study, responsibility, etc.	3.34 (0.77)
Social Learning	hands on, experiment, project, communication, participate, etc.	3.71 (0.31)

Motivation was positively correlated with solitary learning ($R=0.4$ (T) and 0.55 (E); $p<0.05$). It could mean that for T, when one desires to improve oneself, one engages in learning even on one's own. Our result supports a previous finding that motivation and engagement were correlated [3, 10]. Such correlation for E might be because when one enjoys learning with others, their motivation increases. Previous studies showed that people who reported feeling happy were engaged in social activities more often and that sociability was a strong predictor of life satisfaction [2, 7].

Additionally, for E, motivation was positively correlated with social learning ($R=0.42$, $p<0.05$); social learning was positively correlated with peer relationship ($R=0.6$, $p<0.005$), but negatively correlated with negative attitude ($R=-0.44$, $p<0.05$). For T, social learning was positively correlated with positive attitude ($R=0.55$, $p<0.001$). Relationships with peers are very important in helping learners become adaptive in different learning environments [8]. Previous studies showed that students with positive peer relationship were likely to be engaged in academic tasks and perform better in school than students without positive peer relationships [11, 12, 13]. Our finding supports existing literature that learning abilities are related to attitude of learners [5].

However, our approach has some limitations. Our algorithm is a simple frequency counting. However, since less frequently used words have been filtered out, we expected that our results would still be robust even with different weighting methods. Moreover, no sarcasm, negation or irrealis phenomena were considered. This might have a slight effect on our results.

Future work involves testing robustness of our approach with more data. To explore additional emergence, we could also apply adjustments to various clustering algorithms [4]. We are developing a platform to help teachers quantify student's attitudes.

4. REFERENCES

[1] Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., and Graesser, A. C. 2010. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*. 68, 4, 223-241.

[2] Costa, P. T. and McCrae, R. R. 1980. Influence of extraversion and neuroticism on subjective well-being: happy and unhappy people. *Journal of personality and social psychology*. 38, 4 (Apr. 1980), 668-678.

[3] Hsieh, T. L. 2014. Motivation matters? The relationship among different types of learning motivation, engagement behaviors and learning outcomes of undergraduate students in Taiwan. *Higher Education*. 68, 3, 417-433.

[4] Li, G. and Liu, F. 2012. Application of a clustering method on sentiment analysis. *Journal of Information Science*. 38, 2, 127-139.

[5] McMillan, J. H. 1977. The effect of effort and feedback on the formation of student attitudes. *American educational research journal*. 14, 3, 317-330.

[6] Moshinskie, J. 2001. How to keep e-learners from escaping. *Performance Improvement*. 40, 6, 30-37.

[7] Robinson, J. P. and Martin, S. 2008. What do happy people do?. *Social Indicators Research*. 89,3, 565-571.

[8] Rubin, K. H., Bukowski, W., and Parker, J. G. 1998. Peer interactions, relationships, and groups. *Handbook of child psychology*. 3, 5, 619-700.

[9] Sanderson, H. W. 1976. Student attitudes and willingness to spend time in unit mastery learning. *Research in the Teaching of English*. 10,2, 191-198.

[10] Walker, C. O., Greene, B. A., and Mansell, R. A. 2006. Identification with academics, intrinsic/extrinsic motivation, and self-efficacy as predictors of cognitive engagement. *Learning and Individual Differences*. 16, 1, 1-12.

[11] Wentzel, K. R. 2005. Peer relationships, motivation, and academic performance at school. In *Handbook of competence and motivation,* A. J. Elliot and C. S. Dweck, Eds. Guilford Press, New York, 279-296.

[12] Wentzel, K. R., Barry, C. M., and Caldwell, K. A. 2004. Friendships in Middle School: Influences on Motivation and School Adjustment. *Journal of educational psychology*. 96, 2 (Jun. 2004), 195-203.

[13] Wentzel, K. R. and McNamara, C. C. 1999. Interpersonal relationships, emotional distress, and prosocial behavior in middle school. *The Journal of Early Adolescence*, 19, 1, 114-125.

[3] For rating, a five-point score means strongly agree and an one-point score means strongly disagree.

A Neural Network Approach to Estimate Student Skill Mastery in Cognitive Diagnostic Assessments

Qi Guo, Maria Cutumisu, Ying Cui
Department of Educational Psychology, University of Alberta
{qig, cutumisu, yc}@ualberta.ca

ABSTRACT

In computer-based tutoring systems, it is important to assess students' mastery of different skills and provide remediation. In this study, we propose a novel neural network approach to estimate students' skill mastery patterns. We conducted a simulation to evaluate the proposed neural network approach and we compared the neural network approach with one of the most widely used cognitive diagnostic algorithm, the DINA model, in terms of skill estimation accuracy and the ability to recover skill prerequisite relations. Results suggest that, while the neural network method is comparable in skill estimation accuracy to the DINA model, the former can recover skill prerequisite relations more accurately than the DINA model.

Keywords

prerequisite discovery, skills, neural network, student modeling, cognitive diagnosis model

1. INTRODUCTION

In intelligent tutoring systems, assessing students' skill mastery patterns and determining skill prerequisite relationship are two important areas of research. Various approaches are proposed to solve these two problems, including Educational Data Mining (EDM) approaches, such as Bayesian Knowledge Tracing, Learning and Performance Factor Analysis (for a comparison see [5]), and psychometric approaches, such as Cognitive Diagnostic Models (CDMs) [2, 6]. Compared to CDMs, which assess student skill mastery based on their responses to a test administered at one time point (i.e., no learning occurs during the test), the EDM approaches have the advantage of assessing student learning dynamically. However, unlike CDMs, which estimate every item's psychometric properties, the EDM approaches often assume all test items that measure the same set of skills have the same psychometric properties (e.g., same guessing and slipping parameters). This assumption is unlikely to be tenable in practice, and it may lead to less accurate skill estimation and less efficient item selection. While both approaches have their strengths and weaknesses, this study will focus on developing a new CDM approach using the neural networks, and evaluate the proposed approach by comparing it with the current most popular CDM method, the DINA (deterministic inputs, noisy "and" gate) model [2] using simulated data.

2. A BRIEF INTRODUCTION TO NEURAL NETWORKS

A neural network is a supervised classification algorithm that consists of several layers of neurons (i.e., processing units) [4]. Each neuron linearly combines information from previous layers and applies a non-linear *activation* function. The most commonly-used activation function is the logistic/sigmoid function. A typical feedforward neural network consists of a layer of hidden units and a layer of output units. Mathematically, it can be represented as:

$$Y_{n,q} = sigmoid\left(\vec{1}_{n,1}\vec{b}'_{1,q} + sigmoid(\vec{1}_{n,1}\vec{b}'_{1,k} + X_{n,p}W_{p,k})W_{k,q}\right),$$

where $Y_{n,q}$ is the output matrix consisting of n subjects' values on q output variables, $X_{n,p}$ is the input matrix consisting of n subjects' values on p input variables, $\vec{b}'_{1,k}$ is a vector of intercept values for k hidden units, $W_{p,k}$ is the weight matrix between p input variables and k hidden units, $\vec{b}'_{1,q}$ is a vector of intercept values for q output units, and $W_{k,q}$ is the weight matrix between k hidden units and q output units.

One challenge in applying neural networks to estimate students' skill mastery patterns is that students' skill mastery patterns are unobserved. Thus, we only have observed values for the input variables (students' item response patterns) but not for the output variables (students' skill mastery pattern).

3. METHODOLOGY: THE PROPOSED NEURAL NETWORK APPROACH

To overcome the problem mentioned above, we propose a novel neural network model that has the same input and output (i.e., students' item response patterns). The core idea underlying our approach is to first reduce the input (student item response patterns) to a smaller number of hidden units representing students' latent skills and then use these hidden units to best reproduce student item response vectors (i.e., output) with the restriction of the Q-matrix, a matrix that specifies the set of skills measured by each item. A conceptual diagram of the proposed network is shown in **Figure 1**.

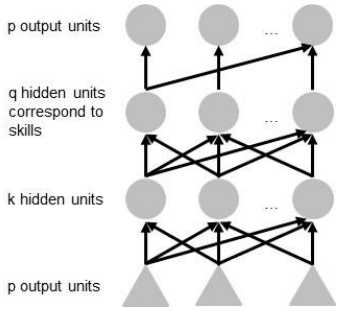


Figure 1. A diagram of the proposed neural network. Relations between skill hidden units and output units are specified based on the Q-matrix.

It is important to note that the relation between the second layer of hidden units and output units is specified based on the Q-matrix, which specifies which skills are required by each item. Intuitively, the network first extracts features from student item response patterns and then it dictates the relations between features and student item response patterns based on the Q-matrix. Mathematically, the model can be represented as follows:

$$Y_{n,p} = sigmoid\left(\vec{1}_{n,1}\vec{b}'_{1,p} + sigmoid\left(\vec{1}_{n,1}\vec{b}'_{1,q} + sigmoid(\vec{1}_{n,1}\vec{b}'_{1,k} + X_{n,p}W_{p,k})W_{k,q}\right)W_{q,p} \odot Q'_{q,p}\right),$$

where $\odot$ represents elementwise multiplication, and $Q'_{q,p}$ is the Q-matrix.

Similar to a regular neural network, the proposed model uses maximum likelihood to define the cost function and it can be optimized using some variants of gradient descent (e.g., rprop [4]). To speed up the optimization, it is important to choose meaningful starting values for the weight matrices. To initialize $W_{q,p}$, we can first train a multivariate logistic regression with all the theoretically possible skill patterns (i.e., expected theoretical plausible skill patterns) as input, and their corresponding expected item response patterns (i.e., item response pattern assuming no slips and guesses) as output, assuming slipping and guessing parameters are 0. Then, we use the weight matrix from this multivariate logistic regression as the starting values of the proposed neural network.

4. EVALUATION

In order to demonstrate the accuracy of the proposed neural network, we conducted a preliminary simulation study. Five thousand students' responses (correct/incorrect) to 28 test items were generated based on a skill prerequisite model shown in Figure 2. Skill prerequisite relations, true model used in the simulation (left); recovered using DINA skill estimates (middle) and neural network skill estimates (right)

To evaluate the recovered prerequisite relationship, we counted the number of estimated causal links that were not in the true model, and the number of missing causal links that were in the true model.

and a Q-matrix (available upon request). The guessing and slipping parameters for all items were set to 0.1. We compared the proposed method with the DINA model in terms of accuracy of 1) student skill pattern estimates and 2) skill prerequisite relation recovery. Accuracy of skill pattern estimates is defined as:

$$accuracy = 1 - \frac{|estimated\ skill\ pattern\ matrix - true\ skill\ pattern\ matrix|}{n * q},$$

where n is the sample size, and q is the number of skills in the Q-matrix. The skill prerequisite relations were recovered by using a Bayesian network to model the relations among estimated student skills. The causal direction in the Bayesian network is determined by the following heuristic [1]:

If P(skill1=0)<P(skill2=0), then skill1 is the prerequisite of skill2.

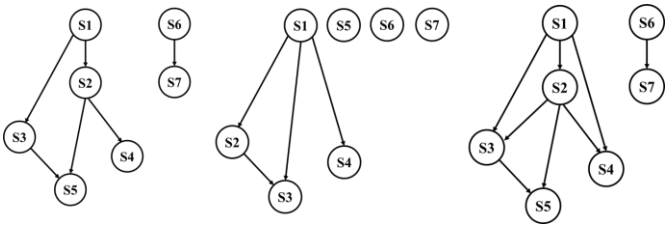


Figure 2. Skill prerequisite relations, true model used in the simulation (left); recovered using DINA skill estimates (middle) and neural network skill estimates (right)

To evaluate the recovered prerequisite relationship, we counted the number of estimated causal links that were not in the true model, and the number of missing causal links that were in the true model.

We programed our proposed neural network using Python. The number of hidden units in the first layer was set to 56. The number of hidden units in the second layer was set to seven, corresponding to seven skills in the Q-matrix. The Rprop algorithm was used to optimize the neural network. For the DINA analysis, we used the *CDM* R package [6]. For the Bayesian network analysis, we used the *bnlearn* R package's mmhc algorithm [7] and *Rgraphviz* R package [3].

The results suggested that the proposed method had similar or slightly better accuracy (89.2%) at estimating skill patterns than the DINA model (87.9%). Moreover, the proposed method was better at recovering the skill prerequisite relations. The recovered skill prerequisite relations by the DINA model and the proposed method are shown in Figure . The prerequisite relations recovered based on the DINA skill estimates only contained two arcs from the true model (i.e., S1 to S2, S1 to S3), and they contained two arcs that were not in the true model (S1 to S4, S2 to S3). The prerequisite relations recovered based on the neural network skill estimates contained all the arcs from the original model, as well as two arcs that were not in the true model (S1 to S4, S2 to S3). Overall, the results suggested that the proposed network had slightly better skill estimation accuracy than the DINA model and it was more accurate at recovering skill prerequisite relations than the DINA model.

5. CONCLUSIONS AND DISCUSSION

This study proposed a novel neural network approach to estimate student skill mastery patterns in CDM. Traditionally, parameter estimation of models with latent variables usually depends on Expectation Maximization or Markov Chain Monte Carlo methods. The proposed neural network approach frames the latent variable model problem as a supervised problem and it solves it using the gradient descent method. Initial evidence suggests that the proposed method has comparable skill estimation accuracy as the DINA model, but it can recover skill prerequisite relations better than the DINA model. Further research is needed to rigorously evaluate this method.

6. REFERENCES

[1] Chen, Y., Gonzlez-Brenes, J. and Tian, J. 2016. Joint Discovery of Skill Prerequisite Graphs and Student Models. In *International Conference on Educational Data Mining*. Raleigh, NC, IEDMS, 46-53.

[2] de La Torre, J. 2009. DINA Model and Parameter Estimation: A Didactic. *Journal of Educational and Behavioral Statistics, 34*, 1, 115-130.

[3] Gentry, J., Long, L., Gentleman, R., Falcon, S., Hahne, F., Sarkar, D. and Rgraphviz, K. H. 2009. Provides plotting capabilities for R graph objects. *R package version, 2*, 0.

[4] Goodfellow, I., Bengio, Y., Courville, A. 2016. *Deep Learning*. MIT Press.

[5] Pavlik, P. I., Cen, H., Koedinger, K.R. 2009. Performance factors analysis - A new alternative to knowledge tracing. *Proceedings of the 2009 conference on artificial intelligence in education: building learning systems that care: from knowledge representation to affective modelling* (Brighton, 2009), 531-538.

[6] Robitzsch, A., Kiefer, T., George, A. and Uenlue, A. 2014. *CDM: Cognitive diagnosis modeling.R package version 3.1-14*, .

[7] Scutari, M. bnlearn: Bayesian Network Structure Learning, R package version 2.7 (2011). *URL* http://www.bnlearn.com.

Automatic Peer Tutor Matching: Data-Driven Methods to Enable New Opportunities for Help

Nicholas Diana
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
ndiana@cmu.edu

Michael Eagle
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
meagle@cs.cmu.edu

John Stamper
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
john@stamper.org

Shuchi Grover
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
shuchi.grover@sri.com

Marie Bienkowski
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
marie.bienkowski@sri.com

Satabdi Basu
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
satabdi.basu@sri.com

ABSTRACT

The number of students that can be helped in a given class period is limited by the time constraints of the class and the number of agents available for providing help. We use a classroom-replay of previously collected data to evaluate a data-driven method for increasing the number of students that can be helped. We use a machine learning model to identify students who need help in real-time, and an interaction network to group students who need similar help together using approach maps. By assigning these groups of struggling students to peer tutors (as well the instructor), we were able to more than double the number of students helped.

Keywords

Introductory Programming; Learning Analytics; Machine Learning; Peer Tutors; Educational Data Mining

1. INTRODUCTION

While a typical classroom may be full of students experiencing the same problem and students who have solved that problem, this expertise is rarely utilized. Instead, often the only source of help is the instructor, who is most likely unable to help all the students who need help within the time constraints of the class period. To address this problem, we propose and evaluate several methods for improving the efficiency of student assistance using machine learning.

Diana et al. [1] showed that low-level log data from the Alice introductory programming environment can be used to accurately predict student grades, and that they could increase the number of students helped by matching struggling students to a peer tutor based on the similarity of their code. A subsequent study [2] found that the accuracy and interpretability of the previously reported predictive model could be improved by increasing the grain size of the features from a vocabulary of terms derived through natural language processing (NLP) to small snippets of code. We explore how this improvement impacts peer tutor matching and the efficiency of providing help more generally. Additionally, we use an interaction network graph to test if students who may benefit from the same kind of help can be grouped together, increasing the efficiency of the instructor or peer tutor.

2. METHODS

The data used in the current study were originally collected by Werner et al. [3] as part of a two year project exploring the impact of game design and programming on the development of computer science skills. The students were asked to complete an assessment task called the *Fairy Assessment*. The current experiment closely follows the data transformation methodology reported in [1] to convert raw log data into program representations called *code-states* and the code-state complexity reduction methodology reported in [2] to reduce code-states to smaller, *code-chunks*.

We used ridge regression to predict students' grades. We compared two methods for generating the features inputted into the regression. In the first method, features were a vocabulary of NLP terms generated from the students' code-states. In the second method, each code-state was first converted into a list of code-chunks, and then into a *chunk-frequency vector*. A chunk-frequency vector is a vector whose length is equal to the total number of features being considered in the model. Each value in the vector corresponds to the frequency of the respective code-chunk.

The predicted grades were also used to estimate which students need help and which students may be able to provide help. We call the students classified as needing help using their actual grades *low-performing students*. This classification serves as the ground-truth that we use to evaluate our predictive model. In a real world implementation, we would not have access to the actual grades, so we must estimate them and use those estimates to classify students as need-

ing help. If a student's predicted grade was in the bottom quartile, and they have not been helped or are not currently being helped ("helped" status persists across time), then that student was added to the group of students who still need to be helped, which we call the *Help Pool*. If a student's predicted grade was in the top quartile, and they are not currently helping a student, then that student was added to the group of students who may be able to help other students, which we call the *Tutor Pool*. For each student in the *Help Pool*, we first checked to see if the instructor was available to help. If so, the instructor was assigned to that student. If the instructor was unavailable (i.e., helping another student), then we searched for a peer tutor. We used a network graph of each code-state (or code-chunk frequencies) for each user to match tutees to tutors. We searched for tutors who shared a common ancestor node (i.e., shared a previous program state) with the tutee. These tutors were added to a pool of potential tutors. From that pool we selected the tutor with the common ancestor node that was closest (i.e., least number of steps away) to the tutee's current node. The same method applied if segmenting was used, except that instead of matching the instructor or peer tutor to one student, the instructor or tutor was matched to a segment of students with a similar problem.

2.1 Efficiency Index

While the primary goal of our previous work [1] was to evaluate how well our model could correctly classify students who would go on to have a low final grade (low-performing students), the primary goal of the current experiment is to evaluate how efficient this intervention would be. That is, we were interested in what percentage of those low-performing students could be helped, and how we can maximize that percentage. We call this ratio the *Efficiency Index* (EI), and define it formally as:

$$EI = \frac{LowPerformingStudentsHelped/BeingHelped}{LowPerformingStudents} \quad (1)$$

The EI can be further broken down into the percentage of low-performing students helped by the instructor (EI_I) and the percentage of low-performing students helped by peer tutors (EI_{PT}).

3. RESULTS

We compared models using a linear mixed model with the measure of interest as the dependent variable, model as a fixed effect, and time bin as a random effect.

We hypothesized that we can use low-level programming data to group similar low-performing students together so that they can be helped as a group. To test this, we first replicated our previously reported model to use as a baseline measure. Then, we generated a new model that incorporated segmenting. Both models used NLP features in a ridge regression and an interaction network graph built using code-states as nodes. We found that the EI (M=0.467, SD=0.210) of the model that incorporated segmenting was significantly higher ($p<.001$) than the baseline model (M=0.305, SD=0.190).

We also hypothesized that using the presence or absence of code-chunks as model features would improve the performance of the model. To test this, we generated a model using a sample of the code-chunks from our previous work that were shown to be good predictors of learning outcomes [2]. We generated a model using these 16 code-chunk features (rather than the NLP-derived terms used in the baseline model), and found that this code-chunk model had a significantly lower ($p<.001$) RMSE (M=0.246, SD=0.064) than the baseline model (M=0.263, SD=0.073).

Finally, we hypothesized that a network graph generated using code-chunks as nodes would lead to greater coverage and a higher EI. To test this, we generated a model using the same 16 code-chunks described above as features in the regression. A network graph was also generated to incorporate segmenting. However, instead of each node corresponding to a code-state, each node corresponded to a chunk-frequency vector. Representing nodes as chunk-frequency vectors more than doubled the coverage (coverage=0.924) compared to the network graph generated using code-states (coverage=0.374). The EI of the model using chunk-frequency vectors to generate the network graph (M=0.813, SD=0.128) also had a significantly higher ($p<.001$) EI than the model using code-states (M=0.428, SD=0.217).

4. CONCLUSIONS

In this paper, we explored a method for increasing the amount of help given in a typical class period. Our previous work demonstrated that we can use a predictive model to accurately identify students who may need help. We built off of this work in two ways. First, we improved the accuracy of the predictive model by using more relevant features. Second, we drastically increased the number of students able to be helped from, on average, 3.72 to 9.92 by grouping low-performing students together to be helped as a group (in combination with better model features). These results suggest that using low-level log data to group and match low-performing students to peer tutors may be an effective way to increase the amount of help given in a classroom.

5. REFERENCES

[1] N. Diana, M. Eagle, J. Stamper, S. Grover, M. Bienkowski, and S. Basu. An instructor dashboard for real-time analytics in interactive programming assignments. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 272–279, New York, NY, USA, 2017. ACM.

[2] N. Diana, M. Eagle, J. Stamper, S. Grover, M. Bienkowski, and S. Basu. Data-driven generation of rubric parameters from an educational programming environment. Submitted.

[3] L. Werner, J. Denner, and S. Campe. The Fairy Performance Assessment : Measuring Computational Thinking in Middle School. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, pages 215–220, 2012.

Short-Answer Responses to STEM Exercises: Measuring Response Validity and Its Impact on Learning

Andrew Waters
OpenStax, Houston, TX
aew2@rice.edu

Phillip Grimaldi
OpenStax, Houston, TX
pjg3@rice.edu

Andrew Lan
Princeton University,
Princeton, NJ
andrew.lan@princeton.edu

Richard Baraniuk
Rice University, Houston, TX
richb@rice.edu

ABSTRACT

Educational technology commonly leverages multiple-choice questions for student practice, but short-answer questions hold the potential to provide better learning outcomes. Unfortunately, students in online settings often exhibit little effort when crafting short-answer responses, instead often produce off-topic (or invalid) responses that are off-topic and do not relate to the question being answered. In this study, we consider the effect of entering on-topic short-answer response on student learning and retention. To do this, we first develop a machine learning method to automatically label student open-form responses as either valid or invalid using a small amount of hand-labeled training data. Then, using data from several high school AP Biology and Physics classes, we present evidence that providing valid short-answer responses creates a positive educational benefit on later practice.

Keywords

Best educational practices, Cognitive psychology, Machine learning, Natural language processing, Mixed effect modeling

1. INTRODUCTION

An important part of the learning process is recalling learned information from memory [3]. In most educational situations, this practice is accomplished by asking students practice questions related to the learning material. In online learning, multiple-choice questions are by far the most common, following by short-answer questions. While multiple choice questions are attractive due to the ease of machine scoring, it is worth asking whether is is the best option for improving learning. Indeed, multiple-choice questions are oft-criticized because they are perceived to require only shallow recognition processes to complete [7]. Short-answer responses, by contrast, are generally believed to have a stronger learning benefit to students as they afford more difficult reconstructive cognitive processes.

Prior experiments examining the relative benefits of multiple-choice and short-answer have been mixed, with short-answer questions generally found to improve learning only when subsequent feedback is provided [2, 4]. One factor that has not been examined in prior research, however, is how the quality of short-answer responses provided by students contribute to learning. In online educational settings where students lack oversight, students do not always take the time to craft thoughtful short-answer responses. Instead, they often opt to to quickly enter an off-topic response to advance their progress or view feedback.

We hypothesize that students derive greater learning benefits when they produce valid short-answer responses than when they do not, even when those valid responses are incorrect. While it is possible to hand-label student responses as valid or invalid for a small number, it is not feasible to do this at large scale. To circumvent this scalability issue, we devise a machine-learning based classifier trained on a small number of hand-labeled exemplars. We then leverage this classifier to analyze the impact of entering valid responses on learning.

2. AUTOMATIC VALIDITY CLASSIFICATION

Due to the large number of words in student responses, our method for automatically classifying student short-answer responses as valid or invalid begins with parsing to reduce the overall size of the feature space. First, we attempt simple spelling correction for each word of a student's response. Following spelling correction, which strip common stopwords (e.g. of, as, is, etc) and replace any nonsensical words (e.g., random keyboard presses) with a specially defined tag, which has the effect of mapping all unknown words to the same label. Finally, we stem acceptable words in a student responses to further reduce the dimensionality of our feature space. Finally, we convert the parsed student response to a numerical feature vector using a bag-of-words model.

Following parsing, we employ a random forest [1] to classify each student response as either valid or invalid. We measured the performance of our method using 5-fold cross-validation on $20,000$ hand-labeled responses and found our accuracy to be 95%.

3. ANALYSIS OF VALID RESPONSES ON LEARNING

We now turn our attention to evaluating the impact of providing valid short-answer responses on future learning outcomes using real-world educational data.

Our dataset is taken from a pilot study of our online learning platform, OpenStax Tutor [6], which was conducted during the 2015–2016 academic year. OpenStax Tutor has two important features relevent to our discussion. First, it uses a hybrid answering format [7] that first requires students to enter a short-answer response to the question and requires the student to select the correct answer from a multiple-choice list. Second, OpenStax Tutor employs a concept known as spaced practice, which automatically assigns questions to students on material that they have learned in previous

assignments. The purpose of this feature is to ultimately improve long-term knowledge retention, but we leverage these spaced practice observations as an opportunity to observe the effects of entering valid short-answer responses on later practice.

The pilot consisted of two separate high school courses, AP Biology and standard (non-AP) Physics. A total of 207 students (74 AP Biology, 154 Physics) and 8 instructors (4 AP Biology, 4 Physics) participated in the pilot. There are roughly 100,000 short-answer responses on initial practice problems, and 20,000 of these answers were hand-labeled by subject matter experts as being valid or invalid responses to the given question. The average spaced practice problem occurs roughly 3 weeks after the initial practice on the topic is complete.

To analyze the impact of entering valid open-form responses we adopt a mixed effect logistic regression model [5]. Our binary outcome is whether or not the student answered the spaced practice question for a given topic correctly. Our random effects (R) are nuisance quantities for student ability, topic difficulty, and instructor quality. We examine two different fixed effects in our model: M, the number of multiple-choice questions that a student answered correctly on a given topic and V, the number of valid short-answer responses that a student provided on a given topic.

We consider four separate models for student success on spaced practice questions. Each model includes the random-effects R. We then separately consider the effects of the fixed effects M and V as well as considering both fixed effects jointly. We fit all four models to the AP Biology and Physics datasets separately. The results for AP Biology and Physics are shown on Table 1 and Table 2, respectively. In order to determine which model provided the best fit, we used the Akaike information criterion (AIC) metric, which imposes a penalty that penalizes modes with too many parameters to prevent overfitting. Models with lower AIC values are deemed better than models with higher AIC values.

For AP Biology, we found that the $R+V$ model achieved the lowest AIC implying that the number of valid responses provided a better predictor of success than the number of correct multiple-choice selections. The coefficient for the number of valid responses is positive and statistically significant, which matches our hypothesis that more valid responses improves student retention. For Physics, we note that $R+M+V$ provides the lowest AIC value, and is significantly better than considering $R+M$ alone. This implies that both factors together produce better modeling fitting.

Table 1: Summary of AP Biology Data Models

	Dependent variable:			
	Correct on Spaced Practice			
	(R)	$(R+M)$	$(R+V)$	$(R+M+V)$
Number Core Correct		0.030^{*}		-0.009
		(0.016)		(0.027)
Number Core Valid			0.034^{**}	0.040^{*}
			(0.013)	(0.023)
Constant	0.613^{***}	0.467^{***}	0.427^{***}	0.437^{***}
	(0.075)	(0.107)	(0.105)	(0.109)
Observations	1,987	1,987	1,987	1,987
Log Likelihood	$-1,278.010$	$-1,276.102$	$-1,274.653$	$-1,274.599$
Akaike Inf. Crit.	2,562.019	2,560.203	2,557.305	2,559.199
Note:				*p<0.1; **p<0.05; ***p<0.01

Table 2: Summary of Physics Data Models

	Dependent variable:			
	Correct on Spaced Practice			
	(R)	$(R+M)$	$(R+V)$	$(R+M+V)$
Number Core Correct		0.082^{***}		0.076^{***}
		(0.013)		(0.013)
Number Core Valid			0.097^{***}	0.078^{***}
			(0.023)	(0.022)
Constant	0.002	-0.316^{***}	-0.105	-0.377^{***}
	(0.074)	(0.087)	(0.079)	(0.089)
Observations	4,000	4,000	4,000	4,000
Log Likelihood	$-2,703.761$	$-2,682.312$	$-2,693.697$	$-2,675.836$
Akaike Inf. Crit.	5,413.522	5,372.623	5,395.394	5,361.672
Note:				*p<0.1; **p<0.05; ***p<0.01

4. CONCLUSIONS

We have developed a machine-learning based method for classifying student open-form responses to questions as being either valid (on-topic) or invalid (off-topic) using a combination of intelligent parsing and supervised classification. We have further presented evidence that students who spend time crafting thoughtful responses show improved learning outcomes when practicing earlier material.

The results that we have derived in this work are the result of searching for patterns in existing data and relied on students deciding of their own volition whether or not to enter a valid short-answer response. Future research in this area will involve more highly controlled study in which the opportunity to enter a short-answer response will be controlled by our learning system. This will allow us greater control over our experimental setup and aid in the interpretation of our final result.

5. ACKNOWLEDGMENTS

Thanks to the Art Ciocca, the Laura and John Arnold Foundation, and John and Ann Doerr for supporting this research. Thanks also to Micaela McGlone, Debshila Basu Malik, and Alicia Chang for their help in conducting the pilot studies, preparing the data, and many helpful discussions regarding this work.

6. REFERENCES

[1] T. K. Ho. Random decision forests. In *Proc. 3rd Intl. Conf. Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995.

[2] S. Kang, K. McDermott, and H. Roediger. Test format and corrective feedback modify the effects of testing on long-term retention. *European J. Cognitive Psychology*, 19:528–558, 2007.

[3] J. Karpicke and P. Grimaldi. Retrieval-based learning: A perspective for enhancing meaningful learning. *Educational Psychology Review*, 24:401–418, 2012.

[4] J. L. Little, E. L. Bjork, R. A. Bjork, and G. Angello. Multiple-choice tests exonerated, at least of some charges: Fostering test-induced learning and avoiding test-induced forgetting. *Psychological Science*, 23:1337–1344, 2012.

[5] C. E. McCulloch and J. M. Neuhaus. *Generalized Linear Mixed Models*. Wiley Online Library, 2001.

[6] OpenStaxTutor. https://openstaxtutor.org/, 2017.

[7] J. Park. Constructive multiple-choice testing system. *British Journal of Educational Technology*, 41(6):1054–1064, 2010.

Using an Additive Factor Model and Performance Factor Analysis to Assess Learning Gains in a Tutoring System to Help Adults with Reading Difficulties

Genghu Shi
University of Memphis
The Institute for Intelligent Systems
365 Innovation Dr.
Memphis, TN, 38152
1 001 901 438 8934
gshi@memphis.edu

Philip Pavlik, Jr
University of Memphis
The Institute for Intelligent Systems
365 Innovation Dr.
Memphis, TN, 38152
1 001 901 678 2326
ppavlik@memphis.edu

Arthur Graesser
University of Memphis
The Institute for Intelligent Systems
365 Innovation Dr.
Memphis, TN, 38152
1 001 901 240 4795
art.graesser@gmail.com

ABSTRACT
After developing an intelligent tutoring system (ITS), or any other class of learning environments, one of the first questions that should be asked is whether the system was effective in helping students learn the targeted skills or subject matter. In this study, we employed two educational data mining models (Additive Factor Model, AFM and Performance Factor Analysis, PFA) which are available in Datashop (LearnSphere) to assess the learning gains on 5 theoretical levels of adults. With AFM, for the KC models tested, the results showed positive learning gains for the Rhetorical Structure knowledge component in contrast, for the PFA model, adults did not learn from either successes or failures.

Keywords
Learning gains, Theoretical Levels, Additive Factor Model, Performance Factor Analysis, CSAL Autotutor

1. INTRODUCTION
One of the first questions that is asked after developing an intelligent tutoring system (ITS) is whether the system was effective in helping students learn the targeted skills or subject matter. Learning gains are based on the performance of the students as they work on the system over time with many opportunities for learning. These learning gains can be assessed at a fine-grained level by tracking the learning of specific knowledge components (KCs), which are particular skills, strategies, concepts, or facts, as articulated in the Knowledge-Learning-Instruction (KLI) framework [2]. In this paper, we analyze the learning of the theoretical components (KCs) which were based on models of comprehension that adopt a multilevel framework in our dialogue-based intelligent tutoring system, called CSAL AutoTutor, that was designed to help struggling adult readers learn reading comprehension strategies. The Graesser and McNamara framework identifies 5 levels [1]: words (W), syntax (S), the explicit textbase (TB), the referential situation model (SM), the discourse genre and rhetorical structure (RS, the type of discourse and its composition). And, the computational models used in the analysis were Additive Factor Model (AFM) and Performance Factor Analysis, both of which were from Datashop (LearnSphere) [3]. 3 questions will be addressed in this paper: 1. When training the adults to read, did the performance of the adults follow the levels of text difficulty? 2. Did adults' learning gains increase after using the Autotutor which just provided some instructions on reading comprehension strategies and some practice? 3. Did adults learn from successes or failures?

2. METHODOLOGY
The adult readers were 52 adults in Atlanta and Toronto who participated in a study of 100 hours of intervention that was conducted by the CSAL team, and they completed up to 30 lessons throughout the intervention. Each lesson had between 10 and 30 multiple choice questions to assess their performance When they answered a question incorrectly, they were given a hint to see whether they selected correctly among the two remaining options. However, in this analysis we only considered performance on their first type, not the follow-up.

The original measures in the AFM model included performance, practice opportunities (the number of questions they answered in a lesson), the knowledge components (KCs were the 5 theoretical components), and subject (participant). For model fitting, pre-test scores and text difficulty (easy, medium, and hard) were entered into the original models (Table 1). Ultimately, we ran 10 models (5 AFM models and 5 PFA models) for the KC approaches, and determined which AFM and PFA models had the best performance, based on AIC, BIC, and Loglikelihood.

Table 1. Models Construction by Adding New Variables

Models	Variables
Model 1	Pre-test score
Model 2	Pre-test score, Text Difficulty
Model 3	Pre-test score, Text Difficulty: KC Model
Model 4	Pre-test score, Practice Opportunity: KC Model
Model 5	Pre-test score, Text Difficulty: Practice Opportunity: KC Model

** These models are basically logit mixed effect models. The ":" refers to interactive effect.*

3. RESULTS AND DISCUSSION
Analyses of the 10 models consistently showed that model 3 was the best model, yielding the lowest AIC BIC and Loglikelihood scores.

Both Table 2 (AFM results) and Table 3 (PFA results) confirm the obvious expectation that pretest score is a strong predictor of adults' performance. Also, only for Rhetorical Structure, performance decreased as a function of text difficulty. This is consistent with the Graesser and McNamara's multilevel

theoretical framework that distinguishes the deeper discourse levels of processing (such as the Situation Model and Rhetorical Structure) from the basic reading levels (such as Words and Syntax) [1]. As shown in table 2, only for Rhetorical Structure, performance significantly got better as the practice opportunity increased, but the case of the other KCs was different. As shown in table 3, although cumulative correctness had significant interactions with Syntax and Situational Model, while cumulative incorrectness had significant interactions with Syntax and Textbase, the estimates of these interactions were all negative, which indicated that the performance got worse, no matter adults experienced more successes or failures on these KCs. And, for other KCs, the coefficients drifted to 0.

Table 2. AFM Output of Model 3 – Theoretical Levels

	Estimate	SE	Z Score	P-value	Sig.
Intercept	0.675	0.25	2.66	0.01	**
Pre-test Score	0.140	0.03	4.97	0.00	***
PO : RS	0.001	0.00	2.27	0.02	*
PO : S	-0.124	0.02	-5.16	0.00	***
PO : SM	-0.003	0.00	-3.69	0.00	***
PO : TB	-0.016	0.00	-4.98	0.00	***
PO : W	-0.004	0.00	-0.95	0.34	
RS : Hard	-1.805	0.19	-9.73	0.00	***
S : Hard	0.822	0.28	2.94	0.00	**
SM : Hard	-0.111	0.18	-0.62	0.54	
TB : Hard	0.014	0.19	0.07	0.94	
W : Hard	-0.204	0.30	-0.69	0.49	
RS : Medium	-1.241	0.18	-7.07	0.00	***
S : Medium	-0.078	0.26	-0.30	0.77	
SM : Medium	-0.035	0.18	-0.20	0.84	
TB : Medium	0.133	0.19	0.71	0.48	
W : Medium	0.529	0.29	1.84	0.07	.

**PO refers to practice opportunity. RS refers to Rhetorical Structure. S refers to Syntax. SM refers to Situational Model. TB refers to Textbase. W refers to Word. Easy, Medium, Hard are three levels of text difficulty.*

Table 3. PFA Output of Model 3 – Theoretical Levels

	Estimate	SE	Z Score	P-value	Sig.
Intercept	0.671	0.26	2.60	0.01	**
pretest	0.145	0.03	4.87	0.00	***
CC : RS	0.000	0.00	-0.12	0.91	
CC : S	-0.127	0.04	-3.47	0.00	***
CC : SM	-0.005	0.00	-2.32	0.02	*
CC : TB	-0.008	0.01	-1.30	0.19	
CC : W	-0.004	0.01	-0.69	0.49	
CI : RS	0.005	0.00	1.37	0.17	
CI : S	-0.123	0.04	-3.14	0.00	**
CI : SM	0.001	0.00	0.41	0.68	
CI : TB	-0.031	0.01	-2.77	0.01	**
CI : W	-0.002	0.02	-0.13	0.90	
RS : Hard	-1.808	0.19	-9.74	0.00	***
S : Hard	0.828	0.37	2.22	0.03	*
SM : Hard	-0.099	0.18	-0.55	0.58	
TB : Hard	-0.069	0.20	-0.35	0.73	
W : Hard	-0.209	0.30	-0.69	0.49	
RS : Medium	-1.248	0.18	-7.10	0.00	***
S : Medium	-0.079	0.27	-0.29	0.77	
SM : Medium	-0.023	0.18	-0.13	0.90	
TB : Medium	0.068	0.19	0.35	0.72	
W : Medium	0.524	0.30	1.77	0.08	.

**CC and CI refer to cumulative correctness and cumulative Incorrectness. Others are the same as Table 2.*

4. CONCLUSIONS

The model comparison revealed that practice opportunity, adults' prior literacy skills, KC model (theoretical levels) and text difficulty were factors influencing adults' performance. From the interactions between theoretical levels and text difficulty, we can draw the conclusion that adults' performance on Rhetorical Structure and Situational Model matched the difficulty levels of the texts used in the lessons of the two KCs, that is, they did better on easy texts and worse on medium and hard texts. But for the basic reading levels (Word, Syntax, and Textbase), situations were different. According to the results of AFM model, the learning gains on deeper discourse levels of processing (Rhetorical Structure) increased, because adults' performance became better when they continuously got practice opportunities. There were no learning gains observed on KCs like Situational Model, Syntax, Textbase, and Word. From results of PFA model, we didn't observe significant learning gains from either successes or failures.

5. ACKNOWLEDGMENTS

This research was supported by the National Center of Education Research (NCER) in the Institute of Education Sciences (IES) (R305C120001) and the National Science Foundation Data Infrastructure Building Blocks program under Grant No. (ACI-1443068).

6. REFERENCES

[1] Graesser AC, Mcnamara DS, Kulikowich JM (2011) Coh-Metrix providing multilevel analyses of text characteristics. Educational researcher 40:223-234

[2] Koedinger KR, Corbett AT, Perfetti C (2010) The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. Cognitive Science

[3] Pavlik Jr PI, Cen H, Koedinger KR (2009) Performance Factors Analysis--A New Alternative to Knowledge Tracing. Online Submission

Identifying Student Communities in Blended Courses

Niki Gitinabard, Collin F. Lynch, Sarah Heckman, Tiffany Barnes
North Carolina State University
Computer Science Department
Raleigh, NC, US
{ngitina, cflycnh, sarah_heckman, tmbarnes}@ncsu.edu

ABSTRACT
Blended courses have become the norm in post-secondary education. Universities use large-scale learning management systems to manage class content. Instructors deliver readings, lectures, and office hours online; students use intelligent tutors, web forums, and online submission systems; and classes communicate via web forums. These online tools allow students to form new social networks or bring social relationships online. They also allow us to collect data on students' social relationships. In this paper we report on our research on community formation in blended courses based on online forum interactions. We found that it was possible to group students into communities using standard community detection algorithms via their posts and reply structure and that the students' grades are significantly correlated with their closest peers.

Keywords
Educational Data Mining, Graph data mining, Social Networks, Blended Courses

1. INTRODUCTION
Improvements in technology have facilitated new models of student and instructor engagement. Students now supplement the traditional course structure with online materials. Instructors can share class material online, have an online discussion forum, or make quizzes and homework submissions online. This in turn provides a wealth of new data on student behaviors that we can use to study students' social relationships. In particular it allows us to study the impact of these social ties on course outcomes.

In prior work Brown et. al. showed that students in MOOCs form pedagogically-relevant, and homogeneous social networks. Brown et. al. has shown that students can be clustered into stable communities based upon their pattern of online questions and replies [1]. They have also shown that students' final grades are significantly correlated with those of their closest peers and community group. They have also shown that these communities, while homogeneous in terms of performance, are not united by their incoming motivations for enrolling in the course nor for their prior experience level [2].

To date these results have only been found in MOOCs where the user forum represents students' primary connection to one-another, and almost all relevant course interactions occur online. Students in blended courses, by contrast, often have preexisting social ties that carry over from prior courses at the same institution. In this paper we show that while forum interactions are not the only means of communication between students, they still define the same communities as was found in MOOCs and that the students' final grades are significantly correlated with those of their community members.

2. DATASET INFORMATION
In this paper we report on studies of three distinct courses, "Discrete Math-2013", "Discrete Math-2015" and "Java Programming Concepts-2015". All three are undergraduate computer science courses, offered at NC State and include significant blended components. Discrete Math-2015 and Java Programming Concepts-2015 occurred contemporaneously during the Fall 2015 semester while Discrete Math-2013, a previous offering of Discrete Math-2015, was offered in Fall 2013.

3. METHODS
3.1 Defining Social Interactions
Each node in our social networks represents an individual participant in the class. In the first class anonymous posting was allowed, so we have an unknown user related to all the anonymous posts. Social relationships are represented as arcs. We define a social relationship based upon direct and indirect replies in the user forum. Our method was similar to that of Brown et. al. [2]. We defined an edge between A and B if B replied to a thread after A had done so. This interaction can include starting the original thread, replying with a follow-up, or posting a feedback on a reply. We then aggregate these edges to form a weighted graph containing arcs for all of the relations. We assume that anyone who posts on a thread has read the prior comments before doing so. Thus it defines a form of social interaction between the participants as the students are expressly choosing to make a public reply to one another. For the purposes of the present analysis we included only students in our network and thus

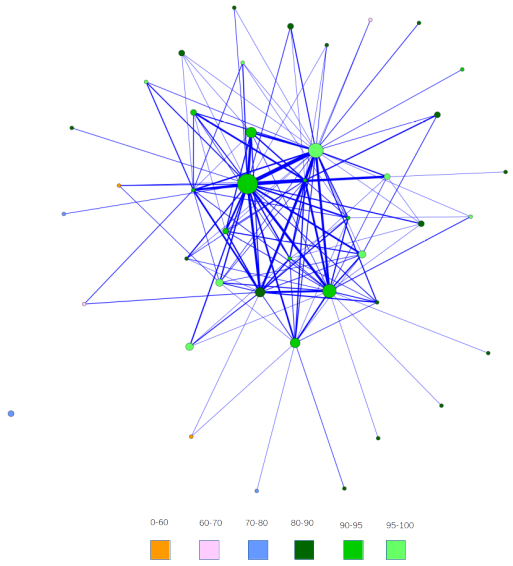


Figure 1: Communities generated on Discrete Math 2013 class

confined our social relationships to between-student connections.

3.2 Graph Analysis

For each of the graphs we generated, we removed the isolated vertices and performed clustering using the method described in [2, 1]. Our clustering method is an iterative process where we evaluate the modularity of graphs with an increasing number of clusters until we find a limit point where the modularity almost stops growing, which indicates the *natural cluster number*. After finding the natural number, on each iteration we generated the clusters via the Girvan-Newman edge-centrality algorithm[3]. On each iteration the algorithm removes the most central edge and and repeats until a set of k disjoint clusters has been produced. We then assessed whether or not the grade distributions in different clusters are significantly different by calculating the Kruskal-Wallis (KW) correlation between cluster assignment and grade. Kruskal-Wallis is a nonparametric analogue to the more common ANOVA test [4].

4. RESULTS

In graphs generated for Discrete Math 2014, we found that the graph reaches its natural cluster number at 42. We performed the Girvan Newman clustering and the resulting clusters can be seen in Figure 1. In this graph, each node represents a community, the size of the nodes shows the number of members and the color shows their average grade. We can observe that the KW correlation between cluster number and the grades is statistically significant ($p = 0.044 < 0.05$), which is similar to the results in MOOCs.

Our results show that, for Discrete Math 2015 ($p = 0.004 < 0.05$) and Java Programming Concepts 2015 ($p = 0.015 < 0.05$) graphs, there is a similar significant KW correlation between student grades and their communities.

5. DISCUSSION, CONCLUSIONS AND FUTURE WORK

In this paper, we generated a social graph between students in three different blended courses based on forum interactions. We found that similar to MOOCs, communities are formed in these graphs whose members tend to have similar grades. This is consistent with prior work which indicates that student communities on forum may be used to predict course outcomes [1, 2].

Having access to these social graphs can help instructors to identify the communities formed among students which can be used to find the students who need more help earlier. Our research does not show causality. Thus more research is needed to find out whether being in the communities makes their grades similar, or students are just likely to interact with others who are more like them. If we find out that the community membership has an effect on students' performance, we can use this information to identify isolated or poorly-performing groups early in the course and intervene by encouraging them to make contact with better students or seek help as a group.

There has been much work done on how forum interactions in MOOCs, being a hub in a social network or how being at the center of the graph could affect students' performance. We can use these graphs to conduct more research on which interaction levels will lead to better grades.

In further work we plan to address whether or not we can identify other types of social ties in blended courses, since the communications are more complicated.

6. ACKNOWLEDGMENTS

This work was supported by NSF grant #1418269: "Modeling Social Interaction & Performance in STEM Learning" Yoav Bergner, Ryan Baker, Danielle S. McNamara, & Tiffany Barnes Co-PIs.

7. REFERENCES

[1] R. Brown, C. Lynch, M. Eagle, J. Albert, T. Barnes, R. S. Baker, Y. Bergner, and D. S. McNamara. Good communities and bad communities: Does membership affect performance? In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain, June 26-29, 2015*, pages 612–613, 2015.

[2] R. Brown, C. Lynch, Y. Wang, M. Eagle, J. Albert, T. Barnes, R. S. Baker, Y. Bergner, and D. S. McNamara. Communities of performance & communities of preference. In *EDM (Workshops)*, 2015.

[3] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[4] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

Automatic Scoring Method for Descriptive Test Using Recurrent Neural Network

Keiji Yasuda
KDDI Research
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
ke-yasuda@kddi-research.jp

Izuru Nogaito
KDDI Research
2-1-15, Ohara, Fujimino city,
Saitama, 356-8502 Japan
iz-nogaito@kddi-research.jp

Hiroyuki Kawashima
KDDI Research
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
hi-kawashima@kddi-research.jp

Hiroaki Kimura
KDDI Research
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
ha-kimura@kddi-research.jp

Masayuki Hashimoto
KDDI CORPORATION
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
mu-hashimoto@kddi.com

ABSTRACT

In this paper, we propose an automatic evaluation method for the descriptive type test. The method is based on Recurrent Neural Networks trained on a non-labeled language corpus and manually graded students' answers. The experimental results show that the proposed method is the second best result among five conventional methods, including BLEU, RIBES, and several sentence-embedding methods. And, the proposed method gives the best performance among several sentence embedding methods.

Keywords

RNN, LSTM, Language Model, Essay Scoring

1. INTRODUCTION

Twenty-first-century skills are advocated in the educational field. Compared to traditional knowledge-based education evaluated by multiple-choice tests, the evaluation of twenty-first-century skills is very difficult. A descriptive test is one solution to the problem, although the cost of scoring is prohibitive. In this paper, we propose a method to automatically score descriptive type tests to solve the problem stated above. The method uses long short-term memory (LSTM) recurrent neural networks (RNN) to score the answers written in natural language. The method requires two kinds of data sets.

One is a large language corpus used for pre-training of RNN. As pre-training, the RNN-based language model is trained using the corpus. A vector given by a hidden layer in the networks is thought to embed the meaning of processed sentences. Thus, the proposed method calculates the similarity between two vectors given by processing model answers and student answers on RNN. The other data set is a small labeled corpus that consists of model answers, student answers, and manually annotated scores of student answers. The labeled corpus is used for training of the RNN.

2. PROPOSED METHOD

The RNN framework used in the paper is shown in Fig. 1. As shown in the figure, the proposed method uses two kinds of corpora and two kinds of training parts. They are the pre-training of word embedding and the main training of the LSTM-type RNN [3].

Here, we express the sentence ($\mathbf{s}$) as the sequence of words $\mathbf{s} = w_1, \cdots, w_t, \cdots, w_T$. The word-embedding part projects the input word of time t (w_t) to high-dimension vector $x_{w_t} \in \mathbb{R}^{d_w}$ as follows:

$$\mathbf{x}_{w_t} = \mathbf{E}^{\mathrm{T}} \mathbf{w}_{w_t} \tag{1}$$

where $w_{w_t} \in \mathbb{R}^{|V|}$ is the one-hot vector of w_t and $\mathbf{E} \in \mathbb{R}^{|V| \times d_w}$ is the lookup table. $x_{w_t}$is used as the input for the LSMT part. The LSTM consists of four components: the forget gate ($\mathbf{f}_t$), input gate ($\mathbf{i}_t$) and output gate ($\mathbf{o}_t$), and the memory state ($\mathbf{c}_t$). These real-valued vectors are calculated by the following formulas:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_{w_t} + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_{w_t} + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_i \mathbf{x}_{w_t} + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_{\tilde{c}} \mathbf{x}_{w_t} + \mathbf{U}_{\tilde{c}} \mathbf{h}_{t-1} + \mathbf{b}_{\tilde{c}}), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t
\end{aligned} \tag{2}$$

where $\mathbf{W}$ and $\mathbf{U}$ are weight matrices, and $\mathbf{b}$ is the bias vector. $\sigma(\cdot)$ and $\tanh(\cdot)$ are an element-wise sigmoid function

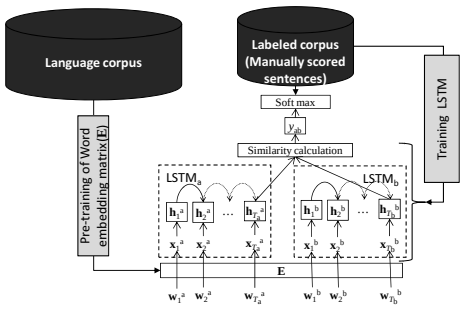


Figure 1: Framework of the proposed method.

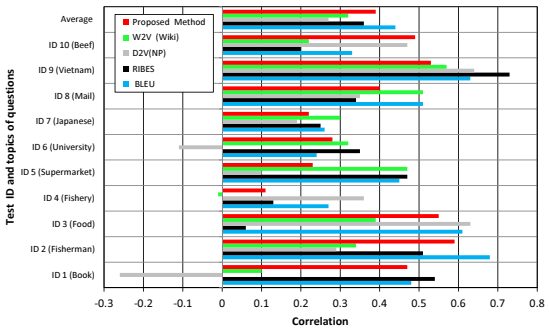


Figure 2: Experimental results.

and a hyperbolic tangent function, respectively. Using these vectors, hidden-layer vector ($\mathbf{h}_t \in \mathbb{R}^{d_s}$) is calculated as follows:

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{3}$$

where $\odot$ is element-wise multiplication. The main training part requires a labeled corpus that consists of model answers, the students' answers, and manually scored results of the students' answers. By using the labeled corpus, the second training part tunes the LSTM whose network configuration was proposed by Mueller et al. [1]. Using pre-trained word-embedding matrix $\mathbf{E}$ from the first training part, LSTM parameters are trained as follows.

First, randomly initialize LSTM parameters in Eq. 2. Then, duplicate the initialized LSTM ($\mathrm{LSTM_a}$ and $\mathrm{LSTM_b}$ in Fig. 1). One of them is used to process the student's answer and the other is used to process the model answer. We regard the hidden-layer vector of the sentence end as sentence embedding. To calculate the sentence similarity between the student's answer and the model answer, we add a new unit between the hidden layers. The unit calculates the L1 norm based on the similarity between the two sentence embeddings ($\mathbf{h}^{\mathrm{a}}_{T_{\mathrm{a}}}$ and $\mathbf{h}^{\mathrm{b}}_{T_{\mathrm{b}}}$ in Fig. 1) by using the following formula [1]:

$$\begin{aligned} g(\mathbf{h}^{\mathrm{a}}_{T_{\mathrm{a}}}, \mathbf{h}^{\mathrm{b}}_{T_{\mathrm{b}}}) &= \exp(-\|\mathbf{h}^{\mathrm{a}}_{T_{\mathrm{a}}} - \mathbf{h}^{\mathrm{b}}_{T_{\mathrm{b}}}\|_1) \\ &= \exp\left(-\sum_{i=1}^{d_s} \left|h^{\mathrm{a}}_{T_{\mathrm{a}}} - h^{\mathrm{b}}_{T_{\mathrm{b}}}\right|\right) \end{aligned} \tag{4}$$

The similarity calculation is performed only when both sentence pairs have been processed by the LSTM. Using the similarity calculated by Eq. 4 and the manually evaluated score, the deviation is back propagated to tune the LSTM weights. Here, we restrict the parameters of $\mathrm{LSTM_a}$ and $\mathrm{LSTM_b}$ to the same values.

3. EXPERIMENTS

The labeled corpus consists of 10 descriptive type questions and their answers. For each question, around 20 answers are manually scored. Additionally, there are also four model answers for each question. For the pre-training of the word-embedding matrix, we use a Mainichi newspaper corpus.

Since the size of the labeled corpus is very small, we carry out a leave-one-out cross-validation test for each question. The cross-validation is carried out only for student answers. The same model answers are used for training and evaluation. The LSTM in the paper can only process a pair of one student answer and one model answer at the same time. Thus, all combinations of student answers and model answers in the training set are used for training. For the scoring of the test set, we calculate the average score of several model answers. The evaluation measure is the correlation coefficients between the manual and the automatic scoring results.

Fig. 2 shows the experimental results. As baseline results, we show the results of BLEU, RIBES, and the Doc2Vec (D2V) cosine similarity method with the NewsPaper(NP) corpus and Wikipedia(Wiki) corpus by referring to the conventional research[2]. As shown in the figure, the proposed method never gives a negative correlation coefficient. Meanwhile the conventional sentence-embedding-based methods give negative correlation coefficients. Additionally, the proposed method gives the best results on average among sentence-embedding methods, which are two kinds of D2V and the proposed method. Compared to all methods, the proposed method offers the second-best performance.

4. CONCLUSIONS AND FUTURE WORKS

We proposed the LSTM-based automatic scoring method for descriptive tests. We carried out experiments using actual learning logs. According to the experimental results, the proposed method gives the best performance among several sentence-embedding methods, and the second-best results among five methods including BLEU and RIBES.

5. ACKNOWLEDGMENTS

This work used model answers, students' answers, and scoring data forms from the Lojim School. (http://lojim.jp/).

6. REFERENCES

[1] J. Mueller et al. Siamese recurrent architectures for learning sentence similarity. In Proc. of AAAI, pages 2786–2792, 2016.

[2] I. Nogaito et al. Study on automatic scoring of descriptive type tests using text similarity calculations. In Proc. of EDM, pages 616–617, 2016.

[3] M. Sundermeyer et al. LSTM neural networks for language modeling. In Proc. of Interspeech, pages 194–197, 2012.

Using Graph-based Modelling to explore changes in students' affective states during exploratory learning tasks

Beate Grawemeyer
Birkbeck, University of London
beate@dcs.bbk.ac.uk

Alex Wollenschlaeger
Birkbeck, University of London
awolle01@dcs.bbk.ac.uk

Sergio Gutierrez-Santos
Birkbeck, University of London
sergut@dcs.bbk.ac.uk

Wayne Holmes
The Open University, UK
wayne.holmes@open.ac.uk

Manolis Mavrikis
UCL Institute of Education
m.mavrikis@ucl.ac.uk

Alexandra Poulovassilis
Birkbeck, University of London
ap@dcs.bbk.ac.uk

ABSTRACT

We describe a graph-based modelling approach to exploring interactions associated with a change in students' affective state when they are working with an exploratory learning environment (ELE). Student-system interactions data collected during a user study was modelled, visualized and queried as a graph. Our findings provide new insights into how students are interacting with the ELE and the effects of the system's interventions on students' affective states.

1. INTRODUCTION

Much recent research has focussed on *Exploratory Learning Environments* (ELEs) which encourage students' open-ended interaction with a knowledge domain, combined with intelligent components that aim to provide pedagogical support to ensure students' productive interaction. The aim of this feedback is to balance students' freedom to explore alternative task solution approaches while at the same time providing sufficient support to ensure that the intended learning goals are being achieved [6]. Here we report on recent work into identifying interaction events that are associated with a change in students' affective state as they interact with an affect-aware ELE called *Fractions Lab*. We adopt a graph-based approach to modelling, querying and visualizing the student-system interactions data, extending preliminary work in this area reported in [8]. In our graphs, nodes represent occurrences of key indicators that are detected, inferred or generated by the ELE, and edges between such nodes represent the "next event" relationship. In contrast, recent work on interaction networks and hint generation (e.g. [4]) uses graphs whose nodes represent states within a problem-solving space and edges represent students' actions in transitioning between states. That work uses the graph-modelled data to automatically generate feedback for the student, whereas we use a graph-based modelling approach to investigate the effects of the system's interventions in order to better understand how students interact with the ELE with the aim of improving its support for students.

2. THE ELE AND USER STUDY

Fractions Lab is an ELE that is part of the iTalk2Learn learning platform targeted at children aged 8-12 years who are learning about fractions. As students interact with Fractions Lab they are asked to talk aloud about their reasoning process. This speech, together with their interactions, are used to detect students' affective states using a combination of Bayesian and rule-based reasoning [5]. Adaptive support is provided based on the student's performance and detected affective state. The affective states detected by Fractions Lab can be ranked according to their effect on learning, based on previous studies (e.g. [7, 3, 1]). For example, being in *flow* is a positive affective state as it indicates that the student is engaging with the learning task well. *Confusion* is mostly associated with realising misconceptions, which also contributes towards learning, while *frustration* and *boredom* are likely to have a negative effect on learning.

We conducted a user study in which iTalk2learn was used by students in a classroom setting. 41 students aged 8-10 took part, with parental consent, recruited from two schools in the UK. Students were given a short introduction to the system. They then engaged with the Fractions Lab ELE for 40 minutes. They then completed an online questionnaire that assessed their knowledge of fractions (the post-test).

The iTalk2Learn platform logged every student-system interaction, such as fractions being created or changed by students, buttons being clicked, feedback being provided by the system, feedback being viewed by students, and the system's detection of students' affective states. This data was then remodelled into a graph form, according to the graph data model shown in Figure 1. We see that the data model comprises two node types: Event nodes, that capture occurrences of key interactions, and EventType nodes, that hold additional metadata about each event. Edges labelled NEXT link together successive Event nodes, allowing us to build up a sequence of events that describe the history of student-system interactions as a student works on a task during a session. An edge labelled OCCURRENCE_OF links each Event node to an EventType node.

The data logged by iTalk2Learn was exported as text, parsed and pre-processed using Python and the Pandas and py2neo

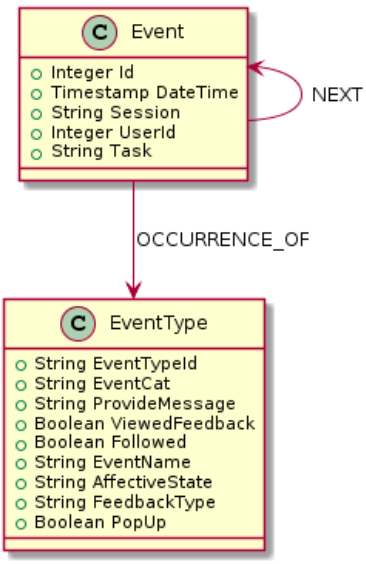


Figure 1: Graph data model for student-system interaction data.

libraries, and then loaded into the Neo4j graph database. To view the resulting data graph we developed a custom visualization tool in JavaScript using the Node.js library. Our tool allows viewing of large-scale changes in affective state as well as details of event sequences. Having interacted with these visualizations, we were interested to explore further the kinds of events that contribute towards changes in students' affective state as they work with Fractions Lab. To do this, we used Neo4j's graph query language, Cypher, to extract the metadata relating to pairs of consecutive events that exhibit a change in a student's affective state. The query below was used to find adjacent Event nodes connected by NEXT, and the EventType nodes they are connected to by OCCURRENCE_OF, such that the affective states associated with the EventType nodes are not equal:

```
MATCH (start_event: Event)-[:OCCURRENCE_OF]->(start_type: EventType),
      (end_event: Event)-[:OCCURRENCE_OF]->(end_type: EventType),
      p = (start_event)-[:NEXT]->(end_event)
WHERE start_type.affective_state in
    ["flow", "boredom", "confusion", "frustration"]
    AND end_type.affective_state in
    ["flow", "boredom", "confusion", "frustration"]
    AND NOT start_type.affective_state = end_type.affective_state
RETURN *
```

3. RESULTS AND CONCLUSIONS

We were interested to explore differences in students' affective states and interactions compared with their performance. Students' performance, based on the post-test score, was on average 3.83 (SD=1.46; min=0; max=6). A median split of students' scores resulted in a higher- and a lower-performing group (high: 27 students; low: 14 students). In order to investigate which interactions moved students into a different affective state we used association rule learning (c.f. [2]) over the data returned by the above Cypher query. We found that students are likely to move from *flow* to *frustration* when provided with reflective prompts in the low-performing group and with open-ended problem solving support in the high-performing group. This might imply that these types of support are imposing too high a cognitive demand on students. Additionally, certain interactions with their fractions may move both categories of student from *flow* to *frustration*. Viewing high-interruption or low-interruption feedback may move low or high performing students, respectively, from *flow* to *confusion*. Finally, we observed a positive effect of Affect Boost messages for both categories of student.

These findings extend earlier ones reported in [5] with a finer-grained analysis of students' affective state changes, identifying several situations where the system's support may need to be modified: (i) reviewing the content of both the high- and the low-interruption messages, to see if the incidences of confusion can be reduced; (ii) considering extending the provision of reflective prompts and open-ended support with additional affect boost messages and hints that students might also select to view, to mitigate against frustration; (iii) considering providing more scaffolds when students are manipulating their fractions, for example additional low-interruption feedback. Exploratory learning environments such as Fractions Lab can generate large volumes of student-system interactions data, making their interpretation a challenging task. We have seen here how modelling such data as a graph can open up new data visualization, querying and analysis opportunities, leading to new insights into how students are interacting with the ELE and the effects of the system's interventions, with the ultimate goal of designing improved support for students.

4. REFERENCES

[1] R. S. J. d. Baker et al. 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. *Int. J. Hum.-Comput. Stud.*, 68, 2010.

[2] D. L. Bazaldua, R. S. J. de Baker, and M. O. S. Pedro. Comparing expert and metric-based assessments of association rule interestingness. In *EDM*, 2014.

[3] S. K. D'Mello et al. Confusion can be beneficial for learning. *Learning & Instruction*, 29(1):153–170, 2014.

[4] M. Eagle, D. Hicks, B. Peddycord III, and T. Barnes. Exploring networks of problem-solving interactions. *LAK*, pages 21–30, 2015.

[5] B. Grawemeyer et al. Affective learning: Improving engagement and enhancing learning with affect-aware feedback. *User Modeling and User-Adapted Interaction - Special Issue on Impact of Learner Modeling*, 2017.

[6] S. Gutierrez-Santos, M. Mavrikis, and G. D. Magoulas. A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments. *J. Research and Practice in Inf. Tech.*, 44:347–360, 2013.

[7] R. Pekrun. The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *J. Edu. Psych. Rev.*, pages 315–341, 2006.

[8] A. Poulovassilis, S. Gutierrez-Santos, and M. Mavrikis. Graph-based modelling of students' interaction data from exploratory learning environments. In *Proceedings of G-EDM, at EDM*, 2015.

Predicting Performance in a Small Private Online Course

Wan Han, Ding Jun, Gao Xiaopeng, Yu Qiaoye, Liu Kangxu

School of Computer Science and Engineering
Beihang University
Beijing, China
+86-10-82338059
{wanhan, dingjun, gxp, yuqiaoye, liukangxu}@buaa.edu.cn

ABSTRACT

In this paper, we describe how we build accurate predictive models of students' performance in a SPOC (small private online course). We document a performance prediction methodology from raw logging data based on OpenEdX platform to model analysis. We attempted to predict students' performance of Computer Structure Lab Course (Fall 2016) offering at Beihang University. 28 predictive features extracted for 377 students, and our model achieved an AUC (area under curve) in the range of 0.62-0.83 when predicting one week in advance. This work would help to identify at-risk students in a SPOC.

Keywords

SPOC, student performance prediction, study behavior analysis, educational data mining, at-risk students

1. INTRODUCTION

EdX has designed and built an open-source online learning platform (OpenEdX) for online education. In addition to offering online courses, participating universities are also committed to researching how students learn and how technology can transform learning both on-campus and online throughout the world.

Some researches focus on how to predict students' performance by using study-related data. Stapel, M. [1] presented an ensemble method to predict students' performance, which includes six classification algorithms. Elbadrawy, A. [2] developed multi-regression models based on regression algorithms for predicting, and Ren, Z. [3] designed different kinds of features based on MOOC courses' characters, which improved the performance of their predictor. In addition to study-related data, social behavior data is helpful in predicting [4].

In this paper, we describe the performance prediction problem, and present models we built. A summary of which features played a role in gaining accurate predictions is presented. The most fundamental contribution is the design, development and demonstration of a performance prediction methodology, from raw logging data to model analysis, including data preprocessing, feature engineering, model evaluation and outcome analysis.

2. PREDICTION PROBLEM DEFINITION

Our SPOC was composed of 3 tutorials and 9 projects in Fall 2016, learners studied the tutorials from week 1 to week 6, and we released project 0 at week 7. We found it was important for learners to move on only after they'd mastered the core concept. Students started one project and as they mastered corresponding content, that they need to pass the test in class, and then they could be awarded to the next project.

Here our performance prediction is to predict whether the learner could pass their test at the end of each week according to their study behavior. We define time slices as weekly units. Time slices started the first week in which in class test was offered (week 7), and ended in the 16^{th} week, after the final test had closed.

So we could use the logging data from week 1 to week 6 to predict the learners' performance at week 7. Furthermore, we used 'lead' represents how many weeks in advance to predict performance. We assign the performance label (x_1, 0 for unpassed the test or 1 for passed the test) of the lead week as the predictive problem label. 'Lag' means use how many weeks of historical variables to classify.

3. PREDICTING WEEK PERFORMANCE

We did not use the non-behavioral attribute such as a leaner's age, gender and others. Instead, we used some features that would show different style of learning habits. One type of behavioral variables is based on the learner's interaction with the educational resources, including time spent on resources and problem / homework. As Colin Taylor described in [5], taking the effort to extract complex predictive features that require relative comparison or temporal trends, rather than using the direct covariates of behavior, is one important contributor to successful prediction. For instance, we create an average number of submissions per problem for each learner (x9). Then we compare a learner's x9 value to the distribution for that week. Feature x16 is the percentile over the distribution and x17 is the percent as compared to the max of the distribution. We also extracted features that related to learners' study habits. For instance, feature to describe whether learners begin doing the problem / homework soon after it was released, and features to characterize the learners that submit problem / homework in timely fashion or at last minute fashion.

To build predictive models, we utilize a common approach of flattening the data- assembling the features from different weeks as separate variables.

We first used logistic regression as our binary predictive model. It calculates a weighted average of a set of variables as an input to the logit function. There are different coefficients for the feature values. For the binary classification problem, the output of the logit function becomes the estimated probability of a positive training example.

When applying the logistic regression to learner week performance prediction. We used 28 features to form the feature vectors, and maintained the week performance value as the label.

3.1 Predicting Performance

When evaluating the classifier's performance. A testing set comprised of untrained covariates and labels evaluates the performance of the model as following steps:

The logistic function learned is applied to each data point and the estimated probability of a positive label is produced. And then a decision rule is applied to determine the class label for each probability estimate. Given the estimated labels for each data point and the true labels we calculate the confusion matrix, true positives and false positives and then obtain an operating point on the ROC curve. Then evaluate the area under the curve and report it as the performance of the model on the test data.

We need to present the results for multiple prediction problems for different week simultaneously. Here means for each week during our course, we want to predict the students' week performance using different historical data. The heat map of a lower right triangular matrix is assembled as shown in figure 1.

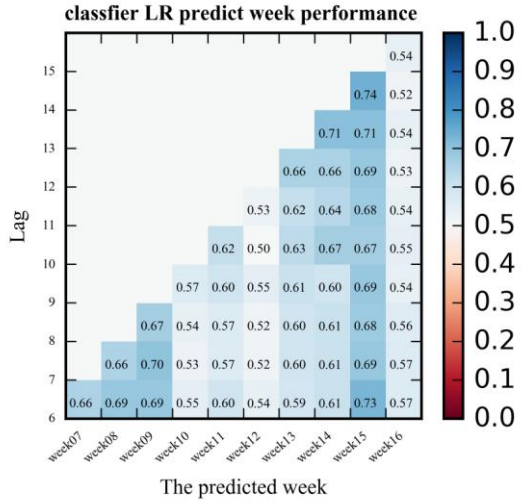


Figure 1. Logistic regression results

The x-axis of figure 1 is the week for which predictions are made in the experiment, while y-axis is the number of the how many week data we use for the prediction (lag). The color shown the area under the curve for the ROC the current model achieved.

We employed cross validation in all of our predictive modelling. Some partitions are used to construct a model, and others are used to evaluate the performance. Considering only 377 samples in our data set, we employed 3-fold cross validation and use the average of the ROC AUC over the folds as evaluation metric.

3.2 Feature Importance

We utilized randomized logistic regression methodology to identify the relative weighting of each features. As shown in figure 2, top features that had the most predictive power include whether learners interact with the resources more time (*max_observed_event_duration*), learners' interaction with the problems (*average_number_of_submissions_percentile*), study habits (*time_first_attempt, problem_finish_time_pre_start24h*, *problem_finish_time_pre_start48h*).

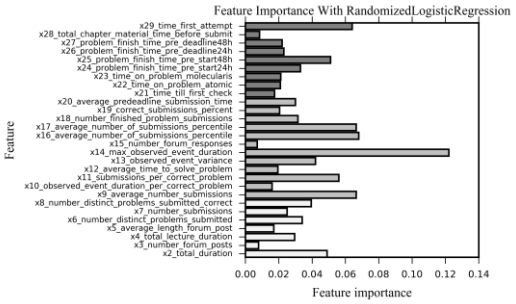


Figure 2. Relative importance of different features across all variants (lag / lead)

4. SUMMARY

We have taken an initial step towards identifying at-risk students in a SPOC, which could help instructors design interventions. Several prediction models are compared, with SVM preferred due to its good performance. The noteworthy accomplishments of our study when compared to other studies including: we extracted variable from the click stream logging data and then generate complex features which explain the learners' study behavior, especially how to describe the learners' study habits. We attributed SVM model to those variables as we achieve AUC in the range of 0.62-0.83 for one week ahead.

In the future, we will collaborate with course instructors to deploy our predictive models. And we will take more attention to why a student is failing, and what strategies make others' success in a SPOC or on-campus course.

5. ACKNOWLEDGMENTS

This research was supported by Teaching Research Funding in Honors College of Beihang University (2017) and Computer Information Specialty Construction Foundation Grant (No.201406025114).

6. REFERENCES

[1] Stapel, M., Zheng, Z., and Pinkwart, N. 2016. An ensemble method to predict student performance in an online math learning environment. In *Proceedings of the 9th International Conference on Educational Data Mining* (June 29 - July 2, 2016, Raleigh, NC, USA), 231-238.

[2] Elbadrawy, A., Studham, S., and Karypis, G. 2014. *Personalized multi-regression models for predicting students' performance in course activities*. Technical Report 14-011. University of Minnesota.

[3] Ren, Z., Rangwala, H., and Johri, A. 2016. Predicting Performance on MOOC Assessments using Multi-Regression Models. *arXiv preprint arXiv:1605.02269*.

[4] Bydžovská, H. 2016. A Comparative Analysis of Techniques for Predicting Student Performance. In *Proceedings of the 9th International Conference on Educational Data Mining* (June 29 - July 2, 2016, Raleigh, NC, USA), 306-311.

[5] Colin Taylor, Kalyan V., and Una-May O., 2014. Likely to stop? Predicting Stopout in Massive Open Online Courses. DOI = http://arxiv.org/pdf/1408.3382v1.pdf.

Social work in the classroom? A tool to evaluate topical relevance in student writing

Heeryung Choi
School of Information
University of Michigan
heeryung@umich.edu

Zijian Wang
Department of EECS
College of Engineering
University of Michigan
zijwang@umich.edu

Christopher Brooks
School of Information
University of Michigan
brooksch@umich.edu

Kevyn Collins-Thompson
School of Information
University of Michigan
kevynct@umich.edu

Beth Glover Reed
Social Work and Women's Studies
University of Michigan
bgr@umich.edu

Dale Fitch
School of Social Work
University of Missouri
fitchd@missouri.edu

ABSTRACT

In a climate where higher education institutions are actively aiming to increase inclusivity [2], we explore how a deep learning-based tool focused on text analysis is able to help assess how students think about issues of privilege, oppression, diversity and social justice (PODS). We created a vocabulary boosting and matching tool augmented with domain-specific corpora and relevance information. We find that the adoption of domain-specific corpora enhances model performance when identifying PODS-related words in short student-written responses to writing prompts, by building a more highly focused PODS vocabulary.

1. INTRODUCTION AND RELATED WORK

Universities are expanding their efforts toward creating more inclusive institutions of higher education [2]. One specific example is the principled blending of curricula with social justice and diversity issues in order to encourage PODS thinking (Privilege, Oppression, Diversity, Social justice) in the School of Social Work at the University of Michigan. PODS principles have been emphasized not only in individual courses but throughout the whole Social Work curriculum. Such a move naturally raises the question of scaled evaluation, both of individual students (e.g. formative or summative assessment) and programmatic evaluation.

In previous work, we explored mechanisms to detect elements of PODS thinking in student writing through semi-supervised machine learning [1]. We adopted the Empath tool [3] to generate an expanded vocabulary from a few seed words for PODS thinking detection, but were extremely limited in our ability to achieve accurate results. The first issue stems from the selection of large but general corpora which, while large in size and topic coverage, were not effective when we attempted to learn domain-specific bigrams. The other issue is how to filter less relevant words while boosting the size of the relevant lexicon. While generating a lexicon for Social Justice on Empath, we found that semantically irrelevant words like "therefore" and "yet" were in the output lexicon [1]. Thus, we expand on previous results and demonstrate a more robust and thorough treatment of the issues of detecting PODS thinking in student writing.

In this work, we consider the specific case of short student writings given in response to a writing prompt. Our goal is to build a technology solution that gives accurately coded responses and that enables instructors to identify quickly which students need elaborated feedback. The system will allow the instructors to focus remediation efforts on those who are of the highest need and to assess how well the overall curricula could increase PODS competency of students. Here we demonstrate the feasibility of using deep learning methods to detect evidence of PODS and apply these methods to a particular writing activity, innovating on the process used by others [3] to improve accuracy and reliability.

2. INSTRUMENTS

We created **Metapath**, a text analysis tool that allows users to use not only general corpora but also domain-specific corpora. Metapath is built on the ability of the Word2Vec model to calculate the similarity of concepts by mapping words and phrases to a vector space via a skip-gram model, and computing the cosine similarity of the corresponding vectors [4]. Given a word, the model gives users a 'most similar' word list ordered by the similarity score. In a preprocessing step, short words ($length \leq 2$), non-English terms, and most stopwords are considered as noise and removed from the corpora. After data cleaning, all words are stemmed using Porter stemming. Common phrases, i.e., multiword expressions, can be detected automatically by calculating mutual information gain within a threshold and minimum count. For example, the words 'Los Angeles' will become the phrase `los_angeles` after phrase detection while the model will return a list of high similarity words like `san_francisco` and `santa_barbara`. The judgment of whether the words are common phrases is based on the formula

$$\frac{cnt(a,b) - min\_count}{cnt(a) \cdot cnt(b)} \cdot N > threshold$$

where $cnt(a,b)$ means the frequency of word a and word b located together and N is the total vocabulary size.

We chose to use domain-specific corpora, i.e., MICUSP (Michigan Corpus of Upper-level Student Papers) and BAWE (British Academic Written English) [5], for detecting common phrases. The general Wikipedia corpus is used to train the model. In addition, considering the contextual nature of the PODS words, existing student responses gathered

from courses were included as a corpus. The domain-specific corpora are able to detect more related phrases on the topics of interest. For example, the proportions (10^{-3}%) of stemmed words like 'prejudic' and 'social_justic' in domain-specific corpora were relatively high (respectively 0.079 and 0.015), compared to the proportions of the same words in the general corpora, which were much lower (0.012 and 0).

3. EVALUATION

We conducted an evaluation to assess how well Metapath can assess PODS-related writing, using our domain-specific corpora, along two dimensions: comparing (1) inter-rater reliability (IRR) for PODS word annotation between human raters and Metapath and (2) IRR for quality evaluation between human raters and Metapath. The latter method is to include percentage of relevance of PODS words, which shows how semantically related each word is to seed words.

3.1 Data

The students' short written responses on PODS topic were used to evaluate Metapath, collected from four sections of a course offered in the School of Social Work ($n = 100$, word counts; $\bar{x} = 695.52$, $\sigma = 434.08$, $min = 115$, $max. = 2747$).

3.2 Approaches

For the evaluation, two expert human coders annotated PODS-related words in the student responses and evaluated overall PODS-relevance of each writing piece with three different marks: high, medium, and low. Their annotations and quality evaluation on student responses were compared with result of Metapath. To build a lexicon to evaluate PODS relevance of student writing, Metapath was boosted by essential PODS words, i.e., privilege, oppression, diversity, and social justice. Furthermore, two keywords from the writing prompt, i.e., "issues" and "actions", were also used to boost the PODS lexicon. After we boosted a lexicon (dim=500), the lexicon was used to calculate the IRR on annotations among two human raters and Metapath. The lexicon and its percentage of relevance were used to assess the overall PODS relevance of each response. After all the responses were ranked based on their percentage of relevance, they were categorized into high, medium, and low. The threshold of the each category was based on the proportion of each category decided by the human raters.

4. RESULTS AND DISCUSSION

We calculated group agreement among the two human raters and Metapath using Krippendorff's alpha (α). For the annotation comparison, IRR among two human raters alone is $\alpha = 0.4480$ ($n = 100$). When we added Metapath the overall group agreement dropped to $\alpha = 0.3804$ ($responses = 100$, $boosted\ words = 4300$, the maximum and minimum possible agreement the 3-rater scenario: $-0.4056 \leq \alpha \leq 0.6324$). IRRs between each human rater individual and Metapath were $\alpha = 0.1622$ and $\alpha = 0.1822$. For the quality evaluation, we achieved $\alpha = 0.3441$ ($responses = 100$, $boosted\ words = 660$) as the level of agreement between human raters and Metapath, which is close to the IRR between the two human raters ($\alpha = 0.4393$, the maximum and minimum possible agreement among 3-rater scenario: $-0.1875 \leq \alpha \leq 0.6223$). IRRs between each human rater individually and Metapath were $\alpha = 0.3702$ and $\alpha = 0.2234$. Overall, the evaluation showed that Metapath could identify PODS-related words and overall PODS relevance. The IRR that Metapath reached was close to those of human raters and not too low, considering the possible minimum and maximum agreement range.

It is worth pointing out that higher agreements in PODS word detection do not align with higher agreements in overall PODS relevance. We varied the size of Metapath's vocabulary by 500 words through setting *the number of boosted words* parameter. Even quite large vocabularies boosted the effectiveness of Metapath in the first task, declining only when values reached $n \approx 4000$. However, the IRR for quality analysis was the highest when $n = 660$.

Further research is needed to explore and improve the performance of Metapath. While identifying PODS-related words, there are still words and phrases in the field of social work that are not detected by Metapath, as noted by the experts. One way to address this is to focus on improved corpora, such as increasing the amount of response data generated by social work students and articles or books curated by PODS experts, or by using corpora based on accumulated Social Work student's writing. Finally, we note that this task is highly multifaceted, and here we have taken just a first pass at addressing it. Issues of personally-lived experiences, intersectionality of topics, and the nature of the writing prompt itself may require more traditional natural language processing techniques in order to capture deeper relationships in the text more fully.

5. ACKNOWLEDGEMENTS

The data used come from the British Academic Written English (BAWE) corpus, which was developed at the Universities of Warwick, Reading and Oxford Brookes, with funding from the ESRC (RES-000-23-0800). This study was funded in part with support from the Michigan Institute for Data Science (MIDAS).

6. REFERENCES

[1] H. Choi, C. Brooks, and K. Collins-Thompson. What does student writing tell us about their thinking on social justice? In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 594–595. ACM, 2017.

[2] E. DeRuy. The complicated process of adding diversity to the college syllabus. *The Atlantic*, Jul 2016.

[3] E. Fast, B. Chen, and M. S. Bernstein. Empath: Understanding topic signals in Large-Scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4647–4657. ACM, 2016.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[5] M. B. O'Donnell and U. Römer. From student hard drive to web corpus (part 2): The annotation and online distribution of the Michigan Corpus of Upper-level Student Papers (MICUSP). *Corpora*, 7(1):1–18, 2012.

Causal Forest vs. Naïve Causal Forest in Detecting Personalization: An Empirical Study in ASSISTments

Biao Yin
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609
byin@wpi.edu

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

Thanaporn Patikorn
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609
tpatikorn@wpi.edu

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

Jian Zou
Worcester Polytechnic Institute
100 Institute Rd.
Worcester, MA 01609
jzou@wpi.edu

ABSTRACT

It is widely understood that students learn in a variety of different ways and what is beneficial for one student may not necessarily help another. This work observes the effectiveness of Causal Forests as they compare to a new method we present called Naïve Causal Forests. This new method, aimed to be a simpler, more intuitive approach to identifying heterogeneous effects, is developed to better understand the strengths and limitations of the Causal Forest method. We apply these techniques to real student data on three RCTs run within the ASSISTments online learning platform.

Keywords

Personalization, Heterogeneous Treatment Effects, Randomized Controlled Trials, Causal Forest, Random Forest

1. INTRODUCTION

The idea that students approach learning in differing ways is not a new concept to researchers in the field of education, but how to leverage these computer-based systems for individualized learning is not always clear. Individualization, also referred to as personalization, also exists outside the field of education as well. In other fields, this idea is described through heterogeneous treatment effects, as the effect of a particular treatment or intervention is not often homologous across all individuals. The introduction of computer-based systems in the classroom makes it feasible to supply aid to individuals allowing the teacher to focus on helping those students struggling most.

Recently, a technique known as a Causal Forest (CF) [8] has been developed, applying random forests to the task of identifying heterogeneous effects. This work explores a new, more intuitive method for identifying heterogeneity as it compares to the more complex CF method. This new method, called Naïve Causal Forest (NCF), attempts to employ a simpler approach based on the structure of CF to answer: 1. To what extent, if any, does the Causal Forest method outperform our simpler, more intuitive approach to identifying heterogeneous treatment effects in real student data? and 2. Do these models converge to large differences when compared using increasing sample sizes?

2. DATASET

The dataset used to build and evaluate our method is comprised of student information on 3 randomized control trials (RCTs) run within the ASSISTments online learning platform [2] from a previously published dataset [5]. ASSISTments is a free web-based platform where a recent efficacy trial found the system to be effective in improving student learning [4], motivating further study to better understand student behavior and measure effects within the platform.

After filtering the data to remove students with missing values, the Experiment 1 contains 519 students, the Experiment 2 contains 833 students, and Experiment 3 contains 1118 students.

3. METHODOLOGY

The Causal Forest (CF) method [8] has established itself as a viable model for identifying heterogeneous effects, for which we do not refute, but rather we wish to explore the benefits of this more complex method to a simpler, more intuitive approach. CF uses estimates of treatment effects within the splitting rule of a random forest algorithm; an "honest" variant uses a holdout set to estimate the effect for each split. Heterogeneous effects can be determined by observing students who then are grouped into different leaves of the generated trees. Our new method, which we have called Naïve Causal Forest, aims to implement a simpler approach that excludes the use of condition from the random forest until students are grouped into each leaf, where then an average treatment effect is calculated across each subgroup. In both methods, each tree has a "vote" as to what condition will benefit the students most.

Figure 1: The 10-fold cross validation results for experiments 1 and 2 comparing NCF to an honest CF model. No reliable differences are found between the two methods, and both appear consistent with increases to the number of generated trees.

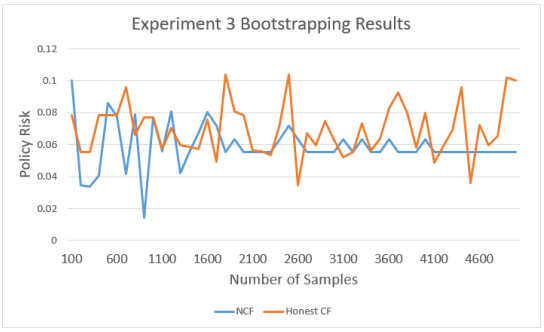


Figure 2: Experiment 3 bootstrapping results comparing NCF to two Causal Forest models.

We compare CF, implemented in R [3] using a Causal Tree package [1], and NCF in their ability to identify heterogeneous effects for the purpose of maximizing completion of the assignment. We calculate the Odds Ratio [7] within each leaf to identify which condition corresponds with the higher student completion rate within each leaf. We evaluate our models using a measure known as policy risk [6], where a lower value indicates better performance. This metric is used to compare the two methods for each experiment as the metric is not directly comparable across experiments.

4. DISCUSSION AND FUTURE WORK

The result of our 10-fold cross validation analysis can be seen in Figure 1. Both models use a minimum leaf size of 30, and are evaluated over several model complexities. In all three experiments, it is found that the CF and NCF model exhibit no reliable differences. It is also the case, however, that no significant heterogeneous effects are found by either method. Figure 2 illustrates how the methods converge with increasing sample sizes using a bootstrapping method of sampling with replacement on the largest experiment.

We compare in this work the Causal Forest method for identifying heterogeneous treatment effects to our Naïve Causal Forest method and find no reliable differences between the simpler and more complex methods. It is expected, and planned for future work, that applying these methods to experiments with larger sample sizes may show statistic reliability.

We also found that the CF model exhibited stable policy risk over increases to model complexity. This is a desirable quality of a prediction model, as it is data driven and less sensitive to changes in model structure. We found that the CF model exhibited non-converging behavior when bootstrapping, but may additionally be caused by insufficient variation or lack of heterogeneity in the dataset.

5. ACKNOWLEDGMENTS

We thank multiple current NSF grants (IIS-1636782, ACI1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428 & DRL-1031398), the US Dept. of Ed (IES R305A120125 & R305C100024 and GAANN), and the ONR.

6. REFERENCES

[1] S. Athey, G. Imbens, and Y. Kong. *causalTree: Recursive Partitioning Causal Trees*, 2016. R package version 0.0.

[2] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[4] J. Roschelle, M. Feng, R. F. Murphy, and C. A. Mason. Online mathematics homework increases student achievement. *AERA Open*, 2(4), 2016.

[5] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184. ACM, 2016.

[6] U. Shalit, F. Johansson, and D. Sontag. Estimating individual treatment effect: generalization bounds and algorithms. *arXiv preprint arXiv:1606.03976*, 2016.

[7] M. Szumilas. Explaining odds ratios. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, 19:227, 2010.

[8] S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *arXiv preprint arXiv:1510.04342*, 2015.

An Offline Evaluation Method for Individual Treatment Rules and How to Find Heterogeneous Treatment Effect

Thanaporn Patikorn, Neil T. Heffernan, Jian Zou
100 Institute Rd.
Worcester, MA 01609
{tpatikorn, nth, jzou} @wpi.edu

ABSTRACT

Heterogeneous treatment effects occur when the treatment affects different subgroups of population differently. In this work, we conducted a large scale simulation study to identify the characteristics of treatments that are more likely to have heterogeneous treatment effects, and to estimate how effective the individual treatment rules are compared to the better conditions. We found that heterogeneous treatment effects are rare. When the overall treatment effect is close to zero, we found that individual treatment rule is very likely to be effective. With large positive or negative overall treatment effect, the heterogeneous treatment effect is less likely to occur, and the individual treatment rules are more likely to be ineffective.

Keywords

Heterogeneous Treatment Effect; Individual Treatment Rule; ASSISTments; Randomized Controlled Experiment.

1. INTRODUCTION

Researchers have been using randomized controlled experiments (RCT) to test their interventions. RCTs are considered the gold standard and are widely used in many fields, from healthcare to education. Traditionally, researchers often look for treatment effects across the population. However, in many experiments, the treatment effect differs systematically from one subgroup of the population to another. For example, patients who are allergic to the treatment drugs may react negatively instead of benefiting from the drug. This type of effect is often called heterogeneous treatment effects, as there are different effects for different types of people. Many machine learning methods have been developed to detect heterogeneous treatment effects. For example, [4] introduced the Causal Forest, a decision tree-based method to determine the treatment effect on each subgroup of the population.

In many cases such as [1], it is better to tutor students with lower prior knowledge using step-by-step hints, while it is better to tutor students with high prior knowledge with full problem solutions. In this case, giving personalized tutoring to each student is better than giving the same tutoring to everyone. This type of condition assignment is often called an individual treatment rule or a personalization policy.

In order to evaluate a personalization policy, the most popular method is to deploy the policy in real time and compare the result. However, the on-line method is often costly and sometimes unavailable to the researchers (e.g. because the data have already been collected). As a result, many researchers conduct an offline policy evaluation using past data. In [3], they use the expected outcome of the policy to evaluate their personalization policy. To calculate the expected outcome using past RCT data, we must first find a subset of subjects whose random condition assignments during the RCT matches the personalized condition assignments of the policy. The expected outcome of a personalization policy is the average outcome of this subset across conditions. Comparing two policies using the expected outcome easy and intuitive; if the larger outcome values are better, the policy with larger expected outcome is better. This method is equivalent to policy risk introduced in [2].

The main goals of this work are 1) to find the characteristics of the experiments that are more likely to have heterogeneous treatment effects, and 2) to compare a personalization method, specifically Causal Forest, against assigning every subject to the best conditions to find out how effective a personalization policy can be.

2. METHODOLOGY

In order to gain a better understanding of expected outcome, we investigated how it is calculated in [3]. They first took the subset of the subjects from the RCT whose random condition assignments are the same as the condition assignments given by a personalization policy. For the rest of this paper, we will refer to this subset as the "congruent subset". Then, the expected outcome of the policy is calculated by taking the average outcome values of the congruent subset regardless of conditions. For example, in Table 1, the congruent subset consists of subject 1, 3, 4, and 5, and the expected outcome of the policy is (0.7 + 0.4 + 0.6 +0.7)/4 = 0.6.

2.1 Simulation Study

We conducted a large-scale simulation study to verify the effectiveness of using the congruent subset as an estimate of real outcome values of the policy, and to find types of experiments that are likely to have personalization. We chose simulation study because it allows us to not only calculate the real outcome values of the policy, but also investigate how different settings impact the personalization.

Table 1: an example data to show how congruent subset works

subject	RCT condition	outcome	personalized condition	Is in congruent subset?
1	C	0.7	C	yes
2	T	0.6	C	no
3	C	0.4	C	yes
4	T	0.6	T	yes
5	T	0.7	T	yes
6	C	0.5	T	no

Table 2: Different Distributions for Effect of Conditions

distribution	parameter	values	number of combinations
normal	mean	0, 1, 2, 5, 10	15
	sd	1, 2, 5	
log normal	meanlog	0, 0.5, 1, 2	16
	sdlog	0.25, 0.5, 1, 2	
gamma	shape	0.5, 1, 2, 5, 10	15
	scale	0.5, 1, 2	
total			46

For the simulation study, we focused only on experiments with two conditions. For each condition, we simulated 46 different settings, as shown in Table 2, resulting in 46 * 46 = 2116 different combinations of experiments. We also include lognormal distributions and gamma distributions because real datasets may not always follow normal distributions, for example the mastery speed in [5] resembles lognormal distribution. For each setting, we generated 1000 datasets, each of which has 1000 data points.

Every data set has 3 covariates: one with a positive, negative, and no effect on the outcome. Every covariate value is generated independently for each subject from a normal distribution with mean = 0 and sd = 1. The true effect is generated using the distribution and parameters in Table 2. The observed outcome is

$$\text{observed} = \text{effect} + \text{cov1} * \text{impact1} - \text{cov2} * \text{impact2} + \text{noise}$$

The impacts are from uniform (0,5) and remains constant within experiment. The noise is drawn from a normal (0,1) distribution.

For each personalization policy, we measured 1) if the outcome values of congruent sets are significantly different from the outcome values of actually assigning everyone using personalization policy, and 2) whether the personalization from the Causal Forest is better than the better of the two conditions.

3. RESULTS

From 2,116,000 simulated dataset, we detected the significant difference between the outcome values of the congruent sets and the real personalized outcome values less than 1% of the time, which is far lower than the threshold of 5%, regardless of parameters of the dataset. As for the effectiveness of the Causal Forest, we look at how often the personalization suggested by Causal Forest are better than assigning subjects to the better of the two conditions. We found that personalization is slightly more common when at least one of the distribution is gamma distribution.

Table 3: the Effectiveness of Personalization Suggested by Causal Forest by Overall Observed Treatment Effect

Rounded average observed treatment effect	Causal Forest suggests personalization	Causal Forest's personalization is the most effective
≤ -5	0.03%	15.26%
-4	0.04%	23.19%
-3	0.12%	22.46%
-2	0.41%	44.44%
-1	2.98%	76.43%
0	8.27%	83.56%
1	3.03%	76.26%
2	0.43%	44.79%
3	0.12%	20.76%
4	0.05%	23.35%
≥ 5	0.03%	14.67%

Table 3 shows that when the treatment effect is close to zero, the personalization suggested by the Causal Forest is very effective. Causal Forest policy is better than assigning subjects to the better of the two conditions more than 3/4 of the times when the treatment effects are between -1 and 1. The effectiveness of the personalization quickly drops as the treatment effect is far from zero. It is important to note that the Causal Forest we used in this study has never been optimized and most of parameters we used are default, except the two we specified earlier in the paper.

4. CONCLUSION

This paper has three main contributions. First, we promoted the study of heterogeneous effects and an offline personalization policy evaluation method to the Educational Data Mining. Second, we investigated several different settings of simulated experiments to find the characteristics of the experiments that are more likely to have heterogeneous treatment effects. We found that, generally heterogeneous treatment effects are not common and typically rare when the treatment effects are very large or very small. Third, we investigated the effectiveness of personalization policies given by Causal Forest. We found that the personalization policy is likely to be effective for the experiments with small treatment effects.

5. FUTURE WORK

We plan to investigate different methods for detecting heterogeneous treatment effects on real dataset from ASSISTments to see if we can detect more experiments like [1]. If we can detect such effects, we would be able to improve our system even further, which will improve student learning.

We also plan to compare different methods for detecting heterogeneous treatment effects to see what are the advantages and disadvantages of each model. We also plan to compare these pre-train models to real-time methods like bandits as well. This result will allow us to be able to choose the right tool for the right personalization task.

6. ACKNOWLEDGMENTS

We thank multiple NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428 & DRL-1031398), the US Dept. of Ed (IES R305A120125 & R305C100024 and GAANN), and the ONR.

7. REFERENCES

[1] Razzaq, L. M., & Heffernan, N. T. (2009, July). To Tutor or Not to Tutor: That is the Question. In *AIED* (pp. 457-464).

[2] Shalit, U., Johansson, F., & Sontag, D. (2016). Estimating individual treatment effect: generalization bounds and algorithms. *arXiv preprint arXiv:1606.03976*.

[3] Vickers, A. J., Kattan, M. W., & Sargent, D. J. (2007). Method for evaluating prediction models that apply the results of randomized trials to individual patients. *Trials*, *8*(1), 14.

[4] Wager, S., & Athey, S. (2015). Estimation and inference of heterogeneous treatment effects using random forests. *arXiv preprint arXiv:1510.04342*.

[5] Xiong, X., Li, S., & Beck, J. E. (2013, May). Will You Get It Right Next Week: Predict Delayed Performance in Enhanced ITS Mastery Cycle. In FLAIRS Conference

MyCOS Intelligent Teaching Assistant

Jiao Guo
MyCOS
Wanliuyichen Center A-18
Beijing China 100083
(+86)1058819001-352
amanda.guo@ mycos.com

Xinhua Huang
MyCOS
Wanliuyichen Center A-18
Beijing China 100083
(+86)1058819001-352
xinhua.huang@mycos.com

Boqing Wang
MyCOS
Wanliuyichen Center A-18
Beijing China 100083
(+86)1058819001-169
boqing.wang@ mycos.com

ABSTRACT
In this preliminary study, we introduce MyCOS Intelligent Teaching Assistant (MITA). It is an open learning platform tailored for a specific challenge of Chinese universities, i.e., undergraduates report less student-faculty interaction than those in the U.S.. Compared with existing classroom tools like Socrative, MITA leverages the app-within-an-app model of WeChat (the largest social app in China) instead of a stand-alone app. Which model is the future is debatable. MITA also uses prompt feedback to engage learners and dashboards to inform teachers and administrators. It now serves more than 3,200 teachers and near 110,000 students from 600+ Chinese universities. What the data from the platform reveal about learning deserves further study.

Keyword
Open learning platform, student engagement

1. INTRODUCTION
Researchers found that the gap in student-faculty interaction (SFI) between Chinese universities and their American peers. Based on a comparative study of 2009 National Survey of Student Engagement (NSSE) results, 27% Tsinghua (a Chinese research university) undergraduates had never received prompt feedback from faculty on academic performance while the average in the American research universities was 7% [1].

MyCOS Intelligent Teaching Assistant (MITA) is an open learning platform tailored to the context of Chinese universities. Different from existing tools such as Socrative, MITA enables teachers to interact with students through the app-within-an-app model of WeChat (the most popular social app in China). Whether this model is better than a stand-alone app to engage college students is debatable. It would be interesting to explore similar learning tools that leverage Facebook or other social apps in different countries and then compare.

Inspired by the 2011 proposal of open learning analytics [2], MITA tracks learner behaviors and provides prompt feedbacks. It has data dashboards for teachers (see Figure 1) and administrators to monitor learning process and take informed actions. Since launched in September 2016, MITA has been used by more than 3,200 teachers and near 110,000 students in 600+ Chinese universities. It is a real case of collaboration across research, industry and education sectors. The fast development and nationwide deployment of MITA can produce data useful for further study.

The rest of the poster sections is organized as follows. In section 2 we describe the data sample; in section 3 we report the learning behavior patterns the data reveal; in section 4, we discuss the need for further analysis.

2. DATA SAMPLE
The sample used in this preliminary study was selected from MITA clickstream data between 2016/09/10 and 2017/02/06. During the time period, 1,599 teachers and 45,383 students registered. Among them, 766 teachers and 32,305 students have verified their institute information and interacted through MITA at least once. They are defined as active teachers and active students in this study.

To assess student engagement, we focus on the related learning patterns the MITA data reveal. Specifically, the patterns discussed below (in section 3) are student attendance, quiz participation and questions answered.

The sample covers 278 Chinese universities, including 199 fouryear universities (71.6%) and 99 three-year vocational colleges.

3. BEHAVIOR PATTERNS
3.1 Student Attendance
Existing studies on student attendance were limited within an institution, e.g., a 2015 research on 2,141 classes of a four-year Chinese university found the average attendance rate of 89% [3]. The student attendance pattern based on the MITA sample extends to nationwide and the numbers fall within a reasonable range. The average attendance rate is higher in three-year vocational colleges (92.8%) than that of four-year universities (89.2%).

Daily attendance behaviors demonstrate a similar pattern: the attendance rate of three-year vocational colleges is higher than that of four-year universities every weekday except Friday. The lowest daily attendance rate for three-year colleges is on Friday (88.9%) while for four-year universities is on Monday (87.9%). Hourly attendance behaviors show a common challenge for both categories of universities: classes scheduled in the evening (6-9 pm) have the lowest attendance rates (85% for three-year vocational colleges and 83.9% for four-year universities).

3.2 Quiz Participation
Quiz participation is one of indicators used by researchers to monitor online learning behaviors [4]. MITA enables us to conduct the similar learning analysis in a real classroom. When students take a quiz in class by MITA, they can view the progress in realtime and get the feedback immediately after submission. With the fine-grained data, the teacher can check who participate, who get the answer wrong and which part of the course content is most challenging.

Based on the MITA sample, the quiz participation rate on average is 84.5% for 3-year vocational colleges and 81.7% for 4-year universities. Both are higher than the quiz participation rate in MOOCs. A 2014 study found that 40%~70% learners completed zero quiz in two live-MOOCs (i.e. in-session, instructor-led course with possibility of obtaining a statement of achievement) [5].

3.3 Questions Answered

Asking questions is one of teaching strategies used in college classroom. In a 2013 study, a researcher observed 30 English classes in a four-year Chinese university for two months. She also surveyed 25 teachers and 237 students to analyze the behaviors of asking and answering questions in class [6]. Data collection becomes more efficient with MITA. Based on the MITA sample data, nearly half teachers in three-year vocational colleges (51.7%) use MITA to ask questions in every class session. The proportion is lower in four-year universities (41.6%).

The proportion of answering questions, however, is quite low for students. The MITA data show that 96.7% students in three-year vocational colleges and 98% in four-year universities never answered a question in class. The result looks plausible given the large class size in the sample: 36.8% classes in three-year vocational colleges and 47.2% classes in four-year universities are larger than 50 students. It indicates that some alternative strategy (e.g., an open question in a quiz) can engage more students.

4. DISCUSSION

The focus of this preliminary study is to enhance student-faculty interaction in a real classroom. Besides, MITA has the data on learning behaviors before class (e.g. viewing the course PPT) and after class (e.g. submitting an assignment) for further exploration.

Further study is using EDM & LA (e.g. user behavior modeling) to explore the MITA data in terms of student motivation, performance and satisfaction. More clickstream data (e.g., the number of attempts students try with a quiz) can be collected and analyzed. Different learning patterns can be compared across not only institutional type (four-year universities vs. three-year vocational colleges) but also class size (small, medium and large) or course type (required courses vs. elective courses). The comparison can provide actionable information for teachers and administrators.

Based on the 2015 IMPACT report from Purdue University, nearly half faculty (48%) chose the ICT-supplemental learning model to redesign their courses, 46% chose the hybrid or flipped model and only 6% chose online-only [7]. It indicates the possibility of developing and deploying MITA or similar learning tools for a real classroom in different countries. Experiments of Facebook in classroom has been explored in the U.S. [8], Canada [9], and Singapore [10], but more third-party applications like MITA are needed to extend the capability of Facebook as a learning tool and more debate on whether we should ban or embrace using such a tool is ongoing.

Figure 1. Teacher Dashboard of MyCOS Intelligent Teaching Assistant (MITA).

5. REFERENCES

[1] Ross, H., Cen, Y. and Zhou, Z. 2011. Assessing Student Engagement in China: Responding to Local and Global Discourse on Raising Educational Quality. Current Issues in Comparative Education, Vol. 14(1): 24-37

[2] Siemens, G., D. Gasevic, C. Haythornthwaite, S. Dawson, S. B. Shum, R. Ferguson, E. Duval, K. Verbert, and R. S. J. d. Baker. 2011. *Open Learning Analytics: An Integrated & Modularized Platform*. SoLAR. DOI= http://www.elearnspace.org/blog/wp-content/uploads/2016/02/ProposalLearningAnalyticsModel_SoLAR.pdf

[3] Yao, L.M., Zhu, L.M. and Hu, J.L.2015. Survey and Analysis on College Student Attendance. Jiangsu Higher Education, Vol.15(3):67-70

[4] Wang, Y. 2014. *MOOC Learner Motivation and Learning Pattern Discovery: A Research Prospectus Paper*. In the Proceedings of the 7th International Conference of Education Data Mining, DOI= http://educationaldatamining.org/EDM2014/uploads/procs2014/YRT/452_EDM-2014-Full-Proceedings.pdf

[5] Campbell, J., Gibbs, A., Najafi, H. and Severinski, C. 2014, A Comparison of Learner Intent and Behavior in Live and Archived MOOCs, The International Review of Research in Open and Distributed Learning, Vol.15(5) DOI= http://www.irrodl.org/index.php/irrodl/article/view/1854/3097 [6] Tian, J. 2013. *A Study on the Pattern of Asking Questions in College English Classes*. Shanxi Finance & Economics University. DOI= http://cdmd.cnki.com.cn/Article/CDMD-10125-1013203176.htm

[7] Purdue University. 2015. *Instruction Matters: Purdue Academic Course Transformation (IMPACT) Annual Report.* DOI=https://www.purdue.edu/impact/assets/documents/IMPACT%20annual%20report%202015(I).pdf

[8] Walsh, K. 2011. *Facebook in Classroom, Seriously.* EmergingEdTech. DOI= http://www.emergingedtech.com/2011/03/facebook-in-theclassroom-seriously/

[9] Malhotro, N. 2013. *Experimenting with Facebook in College Classroom*. Faculty Focus. DOI= https://www.facultyfocus.com/articles/teaching-withtechnology-articles/experimenting-with-facebook-in-thecollege-classroom/

[10] Wang, Q., Woo, H. L., Quek, C. L., Yang, Y. and Liu, M. 2012. Using the Facebook group as a learning management system: An exploratory study. British Journal of Educational Technology, Vol. 43(3):428-438

Towards Automatic Classification of Learning Objects: Reducing the Number of Used Features

Pedro González[1], Eva Gibaja[1], Alfredo Zapata[2], Víctor H. Menéndez[2], Cristóbal Romero[1]
[1]University of Cordoba, Dept. of Computer Science, 14071, Córdoba, Spain
[2]Autonomous University of Yucatan, Faculty of Education, 97305, Mérida, Mexico
{pgonzalez,egibaja,cromero}@uco.es, {zgonzal, mdoming}@correo.uady.mx

ABSTRACT

The automatic classification of LOs into different categories enables us to search for, access, and reuse them in an effective and efficient way. Following this idea, in this paper, we focus specifically on how to automatically recommend the classification attribute of the IEEE LOM when a user adds a new LO to a repository. To do it, we propose the use of the multi-label classification approach, since each LO might be simultaneously associated with multiple labels. An initial problem we have found is that the number of terms or pure text features that characterize LOs tends to be very high. So, we propose to apply a dimensionality reduction process. We have carried out an experiment using 515 LOs from the AGORA repository in order to try to reduce the number of features or attributes used, improving execution time without losing prediction accuracy.

Keywords

Multi-label classification, feature selection, learning object

1. INTRODUCTION

The IEEE Learning Object Metadata standard (IEEE LOM) defines several attributes that may be assigned to each Learning Object (LO). However, manual entering all these metadata is a time-consuming process and automated techniques are required for a wider adoption of the standard [2]. In this paper, we focus on how to automatically recommend the classification attribute of the IEEE LOM when a user adds a new LO to a repository. Our idea is to recommend the user what are the possible categories that a LO belongs to from just user-provided information about the LO (such as the title, keywords and description). In order to do it, we propose to use multi-label classification for automatic categorization of LOs from the terms or pure text features that characterize these LOs. Multi-label classification (MLC) is a variant of the classification problem where multiple target labels can be assigned simultaneously to each instance [1]. In traditional classification classes are mutually exclusive, that is, a specific instance can belong to just a single class. However, there are occasions where classes present overlapping, that is, a specific instance can belong to several classes. In our case, we use MLC because a specific LO could belong to several categories.

2. PROPOSED METHODOLOGY

Our proposed approach for automatically classifying of LOs is represented in figure 1. First, we create the data file starting from the terms or pure text features that characterize LOs extracted from the LOs metadata, and categories to which the LO belongs to. Therefore, our next step consists in performing an attribute selection. The final step is the application of a MLC algorithm that will give us a model for classifying new LOs.

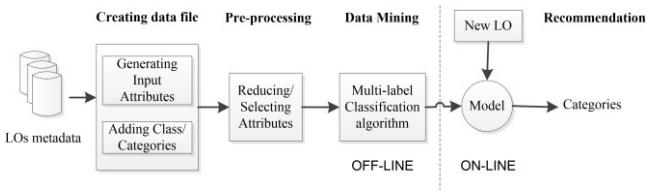


Figure 1. LO multi-label classification approach.

3. EXPERIMENTAL WORK

The data file used in this work has been extracted using 515 LOs from the AGORA repository [3] as follows. When a user adds a new LO to AGORA, he must provide information such as title, keywords, description and other related IEEE LOM metadata. Starting from these information about all the LOs we extracted 1336 terms (features) after removing stop words and stemming (to reduce the terms to their roots). Next, we compute the frequency of these roots for the LO at issue obtaining its term frequency (TF) representation. So, we obtained an example-term matrix, in which each element represents how many times a term appears in an example. We also normalized the count to term frequency to measure the importance of a term. Besides, in AGORA, a user has to specify one or several categories to which the LO belongs to from a predefined set of five academic disciplines: Engineering and Technology; Natural and Exact Science; Social and Administrative Science; Education, Humanities and Art; Health Science. So, we added the 5 labels (in binary format) to each LO as classes to predict. Then, we applied a dimensionality reduction process for reducing the number of attributes in the dataset. The motivation is to reduce training and classification times and removing noisy and irrelevant attributes, which can have a negative impact on accuracy results. Usually, there exists a wide range of possible terms that can refer to LOs of very different topics, and hence, the number of attributes describing LOs tends to be very high. Feature selection has been performed according to a specific method for MLC suggested in [5]. First, the χ^2 feature ranking method was separately applied to each label. Thus, for each label, the worth of each attribute is estimated by computing the χ^2 statistic with respect to the label to determine its independence. The core idea is that, if an attribute is independent on a class, this attribute could be removed. The result of this step is a ranking of all features for each label according to the statistic. Finally, the top-*n* features were selected based on their maximum rank over all labels. Finally, 13 different state-of-the-art MLC algorithms [1] have been applied to the different versions of the data set. They include 3 adaptation algorithms: AdaBoost.MH, Multi-Label k-Nearest Neighbor (MLkNN) and Instance-based Logistic Regression (IBLR), and 10 transformation algorithms in which the J48 implementation of C4.5 decision tree algorithm has been used as base classifier: Binary Relevance (BR), Classifier

Chais (CC), Calibrated Label Ranking (CLR), Label Powerset, Prued Sets (PS), Ensemble of Pruned Sets (EPS), Ensemble of Classifier Chains (ECC), Random-k-LabelSets (RAkEL), Hierarchy Of Mul-tilabel classifiERs (HOMER) and Stacking. The MULAN software for MLC [4] has been used for running both the feature selection method and the MLC algorithms. We have used a 10-fold cross validation with 10 seeds. Our experimentation takes into consideration two main factors: number of attributes and MLC performance. Overall, the time employed by a MLC algorithm to generate a model will be proportional to the number of training instances and the number of attributes describing each instance. So, if we reduce the number of attributes then the computational cost will be reduced as well. However, as a reduction of the number of attributes could discard relevant information, the induced model could perform poorly. This is why we have performed an attribute selection with different reduction levels in order to determine the more suitable reduction level without damaging the classification performance. Our original data set contains 515 LO instances, each one characterized by 1336 attributes. From these, we have selected 1000, 750, 500, 250, 150, 100 and 50 attributes with highest ranking to create different datasets. Next, we have applied 13 MLC algorithms to each different version of the data set, in order to know if there are differences in computational costs and performance by checking some evaluation measures. Therefore, in addition to train time the next five multi-label evaluation measures have been computed: a) Example-based metrics: Hamming loss (H-loss) and Accuracy (E-Acc) b) Label-based measures: Accuracy (L-Acc) and c) Ranking-based measures: Ranking loss (R-loss) and Average precision (A-Pre). On the one hand, we have found a significant reduction of computational costs as the number of features decrease (Figure 2), especially up to 250 features. The algorithms reducing training time at higher degrees are ECC, RAkEL and EPS.

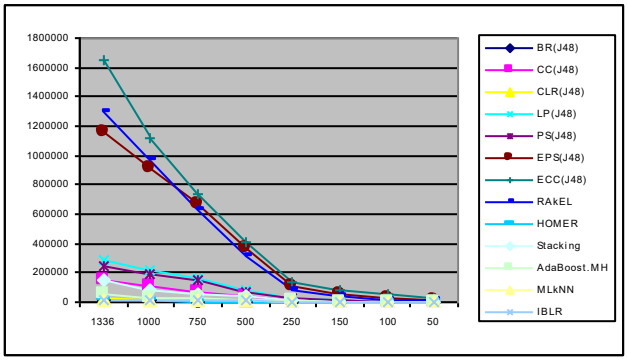


Figure 2. Training time (milliseconds).

On the other hand, in order to compare the classification performance of the algorithms, a Friedman test has been carried out for each evaluation metric by considering results for each feature reduction level. Ranking values and p-values are detailed in Table 1. These p-values (≤ 0,05) show significant differences between reduction levels with high confidence level (95%). We can also observe that for Ranking loss (R-loss) and Average Precision (A-Pre), the best ranking value is obtained for 1000 features instead of the original 1336 features. Besides, a meta-ranking (the rank of rank) of reduction levels was built performing another Friedman test. This way we can evaluate which number of features has the best overall performance in most of the metrics. The last column of Table 1 shows the resulting meta-rank. It is interesting to see that the best ranking does not correspond to the complete feature set. As the test detected significant differences between reduction levels (p-value ≤ 0,01), a Bonferroni-Dunn test was performed. This test found that algorithms performed significantly worst with less than 250 attributes at 95% confidence level. So, we established 250 as the optimum reduction level.

Table 1. Avg. rankings for all metrics and reduction levels.

Number Features	↓H-loss	↑E-Acc	↑L-Acc	↓R-loss	↑A-Pre	Meta Rank
1336	**2,92**	**3,07**	**2,92**	4,50	4,19	2,60
1000	3,76	3,23	3,76	**3,11**	**3,11**	**2,40**
750	3,11	3,57	3,11	3,88	3,42	2,80
500	2,96	3,34	2,96	4,42	3,76	2,80
250	4,19	3,96	4,19	3,96	3,88	4,40
150	5,73	5,57	5,73	4,88	5,46	6,00
100	6,50	6,50	6,50	5,96	6,23	7,40
50	6,80	6,73	6,80	5,26	5,92	7,60
p-values	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Finally, a comparison of 13 MLC algorithms when using the optimum reduction level (250 features) has been performed. The goal was to identify which algorithm yields the best results in this specific dataset considering the previous 5 evaluation metrics. The algorithm with the overall best results in the five evaluation measures (higher in E-Acc, L-Acc and A-Pre; and lower in H-Loss and R-Loss) was RAkEL. So, this algorithm will be used in our proposed approach for recommending the categories to which the new LOs belong. In the future we want to use more evaluation measures and also information about LO usage in order to try to improve classification performance.

4. ACKNOWLEDGMENTS

Authors gratefully acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2014-55252-P.

5. REFERENCES

[1] Gibaja, E., Ventura, S. 2014. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4, 6, 411-444.

[2] Kannampallil, T. G., Farrell, R. G. 2005. Automatic Learning Object Categorization For Instruction Using An Enhanced Linear Text Classifier. *Knowledge Management: Nurturing Culture, Innovation, and Technology*, 299-304.

[3] Menéndez, V., Prieto, M., Zapata, A. 2010. Sistemas de gestión integral de objetos de aprendizaje, *Revista Iberoamericana de Tecnologias del Aprendizaje*, 5, 2, 56-62.

[4] Tsoumakas, G., Spyromitros-, E., Vilcek, J., Vlahavas, I. 2011. Mulan: a java library for multi-label learning", *Journal of Machine Learning Research*, vol. 12, 2411-2414.

[5] G. Tsoumakas, I. Katakis, I. Vlahavas. 2011. Random k-labelsets for multilabel classification", *IEEE Transactions on Knowledge and Data Engineering*, 23, 7, 1079-108.

The Reading Ability of College Freshmen

Andrew M. Olney
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
aolney@memphis.edu

Breya Walker
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
bswlker2@memphis.edu

Raven N. Davis
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
rndavis2@memphis.edu

Art Graesser
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
graesser@memphis.edu

ABSTRACT
Over the past 50 years, an increasing proportion of student graduating high school attend college, but literacy levels in the United States have remained largely unchanged. We present preliminary results that suggest the literacy levels of assessed first year college freshmen are above 5th grade but below 12th grade, that only 32% of these freshmen are reading at a 12th grade level, and that this high-performing group has only a 69% chance of passing the reading portion of the GED high school equivalence test.

Keywords
adult literacy, higher education, NAEP, TABE

1. INTRODUCTION
The percentage of high school graduates immediately attending college has steadily increased from 60% in 1990 [5] to 69% in 2015 [2]. However, during this same period the average reading score of 12th grade students on the National Assessment of Educational Progress (NAEP) has declined slightly, such that in 2015, only 37% of students were deemed proficient readers [6]. If all proficient readers immediately attend college, then only 54% of college freshmen are proficient readers. Accordingly, the remaining 46% of college freshmen are either basic or below basic readers.

While it is alarming to think that approximately half of college freshmen are not proficient readers, the NAEP proficiency criteria and cut scores are not without controversy [1]. For example, in a recent mapping of NAEP standards to state standards for 8th grade reading (the highest grade available), only one state was found to have standards aligned with NAEP's proficient category. Given the controversy, it is not clear if the NAEP standards are too high or the state standards are too low.

To better understand the relationship between NAEP reading scores and college freshmen reading ability, we conducted a pilot study using questions from the Reading section of the Tests of Adult Basic Education (TABE). The TABE [3, 4] is useful for exploring the question of reading proficiency of college freshmen because i) TABE items have national norms and are aligned with grade equivalences, allowing us to categorize freshmen reading ability according to grade level and ii) TABE can be used to predict General Educational Development (GED) test performance, which is a proxy for determining whether a participant's reading ability is high school equivalent.

2. METHOD
2.1 Participants
Participants (N = 1062) were recruited through the psychology subject pool at an urban university in the southern United States in two waves of online data collection. The first wave (N = 313), which took place during the spring semester of 2015, was conducted as a regular online study, but the second wave (N = 749), which took place during the fall semester of 2015, was conducted as a screening component for the entire subject pool. Subject pool screening is used to determine eligibility for other studies later in the semester and therefore represents an even more diverse group of participants, as it largely eliminates the self-selection bias of experimental sign up. No demographics of participants were collected.

2.2 Materials
Ten items (#4-13) were selected from the nationally-normed, TABE 10 Form D Reading Survey. Form D (Difficult) is designed to assess reading ability in grade ranges 6.0 - 8.9 and therefore may seem a less obvious choice for assessing college freshmen. However, Form D items cover the widest range of grade equivalents (grades .7 - 12.9) of all TABE 10 forms and therefore has some additional utility when the underlying grade level is unknown. Because the 10 items used in the present study were selected from the 25-item TABE 10 Form D Reading Survey, the distribution of grade equivalents for items does not match the distribution of the complete survey and instead falls into three clusters: five items are at grades 4-5 (3.9, 4.4, 4.8, 5.1, and 5.2), three items are at grades 11-13 (11.4, 12, and 12.9), and two items are at grades 6-7

(6.2 and 7). All items had multiple choice format with four response options.

2.3 Procedure

Participants completed the informed consent and the 10 items using a web browser. Because the study was online and not proctored, the time guidelines of the TABE (approximately 1 minute per question) were not enforced, and due to technical problems, the time participants spent on the items could not be determined. Participants read each of three text passages in turn and answered three to four items after each passage by selecting a multiple-choice response option.

3. RESULTS

Overall, 75% of participants answered 80% or more items correctly, suggesting that the 10 items were overall too easy, as recommendations for TABE specify that participants answer 40% to 75% of the items correctly [4]. Participant performance varied across item difficulty cluster, however. While 73% of participants answered all five items correctly in the 4-5th grade cluster, only 32% answered all three items correctly in the 11-13th grade cluster. Furthermore 30% of participants answered one item or less correctly in the 11-13th grade cluster. Using the TABE guidelines above, this differential cluster performance suggests that 4-5th grade items are too easy but that 11-13th grade items are too hard for the participants assessed.

These results may also be considered in terms of scale scores and GED equivalence. According to previous work mapping TABE Reading scale scores to GED Reading test scores [3], a TABE scale score of 523 corresponds to the passing GED score of 450. Scale scores for each item cluster and items overall were calculated and compared to the GED criterion. Only participants who answered all 10 items correctly (248 participants) or all of the 11-13th grade items correctly (335 participants) surpassed the GED criterion. Using the TABE-GED mapping [3], participants who answered all of the 11-13th grade items correctly had a 69% chance of passing the GED Reading test. Thus while 32% of all participants answered the 11-13th grade items correctly, only 22% of all participants are likely to pass the GED Reading test.

4. DISCUSSION

Our preliminary results suggest that college freshmen reading ability overall is between 5th and 12th grade. This finding is plausible given NAEP results that only 37% of 12th grade students are proficient readers [6]. The lack of a more specific grade-level assessment of freshmen reading ability is attributable to the 10-item assessment used, which lacked medium difficulty items. In the present study, the duration of the complete 25 item TABE Survey was beyond what could be accommodated logistically; however, our results indicate that such logistic considerations must be overcome to assess the reading ability of college freshmen adequately.

Analysis of the 11-13th grade cluster offers suggestive results regarding freshmen reading ability, but must be treated with caution given that there were only three items in this cluster. Participants who answered all three items in this cluster correctly could reasonably be assumed to be proficient readers, and the difference between this percentage (32%) and NAEP's percentage of proficient readers (37%) could be easily explained by regional differences. Although demographic data was not collected for this study, the freshman demographics for the university where the study was conducted suggest that approximately half of students are white and half are African-American. These two groups have NAEP 12th grade Reading Proficiency rates of 46% and 17% respectively, averaging 32% as found in the present study.

However, as previously noted, only 69% of graduating seniors went straight to college in 2015 [2], suggesting that 54% of college freshmen should be proficient readers, assuming that all NAEP Proficient readers attend college. The present finding that reading proficiency is closer to the high school rate than the projected college rate could reflect a self-selection effect whereby the most proficient readers attend schools with more stringent admissions criteria on standardized tests.

The projection that only 69% of participants who answered all three items in the 11-13th grade cluster would pass the GED Reading test gives a strikingly different assessment of freshman reading proficiency (22% vs. NAEP's 37%) that cannot be easily explained by regional differences and may be a useful target for future research.

Altogether, our findings suggest that two-thirds of college freshman assessed have reading ability corresponding with below Proficient as described by NAEP. More accurate assessment and determination of regional differences are important areas of future research, as reading proficiency plays a large role in college success.

5. ACKNOWLEDGMENTS

This research was supported by the Institute of Education Sciences (IES; R305C120001). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of the IES.

6. REFERENCES

[1] Grading the nation's report card: Evaluating NAEP and transforming the assessment of educational progress, 1999. DOI: 10.17226/6296.

[2] Bureau of Labor Statistics, U.S. Department of Labor. College enrollment and work activity of high school graduates news release, 2016.

[3] CTB/McGraw-Hill. Tests of adult basic education, forms 9 and 10 norms book, complete battery and survey, all levels. Technical Report 91496, The McGraw-Hill Companies, Inc., 2004.

[4] CTB/McGraw-Hill. Tests of adult basic education, forms 9 and 10 technical report, all levels. Technical Report 91495, The McGraw-Hill Companies, Inc., 2004.

[5] National Center for Education Statistics, U.S. Department of Education. The condition of education - elementary and secondary education - transition to college - immediate college enrollment rate - indicator, 2016.

[6] The Nation's Report Card, U.S. Department of Education. NAEP - 2015 mathematics & reading at grade 12 - reading - national average scores, 2015.

Discovering Skill Prerequisite Structure through Bayesian Estimation and Nested Model Comparison

Soo-Yun Han
Dept. of Mathematics Education
Seoul National University
Seoul, South Korea
ssu1205@snu.ac.kr

Jiyoung Yoon
Dept. of Mathematics Education
Seoul National University
Seoul, South Korea
torol2@snu.ac.kr

Yun Joo Yoo
Dept. of Mathematics Education
Seoul National University
Seoul, South Korea
yyoo@snu.ac.kr

ABSTRACT
Identifying prerequisite relationships among skills is important for better student modeling in many educational systems. In this paper, we propose a new method to discover prerequisite structure from data using nested model comparisons in the context of Bayesian estimation. We evaluate our method with simulated data and real math test data.

Keywords
Prerequisite structure discovery, Bayesian Network, MCMC estimation, nested model comparison, pseudo-Bayes factor.

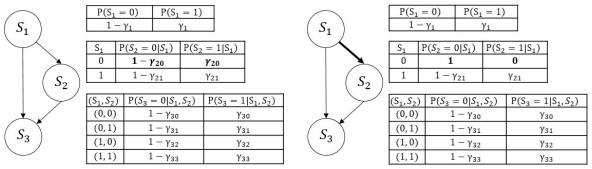
1. INTRODUCTION
In many educational systems, the process of learning usually proceeds sequentially according to a predetermined order that reflects cognitive theories about student learning. In this learning sequence some knowledge skills must be acquired prior to learning advanced skills. In this study, we refer to *prerequisite structure* as the relationships among skills that put strict constraints on the order in which these skills can be mastered.

Identifying skill prerequisite structure is a crucial step to construct a valid and accurate student model in adaptive tutoring system or other educational system for estimation of student's skill mastery status and provision of appropriate remediation for them. Prerequisite structure can be specified by domain experts, but such process may be time-consuming and could produce subjective models lacking validity. Using large educational data and data mining techniques, several previous studies have tried to find prerequisite relationships among knowledge skills [1,2,3,7]. To derive prerequisite structure from student performance data is somewhat challenging in that a student's mastery status of skills cannot be directly observed, but can only be estimated, i.e, is latent in nature. Previous works mostly used Expectation-Maximization (EM) estimates for latent skill variables [1,2,3].

In this paper, we present a new method for discovering prerequisite structure from student performance data using Bayesian Markov Chain Monte Carlo (MCMC) estimation and nested model comparison. For nested model comparison, we use pseudo-Bayes factor (PsBF) [4], one of the Bayesian model selection criteria.

2. METHOD
In our method, it is assumed that student performance (item response) data at a certain point in time is given and skills related to items are specified. Skills and items are considered as binary random variables and the item-skill relationships are given by Q-matrix (a binary matrix that represents the mapping of items to skills) [9]. DINA model is used for modeling the probability of correct response to an item as a function of whether all the skills required are mastered and of slip and guess parameters [5]. To represent skill prerequisite structure, (static) Bayesian Network is used as student model. Bayesian network is a probabilistic graphical model representing the relationship of a set of random variables as a directed acyclic graph (DAG) with conditional probability tables (CPTs).

We now focus on the discovery of prerequisite relationship, that is, *strict hierarchical order* between mastery of two skills. To this end, we set two types of models: a *full model*, which parameterizes all possible dependencies between skills, and a *strict model*, which assumes prerequisite relationship between a pair of skills. For example, Figure 1 illustrates DAGs and CPTs of a full model consisting of three skills (S_1, S_2, S_3) and a strict model assuming prerequisite relationship between skill S_1 and S_2 (S_1 is a prerequisite for S_2). The difference between two models is that, while the full model contains the parameter γ_{20} related to the probability $P(S_2 = 1 \mid S_1 = 0)$, the strict model put a constraint that this probability is zero (that is, the strict model is nested within the full model). If skill S_1 is a true prerequisite for S_2, the parameter γ_{20} in the full model will be estimated to be closed to zero and there will be no significant difference in the degree to which the two models explain the data. The idea of nested model comparison is to statistically test the null hypothesis that the two models present the same likelihood on the data.



(a) Full model (b) Strict model

Figure 1. DAGs and CPTs of (a) a full model and (b) a strict model of skills S_1, S_2, S_3. The bolded directed edge from S_1 to S_2 in DAG of the strict model (b) means that S_1 is a prerequisite for mastery of S_2.

When two models are fitted to the data using maximum likelihood, the likelihood ratio test is used for hypothesis testing. In the context of Bayesian estimation, Bayes factor or its variants can be considered as the test method. We use *pseudo-Bayes factor*, which can be calculated by the MCMC estimation process, as the test statistic to contrast two models. The pseudo-Bayes factor for model M_1 relative to M_2 is the ratio of approximations of marginal likelihood based on predictive distributions and cross-validation strategies and defined as

$$\mathrm{PsBF}_{12} = \frac{\hat{p}(X \mid M_1)}{\hat{p}(X \mid M_2)} = \frac{\prod_{i=1}^{n} p(X_i \mid X_{-i},\ M_1)}{\prod_{i=1}^{n} p(X_i \mid X_{-i},\ M_2)} = \frac{\prod_{i=1}^{n} \int p(X_i \mid \Theta, M_1) p(\Theta \mid X_{-i}, M_1) d\Theta}{\prod_{i=1}^{n} \int p(X_i \mid \Theta, M_2) p(\Theta \mid X_{-i}, M_2) d\Theta}$$

where X_i is the response data of student i, X_{-i} is the complement of X_i in the data X, and Θ is the set of free parameters. The

calculated PsBF value in MCMC estimation is compared to a critical value to decide whether to reject the null hypothesis or not. If the null hypothesis is not rejected, then the strict model is accepted, thus concluding that the prerequisite relationship exists.

3. EVALUATIONS

To evaluate the efficiency of our method in discovering prerequisite structures, we first conducted a simulation study and then applied our method to a real dataset. In this process we faced a problem that PsBF values are dispersed from the known distribution of Bayes Factor [6]. To address this problem, we derived the critical value from the empirical distribution of PsBF values under the null hypothesis.

In our evaluation steps, all MCMC estimation algorithms were implemented using R package R2OpenBUGS [8]. For MCMC estimations, we set the priors as follows: a uniform prior $\text{Unif}(0,1)$ on each structural parameters (γ_{ij}) and a beta prior $\text{Beta}(6,21)$ on slip and guess parameters for each items.

3.1 Simulated Data

In this simulation part, we considered five prerequisite structures of latent skills (Figure 2). For each structure, we generated 500 datasets consisting of 1000 students' skill mastery status and their responses for test items using a balanced Q-matrix (each skills are measured with the same number and types of items) under the DINA model with low slip and guess probabilities randomly drawn from $\text{Unif}(0, 0.05)$.

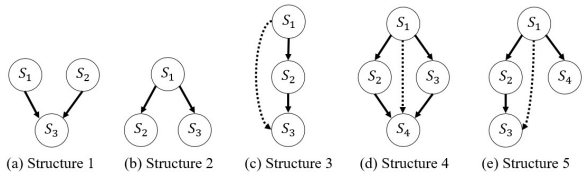


Figure 2. Five prerequisite structures of skills used in simulation study

We evaluate our method using two metrics: *true positive structure rate* (TPSR; # of correct structure recoveries in the output / # of true structures) and *true positive adjacency rate* (TPAR; # of correct adjacency recoveries in the output / # of adjacencies in true model).

The results show that our method can efficiently discover prerequisite structure (Table 1). In all cases recovery rates of true structure are over 80% (the worst rate is 81.6% in structure 4). The recovery rates of true prerequisite relationship between two skills (edges) are even higher such as over 90%.

Table 1. TPSR and TPAR results for each structure

Structure	1	2	3	4	5
TPSR	0.926	0.840	0.872	0.816	0.874
TPAR	0.937	0.942	0.943	0.942	0.962

3.2 Real Data Application

We used mathematics cognitive diagnosis assessment data from 936 eighth grade students over a set of 16 items measuring four skills related to linear equation and linear inequality (Figure 3-a). The prerequisite structure of these skills (Figure 3-b) was initially set by knowledge experts.

Figure 3-c shows the prerequisite structure discovered by applying our method to the real data. All prerequisite relationships set by experts are well discovered, and one additional prerequisite

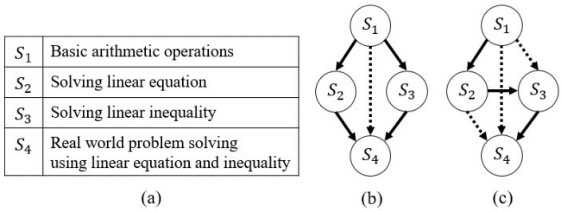


Figure 3. (a) Four skills in math test; (b) Prerequisite structure from knowledge experts; (c) Discovered prerequisite structure

relationship ($S_2 \rightarrow S_3$) is found. A possible explanation for this is that while knowledge experts judge that either linear equation or linear inequality can be learned first, students usually learn to solve linear equation first following the sequence in the curriculum.

4. CONCLUSION AND FUTURE WORK

We presented a method to discover skill prerequisite structure from data based on nested model comparison and evaluated the method using simulated data and real data. The performance of our prerequisite structure learning method was good within the settings used in our experiments. Since we used only low number of skills and certain assumptions for the evaluation, we need to further explore our method in various conditions.

In future work, we will investigate the idea of nested model comparison in the context of frequentist estimation (e.g., EM estimation) and compare with other previous methods. In this paper the focus is only on the prerequisite relationship between skills, but there may be other dependence relationships between them along with different types of response models. It would be interesting to study how to discover skill structures considering various dependency relationships in Bayesian Network modeling of skill mastery.

5. REFERENCES

[1] Brunskill, E. 2011. Estimating prerequisite structure from noisy data. In *Proceedings of the 4th International Conference on Educational Data Mining*.

[2] Chen, Y., González-Brenes, J. P., and Tian, J. 2016. Joint discovery of skill prerequisite graphs and student models. In *Proceedings of the 9th International Conference on Educational Data Mining*.

[3] Chen, Y., Wuillemin, P. H., and Labat, J. M. 2015. Discovering prerequisite structure of skills through probabilistic association rules mining. In *Proceedings of the 8th International Conference on Educational Data Mining*.

[4] Gelfand, A. E. 1996. Model determination using sampling-based methods. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in practice* (pp. 145-161). London: Chapman & Hall.

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

[6] Kass, R. E., and Raftery, A. E. 1995. Bayes factors. *Journal of the American Statistical Association, 90*(430), 773-795.

[7] Scheines, R., Silver, E., and Goldin. I. 2014. Discovering prerequisite relationships among knowledge components. In *Proceedings of the 7th International Conference on Educational Data Mining*.

[8] Sturtz, S., Ligges, U., and Gelman, A. 2010. *R2OpenBUGS: a package for running OpenBUGS from R.* http://cran.r-project.org/web/packages/R2OpenBUGS/vignettes/R2OpenBUGS.pdf

[9] Tatsuoka, K. K. 1983. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement, 20*(4), 345-354.

Text analysis with LIWC and Coh-Metrix: Portraying MOOCs Instructors

Junyi Li
Central China Normal University
University of Pennsylvania
junyili@mails.ccnu.edu.cn

Yun Tang
Central China Normal University
tangyun@mail.ccnu.edu.cn

Lijun Sun
Central China Normal University
lijunsunccnu@gmail.com

Xiangen Hu
Central China Normal University
University of Memphis
xiangenhu@gmail.com

ABSTRACT
To date, most MOOCs in major platforms (e.g. Coursera and edX) are xMOOCs, which means teacher speech is still the major part of these MOOCs. Therefore, it is necessary to evaluate the quality of lecture and to explore the relationships between lecture quality of MOOCs and learning outcomes. The present study attempted to explore the lecture styles of instructors in MOOCs by using text analysis. One hundred and twenty-nine course transcripts were collected from Coursera and edX. We also collected public data of course evaluation from the largest MOOC community in China (mooc.guokr.com) Linguistic inquiry and word count (LIWC) and Coh-Metrix were used to extract text features including self-reference, tone, affect, cognitive words, and cohesion. After combined students' comments with clustering analysis, results indicated that four different lecture styles emerged from 129 courses: "mediocre", "boring", "perfect" and "enthusiastic". Significant difference was found between four lecture styles for the notes taken, but significant differences were not found for the course satisfaction and discussion posts. Future studies should exam whether different lecture styles have impacts on students' engagement and learning outcomes in MOOCs.

Keywords

MOOCs; Lecture styles; Instructors; Text analysis

1. INTRODUCTION
Massive open online courses (MOOCs) have attracted much attention in the recent years. They provide not only free courses from high prestige universities, but also the freedom of learning for learners all over the world. Major MOOC platforms, such as Coursera, FutureLearn, edX, and Open2Study, are well received by most learners. The reason why MOOCs become a popular way to learn is that it provides each individual learner with opportunities to engage with the materials via formative assessments and the ability to personalize her learning environment (Evans, Baker & Dee, 2016).

Researchers from different discipline have conducted many studies focused on MOOCs learners, including course completion, quality of interaction, student engagement, and collaborative learning in MOOCs (Andres et al., in press; Wang & Baker, 2015). However, the complexities of teaching have been largely absent from emerging MOOC debates (Ross et al., 2014). After all, MOOC is quite different from traditional class in many aspects. For example, MOOC instructors were motivated by a sense of intrigue, the desire to gain some personal rewards, or a sense of altruism; they were challenged by difficulty in evaluating students' work, encountering a lack of student participation in online forums, being burdened by the heavy demands of time and money, and having a sense of speaking into a "vacuum" due to the absence of student immediate feedback (Hew & Cheung, 2014). Some instructors found it difficult to teach when not facing a real audience of students (Allon, 2012). To date, most MOOCs in major platforms (e.g. Coursera and edX) are xMOOCs, which is a highly structured, content-driven course and designed for large numbers of individuals working mostly alone, teacher speech is still the major part of these MOOCs. Therefore, it is necessary to evaluate the quality of lecture and to explore the relationships between lecture quality of MOOCs and learning outcomes. Some researchers have tried to build models to automatically predict if certain course content would show up by using natural language processing (Araya et al., 2012). Based on the mentioned above, the present study attempted to explore the lecture styles of instructors in MOOCs by using text analysis.

2. METHOD
2.1 Data Collection
Transcripts from 129 courses (humanities: 24.8%, social science: 38%, science: 37.2%) were collected from Coursera and edX. We also collected public data of course evaluation from the largest MOOC community in Mainland China (mooc.guokr.com). This community offered online learners a platform on which they could voluntarily evaluate MOOCs and share their opinions with fellow online learners. The data set we used included course satisfaction, the number of asynchronous discussion posts per course, notes taken per course, the number of followers per course, to name a few.

2.2 Extracting Text Features
Two text analysis tools (i.e. LIWC and Coh-Metrix) were used to extract text features from 129 course transcripts. According to previous studies, self-reference (I, me, my), affect (positive emotion and negative emotion), tone, cognitive words, and cohesion were extracted. Other features like words per sentence and big-words (words are longer than 6 letters) were also viewed as complexity measure of teacher speech.

2.3 Data Analysis

Clustering analysis and ANOVA were conducted by using RapidMiner and SPSS. We first transformed all the text features into Z score, then performed k-means algorithm with euclidean distance in RapidMiner. The k value was assigned with a value from 2 to 6, because of comprehensibility.

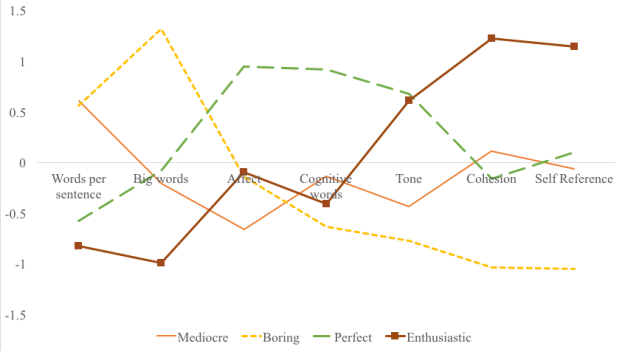


Figure 1 The four lecture styles in MOOCs

3. RESULTS AND DISCUSSION

Four clusters were found, and there were 42, 27, 36, and 24 courses in each cluster respectively. We then checked the students' comments of these courses in Guoke MOOC community, and assigned label to these clusters (Figure 1).

Concretely, instructors who used the most self-reference words (I, me, my), short sentences, and the least big-words were perceived as agreeable and enthusiastic by students (Cluster 4: Enthusiastic). Instructors who used the least self-reference words, long sentences, the most big-words, and showed a low cohesion were perceived as boring by students (Cluster 2: Boring). Instructors who used the most cognitive words to help students to understand and used medium level of self-reference words, big-words and showed medium cohesion were labeled as "perfect" (Cluster 3). Courses used the most of long sentences and showed average level in other dimensions were labeled as "mediocre" (Cluster 1). No significant differences were found between four lecture styles for the course satisfaction (F = .76, p = .52, $\eta 2$ = .02) and discussion posts (F = 1.39, p = .25, $\eta 2$ = .03). However, significant difference was found for notes taken (F = 2.80, p = .4, $\eta 2$ = .06). Concretely, the number of notes taken in "perfect" style was much more than "mediocre". Notes taken can stand for the cognitive processing of learners to some extents. These results suggested that the "perfect" lecture style may be more likely to encourage students' engagement. Since the discussion posts, notes taken and course satisfaction data in the present study were acquired from a third-party platform, further evidence are needed to verify these results. Future studies should examine whether the four lecture styles have different impacts on students' engagement and learning outcomes (e.g. academic performance and course completion) in MOOCs.

4. ACKNOWLEDGEMENTS

This research is supported by China Scholarship Council and Excellent Doctoral Dissertation Program (Central China Normal University). We appreciated these support both in finance and in spirit.

5. REFERENCES

[1] Evans, B. J., Baker, R., & Dee, T. S. 2016. Persistence Patterns in Massive Open Online Courses (MOOCs). *Journal of Higher Education,* 87, 206-242.

[2] Andres, M., Baker, R., Gasevic, D., & Spann, G. in press. Replicating 21 Findings on Student Success in Online Learning. *Technology, Instruction, Cognition, and Learning.*

[3] Wang, Y., & Baker, R. 2015. Content or Platform: Why do students complete MOOCs? *Journal of Online Learning and Teaching,* 11, 17-30.

[4] Ross, J., Sinclair, C., Knox, J., Bayne, S., & Macleod, H. (2014). Teacher experiences and academic Identity: The missing components of MOOC pedagogy. *Jounral of Online Learning and Teaching*, 10, 57-69.

[5] Hew, K. F., & Cheung, W. S. 2014. Students' and instructors' use of massive open online courses (MOOCs): Motivations and challenges. *Educational Research Review,* 12, 45-58.

[6] Allon, G. 2012. Operations Management, Udemy. *Chronicle of Higher Education,* 59, B10-11.

[7] Araya, R., Plana, F., Dartnell, P., Soto-Andrade, J., Luci, G., Salinas, E. and Araya, M. 2012. Estimation of teacher practices based on text transcripts of teacher speech using a support vector machine algorithm. *British Journal of Educational Technology*, 43, 837-846.

Identifying relationships between students' questions type and their behavior

Fatima Harrak
Sorbonne Universités
UPMC Univ Paris 06,
CNRS, LIP6 UMR 7606,
4 place Jussieu, 75005 Paris, France
fatima.harrak@lip6.fr

François Bouchet
Sorbonne Universités
UPMC Univ Paris 06,
CNRS, LIP6 UMR 7606,
4 place Jussieu, 75005 Paris, France
francois.bouchet@lip6.fr

Vanda Luengo
Sorbonne Universités
UPMC Univ Paris 06,
CNRS, LIP6 UMR 7606,
4 place Jussieu, 75005 Paris, France
vanda.luengo@lip6.fr

ABSTRACT

We present the process of categorization of students' questions, and through a clustering on students, we show the relevance of this classification to identify different profiles of students. It opens perspectives in assisting teachers during Q&A sessions.

Keywords

Clustering, question taxonomy, students' behavior.

1. INTRODUCTION

Studying learners' questions while they learn is essential [1], not only to understand their level and eventually help them learn better [2] but to help teachers in addressing these questions. Analyzing students' questions can help for instance in distinguishing deep learning vs. shallow learning [3]. In this paper, we are interested in whether the type of questions asked by students on an online platform is characteristics of their classroom behavior. We investigate this question in the context of an hybrid curriculum (like [4]), where students have to ask questions before the class to help professors prepare their Q&A session. Our goal here is threefold: (RQ1) Can we define a taxonomy of questions relevant to analyze students' questions? (RQ2) Can we automatize the identification of these questions? (RQ3) Can annotated questions asked by a student inform us about their performance, attendance and questioning behavior?

2. RESEARCH METHODOLOGY

We addressed these research questions in 3 successive steps: (1) we conducted a manual process of categorization of students' questions, which allowed us to propose a taxonomy of questions, (2) we used this taxonomy for an automatic annotation of a corpus of students' questions, (3) to identify students' characteristics from the typology of questions they asked, we used clustering technique over two courses and then characterized the obtained clusters using a different set of features, as in [5].

The dataset used for this work is made of questions asked in 2012 by 1st year medicine/pharmacy students from a major public French university (Univ. Joseph Fourier). Each course is made of 4 to 6 4-week sequences on the PACES[1] platform. After a 1st week dedicated to learning from online material, during week 2 students must ask questions and vote for questions asked by other students on an online forum to help professors prepare their Q&A session in week 3. Therefore, for each of the 13 courses, we have 4 to 6 sets of questions asked by students (6457 questions overall) during the 2nd week of each sequence.

[1] paces.medatice-grenoble.fr

3. RESULTS

3.1 Categorization of questions

To answer to RQ1, we took a sample of 600 questions (around 10% of the corpus size) from two courses (biochemistry [BCH], histology & developmental biology [HBDD]), which are considered to be among the most difficult courses and had the highest number of questions asked. This sample was randomly divided in 3 sub-samples of 200 questions to apply 3 different categorization steps: a discovery step, a consolidation step and a validation step. Step 1 consisted in grouping sentences with similarities to extract significant concepts. Then we segmented the combined questions to standardize the previous annotation and we grouped the extracted categories into independent dimensions, where each dimension grouped similar concepts in sub-categories. Step 2 consisted in annotating the second sub-sample to validate the dimensions previously identified and to make sure they were indeed independent from each other. In step 3, we performed a double annotation to validate the generality of our categories on the remaining sub-sample of 200 sentences. Two human annotators used as a unique reference the taxonomy previously created. They annotated independently each dimension (average kappa = 0.70) – discussions to fix discrepancies led to a final refinement of the categories' description. Finally, a re-annotation was performed on the entire sample (600 sentences) to consider the changes and to provide a grounded truth for the automatic annotation. The final taxonomy is provided in Table 1.

Table 1. Final question taxonomy from manual annotation

Dim1	**Type questions**	**Description**
1	Re-explain / redefine	Ask for an explanation already done in the course material.
2	Deepen a concept	Broaden a knowledge, clarify an ambiguity or request for a better understanding
3	Validation / verification	Verify/validate a formulated hypothesis
Dim2	**Modality explanation**	**Description**
0	N/A	None – attributed when neither of the other values below applies
1	Example	Example application (course/exercise)
2	Schema	Schema application or an explanation about it
3	Correction	Correction of an exercise in course/exam
Dim3	**Type of explanation**	**Description**
0	N/A	None – attributed when neither of the other values below applies
1	Define	Define a concept or term
2	Manner (how?)	The manner how to proceed
3	Reason (why?)	Ask for the reason
4	Roles (utility?)	What's the use / function
5	Link between concepts	Verify a link between two concepts

Dim4	Optional: if question is a verification	Description
1	Mistake / contradiction	Detect mistake/contradiction in course or in teacher's explanation.
2	Knowledge in course	Verify knowledge
3	Exam	Check exam-related information

3.2 Automatic annotation

To answer to RQ2 and to annotate the whole corpus (and on the long term, to use it online to analyze the questions collected), we identified keywords representative of each value in each dimension (e.g. the word "detail" is representative of a "deepen a concept" question). Then we developed an automatic tagger which identifies for each question the main value associated to each dimension and tags the question as such. We validated the automatic annotator by comparing its results on the manually annotated subsample of 600 questions and obtained a kappa value of 0.74, enough to consider applying it to the full corpus.

3.3 Links between questions and behavior

To identify whether the type of questions asked can inform us on students' characteristics, first we performed two clustering analyses using K-Means algorithm (with k varying between 2 and 10) over two datasets: students who asked questions in the BCH course (1227 questions asked by N_1=244 students) and in the HBDD course (979 questions asked by N_2=201 students). We performed the clustering using as features for each student the proportion of each question asked in each dimension (*e.g.* the proportion of questions with value 1 in dimension 1) asked (a) overall, (b) during the first half of the course, and (c) during the second half of the course (44 features overall). Distinguishing (b) and (c) in addition to (a) allowed us to take into account whether it was a change in questions asked that could be meaningful, more than the overall distribution. We obtained 4 clusters in both cases.

The second step consisted in characterizing the clusters by considering attributes not used for the clustering: students' grade on the final exam on this course (out of 20), attendance ratio (from 0 [never there] to 1 [always there]), the number of questions asked in this course, and the number of votes from other students on their questions in this course. Students for whom this data was not available were excluded from the datasets, leading to two smaller sample sizes (N'_1=173 and N'_2=161). We performed two one-way ANOVA for grades on these two clusterings and found statistically significant differences (p<0.001 and p<0.001). For the other variables, the distribution did not follow a normal law and we therefore performed a Kruskal-Wallis H test on ranks associated to each variable. The test showed that there was a statistically significant difference for attendance (p=0.04 and p=0.02), number of questions asked (p<0.001 and p<0.001) and number of votes received (p=0.04 and p<0.001) for BCH and HBDD respectively. Results are summarized in Table 2.

Table 2. Differences between the 4 BCH and HBDD clusters

Course	Cluster	*N*	Grade (/20)	Attendance	# quest.	# votes
BCH	A	53	7.97	0.86	2.83	3.06
	B	63	8.54	0.90	2.92	2.69
	C	86	9.38	0.93	6.23	2.61
	D	42	11.2	0.93	11.74	1.22
HBDD	A	59	7.43	0.89	3.53	5.57
	B	34	9.78	0.92	2.44	2.47
	C	72	10.11	0.92	6.54	3.69
	D	36	11.78	0.95	7.00	1.71

4. DISCUSSION AND CONCLUSION

Overall, when considering the results presented in Table 2, we see two similar clusters in both cases: A and D. Cluster A is made of around 28-41% of the students with grades lower than average, attending less to classes, asking less questions than average but which are particularly popular (probably because of votes from similar students, but that information was unfortunately not available). In terms of questions asked, they had a higher number of "how to" questions (*cf.* dim3-2 in Table 1) than any other cluster. On the other end of the spectrum, cluster D is made of around 21% of the students with grades above average, high attendance, who ask more questions than average that are fairly unpopular – we can assume these must be very precise questions that already require a good understanding of the content of the course, and are thus not deemed as important by other students. Interestingly, when comparing the proportion of questions asked in the first vs. second half of the class, cluster D students are the only ones who asked more questions in the 2nd half of the 4-6 sequences than in the 1st half, presumably because the concepts presented at the beginning were simpler and easier for them to understand. In between, clusters B and C represent more average students who differ mostly in terms of number of questions asked.

Therefore, to answer to RQ3 we have shown that although the clustering was performed exclusively on semantic features (*cf.* taxonomy in Table 1), it correlates with information relative to students' performance, attendance and questioning/voting behavior. Our work has some limits: we have applied it only to 2 courses (because a minimum number of questions is required) and we have not considered if it would be possible to classify students in clusters online or even if the same clusters could be found in the same courses on different years. Furthermore, not all questions could be automatically annotated, which reduced the dataset size and is particularly problematic for students who asked few questions. However, this work demonstrates the validity and the usefulness of our taxonomy, and shows the relevance of this classification to identify different students' profiles. It also suggests the taxonomy could be useful for our long-term goal which is to assist teachers in choosing questions to be explained in Q&A sessions. We also intend to apply this taxonomy to different datasets (*e.g.* questions asked in a MOOC) to see if it can also be useful in these contexts and if similar patterns appear.

5. REFERENCES

[1] A. C. Graesser and N. K. Person, "Question Asking During Tutoring," *Am. Educ. Res. J.*, vol. 31, no. 1, pp. 104–137, 1994.

[2] J. Sullins *et al.*, "Are You Asking the Right Questions: The Use of Animated Agents to Teach Learners to Become Better Question Askers.," in *FLAIR*, 2015, pp. 479–482.

[3] C. Chin and J. Osborne, "Students' questions: a potential resource for teaching and learning science," *Stud. Sci. Educ.*, vol. 44, no. 1, pp. 1–39, Mar. 2008.

[4] Q. Liu, W. Peng, F. Zhang, R. Hu, Y. Li, and W. Yan, "The Effectiveness of Blended Learning in Health Professions: Systematic Review and Meta-Analysis," *J. Med. Internet Res.*, vol. 18, no. 1, Jan. 2016.

[5] F. Bouchet, J. M. Harley, G. J. Trevors, and R. Azevedo, "Clustering and Profiling Students According to their Interactions with an Intelligent Tutoring System Fostering Self-Regulated Learning," *J. Educ. Data Min.*, vol. 5, no. 1, pp. 104–146, May 2013.

Metacognitive Prompt Overdose: Positive and Negative Effects of Prompts in iSTART

Kathryn S. McCarthy, Amy M. Johnson, Aaron D. Likens, Zachary Martin, Danielle S. McNamara
Arizona State University
Tempe, AZ, USA
{ksmccar1, amjohn43, alikens, zsmartin, dsmcnama}@asu.edu

ABSTRACT
Interactive Strategy Training for Active Reading and Thinking (iSTART) is an intelligent tutoring system that supports reading comprehension through self-explanation (SE) training. This study tested how two metacognitive features, presented in a 2 x 2 design, affected students' SE scores during training. The *performance notification* feature notified students when their average SE score dropped below an experimenter-set threshold. The *self-rating* feature asked participants to rate their own SE scores. Analyses of SE scores during training indicated that neither feature increased SE scores and, on the contrary, seemed to decrease SE performance after the first instance. These findings suggest that too many metacognitive prompts can be detrimental, particularly in a system that provides metacognitive strategy training.

Keywords
intelligent tutoring systems; metacognition; educational games; system interaction logs

1. INTRODUCTION
Intelligent tutoring systems (ITSs) provide an opportunity for extended training and individualized feedback to support the development of skills and strategies. One such ITS, Interactive Strategy Training for Active Reading and Thinking (iSTART) uses self-explanation (SE) training as a means of increasing students' comprehension of complex texts [4]. iSTART provides instruction on SE strategies through lesson videos, guided demonstration, and practice. Research indicates that prompting metacognition, or reflection on one's own knowledge, can enhance the benefits of training within computer-based learning [1]. In this study, we expand upon previous research to investigate how two metacognitive features affect the SE scores during iSTART practice.

In iSTART's generative practice, students write their own SEs and a natural language processing (NLP) algorithm immediately provides a score of poor (0), fair (1), good (2), or great (3). The two metacognitive features were implemented within this generative practice. The first feature is a *performance notification* that alerts students that their SE score is below 2.0 and sends them to Coached Practice for remediation. The second feature is a *self-rating* that prompts students to rate the quality of their SE before receiving the computer-generated score. The performance notification encourages metacognition indirectly, whereas the self-rating is a direct metacognitive prompt [6]. The current study expands on data reported in [3], which further demonstrated the positive effects of iSTART on deep comprehension, but also indicated that neither metacognitive feature affected post-training learning outcomes. In this study, we explore the log-data to investigate how these two metacognitive features, both individually and in combination, affect SE scores during iSTART generative practice.

Based on previous work [6], we predicted that the performance notification would increase SE scores immediately after the first instance of the notification. In [6], however, the instruction was brief, and did not allow examining further instances of the notification. In this study, we examine the effects of the notification after the initial instance during a longer duration study. Consistent with previous research [5], we had predicted that self-ratings would improve performance. Of particular interest was the interaction of the two features. One hypothesis is that there would be an additive effect such that having both features would yield the greatest SE score improvement [2]. An alternative hypothesis is that the redundancy of the two features would result in an interactive, and possibly negative effect [4].

2. METHODS
2.1 Participants
As part of the larger study reported in [3], 116 high school students (M_{age}=17.67, SD=1.30) received monetary compensation for their participation.

2.2 Design and procedure
The study employed a 2(performance notification: off, on) x 2(self-rating: off, on) between-subjects design. Participants completed iSTART training in three 2-hour sessions. Participants first watched iSTART video lessons that provide instruction on the purpose of SE training and five comprehension strategies (comprehension monitoring, paraphrasing, prediction, elaboration, and bridging). Next, participants completed one round of Coached Practice, in which a pedagogical agent provides individualized feedback on students' self-explanations. Participants were then allowed to move freely throughout the system to interact with videos, Coached Practice, identification games, and generative games for the remainder of the training sessions. The metacognitive features were implemented only during generative games. Performance notifications were triggered each time the average SE score was less than 2.0 and self-rating prompts were triggered on randomly-determined self-explanations approximately 1/3 of the time.

3. RESULTS

We calculated a *gain score* to compare the average SE score in the game before and immediately following an average generative game score of 2.0 indicative of when the performance notification was triggered (or *would have triggered* in the notification off conditions). We used log-data to identify participants who completed at least one game in which their average SE score was less than 2.0 (n=78). Though the performance notification could be triggered as many times as necessary, most participants had no more than two instances of less than 2.0 average SE scores (Fig. 1). As participants were able to move freely through the system, only 48 participants (across all conditions) followed the *generative game, notification, generative game* sequence needed to calculate a gain score. These participants were relatively evenly distributed across the conditions. We analyzed the first two instances of average SE scores less than 2.0 for these 48 participants.

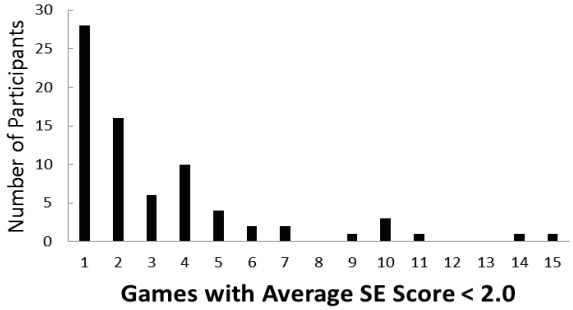


Figure. 1 *Frequency of Games with Average SE Scores < 2.0*

For the first instance of notification, the average gain scores in all conditions were positive. Though the pattern of gain scores for the performance notification is consistent with previous findings [3], an ANOVA indicated no effect of notification, of self-rating, and no interaction, all $F(1, 47) < 2.00$ (Fig. 2, *left*).

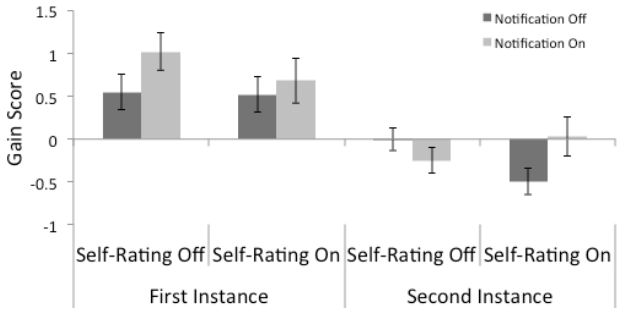


Figure 2. *Gain score in 1st and 2nd instance of avg. SE score < 2.0 as a function of performance notification and self-rating*

Fewer participants (n=27) had a second instance of notification. Contrary to the scores following the first instance, in this second instance, average gain scores were either near zero or negative, indicating that the scores after notification were the same or lower than before the notification. An ANOVA revealed no main effect of performance notification or self-rating, Fs < 1.00, *ns.* There was a significant notification by self-rating interaction indicating that having neither feature or both features did not affect SE score, but that the presence of only one metacognitive feature was detrimental to SE score, $F(1, 26)=5.46$, $p < .05$, $\eta^2_p=.17$ (Fig. 2, *right*).

4. CONCLUSIONS

These findings indicate that neither metacognitive feature had a consistent effect on SE quality during iSTART training. Though there was an overall increase in SE score in the first instance (as indicated by positive gain scores), there was no significant effect of either performance notification or self-rating compared to control. In the second instance, the interaction should be interpreted with caution given the small sample size. Nonetheless, the features did not improve SE score, and were potentially detrimental to performance. One explanation for these findings is that iSTART intrinsically instructs on metacognitive strategies. Hence, the inclusion of additional metacognitive prompts may be redundant, if not overwhelming, at least after the first instance.

These results were not consistent with extant research, and may be particular to iSTART. Certainly further analyses and studies are merited and will be explored. Nonetheless, given that neither prompt showed post-training learning outcomes [3] or sustained training benefits, we do not intend to include these features in future implementations of iSTART, and we would caution other researchers to consider the possibility of potential metacognitive prompt over-dosages.

5. ACKNOWLEDGMENTS

This research was supported in part by IES Grant R305A130124. Opinions, conclusions, or recommendations do not necessarily reflect the views of the IES.

6. REFERENCES

[1] Azevedo, R., Hadwin, A.F.: Scaffolding self-regulated learning and metacognition–Implications for the design of computer-based scaffolds. *Instructional Science 33*: 367-379 (2005)

[2] Flavell, J. H.: Metacognition and cognitive monitoring: A new area of cognitive– developmental inquiry. American Psychologist. 34(10), 906 (1979)

[3] McCarthy, K.S., Jacovina, M. E., Snow, E.L. Guerrero, T. A., & McNamara, D.S.. *iSTART therefore I understand: But metacognitive supports did not enhance comprehension gains.* In R. Baker, E. André, X. Hu, M.T. Rodrigo, B. du Boulay (eds.) *Proceedings of the 18th International Conference on AIED.* Wuhan, China (2017).

[4] McNamara, D.S., Levinstein, I.B., & Boonthum, C. (2004). iSTART: Interactive strategy trainer for active reading and thinking. *Behavioral Research Methods, Instruments, & Computers, 36*, 222-233.

[5] Schraw, G. & Dennison, R.S.: Assessing metacognitive awareness. *Contemporary Educational Psychology, 19*: 460-475 (1994)

[6] Snow**,** E.L., McNamara, D. S., Jacovina, M. E., Allen, L. K., Johnson, A. M., Perret, C. A., Dai, J., Jackson, G. T., Likens, A. D., Russell, D. G., & Weston, J. L. Promoting metacognitive awareness within a game-based intelligent tutoring system. In Mitrovic, A., Verdejo, F., Conati, C., Heffernan, N. (eds), *Proceedings of the 17th International Conference on AIED 2015.* Madrid, Spain: Springer, pp 786-789 (2015)

Tracking Online Reading of College Students

Andrew M. Olney
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
aolney@memphis.edu

Eric Hosman
Department of Counseling, Educational Psychology and Research
University of Memphis
Memphis, TN 38152
ehosman@memphis.edu

Art Graesser
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
graesser@memphis.edu

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

ABSTRACT

We conducted a pilot study that used kernel-level packet capture to record the web pages visited by college students and the reading difficulty of those pages. Our results indicate that i) no students were fully compliant in their participation, ii) the number of texts encountered by participants was highly skewed, iii) the reading difficulty of texts was about 7th grade, $M = 7.24$, $CI_{95}[7.04, 7.43]$, though difficulty varied by participant, and iv) the increasing use of encryption is likely a limiting factor for using kernel-level packet capture to measure online reading in the future.

Keywords

reading, Internet, measurement, text difficulty

1. INTRODUCTION

A recent survey revealed that approximately 90% of undergraduate respondents used laptops for their electronic course readings even though 68% did not prefer electronic textbooks to print [3]. The increase in online reading behavior has created new opportunities for researchers to track ecologically valid reading behavior. Online reading reflects true interests and goals (unlike artificial experimental paradigms) and further allows measures of the time spent reading and of the text itself over extended periods of time.

To better understand the online reading behavior of college freshmen, we conducted a pilot study using custom-designed online reading tracking software based on kernel-level packet capture. Tracking naturalistic online reading behavior appears to be novel to the literature, as most studies of online reading behavior either use lab-based methods like eye-tracking or self-report methods like surveys. Our main research objectives were to determine whether i) participants would comply with the tracking, ii) the reading behavior of participants was measured consistently, and iii) the text difficulty of measured texts was in a reasonable range.

2. METHOD

2.1 Participants

Participants (N = 7) were recruited through the psychology subject pool at an urban university in the southern United States. Self-reported ACT scores ($M = 21.29, SD = 3.64$) ranged from 18 to 29. Participants were required to own and bring a laptop to the study when they enrolled.

2.2 Materials

Kernel-level packet capture software for tracking online reading behavior was developed in C♯ using the WinPcap and PcapDotNet packet capture libraries. The resulting software, called Snarf, runs as a Microsoft Windows service in the background whenever the computer is turned on. Snarf monitored all http packet traffic on all network devices and sent anonymized timestamped records of web page URLs to an online Google Fusion Tables service for collection. Records were anonymized by using the media access control (MAC) address of the participant's network card as an identifier. To minimize data traffic, Snarf sent only URLs that did not match a blacklist of known non-reading-related URLs, such as Windows Update and image/audio/video filetypes. Also excluded from collection was any service using the encrypted https protocol. Encrypted traffic was excluded for two reasons. First, it is highly likely that encrypted traffic is of a personal nature that the participants would prefer not to share, e.g. email, banking, or health information. Secondly, breaking encryption could potentially introduce security vulnerabilities and put participants at significant risk.

2.3 Procedure

Approval for the research protocol was obtained from our institutional review board. Participants were enrolled in the study in the fall of 2015. After consent was obtained,

Table 1: Participant reading behavior

			Flesch-Kincaid Grade Level				Word Count			
					95% CI				95% CI	
Id	Texts	Days	M	(SD)	LL	*UL*	*M*	(*SD*)	*LL*	*UL*
1	1	0^{-}	-							
2	23	4^{-}	9.30	(8.05)	6.01	12.59	1137.10	(1985.10)	325.83	1948.30
3	170	100^{+}	6.98	(5.74)	6.12	7.85	509.72	(1578.30)	272.46	746.97
4	210	101^{+}	9.20	(6.67)	8.30	10.11	1152.50	(2086.00)	870.37	1434.60
5	829	94^{+}	7.15	(5.57)	6.77	7.53	963.39	(1778.20)	842.34	1084.40
6	4	50^{+}	7.28	(7.13)	0.29	14.26	14.00	(8.98)	5.20	22.80
7	3116	119^{+}	7.10	(6.76)	6.86	7.34	417.77	(1236.40)	374.36	461.18

Note: CI = confidence interval; LL = lower limit; UL = upper limit; -/+ indicates under/over study length.

an experimenter installed the SNARF online reading behavior tracker onto the participant's laptop and confirmed that SNARF was logging data to the Google Fusion Table service. At the end of the study, each recorded URL was queried and, if it was accessible, downloaded. Text from downloaded files was extracted using the Apache Tika library, tokenized into sentences using the Stanford CoreNLP tools [2], and then measured for word count and text difficulty using the Flesch-Kincaid Grade Level metric [1].

3. RESULTS & DISCUSSION

Of the 327,179 timestamped URLs collected, only 87,029 were unique, and of those unique URLs, only 26,762 (31%) were downloadable at the end of the study. Inspection of the timestamped URLs revealed that, despite efforts to blacklist non-reading-related web traffic, many URLs were not reading-related, e.g. antivirus updates, ads, and video websites.

Texts from downloadable URLs had extreme Flesch-Kincaid Grade Level (FKGL) values ranging from -3.40 to 7431, and extreme word count values ranging from 0 to approximately 10 million. Inspection of the data revealed that the FKGL frequency distribution dropped precipitously at grade level 20 and that the word count frequency distribution likewise dropped at 10,000 words. These values would be possible if a participant read a document with an average sentence length of 22 and average syllables per word of 2.3 (FKGL) or a 20-page single spaced paper (word count); thus these values are plausible but may be overly generous. Descriptive statistics for the texts and downloadable URLs after applying these filtering criteria are shown in Table 1.

Table 1 presents evidence addressing our research objectives. First, participants did not comply with tracking: two participants uninstalled the software within a week (one within the same day) and the remaining five participants failed to uninstall the software or meet the experimenter to uninstall the software after being reminded by email. Secondly, participant's online reading behavior was not measured evenly: the number of texts (as measured by downloadable URLs) read by participants was highly skewed, ranging from 1 to over 3,000. This skewed distribution could be caused by some participants mostly using encrypted sites like Wikipedia or the New York Times which, by virtue of being encrypted, SNARF would not record. Finally, the reading difficulty of texts was in a reasonable range, generally 7th grade, $M = 7.24$, $CI_{95}[7.04, 7.43]$, and word count on average was comparable to a page of single spaced text, $M = 564$, $CI_{95}[521, 507]$, though both varied somewhat by participant as shown in Table 1. These results are slightly lower than might be expected when reading for academic purposes, but for general reading seem reasonable.

4. CONCLUSIONS

Our results indicate that kernel-level packet capture is a viable means for measuring online reading behavior save for the increasingly prevalent use of encryption on all web sites. While it would be possible to modify a browser to record the text displayed to the user, this alternative could inadvertently collect email, banking, or health information that should remain private. Thus it may be that the balance between privacy concerns and reading research is best struck by avoiding general purpose reading applications like web browsers and instead focusing on reading-specific applications that are not otherwise used to access personal information.

5. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF; 1235958 and 1352207) and Institute of Education Sciences (IES; R305C120001). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of the NSF or IES.

6. REFERENCES

[1] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease Formula) for Navy enlisted personnel. Research branch report 8-75., Naval Air Station, Memphis, 1975.

[2] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[3] D. Mizrachi. Undergraduates' academic reading format preferences and behaviors. *The Journal of Academic Librarianship*, 41(3):301 – 311, 2015.

Dropout Prediction in MOOCs using Learners' Study Habits Features

Han Wan Jun Ding Xiaopeng Gao
School of Computer Science and Engineering
Beihang University
Beijing, China
+86-10-82338059
{wanhan, dingjun, gxp}@buaa.edu.cn

David Pritchard
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA, 02139
617-253-6812
dpritch@mit.edu

ABSTRACT
Many educators have been alarmed by the high dropout rates in MOOC. There are various factors, such as lack of satisfaction or attribution, may lead learners to drop out. Educational interventions targeting such risk may help reduce dropout rates. The primary task of intervention design requires the ability to predict dropouts accurately and early enough to deliver timely intervention. In this paper, we present a dropout predictor that uses student activity features and then we add learners' study habits features to improve the accuracy. Our models achieved an average AUC (receiver operating characteristic area-under-the-curve) as high as 0.838 (if lacking study habits is 0.795) when predicting one week in advance. The model with learners' study habits features attained average increase in AUC of 0.03, 0.06, 0.08 and 0.05 in different cohorts (passive collaborator, wiki contributor, forum contributor, and fully collaborative).

Keywords
MOOC, dropout prediction, study habits

1. INTRODUCTION
One way to solve the high dropout rates in MOOC is to deliver timely intervention by predicting the dropout probability. Some researchers focused on extracting features of learners' study activities (such as resource accessing) from MOOCs' log, and then building machine learning models. Balakrishnan [1] used the discrete single stream HMMs model to predict whether a student would dropout or not. [2] tried to establish an extensible real-time predicting model, which is fit for any different courses. Loya [3] demonstrated that who executed their learning process on schedule has greater probability to finish the course in MOOCs. Liang J [4] predicted a student's dropout state 10 days later with 3 months' data into four typical machine learning models (LR/SVM/GBDT/RF).

Taylor C. [5] used the dataset of 6.002x: Circuits and Electronics taught in Fall of 2012 on edX, includes course information and students' activity data. In addition to the common simple features, they produced some complex, multi-layered interpretive features, and then used them as the input of predicting models. They divided the students into four groups according to their participation: *passive collaborator* are those learners never actively participated in either the forum or the Wiki, they just view the resources, but did not have contributions; *wiki contributor* are those learners generated Wiki content, but never posted in the forum; *forum contributor* are those learners posted in the forum, but never actively participated in the Wiki; *fully collaborative* are those learners actively participated by generating Wiki content and posting in the forum. Their results shown that if the sample size of the students group is small (especial for wiki contributor, forum contributor and fully collaborative), the predicting accuracy is relative low.

In our work, we focus on extracting more important features of learners' study habits features to improve the accuracy of predicting models, particularly for the small sample size group.

2. PREDICTION PROBLEM DEFINITION
Our data obtained from the 2014 instance of the introductory physics MOOC 8.MReV through the edX platform. We considered defining the dropout point as the time slice (week) a learner fails to submit any further assignments or problems / exam.

The instructor could use the data from week 1 to the current week i to make predictions. The model will predict existing learner dropout during week (i + 1) to week 16. For example, current week is week 7, and we use the logging data from week 1 to week 7 to predict the learners' performance at week 12 with *lead* equals to 4 and *lag* equals to 7.

3. FEATURES ENGINEERIN
Table 1. Self-proposed covariates

	NAME	Definition
x1	stopout	Whether the student continue submit problem
x2	total_duration	Total time spent on all resources
x3	number_forum_posts	Number of forum posts
x4	number_wiki_posts	Number of wiki posts
x5	average_length_forum_post	Average length of forum posts
x6	number_distinct_problems_submitted	Number of distinct problems attempted
x7	number_submissions	Number of submissions
x8	number_distinct_problems_submitted_correct	Number of distinct correct problems
x9	average_number_submissions	Average number of submissions per problem (x7 / x6)
x10	observed_event_duration_per_correct_problem	Total time spent /number of distinct correct problems (x2 / x8)
x11	submissions_per_correct_problem	Number of problems attempted / number of correct problems (x6 / x8)
x12	average_time_to_solve_problem	Average time between first and last problem submissions for each problem (average(max(submission.timestamp) -min(submission.timestamp) for each problem in a week))
x13	observed_event_variance	Variance of a student's observed event timestamps
x14	number_collaborations	Total number of collaborations (x3 + x4)
x15	max_observed_event_duration	Duration of longest observed event
x16	total_lecture_duration	Total time spent on lecture resources
x17	total_book_duration	Total time spent on book resources
x18	total_wiki_duration	Total time spent on wiki resources

We extracted 18 self-proposed features, 7 crowd-proposed features (according to Taylor's work [5]) and 6 study habits related behavioral features on a per-learner basis, these features are list in table 1, table 2 and table 3. And then these features are

assembled from different weeks as separate variables to build predictive models.

Table 2. Crow-proposed covariates

	NAME	Definition
x201	number_forum_responses	Number of forum responses
x202	average_number_of_submissions_percentile	A student's average number of submissions / the average of all the students' submission
x203	average_number_of_submissions_percent	A student's average number of submissions / maximum average number of submissions
x204	pset_grade	Number of the week's homework problems answered correctly / number of that week's homework problems
x205	pset_grade_overtime	Difference in grade between current pset grade and average of student's past pset grade
x206	correct_submissions_percent	Percentage of the total submissions that were correct (x 8 / x 7)
x207	average_predeadline_submission_time	Average time between a problem submission and problem due date over each submission

Table 3. Study habits related behavioral features

	NAME	Definition
x301	problem_finish_percent _pre_start24h	The number of problem learner finished correctly in the first 24h after the problem issued
x302	problem_finish_percent_pre_deadline24h	The number of problem learner finished correctly in the last 24h before the problem due
x303	time_first_visit	Min(time_first_problem_get, time_first_html etext_access) - project_issue_time
x304	time_till_first_check	Average of all problem the time between problem_first_check and problem_first_get
x305	study_before_submit	Total book duration before problem submit + total video duration before problem submit
x306	discussion_duration_after_incorrect_submit	Total discussion duration after incorrect submitssion

4. RESULTS

As shown in figure 1, for all learners, our models achieved an average AUC as high as 0.838 (and lacking study habits features is 0.795) when predicting one week in advance.

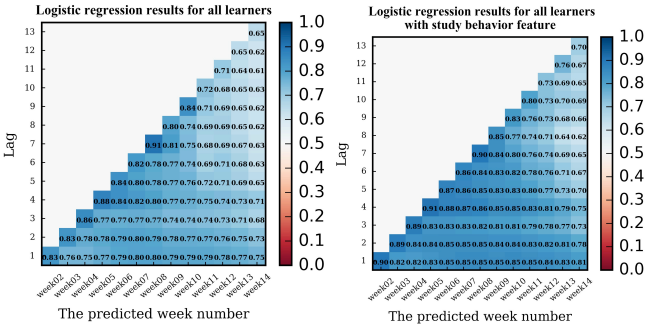


Figure 1. Heatmap for the logistic regression dropout prediction problem

From feature importance analysis as shown in figure 2, the study habits related behavioral features (x301-306) had played more important roles in the dropout prediction. Top features that had the most predictive power including *problem_finish_percent_pre_deadline24h*, *study_before_submit*, and *time_first_visit*.

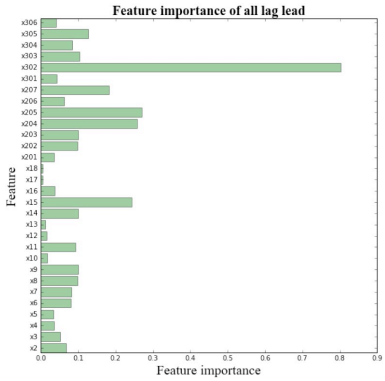


Figure 2. Feature importance

With new features related to study habits, the AUC of our predicting improved (figure 3), especially for the small sample size group (wiki / forum contributor and fully collaborative).

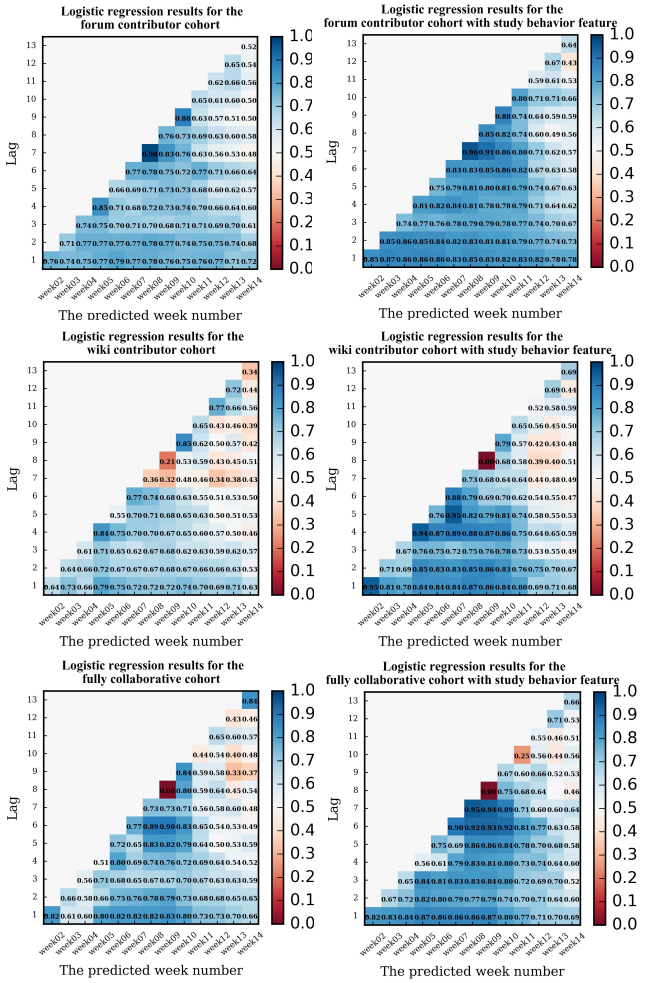


Figure 2. Heatmap for the logistic regression dropout prediction problem for three groups

In the future, we will try to using improved predictor each week within the course progress to deliver the intervention into small private online course.

5. REFERENCES

[1] Balakrishnan, G., & Coetzee, D. (2013). Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*.

[2] Whitehill, J., Williams, J. J., Lopez, G., Coleman, C. A., & Reich, J. (2015). Beyond prediction: First steps toward automatic intervention in MOOC student stopout.

[3] Loya, A., Gopal, A., Shukla, I., Jermann, P., & Tormey, R. (2015). Conscientious behaviour, flexibility and learning in massive open on-line courses. *Procedia-Social and Behavioral Sciences*, *191*, 519-525.

[4] Liang, J., Yang, J., Wu, Y., Li, C., & Zheng, L. (2016, April). Big Data Application in Education: Dropout Prediction in Edx MOOCs. In *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on* (pp. 440-443). IEEE.

[5] Taylor, C., Veeramachaneni, K., & O'Reilly, U. M. (2014). Likely to stop? Predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*.

Exploring the Relationship Between Student Pre-knowledge and Engagement in MOOCs Using Polytomous IRT

Jingxuan Liu and Hongli Li
Georgia State University, USA
jliu56@student.gsu.edu, hli24@gsu.edu

ABSTRACT

One of the issues that MOOCs face since its emergence is the low engagement rate and accomplish rate. As an open and free education source, MOOCs are available for people around the world with different motivations and previous knowledge to join. It is a challenge to keep students engaged in a MOOC environment. In the present study, we implement a polytomous item response model (IRT) to explore the relationship between students' self-evaluation of their previous knowledge and students' engagement behaviors in a Geography MOOC. Specifically, we estimate students' latent trait, pre-knowledge, through 15 likert-scale items. Engagement behaviors include assignment, peer review, forum, comment, quiz, and lecture. Each of them is quantified by the aggregated frequency. Then we examine the correlation between pre-knowledge and each type of engagement behavior. We find self-evaluation on previous knowledge cannot predict students' engagement behaviors for any type of engagement. This application indicates that the self-evaluation of pre-knowledge does not predict student engagement in MOOC environment. However, it shows that traditional psychometric models used for standardized tests may be useful and promising in the MOOC context.

Keywords

MOOC, engagement, pre-knowledge, Polytomous IRT

1. INTRODUCTION

A massive open online course (MOOC) is a model for delivering learning content online to anyone who wants to take a course, with no limit on attendance. MOOC engagement is a concept to describe students' involvement of a MOOC. Usually it includes behaviors like posting questions and comments in the MOOC system, submitting assignment and quiz, and other behaviors, which can directly predict students' achievement. Although during the past decade, the number of MOOC students increased tremendously across the world, the low accomplishment and low level of active engagement is always a problem for MOOC development [1]. MOOC engagement is important to predict students' achievement and to show whether students really learned something from the course or not. Students' prior knowledge, which was defined by first two assignments' performance, in computer science and problem solving had impact on their MOOC performance [3]. In the current research, we used pre-course survey data to define pre-knowledge of Geography and to explore if it can predict students' MOOC engagement. Also we use a polytomous IRT model to exam each item and their performance.

2. POLYTOMOUS IRT

Polytomous IRT model is an important model in the IRT family, which is designed for items with more than 2 possible options. Within polytomous IRT models, there are mainly four types: the partial credit model, the rating scale model, the generalized partial credit model, and the graded response model. One example of the application of the graded response model is attitude survey data. Usually the format of item in an attitude survey is likert-scale. For example, for question, "how much do you think you like this opera?", the options can be 5 likert scale from "I like it very much" to "I don't like it at all". The mathematic equation for polytomous IRT model is the following:

$$P^*_{x_{ij}}(\theta_i) = P(X_{ij} \geq x_{ij}|\theta_i) = \frac{e^{Da_j(\theta_i - b_{ij})}}{1 + e^{Da_j(\theta_i - b_{ij})}}$$

In the above equation, D equals to 1.7. For each item j, a_j is a discrimination parameter, and b_{ij} is the difficulty parameter for each option i in each item j ($b_1 < b_2 < ... < b_n$) [2]. Figure 1 indicates a graded response function of a polytomous item. Take the blue line as an example, people with higher theta level seldom choose this option, since the slope is roughly negative.

3. METHOD

3.1 Data

Data comes from a MOOC in Geography. It has enrolled over 100,000 students from 200 countries to date. Data from its 2014 class was used in the present study. In total, after excluding students with little data, there were 3058 students in the current analysis.

3.2 Measure

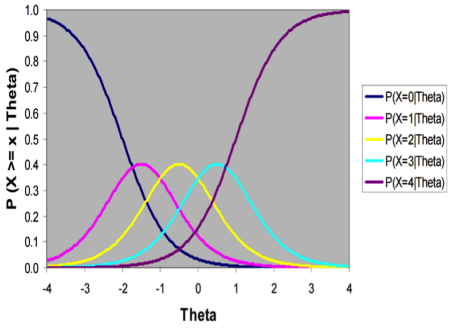


Figure 1: Graded response function

Table 1: Factor loading for each item

Item	1	2	3	4	5
Factor Loading	0.630	0.427	0.608	0.782	0.522
Item	6	7	8	9	10
Factor Loading	0.726	0.705	0.769	0.798	0.697
Item	11	12	13	14	15
Factor Loading	0.668	0.657	0.656	0.698	0.800

There are 15 seven-point likert-scale items, from "strongly agree" to "strongly disagree" designed for students to evaluate their pre-knowledge of Geography. One example is "I enjoy reading maps." In terms of the students' engagement behavior, there are six criteria including assignment, peer review, forum, comment, quiz, and lecture. The method for quantify them is to aggregate the number of times they participate in each type of behavior.

3.3 Procedure

The graded response model was applied using package mirt in R to estimate students' pre-knowledge of Geography. Then the Pearson correlation coefficients between pre-knowledge and each type of engagement behaviors were calculated respectively to examine if students' pre-knowledge influence their engagement behaviors in the MOOC environment.

4. RESULTS

The model fit indices verify a good model fit (RMSEA=0.047, RMSEA_5=0.041, RMSEA_95=0.053, CFI=0.959). The factor loading estimation shows that these 15 items can be used to measure the latent trait, pre-knowledge of Geography (table 1). The parameter estimates are presented in table 2, and the graded response function for each items is shown in the following figure 2. Additionally, table 3 presents the correlation coefficients between pre-knowledge of Geography and each type of engagement behavior.

5. CONCLUSIONS

Table 2: Parameter estimation for each item.

item	a	b1	b2	b3	b4
1	1.38	-1.741	-0.454	1.074	N/A
2	0.804	-2.746	-0.255	2.151	N/A
3	1.302	-6.566	-1.68	0.12	1.842
4	2.133	-5.112	-1.695	-0.553	0.581
5	1.041	-7.801	-2.436	-0.504	0.969
6	1.795	-5.465	-1.72	-0.199	1.096
7	1.693	-5.613	-2.494	-1.345	-0.301
8	2.049	-5.19	-1.768	-0.606	0.639
9	2.257	-5.027	-1.878	-0.757	0.23
10	1.654	-5.692	-1.828	-0.3	0.986
11	1.529	-5.932	-1.553	-0.026	1.268
12	1.482	-6.049	-2.67	-1.278	0.001
13	1.478	-6.048	-2.213	-0.933	0.253
14	1.66	-1.771	-0.307	0.898	N/A
15	2.268	-5.02	-1.621	-0.545	0.52

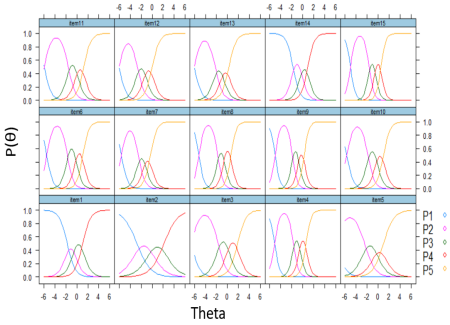


Figure 2: Graded response function for each item.

Table 3: The Pearson correlation coefficient between pre-knowledge and Engagement Behavior Type (EBT)

EBT	Pre-knowledge of Geography
	The Pearson correlation coefficient
assignment	-0.018
peer review	-0.022
forum	-0.016
comment	-0.022
quiz	-0.019
lecture	-0.025

All of the 15 items have relatively good loading on one factor, so it is reasonable to use one-dimensional IRT model. Also, the fit indices show that this graded response model fit well with the data. In terms of the discrimination index, item 8, item 9, item 15 have very good discrimination level. It indicates that these three items can provide more information in terms of students' pre-knowledge of Geography than other items. In terms of the difficulty parameter, b4 cannot be estimated for item 1, item 2, and item 14. This indicates that these items might be problematic.

All of the correlation coefficients are negative and nonsignificant (p-value>.05). This results indicates that although the general trend is students with less pre-knowledge of Geography will have less frequency of engagement behavior, none of them are statistically significant. In other words, whether students report a relative rich or poor pre-knowledge of Geography cannot predict their engagement behaviors. One of the explanation may be the pre-knowledge here is measured by self-evaluation, which relates to the meta-cognitional ability of students. This subjective report is different from objective questions, such as "have you taken any university level courses related to this MOOC course?" In further research, more direct measure of pre-knowledge is needed.

6. REFERENCES

[1] J. Daniel. Making sense of moocs: Musings in a maze of myth, paradox and possibility. *Journal of interactive Media in education*, 2012(3), 2012.

[2] R. J. De Ayala. *The theory and practice of item response theory*. Guilford Publications, 2013.

[3] G. Kennedy, C. Coffrin, P. De Barba, and L. Corrin. Predicting success: how learners' prior knowledge, skills and activities predict mooc performance. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 136–140. ACM, 2015.

An Analysis of Students' Questions in MOOCs Forums

Meng Cao
School of Psychology, Central China Normal University, Wuhan, 430079
caomeng@mails.ccnu.edu.cn

Yun Tang
School of Psychology, Central China Normal University, Wuhan, 430079
tangyun@mail.ccnu.edu.cn

Xiangen Hu
School of Psychology, Central China Normal University, Wuhan, 430079
xiangenhu@mail.ccnu.edu.cn

ABSTRACT

When learners become frustrated or confused, they can ask for help by posing questions in MOOCs forums. Students' questions reveal their needs and learning problems. If not answered timely and effectively, they may drop out. In the present study, students' questions from one Chinese MOOCs forum were collected and classified. Results showed that most of the posts in the forum were questions and the quantity of questions decreased over time although in some weeks the number of questions increased. Different types of questions have their own variation characteristics which means that the instructors need to focus on certain types of questions in the corresponding period.

Keywords

Student questions, MOOCs forum, classification, time-variation.

1. INTRODUCTION

Educators think highly of students' question asking. Questions posed by students can reflect active learning, knowledge construction, curiosity and the depth of the learning process [1]. Through analysis of these questions, instructors can better understand a student's thinking, so as to make more targeted teaching decisions [2]. Besides, students' questioning asking has association with their achievement. Learners with good performance behave better in the frequency or quality of questioning [3][4]. Thus, Teachers can also assess students learning based on their questions.

Researchers have investigated students' questioning behavior in a variety of educational settings, such as classroom, tutoring, online learning environments[1]. MOOCs allow students to pose their questions in a forum format and then wait for their questions to be answered by instructors and peer students. This online learning mode and asynchronous discussion pattern influences students' questioning behavior. Students may pose different kinds of questions at any time and at any place anonymously. The present study investigated students' questioning behaviors in the MOOCs forums including the quantity, classification and variations over time. According to previous research and forum data, we first establish standards to screen question posts, then classify and count the quantity of them, and finally observe the variation in the entire course.

2. DATA AND ANALYSIS

2.1 Platform and Data

We analyzed a forum of the course *The Introduction to Psychology* on the Chinese MOOCs platform XuetangX, which was launched in October 2013. This course has been opened for several sessions and has a large enrollment with tens of thousands learners. We chose the data for the 2015 Spring Session as it had the largest number of posts in the forum, starting from March 4th to September 15th. The whole course had 12-week lectures and two exams. The mid-term test took place between the 10th week and the 12th week. The final exam period ran from the 15th to 16th week. All the data came from www.kddcup2015.com and www.xuetangx.com.

2.2 Question Selection and Classification

First, we selected question posts from all the data. We regarded the question mark in the sentence as a marker feature. Some modal words and question words were also taken into consideration, such as" 是不是 (whether or not)", " 什么 (what)","怎么(how)","为什么(why)". And there are some fixed expression of questions, such as "我不懂(I do not know)","我很困惑/疑惑(I am confused)"[4]. Two researchers labeled the posts separately, then compared and made an agreement on the differences. The inter-rater agreement was 86% (representing agreement on 880 items out of 1029 opportunities for agreement multiplied by 100).

After filtering posts, a taxonomy of the questions was created based on Brinton's[5] classification on MOOCs discussion threads and question posts in the forum, including five categories: (1) Course management questions, relating to course design, time arrangement, learning resources, etc.; (2) Course content questions, involving learner's understanding of the learning materials or exercises; (3) Interaction questions, where learners ask and exchange experiences, learning methods and emotions; (4) Platform operation questions, students encounter when operating the platform; (5) Other, including vague expression and irrelevant questions. Two researchers classified the question posts separately and then reached an agreement. The inter-rater agreement was 82% (representing agreement on 613 items out of 751 opportunities for agreement multiplied by 100).

We calculated the total amount of students' question posts, the distribution of different classifications and different types of question variation over the weeks of the course.

3. RESULTS

3.1 The Quantity of Students' Question Posing

In the forum, 1002 people participated in the discussion, accounting for only 3 per cent of the total registers. Among them, 569 students posed 1029 posts, getting 3165 replies, which means that the average reply per post is 3.1. Two researchers screened 751 question posts, accounted for about 73% of the total posts,

indicating that learners' main activity in the MOOCs forum was question asking and answering. Figure 1 shows the quantity of students' questions over the course weeks. The number of posts decreased in general with a few fluctuations.

Figure 1: The quantity of students' questions over course weeks

3.2 The Distribution of Five Categories

Table 1 shows the amounts and proportions of five categories, as well as number of replies and average reply per question on each category. The quantity of course management questions are the most while course content questions are only the second. This may due to instructors' low participation in the forum. In the whole course, only some community assistants and administrators posed a limited number of posts and answers. However, course management questions and platform operation questions mainly rely on instructors' answers. As for the course content questions and interaction questions, they can be answered by both instructors and peer learners. Without prompt and proper replies, the first and forth kinds of questions will be repeatedly asked. So the average reply of them are lower than course content questions and interaction questions.

Table 1. The quantity of questions and their replies

Question type	Quantity	Proportion of the total questions	Replies	Average reply per question
Course management questions	334	44.5%	827	2.5
Course content questions	248	33.0%	875	3.5
Interaction questions	49	6.5%	218	4.4
Platform operation questions	111	14.8%	274	2.5
Others	9	1.2%	20	2.2

3.3 The Time-variation of Three Categories

As only a very small number of questions belong to the third and fifth category, we removed them from further analysis and calculated the quantity of the other three categories by course week. Figure 2 shows the relationship between course weeks and question quantity, suggesting a decreasing trend for all the types of questions. However, each type also has its specific characteristics. Course management questions existed throughout the course, because learners will generate a series of questions on textbook, exam, and certificate from start to end. At some time, these questions increased significantly. In contrast, course content questions disappear after the lectures are over. Questions mainly emerge in certain chapters. As to the platform operation questions, the proportion is lower while students may encounter more problems in some weeks on the practice submission.

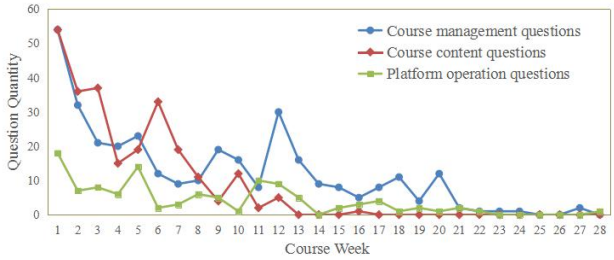


Figure 2: Question quantity of three categories in every course week

To summarize, through the analysis of students' questions in the forum, we can learn the patterns of their questioning behavior and in turn improve instructions in MOOCs. Instructors need to focus on certain kind of questions during different periods and provide appropriate guidance and answers. Course management questions and platform operation questions will influence learners' learning progress, so instructors should clearly describe details of course arrangement to avoid misunderstanding and confusion. When platform errors occur, they need to solve the problem as quickly as possible or give suggestions to learners. As to the course content questions, even without instructors' replies, learners and peers will try to discuss and find answers by themselves. So the main task of instructors are guiding their discussion and giving answers at the proper time.

The current study is part of a larger project studying the long-term impact of question asking/answering in MOOCs. We expect a significant relation between student's completion rate and the way students questioning/answering behaviors. Further study will be reported in the future.

4. REFERENCES

[1] Li, H., Duan, Y., Clewley, D. N., Morgan, B., Graesser, A. C., & Shaffer, D. W., et al. 2014. *Question Asking During Collaborative Problem Solving in an Online Game Environment. Intelligent Tutoring Systems*.

[2] Colbert, J. T., Olson, J. K., & Clough, M. P. 2007. Using the web to encourage student-generated questions in largeformat introductory biology classes. *Cbe Life Sciences Education, 6*(1), 42-48.

[3] Harper, K. A., Etkina, E., & Lin, Y. 2003. Encouraging and analyzing student questions in a large physics course: meaningful patterns for instructors. *Journal of Research in Science Teaching, 40*(8), 776-791.

[4] Graesser, A. C., & Person, N. K. 1994. Question asking during tutoring. *American Educational Research Journal, 31*(31), 104-137.

[5] Brinton, C. G., Chiang, M., Jain, S., Lam, H., Liu, Z., & Wong, F. M. F. 2014. Learning about social learning in moocs: from statistical analysis to generative model. *IEEE Transactions on Learning Technologies, 7*(4), 346-359.

Tutorials

Real-time programming exercise feedback in MOOCs

Zhenghao Chen, Andy Nguyen, Amory Schlender, Jiquan Ngiam
Coursera
381 East Evelyn Ave
Mountain View, CA, USA
{zhenghao, anguyen, aschlender, jngiam}@coursera.org

ABSTRACT

We present an active learning system for coding exercises in Massively Open Online Courses (MOOCs) based on real-time feedback. Our system enables efficient collection of personalized feedback via an instructor tool for automated discovery and classification of bugs.

1. INTRODUCTION

Active learning is a learning approach that "requires students to do meaningful learning activities" in contrast to traditional lecture-based approaches where "students passively receive information from the instructor" [2]. In active learning, timely feedback is important as it helps learning and reduces the risk of learner disengagement due to repeated failure to complete learning activities.

MOOCs have leveraged in-videos quizzes as an active learning strategy, but these quizzes have traditionally been limited to multiple choice questions. One reason that introducing higher order tasks, such as coding exercises, has been challenging is that it is difficult to provide good feedback. Most automated code grading systems allow for efficient grading through unit testing, but these methods are often limited in the forms of feedback they can provide.

Feedback that helps learners understand their errors can improve learning outcomes. Stamper et al. [5] demonstrated significant problem completion rate improvements in a logic course when feedback was available to learners. This has motivated related developments in data-driven methods to generate such feedback [3, 4, 1].

In this demo, we will show a system that enables instructors to efficiently generate and provide real-time feedback for programming exercises in MOOCs through extensions to Executable Code Blocks (ECBs) [6] and the Codewebs engine [1]; these exercises can be embedded throughout the learning experience to enable rich active learning.

2. EXECUTABLE CODE BLOCKS

Executable code blocks (ECBs) [6] enable learners to write and execute code directly in their web browser. The primary advantage of ECBs is that they can be tightly integrated into the course experience. For example, immediately after a concept is explained in a video, a learner can be asked to implement the specific concept in an ECB.

ECBs usually employ unit testing strategies to evaluate if a learner's implementation is correct. We extend ECBs such that when a learner makes an incorrect submission, they can request additional feedback that highlights potential errors in their submission and provides hints that guide the learner towards correcting these errors (see figure 1). These hints are provided efficiently by an instructor through an extension of the Codewebs engine.

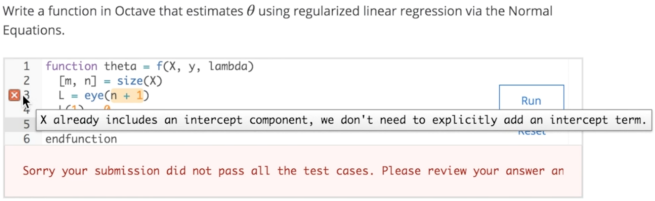


Figure 1: Hints provided in an ECB for an incorrect submission.

3. CODEWEBS ENGINE

We use the Codewebs engine [1] to localize errors in learner code submissions and identify common classes of errors. We describe here the relevant process of doing so automatically at a high level, and refer the reader to [1] for details.

The Codewebs engine operates on the abstract syntax tree (AST) representation of code submissions. Let n be a node in the AST, T_n be the subtree rooted at n, and P_n be the subtree rooted at the *parent of* n. The local context of T_n, denoted by T_n^c, is P_n with T_n removed (see figure 2).

We say that T_n^c is a buggy context if submissions containing T_n^c are more likely to be incorrect than by random chance. The Codewebs engine declares that P_n is a bug if T_n^c is a buggy context but no subtree of T_n has a buggy context. Given a bug P_n, the Codewebs engine then searches for a correction C such that replacing P_n with C results in a correct program.

We extend Codewebs in two ways. First, we modify the localization process to consider local contexts that are semantically equivalent[1]. This allows us to discover more bugs across submissions that might have *syntactically* distinct but *semantically* equivalent contexts. We also use this to improve correction discovery in a similar way (see figure 3) and improve correction searching to handle instances where multiple bugs occur within a submission.

Second, we introduce the concept of bug groups or error modes. Two bugs B and B' belong to the same group *iff* B

[1]We follow the definition of semantic equivalence used in [1].

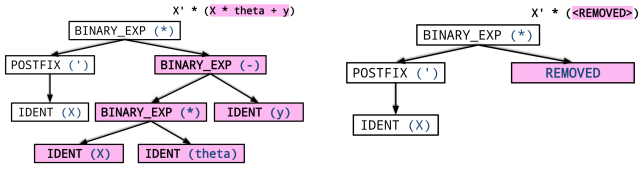


Figure 2: Left: Subtree P_n containing subtree T_n in pink. Right: T_n^c, the local context of subtree T_n.

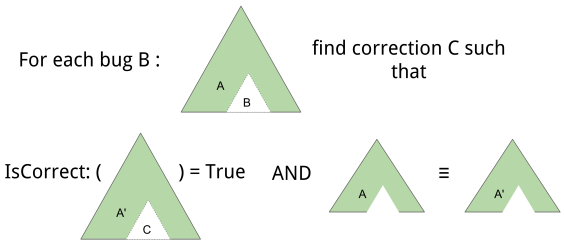


Figure 3: Visual illustration of finding corrections for bug B, C is a correction for B if we can find a correct submission where C is surrounded by A' and A' is semantically equivalent to A.

is semantically equivalent to B' and the correction for B is semantically equivalent to the correction for B'.

4. INSTRUCTOR ANNOTATIONS

By grouping bugs together, instructors can provide a hint for each error mode (instead of for individual submissions). These hints power the feedback features mentioned in section 2 (see figure 1).[2]

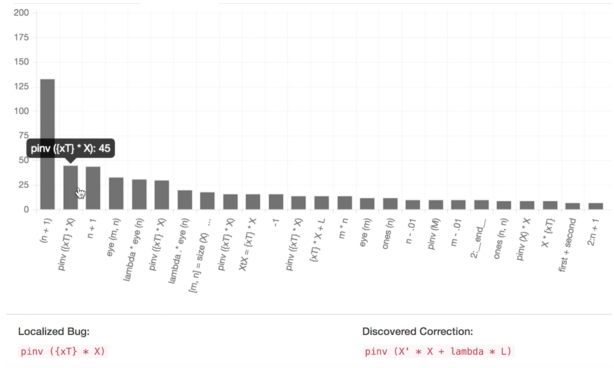


Figure 4: Instructor tool for exploring common errors based on bug equivalences classes.

Furthermore, we can provide instructors with a tool (see figure 4) to explore these common error modes. This tool orders bug groups by the frequency at which they appear in learner submissions. This enables instructors to quickly understand the most common errors made by learners. This breakdown is useful for course material improvement as they can expose common learner misconceptions.

5. RESULTS

We introduced 3 ECBs into the Machine Learning MOOC on Coursera involving tasks of varying levels of complexity (e.g., implementing the cost function for regularized linear regression). Each ECB required between 10 and 20 lines of code each to solve.

For each ECB we collected between 3,118 and 5,550 submissions, consisting of between around 1,000 and 3,000 distinct ASTs (see table 1). These submissions were used to train the Codewebs model. We find that a relatively small number of error groups (40) is required to achieve good coverage

[2]It is also possible to show learners automatically generated corrections when instructor input is not available.

	Submissions	% Correct	Unique ASTs	% Coverage (40 bug groups)
RLR Normal Eqn	3,118	52.8%	1,338	61.0%
Matrix Inv Cost Fn	3,892	19.5%	1,440	49.5%
Matrix Inv Grad	5,550	11.5%	3,050	36.7%

Table 1: 3 ECBs added to the Machine Learning MOOC on Coursera

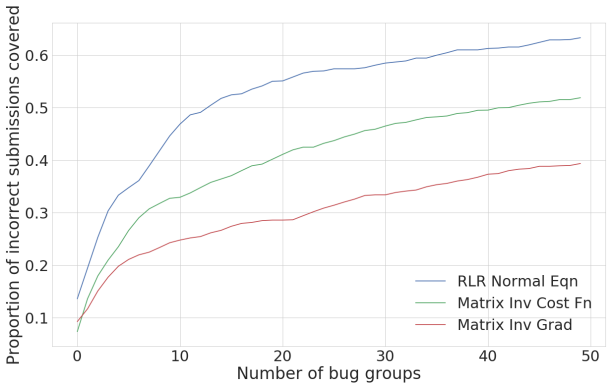


Figure 5: Percentage of incorrect submissions by number of error modes.

of a large fraction of incorrect submissions (see figure 5). Between 28.6% and 55.0% of incorrect submissions contain at least 1 of the 20 most common error modes, and between 36.7% and 61.0% contain at least 1 of the 40 most common error modes (see figure 5).

A teaching assistant was recruited to label the top 40 discovered error groups, and we are now running tests to understand the effects of this intervention on learning outcomes.

6. REFERENCES

[1] A. Nguyen, C. Piech, J. Huang, and L. Guibas. Codewebs: Scalable homework search for massive open online programming courses. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 491–502, New York, NY, USA, 2014. ACM.

[2] M. Prince. Does active learning work? a review of the research. *J. Engr. Education*, pages 223–231, 2004.

[3] K. Rivers and K. R. Koedinger. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, 27(1):37–64, 2017.

[4] R. Singh, S. Gulwani, and A. Solar-Lezama. Automated feedback generation for introductory programming assignments. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '13, pages 15–26, New York, NY, USA, 2013. ACM.

[5] J. C. Stamper, M. Eagle, T. Barnes, and M. Croy. *Experimental Evaluation of Automatic Hint Generation for a Logic Tutor*, pages 345–352. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[6] C. Wong. Active learning experiences with code executable blocks. `https://building.coursera.org/blog/2016/09/30/active-learning-experiences-with-code-executable-blocks/`.

Tutorial: Why data standards are critical for EDM and AIED

Robby Robson
Eduworks Corporation, Inc.
IEEE Learning Technology Standards
robby.robson@eduworks.com

Avron Barr
Aldo Ventures, Inc.
IEEE Learning Technology Standards
avron@aldo.com

Xiangen Hu
The University of Memphis
Central China Normal University
xhu@memphis.edu

SUMMARY

As EDM and AIED innovations proliferate, the ability for diverse products to consistently interpret each other's data will emerge as a critical issue. Formal data interoperability standards that enable diverse datasets to be curated, accessed, merged/compared and fruitfully analyzed will play a crucial role in research and in the successful mass adoption of products based on that research, as will standards that enable systems to produce data that can be mined by existing and yet-to-be-invented algorithms. Yet this important topic is often neglected by researchers and system developers, who naturally focus on the specific problems they set out to solve and do not consider how they can either contribute or consume data produced by other systems or how their innovations will fit into larger ecosystems. This tutorial is intended to:

- Raise awareness of the role of standards and their criticality for EDM and AIED;
- Provide participants with an understanding of the nature, status, and current activity of multiple international standards development effort relevant to educational data;
- Provide participants with insight into how they can beneficially apply standards and, in some cases, contribute to their development.

TOPICS

This tutorial will cover following topics:

- **Why schools, corporations, and government agencies require standards conformance in procurement**: How standards interact with regulations and requirements to facilitate the free exchange of information and data, to prevent "lock-in" and thereby lower costs, to ensure quality and minimal levels of functionality, and to protect the integrity and privacy of data.
- **How standards shape product categories and markets**: How standards can define functionality, product capabilities, and market segmentation. In many instances, standards determine which of a number of competing approaches will dominate. They can shape markets and lead to winners and losers and long-term consequences for producers, consumers, and researchers alike. There are obvious examples in areas such as telecommunications and manufacturing, but there are also examples in educational technology relevant to EDM and AIED.
- **How standards can support research and lower market entry barriers for innovative products**: How standards make it possible for innovative component technologies to be independently developed without requiring a vertical monopoly, and how they support research by making it possible for data produced by one system to be understood by another.
- **Types of standards (governance, process, and data interoperability)**: People often think of standards as relevant only to technical interoperability, e.g. to determining data formats, sizes, shapes, tolerances, and the like. But there are other types of standards as well, including process standards such as ISO 9001 and Software Engineering Standards and governance standards that address issues such as data preservation, curation, ethics, and privacy. All of these will play a critical role for EDM and AIED.
- **International standards organizations**: A survey of standards development organizations (SDOs). This segment will briefly explain the structure of international standardization, the principles by which ISO, IEC, IEEE, W3C, and similar SDOs abide (openness, consensus, balance, due process, right of appeal), the differences (and similarities) between these and industry consortia, and the SDOs that are most relevant to EDM and AIED.
- **How standards are made**: The standards development process has been refined over many years to ensure that each SDO can be productive within its principles and goals. This segment will describe how standards development works so that participants have an idea of what it entails and how to participate.
- **A brief history of standards related to educational and training technology**: Starting circa 1996, various organizations and consortia began developing standards, some better known and more widely adopted than others. We will briefly survey this history with a view towards extracting some key "lessons learned" that apply generally to standards development: The perfect is the enemy of the good; standards are a poor way to define systems but a great way to define how they interoperate; simplicity and modularity leads to adoption; industry participation is vital; and how to avoid standards wars.
- **Current international standards activity relevant to EDM and AIED**: This is a major segment that will touch on a large number of relevant standards, including:
    - Metadata standards
    - Format standards (e.g. data shop)
    - Competency and learner information standards
    - Data reporting and curation standards

- Platform standards
- Big data and AI ethics
- Student data governance
- Possibly needed additional standards

Each standard will be summarized and described in terms of what problem(s) it solves, how it works, who developed it, who uses it, how it fits in with other standards, and what the presenters see as its future.

- **Tools for applying standards to EDM and AIED**: This segment will focus in on a few high-value standards and applications of standards to EDM and AIED. This segment is the punchline of the tutorial and will cover the standards that the presenters feel are most important. It will focus on existing or emerging technologies that participants can apply now or in the near future and will provide concrete examples of how standards are applied in software.
  - Using standards to report and collect data
  - Data set efforts (Datashop, Dataport)
  - The US DoD's Total Learning Architecture and related unification efforts
- **How to get involved in the standards development process**: This last, short segment will provide participants with information on how to get involved if they are interested, to be followed up offline.
- **Questions and Answers**: Adequate time will be set aside to address participants' questions and issues.

Presenter Relevant Bios:

- http://transformingedu.com/speakers/avron-barr/
- http://eduworks.net/robby/
- http://www.xiangenhu.info/

Tutorial: Principal Stratification for EDM Experiments

Adam C Sales
University of Texas College of Education
1912 Speedway Stop D5700
Austin, Texas, USA
asales@utexas.edu

ABSTRACT

Principal stratification (PS), which measures variation in a causal effect as a function of post-treatment variables, can have wide applicability in educational data mining. Under the PS framework, researchers can model the effect of an intelligent tutor as a function of log data, can account for attrition, and study causal mechanisms. Participants in this tutorial will learn how and when PS works and doesn't work, and will learn three methods of estimating principal effects.

1. PRINCIPAL STRATIFICATION IN EDM RESEARCH

Educational data miners are increasingly interested in causal questions—what interventions work, for whom, and how. Accompanying this interest is the widespread realization that there is no such thing as "the effect": actually, effects can vary widely between individuals. Estimating the differences in effects between types of learners is (in principal) straightforward for types defined prior to the onset of an experiment. But what about learners who use the software in different ways—or, even given the opportunity, don't use it at all? Traditionally, "post-treatment" variables, observed subsequent to treatment assignment, are treated as mediators whose analysis requires the kind of untestable assumptions randomization is supposed to avoid.

Principal stratification (PS) [2] offers a different approach: categorizing learners based on how they *would* (or would not) use the software if given the opportunity. Under the PS approach, an analyst begins by defining types, or "principal strata" of learners based on post-treatment measurements, then estimates the probability each learner is a member of each stratum (conditional on baseline covariates), and finally the average effect of the treatment within each stratum. In a randomized experiment, the final step of the process proceeds from the randomization (and, possibly, testable modeling assumptions). That is, researchers need not assume unconfoundedness, or that all relevant variables have been measured. The result is a principal effect, or separate estimate of an average treatment effect for each usage mode of interest; these may be used to explore causal mechanisms, study the conditions under which software might work better (or worse), learn dosage effects (i.e. does more usage translate to larger effects), and many other applications.

1.1 EDM Questions PS may Help Answer

PS could help address a wide range of research questions in EDM. Some examples are:

- Does the effect of an intervention depend on learners' (measured) emotional state?
- Are some sections of a software more effective than others?
- Do some learner strategies—such as hint usage or mastery learning—correspond to larger effects than others?
- Are there intermediate outcomes, such as mastery speed or error rate, that can serve as good surrogates for a final outcome, such as a post-test?
- Estimating treatment effects after attrition

Each of these questions estimates an average treatment effect for a group of learners which is defined based on variables measured only after the intervention began. This is the type of question principal stratification was designed to answer.

1.2 Estimating Principal Effects

The catch is that principal effects can be difficult to estimate. Estimating effects within principal strata depends on knowing who is in which stratum—for instance, which students in the control condition *would have* been frustrated, had they been assigned to treatment, or which students would have attritted, had they been assigned to the opposite condition—which is unobserved and must be inferred. The most popular and powerful approach begins by assuming a model (typically the normal distribution) for the outcome within each stratum and a model for who is in which stratum (typically logistic regression). Next, it fits a mixture model for those subjects with unobserved stratum membership. For instance, in an experiment comparing students assigned to use an intelligent tutor with students assigned to

use traditional curricula, a researcher looking to estimate average effects for high-hint users might model post-test scores for subjects in the control condition as a mixture of two distributions: one for students who would use many hints, and one for students who would not. The success of this approach depends on the fit of the model—misspecified models may yield misleading results—so extensive model checking is necessary. Further, even when the model is correctly specified, its success can depend on factors beyond the researcher's control [1].

Two other approached depend less on modeling assumptions, but may yield less precise estimates. One approach [3] estimates bounds for principal effects, rather than estimating the effects themselves. Another [4], applicable in some PS studies but not others, uses non-parametric techniques to identify plausible candidates for unobserved principal strata, and estimates effects based on those. These approaches are more "automatic" than the model-based approach, in that they do not require careful model fitting and checking, but still require researchers to specify the problem carefully.

1.3 My Expertise

For the past three years, I have been working on an NSF-funded project to use the PS framework to study data from the Cognitive Tutor Algebra I effectiveness study. With Dr. John Pane of the RAND Corporation, I have estimated various associations between Cognitive Tutor treatment effects and student usage. This has produced two EDM proceedings papers, [5] and [6]. As part of the project, I have developed a new method for estimating principal effects which expands on [4] and set of new diagnostic and model checking techniques. I have also worked extensively with Neil Heffernan's lab using PS to model data from ASSISTments experiments.

2. TUTORIAL PLAN

2.1 Introduction to Principal Stratification

The beginning of the tutorial will introduce the PS framework. First, we will discuss why principal stratification is necessary: participants will learn to distinguish post-treatment from pre-treatment variables and understand the conceptual and methodological issues with conditioning causal inference on post-treatment variables. Next, we will describe PS framework, so participants understand how it solves the problems with post-treatment conditioning. Finally, we will discuss methods for estimating effects within principal strata: what assumptions they depend on and the source for their identification. We will give a brief overview of the various PS methods that we will explore hands on, in more depth, during the remainder of the tutorial.

2.2 Hands on PS Estimation

The second half of the tutorial will focus on three classes of methods to estimate principal effects: nonparametric bounds, nonparametric randomization inference, and model based PS.

I will provide two real EDM datasets that participants can use for exercises. The first will be a subset of the data from the Cognitive Tutor effectiveness study, comparing subjects assigned to use the Cognitive Tutor to those assigned to traditional curricula. The study produced rich log-data—PS can be used to compare treatment effects between sets of learners who used, or would have used, the tutor differently. The second dataset will come from an experiment run on the ASSISTments platform [7]. I will also give participants the opportunity to bring their own datasets to the tutorial.

The methods will be taught in `R`, a free, open-source language for statistical computing. We will begin with a brief introduction to the software: how to read in data, and how to write and execute simple code.

The bounding portion will be based on [3], which describes a set of bounds on principal effects, depending on available covariates and certain identification assumptions. We will set out a number of real or realistic data scenarios and discuss which bounds may be appropriate when. Next, we will use `R` to calculate the appropriate bounds for principal effects.

The randomization inference portion will be based on [4] and extensions I have developed. They depend on the assumption of monotonicity—that principal stratum membership is directly observable for all members of either the treatment or the control group. I will provide code in `R` to estimate confidence intervals for principal effects with and without covariates the predict stratum membership.

The model based portion will use Bayesian methods, with the JAGS language, via `R` and the `R2Jags` package. We will practice estimating principal effects with pre-written JAGS code (which I will explain) as well as discuss diagnostic tools: model checking, convergence diagnostics, and small simulation studies.

References

[1] A. Feller, E. Greif, L. Miratrix, and N. Pillai. Principal stratification in the twilight zone: Weakly separated components in finite mixture models. *arXiv preprint arXiv:1602.06595*, 2016.

[2] C. E. Frangakis and D. B. Rubin. Principal stratification in causal inference. *Biometrics*, 58(1):21–29, 2002.

[3] L. Miratrix, J. Furey, A. Feller, T. Grindal, and L. C. Page. Bounding, an accessible method for estimating principal causal effects, examined and explained. *arXiv preprint arXiv:1701.03139*, 2017.

[4] T. L. Nolen and M. G. Hudgens. Randomization-based inference within principal strata. *Journal of the American Statistical Association*, 106(494):581–593, 2011.

[5] A. C. Sales and J. F. Pane. Exploring causal mechanisms in a randomized effectiveness trial of the cognitive tutor. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.

[6] A. C. Sales, A. Wilks, and J. F. Pane. Student usage predicts treatment effect heterogeneity in the cognitive tutor algebra i program. In *Proceedings of the 9th International Conference on Educational Data Mining*, 2016.

[7] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184. ACM, 2016.

Whitebox: A Device To Assist Group Work Evaluation

Daisuke Yukita
Keio University Graduate School of Media Design
4-1-1 Hiyoshi Kohoku
Yokohama, Japan
daisuke@kmd.keio.ac.jp

ABSTRACT

With the growing trend of Active Learning, group work is becoming increasingly common among education of all ages. Among the many advantages of group works, we have also witnessed how difficult it is for teachers to keep an eye on the activities within each group, thereby turning the group work process itself into a black box from the teachers' perspective. In order to propose a solution for this problem, this study introduces Whitebox, a device that discreetly gathers several types of data within group work, which are then visualized for the teacher to reference after the group work. The user study with high shcool students showed that group work analysis by Whitebox led to deeper understanding of how each student performed within their group.

1. INTRODUCTION

Considering the fact that there can be more than 30 students in a typical high school class in Japan, it is highly difficult for teachers to look over the activities within each group during group work. In other words, the students' processes of their group work remain a blackbox for teachers. In addition, how we evaluate group work is still an often debated issue, especially in formal education where a standard evaluation method is required. Whitebox was developed in order to suggest a solution towards such obstacles for schools in adopting group work. By placing the Whitebox in the middle of a group work table, it tracks the activities within the group. Later the recorded data will be visualized for the teacher to check, enabling teachers to get a rough idea of what kind of process each group went through without being physically present all the time. Furthermore, Whitebox quantifies the group work process by measuring talking ratios, volumes, etc., suggesting novel evaluation measurement units for group work, which can be used as the future standard.

2. LITERATURE REVIEW

While many of EDM / LA related researches have been limited to online or digital learning environments, recent studies have stepped in to face-to-face classroom activities with the help of advanced sensors and devices. Martinez-Maldonado et al. [1] created a realtime feedback system for teachers to provide feedback just at the right time using the data obtained from MTClassroom, a multi-touch tabletop that analyzes the strategies of student groups. Evans et al. [2] also proposed to identify touch patterns of students on an interactive tabletop to analyze the quality of collaboration. Whitebox aims to provide similar feedback to the teachers without relying heavily on each hardware. In terms of providing measurement units for conversation and collaboration, Lederman et al. [3] proposed Open Badges, an open source toolkit to measure face to face interaction and human engagement in real-time with custom hardware. Olguin et al. [4] states that such sociometric badges can make group collaborations more efficient by providing context, but such badges are mainly used for business and work environments, and they must be designed alongside students and teachers if it were to be used in a classroom setting.

3. SYSTEM DESCRIPTION

Initially, Whitebox used Kinect's mic arrays to determine which direction the audio is coming from, thereby distinguishing who is currently speaking. Following the feedbacks from a pilot test, however, audio recording was also done with separate pin microphones attached to the students' clothing. The attained audio is processed to obtain the volume as well. Using Kinect's depth camera, Whitebox also obtains the participants' body skeletons, allowing it to track their hand coordinates and their posture angles. Due to the way the current system is designed, Whitebox can only track the participants' data when they are sitting down and are not moving around or switching positions. The entire group work is also recorded, and when the group work is finished the audio data is converted into text using Google Cloud Speech API.

4. USER STUDY

A user study was conducted during a 4 day Design Thinking workshop at Tokyo Metropolitan College of Industrial Technology high school. In this user study, we especially focused on one group of four students, student A, B, C and D, and recorded only those 4 students' activities. After the workshop, the 4 visualizations and speech-to-texts were shown to both teachers and students separately, followed by an hour long discussion each on what those data meant to them.

Figure1 shows the study setup.

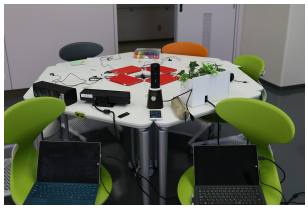


Figure 1: User study setup

The data acquired from the 4 workshops was processed, then visualized in to A4 infographic posters as shown in Figure 2.

Figure 2: visualizations from user study

To provide a more fine grained analysis of each session, we also provided additional visualization that plotted the students' audio data, posture data and hand position data along the timeline of the workshop. Figure 3 is an examples of the additional visualization.

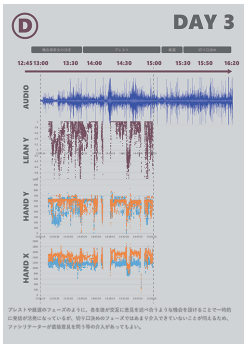


Figure 3: additional visualization of student D from day 3

By visualizing the data from all four sessions, it was possible to get a grasp of how each student behaved in the workshops. It is important to note here that what the visualizations suggested matched with the thoughts of facilitators who were in charge of this group (e.g. that student D would speak the least and student B would take charge of the overall discussion), meaning that Whitebox would be able to assist teachers to evaluate group work without them having to be present at each group's table all the time. As for the speech-to-text, it helped the teachers to see what words were mentioned most frequently. With improved conversion accuracy, it would become possible to process the text to search the most frequenty mentioned conjunctional phrases per student in order to see the characteristics of their contributions.

By post processing the audio data recorded, we were also able to provide visualizations on the order of conversational turn taking during the discussion. The data was plotted for each 30 seconds of conversation. This enables the teacher to examine specific points in a discussion and analyse how it transitioned between the group members. An example is shown in Figure 4.

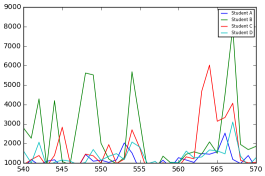


Figure 4: conversation transition

5. CONCLUSION

In this study we proposed Whitebox, a device that tracks the activities within a group work. Through the discussions with the teachers, we were able to see that Whitebox analysis certainly functioned as a guideline for a deeper understanding of the group and its students, and it also functioned as signs for what was and was not working in the group work, ultimately leading to improvements in the design of the class. Although not all the data we recorded seemed useful to the teachers, the measurements that Whitebox proposed, especially talking ratios, volumes and posture were valuable information for the teachers, uncovering the activities within the group that they otherwise would have missed. By using these measurements continuously, they can become a standard measurement unit in assessing group work.

6. REFERENCES

[1] R. Martinez-Maldonado, A.Clayphan, K.Yacef and J. Kay. "MTFeedback: providing notifications to enhance teacher awareness of small group work in the classroom." IEEE Transactions on Learning Technologies, 8(2): 187-200, 2015.

[2] A.C. Evans, J.O. Wobbrock, K.Davis. "Modeling Collaboration Patterns on an Interactive Tabletop in a Classroom Setting." Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work Social Computing, 860-871, 2016.

[3] O. Lederman, D. Calacci, D.C. Fehder, F.E. Murray, A. (Sandy) Pentland. "Open Badges: A Low-Cost Toolkit for Measuring Team Communication and Dynamics ." 2017 International Conference on Social Computing, Behavioral-Cultural Modeling, Prediction and Behavior Representation in Modeling and Simulation, 2017.

[4] D.O. Olguin, A. (Sandy) Pentland . "Sociometric Badges: State of the Art and Future Applications." IEEE 11th International Symposium on Wearable Computers, 2007.

Understanding Student's Reviewing and Reflection Behaviors Using Web-based Programming Grading Assistant

Yancy Vance Paredes
Arizona State University
699 S Mill Ave
Tempe, AZ 85281
yvmparedes@asu.edu

Po-Kai Huang
Arizona State University
699 S Mill Ave
Tempe, AZ 85281
phuang24@asu.edu

I-Han Hsiao
Arizona State University
699 S Mill Ave
Tempe, AZ 85281
sharon.hsiao@asu.edu

ABSTRACT
Paper-based assessment is still one of the most preferred methods in assessing students in a blended learning environment. However, it has several drawbacks such as having a high turnaround time before feedback is provided to the students. Furthermore, understanding how students attend to their graded papers is difficult to investigate because of the absence of empirical evidence. We describe in this paper a web-based system we developed that addresses some key issues when trying to understand the reviewing and reflection behaviors of the students. This system also aims to help instructors to efficiently and effectively grade paper-based assessments.

Keywords
Reviewing Behavior, Paper-Based Assessment, Educational Technology

1. INTRODUCTION
Paper-based assessment is still one of the most preferred methods in assessing students in a blended learning environment. Aside from being convenient to prepare, the possibility of students committing academic dishonesty is lower. However, it also has its drawbacks. Evaluating large amounts of test paper gives rise to the possibility of inconsistency among or even within graders [2]. Additionally, the feedback is limited [5]. Moreover, there is a high turnaround time before students receive their graded papers [1]. In terms of understanding the reviewing and reflecting behaviors of the students, it is difficult to systematically estimate how students review their paper-based assessments because of the absence of empirical evidence. It is not possible to determine whether students really do review their graded test papers. Thus, it is challenging to estimate the impacts of reviewing on learning.

2. WEB-BASED PROGRAMMING GRADING ASSISTANT (WPGA)
A web-based system was developed to address the above-mentioned issues. More specifically, it is designed to help students to review effectively. In addition, it aims to help instructors to efficiently and effectively grade paper-based assessments. The name of the system is Web-based Programming Grading Assistant (WPGA). The system is capable of capturing all activities performed by the users, which is mostly comprised of students' clickstream.

2.1 Documentation of Paper-Based Assessments
WPGA uses quick response (QR) codes to label the paper exam of a student. These generated codes are manually placed on the students' papers prior to scanning. Using an automatic document feeder, all the papers are scanned and uploaded to the system. The system automatically associates the scanned image to the corresponding student and the corresponding assessment. There are instances where the system may not accurately associate an image to a student. One possible reason would be due to the QR code being not readable. It could also be because the student is not registered in the system. When this happens, the instructor can just manually label the images.

2.2 Interface for Grading Assessments
After the exams are digitized, instructors can distribute the questions to be evaluated by different graders. The system allows multiple graders to work on the same assessment simultaneously. In effect, the turnaround time in the distribution of grades is reduced. The grading coherence will improve since graders will only be working on the question assigned to him or her.

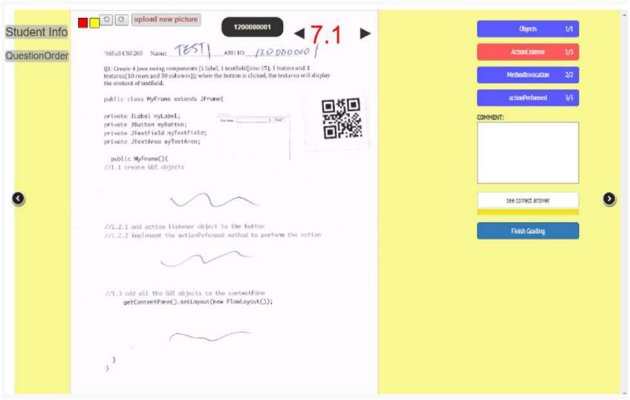


Figure 1. The grading interface of WPGA

The grading interface is shown in Figure 1. Buttons on the upper right portion represent a learning concept or a rubric that is used to evaluate a question. Every rubric default to a perfect score, which

translates to a full understanding of the concept. Whenever the button is clicked, the grade for the rubric is decremented and the overall score is recalculated. Also, the color of the button changes depending on the grade for the rubric. It could be blue (full understanding), red (partial understanding), or grey (missed the concept). The overall score can also be overridden, if necessary. The graders can also add markings on top of the student's paper. This will enable them to highlight the mistakes. Lastly, using the comment section, the graders can provide free form feedback. In previous studies [2,3], we found out that graders prefer to type their feedback rather than physically writing them on paper. One advantage of this over the traditional way of checking is the ability to copy and paste feedbacks of common and similar mistakes.

2.3 Interface to Encourage Student Reflection

After the instructor publishes the results of an assessment, the students can log in to the system and review it. There are two levels how the students can view the results: assessment level and question level. In the assessment level (shown in Figure 2), the general result is displayed. This includes the overall score obtained by the student along with the individual scores for each question. In the question level (shown in Figure 3), a detailed feedback for the particular question is provided. This includes the scores for all the rubrics, the markings on the student's paper, and the free form text provided by the grader.

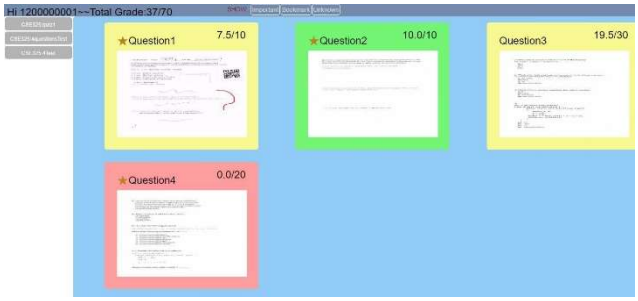


Figure 2. The assessment level view of the student interface

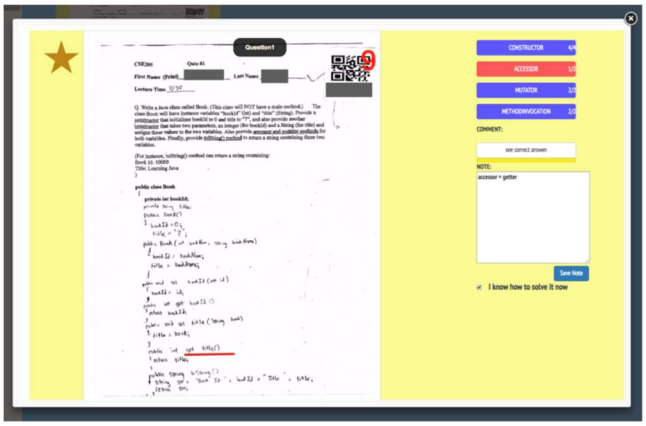


Figure 3. The question level view of the student interface

In addition to letting the students access a digital copy of their paper assessments, the system also allows them to reflect on the feedback given to them by the graders. We incorporated some features that help students track and monitor their learning. For example, in the question level, there is a checkbox that the students can tick to indicate whether they already know how to solve the problem after reviewing it. This is particularly useful for questions where they committed mistakes. Another feature is the bookmark which enables students to highlight the importance of a question. This could be used in future targeted reviews along with the use of filters. We also provided a free form text area to allow the students to type in his or her personal notes. The collection of these bookmarks, checkbox ticks, and notes are externalization of what the student knows. Through these features, it is hoped that students will be encouraged to reflect on their answers.

3. CASE STUDY

Using the system, we designed a classroom study and analyzed the logs collected from an *Object-Oriented Programming and Data Structures* class. We tracked and modeled students' reviewing and reflecting behaviors. Results show that students demonstrated an effort and desire to review assessments regardless whether they are graded or not [4].

4. FUTURE WORK

We intend to improve the system by using the feedback obtained from the users. For the next iteration, we are integrating the analytics module that will enable the instructors to quickly see a snapshot of the class performance and will enable them to gain insight on the assessments they gave to the students. Furthermore, we intend to do more research in understanding the reviewing behaviors of the students. This would allow us to create personalized review sessions that will help students do effective reviews.

5. REFERENCES

[1] Susan A. Ambrose, Michael W. Bridges, Michele DiPietro, Marsha C. Lovett, and Marie K. Norman, *How Learning Works: Seven Research-Based Principles for Smart Teaching*.: John Wiley & Sons, 16 April 2010.

[2] I.-Han Hsiao, "Mobile Grading Paper-Based Programming Exams: Automatic Semantic Partial Credit Assignment Approach," in *Lecture Notes in Computer Science*., 2016, pp. 110-123.

[3] I.-Han Hsiao, Sesha Kumar Pandhalkudi Govindarajan, and Yi-Ling Lin, "Semantic visual analytics for today's programming courses," in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (LAK'16)*, 2016.

[4] I.-Han Hsiao, Po-Kai Huang, and Hannah Murphy, "Uncovering reviewing and reflecting behaviors from paper-based formal assessment," in *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, 2017, pp. 319-328.

[5] Hannah E. Murphy, "Digitalizing Paper-Based Exams: An Assessment of Programming Grading Assistant," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, New York, NY, USA, 2017, pp. 775-776. [Online]. http://doi.acm.org/10.1145/3017680.3022448