STUDENT MODELING FROM DIFFERENT ASPECTS

by

Yan Wang

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

April 2016

APPROVED:

Dr. Neil T. Heffernan	Dr. Joseph E. Beck
Advisor – WPI	Reader – WPI

Contents

Abstract

With the wide usage of online tutoring systems, researchers become interested in mining data from logged files of these systems, so as to get better understanding of students. Varieties of aspects of students' learning have become focus of studies, such as modeling students' mastery status and affects. On the other hand, Randomized Controlled Trial (RCT), which is an unbiased method for getting insights of education, finds its way in Intelligent Tutoring System. Firstly, people are curious about what kind of settings would work better. Secondly, such a tutoring system, with lots of students and teachers using it, provides an opportunity for building a RCT infrastructure underlying the system. With the increasing interest in Data mining and RCTs, the thesis focuses on these two aspects. In the first part, we focus on analyzing and mining data from ASSISTments, an online tutoring system run by a team in Worcester Polytechnic Institute. Through the data, we try to answer several questions from different aspects of students learning. The first question we try to answer is what matters more to student modeling, skill information or student information. The second question is whether it is necessary to model students' learning at different opportunity count. The third question is about the benefits of using partial credit, rather than binary credit as measurement of students' learning in RCTs. The fourth question focuses on the amount that students spent Wheel Spinning in the tutoring system. The fifth questions studies the tradeoff between the mastery threshold and the time spent in the tutoring system. By answering the five questions, we both propose machine learning methodology that can be applied in educational data mining, and present findings from analyzing and mining the data. In the second part, we focused on RCTs within ASSISTments. Firstly, we looked at a pilot study of reassessment and relearning, which suggested a better system setting to improve students' robust learning. Secondly, we proposed the idea to build an infrastructure of learning within ASSISTments, which provides the opportunities to improve the whole educational environment.

Students vs. Skills:
Partitioning Variance Explained in Learner Models

Joseph Beck, Korinn Ostrow, Yan Wang
Worcester Polytechnic Institute
Worcester, MA 01609
{josephbeck, ksostrow, ywang14 } @wpi.edu

This section has been submitted to:

Beck, J., Ostrow, K. & Wang, Y. (2016) Students vs. Skills: Partitioning Variance Explained in Learner Models. *The 9th International Conference on Educational Data Mining*. ACM

ABSTRACT

Learner modeling is a significant tool within the Educational Data Mining (EDM) community that can drive system implementation and learner analytics. Students and skills are often modeled together, and yet the proportion of variance attributed to each is typically overlooked. The present work examines how student and skill variance are partitioned across large-scale datasets from three popular learning platforms while considering four popular constructs for learner modeling. Results suggest that variance attribution is largely system and construct specific. Further, findings suggest that many researchers in the EDM community are working in an overly complex portion of the space by modeling next item correctness. These novel observations offer a strong contribution to the field. Limitations and future work are also discussed.

Keywords

Learner modeling, student variance, skill variance, ASSISTments, Cognitive Tutor, Andes, next item correctness, first item correctness, mastery speed, wheel-spinning.

INTRODUCTION

Learner Modeling

Student and skill modeling are primary focuses within the Educational Data Mining (EDM) community that have shifted from tools for the development of learning technologies to features driving adaptive tutors in real time [9]. Learner modeling allows designers of educational technologies to fine-tune learning materials, reform skill compositions, and predict student skill mastery to guide adaptive content provision. Despite persistent attempts to strengthen learner models, the majority of methods for guiding student and skill models have remained largely stagnant. For instance, one of the most popular forms of student modeling, Bayesian Knowledge Tracing (BKT), was conceived over 20 years ago to predict skill mastery using four parameters per skill [8]. By considering the probability of *prior knowledge* alongside probabilities at each skill opportunity for *slip*, *guess*, and *learning*, knowledge tracing calculates the likelihood of skill mastery with a swift and generally accurate quaintness that has sustained the test of time [9]. Still, researchers have shown that individualizing BKT in an attempt to account for student or skill variance can produce more robust models with predictions that are more generalizable to unseen students or skills [17; 23; 24]. This leaves researchers questioning what portion of the variance explained by their models can be attributed to individualized parameters.

Learner models are also versatile in terms of constructs of interest. Arguably the most common construct for the prediction of skill mastery, next item correctness drives models like Knowledge Tracing and (in a sense) Performance Factors Analysis [9]. However, researchers have also modeled student performance by predicting first item correctness, or an estimate of prior knowledge [6], mastery speed, or the number of skill opportunities required to reliably learn a skill [23], and wheel-spinning, or a state of perpetual struggle within skill acquisition [4]. Numerous constructs can be considered when examining variance within learner models.

Partitioning the Variance

It is typical for modeling approaches to be compared against one-another within the same dataset to examine effectiveness in predicting outcomes. However, the present work was inspired by a question posed by Ken Koedinger during a conference presentation meant to explain a model of wheel-spinning within ASSISTments: "What portion of the variance was due to the student and what portion was due to the skill?" [10]. It is true that learner models are often comprised of both student variance and skill variance, yet few researchers have taken a broad enough stance to examine how these sources of variance are partitioned within datasets [15]. Further still, no one (to the best of our knowledge) has yet pushed the boundary to examine trends in student and skill variance across systems, skill domains, modeling constructs, or longitudinally within systems. Just as Brahe and Kepler would have had far more difficulty discovering heliocentric orbits without the printing press that expanded access to astronomical tables [14], educational technologies had to reach a particular scale before student and skill variance could be compared across platforms and constructs.

The present work seeks to partition the variance across systems and predictive constructs. Specifically, the following research questions guide this work:

1. How much variance across systems and constructs can be attributed to differences between students?
2. How much variance across systems and constructs can be attributed to differences between skills?
3. Within systems, how do student and skill variability change over time?

The following sections detail three popular tutoring systems that are commonly used for learner modeling, as well as four constructs that are common resources within the field. Then, remaining sections highlight the methods used in the present work, results observed and their potential implications, limitations of our approach, suggestions for future work, and the overall contribution of this work to the EDM community.

SYSTEMS & CONSTRUCTS

Systems of Interest

The present work highlights three tutoring systems that produce datasets commonly used for student modeling. These systems cover different domains, reach qualitatively different student populations, and were designed using different protocols. The following subsections briefly describe each system and specify the datasets analyzed herein.

ASSISTments

ASSISTments is an online learning platform focused primarily on middle school mathematics and used by more than 50,000 students around the world. The system aims to provide students with *assistance* and teachers with *assessment* within a variety of assignments mapped to the Common Core State Standards and popular mathematics textbooks [11]. As students work through classwork and homework, ASSISTments logs student performance that can be used to construct student models.

The most common type of assignment within ASSISTments is the Skill Builder, a skill driven mastery-based problem set. Students must complete a series of problems randomly selected from a skill pool until meeting a predefined threshold for skill mastery (i.e., the system default requires that students accurately answer three consecutive problems). The ASSISTments dataset considered herein is comprised of all data available from Skill Builders spanning five academic years (2009-2014). As shown in Table 1, this dataset contained performance details on almost 6.5M problems representative of 54,570 students and 645 skills. This dataset was accessed by querying the ASSISTments database and has been made publicly available at [22].

Cognitive Tutor - Algebra 1

Cognitive Tutors are a series of commercialized tutoring systems distributed by Carnegie Learning for students in grades 9-12 [7]. These systems are built around the ACT-R theory of cognition, allowing each system to enlist humanistic problem solving techniques and compare automated solution steps against student solutions to provide appropriate feedback and assistance [2; 18]. Cognitive Tutors are developed as a part of broader curriculum reform, with courses spanning mathematics and language domains [20; 7]. As students work through units and fluency challenges within modules, the tutor logs details on student performance useful for constructing student models.

The Cognitive Tutor dataset used in the present work is composed of data from the Algebra 1 Course and was promoted as the Knowledge Discovery and Data Mining (KDD) Cup dataset in 2010 [12]. This dataset spans two academic years (2005-2007), with over 2.5M problems completed by 1,857 students working within 445 Algebra skills (see Table 1). This dataset was retrieved from the PSLC DataShop [19] where it was split by academic year. Given its breadth, Cognitive Tutor surely houses far larger datasets, but they are not readily available in the PSLC DataShop.

Andes2 Physics

The Andes Physics tutoring system was created as a minimally invasive web-based homework tool for college students at the U.S. Naval Academy [20]. The platform was intended to supplement existing curriculum by replacing pencil and paper homework when solving physics problems. Andes provides feedback following each step within the derivation of a single problem; a far more finite granularity than the other systems considered herein [20]. The rule-based cognitive modeling behind Andes stemmed from the Cascade and Olae projects, with additions to incorporate immediate feedback and various types of tutoring assistance meant to guide students' reasoning while problem solving [20]. As students work through

Table 1. Descriptive statistics across systems years

	Students	Skills	Student-Skill Pairs	Problem Logs	Problems Per Student	Problems Per Skill
AS 2009-2010	2,028	104	25,263	265,821	131	2,556
AS 2010-2011	7,317	130	89,525	931,798	127	7,168
AS 2011-2012	14,971	131	186,352	1,815,054	121	13,855
AS 2012-2013	15,400	139	203,271	1,624,007	105	11,684
AS 2013-2014	14,854	141	219,024	1,824,295	123	12,938
ASSISTments Totals/Ave	54,570	645	723,435	6,460,975	121.4	9,640.2
CT-A 2005-2006	559	106	20,622	879,561	1,573	8,298
CT-A 2006-2007	1,298	339	78,991	1,828,055	1,408	5,392
Cognitive Tutor Totals/Ave	1,857	445	99,613	2,707,616	1,490.5	6,845.0
Andes2 – Fall 2005	76	150	7,589	118,822	1,563	792
Andes2 – Fall 2006	66	157	7,142	119,196	1,806	759
Andes2 – Fall 2007	79	143	4,851	73,744	933	516
Andes2 – Fall 2008	64	99	3,585	36,532	571	369
Andes2 – Fall 2009	63	88	2,274	23,840	378	271
Andes2 – Fall Totals/Ave	348	637	25,441	372,134	1,050.2	541.4
Andes2 – Spring 2005	72	128	6,117	59,834	831	467
Andes2 – Spring 2006	71	144	7,162	82,923	1,168	576
Andes2 – Spring 2007	93	120	6,362	58,212	626	485
Andes2 – Spring 2008	42	34	903	22,588	538	664
Andes2 – Spring 2009	71	108	4038	38,001	535	352
Andes2 Totals/Ave	349	534	24,582	261,558	739.6	508.8

Note. System totals do not represent unique students or skills, as overlap is possible across years. Assumptions of independence do not apply. Averages are presented for total Problems Per Student and Problems Per Skill.

homework problems within Andes, performance details are primarily collected to assist professors in grading, but also prove useful for student modeling.

Specifically, the dataset used in the present work was collected from Andes2, the second iteration of the platform, and spans five academic years (2005-2009). This data was retrieved from the PSLC DataShop [19], where it was split by academic semester. The full dataset included over half a million problems solved by 650 students spanning 1,044 skills, as shown in Table 1. For the analyses presented herein, the academic semester split was retained across years as variance attributed to students differed greatly across semesters, suggesting wualitative differences between semesters. Although the Andes dataset had far fewer students in comparison to the ASSISTments and Cognitive Tutor datasets, it is included because the sample sizes were large enough to support the modeling approach used without over fitting the data. Few parameters were necessary to partition student and skill variance, and cross validation was employed for reliability (see Section 3.2).

Constructs of Interest

While considering the distribution of student and skill variance across datasets from three qualitatively different platforms, it was also of interest to define these distributions across numerous constructs that are commonly used in learner models. The following subsections highlight the constructs examined herein.

First Item Correctness

Models focused on first item correctness seek to isolate what students know when they first sit down to complete an assignment, or essentially, the prior knowledge they bring to a skill. Recent research has examined the prediction of first item correctness, or initial knowledge, within BKT to enhance the individualization of learner modeling [17; 6]. Models have also been constructed using first item correctness to examine the influence of prerequisite performance, or initial skill knowledge, on wheel-spinning [21]. Determining the knowledge a student brings to the table can be critical for predicting whether he or she will succeed in mastering a skill.

Within the present analyses, first item correctness is traditionally defined as the prediction of whether or not a student will accurately solve the first item within a given skill.

Next Item Correctness

Models focused on next item correctness seek to predict what students will come to know as they progress through an assignment, or essentially, whether they ultimately learn a skill. Next item correctness is one of the most popular constructs in the field, as determining whether a student will answer the next item accurately is key in predicting precisely when a student will master a given skill. Knowledge Tracing relies largely on predictions of next item correctness [8], and other common learner models like Performance Factors Analysis consider the accuracy of sequential skill items in a similar nature [9]. Leaders within the field have long argued that predicting skill mastery or overall performance is impossible without tracking a student's performance at item-level [2].

Within the present analyses, next item correctness is traditionally defined as a prediction of whether or not a student will accurately solve the next item opportunity within a given skill, considering their performance on previous items.

Mastery Speed

Models focused on mastery speed seek to gain insights from how quickly students learn or master a skill by considering the number of skill opportunities or problems that a student receives [23]. Some systems define skill mastery using predictive models while others define mastery through consecutive, *n* right-in-a-row, problems solved. In some senses, being able to predict when a student will master a skill, or how much additional practice would be necessary to reach mastery, can be as helpful as incremental predictions of next item correctness.

Within the present analyses, mastery speed is defined across all platforms (regardless of their internal definitions of mastery) as accurate responses to three consecutive questions. Although this approach is simple, it is easy to replicate and produces results similar to skill mastery as defined by Knowledge Tracing ($P(T) = 0.95$). Prior work has shown that within the context of ASSISTments Skill Builder data, similar predictions for mastery can be obtained from KT to those observed using the system's default approach requiring correct answers on three consecutive skill items. Comparing predictions of next item correctness for a transfer item of greater difficulty, when guess rate was low (<0.1) and slip rate was low (<0.3), three consecutive items reached the 95% threshold of KT [13]. Knowledge Tracing also presents an identifiability issue that results in models with equivalent statistical fit but mixed predictions of student knowledge [3]. This issue can be avoided by defining mastery by a series of accurate responses to consecutive skill items.

Wheel-Spinning

Models focused on wheel-spinning seek to determine whether struggling students will eventually master a skill, even when they may fail to initially master or master in a timely manner [4]. The mastery-based learning approach to skill acquisition that is taken by most Intelligent Tutoring Systems and online learning platforms may be too strict for students that are not capable of reaching proficiency, especially considering potential variation in content difficulty. Recent models have predicted whether or not students will wheel-spin by considering the student's performance on prerequisite skills, or essentially, a measure of their prior knowledge [21].

Within the present analyses, wheel-spinning is defined as it was presented in [4]: failure to attain skill mastery following ten item opportunities within a given skill.

METHODS

Data Preprocessing

The datasets were retrieved and the constructs were isolated, as described in previous sections. Datasets from all three systems included information that would allow for the modeling of first item correctness, next item correctness, mastery speed, and wheel-spinning. Each dataset required preprocessing to format universal constructs for modeling. First, the data was filtered to include only skills with performance information from at least ten students. The data was then filtered to include only students that had worked on at least three skills. Additionally, data was filtered such that only student/skill pairs with at least three item opportunities were included. The intuition behind this filtration process was to

Table 2. The process for calculating student and skill covariates

Student ID	*Skill ID*	*Opportunity Order*	*Correct*	*Student Covariate*	*Skill Covariate*
1	A	1	0	0.75	0.25
1	A	2	1	0.75	0.25
1	B	1	1	0.75	0.75
1	B	2	1	0.75	0.75
2	A	1	0	0.25	0.25
2	A	2	0	0.25	0.25
2	B	1	0	0.25	0.75
2	B	2	1	0.25	0.75

Note. A covariate is calculated per student given the overall average accuracy across skills and item opportunities (i.e., Student 1 scores 0 + 1 + 1 + 1 = 0.75). A second covariate is calculated per skill given average accuracy across all item opportunities by all students (i.e., Skill A has accuracy of 0 + 1 + 0 + 0 = 0.25).

arrive at a subset of data for each system that was robust enough to model the proportions of skill and student variance. Rather than applying a complex algorithm, this simple iterative filtering process ensured enough data for each skill and for each student. Further, within the ASSISTments dataset, tutoring problems (i.e., Scaffolds) were excluded as to only retain primary skill item opportunities. This issue was not apparent in the Cognitive Tutor or Andes2 datasets. All resulting datasets and the filtration code are available at [22] for further reference.

Following filtration, it was necessary to develop weighted covariates for student and skill to help partition the variance attributed to each predictor. These covariates were calculated using identifiers for the student and the skill, the number of items and their opportunity order, and the item's accuracy, as shown in Table 2. To process the student covariate, accuracy was averaged across all problems that the student answered, regardless of skill. For example, in Table 2, Student 1 answered four items spanning two skills, with an average overall accuracy of 0.75. Student 2 also answered four items spanning two skills, but her average accuracy was 0.25. The student covariate provides insight into overall student performance, regardless of skill, or essentially a student-level characteristic inherent to ability. To process the skill covariate, a similar approach was taken using skill as the unit of analysis. For example, in Table 2, both students solved two items pertaining to Skill A. Looking across students, the average accuracy on Skill A items was 0.25. Both students also solved two items pertaining to Skill B, which carried an average accuracy of 0.75. The skill covariate provides insight into overall skill difficulty, as experienced by all students.

Modeling Approach

The modeling approach presented herein is simple in nature, with a focus on how student and skill variance are partitioned across systems and constructs. Linear or Logistic Regression models were constructed (for continuous and binary constructs, respectively) to predict the constructs of interest while examining R^2 as a core metric for variance explained. For example, as shown in the logit equation (2) and resulting probability equation (3) below, a Logistic Regression model was built to predict the probability of next item correctness (Y), with student and skill covariates (X) as independent variables.

Linear Regression:

$$\hat{Y} = \alpha + \beta X \tag{1}$$

Logistic Regression:

$$logit(Y) = \ln\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta X \tag{2}$$

$$Prob_Y = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}} \tag{3}$$

Table 3. Correlations of constructs across systems

	First Item	*Next Item*	*Mastery Speed*	*Wheel-Spinning*
ASSISTments				
First Item	1.0			
Next Item	0.28	1.0		
Mastery Speed	-0.43	-0.33	1.0	
Wheel-Spinning	-0.29	-0.32	0.78	1.0
Cognitive Tutor				
First Item	1.0			
Next Item	0.17	1.0		
Mastery Speed	-0.46	-0.22	1.0	
Wheel-Spinning	-0.28	-0.22	0.79	1.0
Andes2 - Fall				
First Item	1.0			
Next Item	0.27	1.0		
Mastery Speed	-0.45	-0.21	1.0	
Wheel-Spinning	-0.32	-0.31	0.73	1.0
Andes2 - Spring				
First Item	1.0			
Next Item	0.27	1.0		
Mastery Speed	-0.46	-0.24	1.0	
Wheel-Spinning	-0.31	-0.30	0.71	1.0

The model also included inherent error, α. This model was run once while considering only the student covariate, again while considering only the skill covariate, and a final time considering the compound effect of student + skill. Through this approach, resulting R^2 values can be interpreted as variance explained by the variable(s) included in each model. Within the iterations of Linear and Logistic Regression models, and regardless of the covariate or construct being modeled, ten-fold cross validation based on student-skill pairs was used to promote robust outcomes. These models were not designed to examine the error inherent to resulting predictions, but simply to gauge the overall variance explained by variables within the model.

RESULTS

Correlations of Constructs

Prior to running the necessary Linear and Logistic Regressions across systems and constructs, it was first of interest to briefly examine the correlations between constructs within systems. Correlations are presented in Table 3, showing relatively stable trends in the relationships between constructs across platforms. The values presented are the average of Pearson's r correlations collected from each academic year within each system (e.g., 5 years for ASSISTments, 2 years for Cognitive Tutor, and 5 years split by semester for Andes2). In order to collect these correlations, first item correctness, mastery speed, and wheel-spinning (logged at the level of student/skill pairs) were replicated across each item as necessary, such that the number of items represented, *n*, was stable across constructs. While most correlations were mild (all were significant prior to averaging across years), mastery speed and wheel-spinning maintained a strong positive correlation across platforms, suggesting that the nature of this relationship is linked to how these constructs are defined. Given this strong correlation, variance explained should look similar within these constructs across systems.

Variance Explained

After examining correlations amongst constructs within systems, the Linear and Logistic Regressions were modeled with cross-validation employed. Immediate results were intriguing, as to our knowledge, partitioning the variance within learner models across systems and constructs is a novel task.

Table 4. Variance explained (R^2) by Student, Skill, and Student + Skill across systems and constructs

	Student					Skill					Student + Skill				
	FI	*NI*	*MS*	*WS*	*Ave*	*FI*	*NI*	*MS*	*WS*	*Ave*	*FI*	*NI*	*MS*	*WS*	*Ave*
ASSISTments	0.17	0.10	0.19	0.18	0.16	0.08	0.04	0.09	0.05	0.07	0.25	0.14	0.27	0.25	0.23
Cognitive Tutor	0.07	0.04	0.10	0.09	0.08	0.19	0.11	0.17	0.16	0.16	0.25	0.15	0.27	0.26	0.23
Andes2 - Fall	0.11	0.09	0.10*	0.17*	0.12 (0.10)	0.20	0.11	0.30*	0.22*	0.21 (0.16)	0.31	0.20	0.37*	0.41*	0.32 (0.26)
Andes2 - Spring	0.17	0.14	0.16*	0.24*	0.18 (0.16)	0.13	0.06	0.26*	0.15*	0.15 (0.10)	0.30	0.19	0.41*	0.41*	0.33 (0.25)
Average	0.13	0.09	0.14 (0.15)	0.17 (0.14)		0.15	0.08	0.21 (0.13)	0.15 (0.11)		0.28	0.17	0.33 (0.27)	0.33 (0.26)	

Note. *FI* = First Item Correctness, *NI* = Next Item Correctness, *MS* = Mastery Speed, *WS* = Wheel-Spinning. Averages are provided for each system across constructs, and for each construct across systems. *As Andes2 was found to be a qualitatively different system in which measures of mastery speed and wheel-spinning were less reliable, averages are corrected to include only ASSISTments and Cognitive Tutor and presented in parentheses.

Student Variance

The proportion of variance in each model that could be attributed to students differed considerably across systems and constructs. Results are depicted in the Student section of Table 4. Further investigation shows that when modeling first item correctness, student characteristics explained anywhere from 7% to 17% of variance in models across systems (M = 0.13, SD = 0.05). Considering next item correctness, student characteristics explained between 4% and 14% of variance in models across systems (M = 0.09, SD = 0.04). When examining mastery speed, between 10% and 19% of variance in models across systems was attributed to students (M = 0.14, SD = 0.05). Finally, wheel-spinning was more reliant on student characteristics yet showed greater variability across systems, with between 9% and 24% of variance attributed to students (M = 0.17, SD = 0.06).

Skill Variance

The proportion of variance in each model that could be attributed to skills also differed considerably across systems and constructs. Results are depicted in the Skill section of Table 4. When modeling first item correctness, skill explained anywhere from 8% to 20% of variance in models across systems (M = 0.15, SD = 0.06). When examining next item correctness, skill showed less variability across systems, explaining between 4% and 11% of the variance in models (M = 0.08, SD = 0.04). Alternatively, skill was exceptionally variable when examining mastery speed, explaining between 9% and 30% of variance in models across systems (M = 0.21, SD = 0.14). The variance explained by skill was also highly variable in wheel-spinning, with between 5% and 22% of variance attributed to skill (M = 0.15, SD = 0.07).

Student + Skill Variance

Briefly examining the compound effects of student and skill, variance explained was not always strictly summative when these covariates were modeled together. Referring to Table 4, within ASSISTments, student explained 10% of the variance when used to model next item correctness alone, while skill explained 4% of the variance when used to model the same construct alone. When taken together, student and skill did come together in perfect summation to explained 14% of the variance in the model. However, when modeling the construct of wheel-spinning within Andes2 - Spring, student alone explained 24% of the variance and skill alone explained 15% of the variance, and yet together they explain 41% of the variance in the model (gaining strength by 2%, perhaps through a moderating latent construct).

Findings Across Systems

Overall, student and skill were equally informative in terms of average variance explained. However, trends in the attribution of variance were impressively different across constructs and systems. Considering averages across constructs but within systems is perhaps more crucial to the field. On average within ASSISTments, a greater proportion of variance was attributed to student (M = 0.16, SD = 0.04), while skill was about half as powerful in terms of variance explained (M = 0.07, SD = 0.02). The Cognitive Tutor data actually showed the reverse. On average, a greater proportion of variance was attributed to skill (M = 0.16, SD = 0.03), while student was about half as powerful in terms of variance explained (M = 0.08, SD = 0.03). Oddly, this flip also occurred within the Andes2 system, with skill claiming a greater portion of the variance explained on average in the Fall (M = 0.21, SD = 0.08), and student explaining a greater proportion of the variance explained on average in the Spring (M= 0.18, SD = 0.04).

Findings Across Constructs

Across constructs, next item correctness is perhaps the most popular for learner models and yet appeared to be the most difficult to predict. While student and skill were fairly well balanced in importance, only 9% of the variance (on average) in next item correctness was explained by student, and only an additional 8% (on average) was explained by skill. Other constructs carried more accurate predictions. Models of first item correctness attributed 13% of variance to student and 15% to skill, while models of mastery speed and wheel-spinning also carried high proportions of variance explained by both student and skill, as shown in Table 4. Skill held more variance in these constructs, suggesting they may also provide avenues for driving instructional interventions to improve modeling outcomes through curriculum design.

A Deeper Look into Andes2

Additional tests were run to examine why Andes2 (Fall and Spring) resulted in impressively different R^2 values for the constructs of mastery speed and wheel-spinning, as shown in Table 4. It was thought that perhaps the issue was linked to the definition of mastery used here (i.e., accurate responses on three consecutive skill items). As such, the percentage of students reaching mastery within 15 attempts was graphed in Figure 1, a classic wheel-spinning curve. Results suggested that students within Andes2 were only mastering at chance levels by 10 attempts (i.e., 52% mastery in the Fall, 46% mastery in the Spring). In comparison, 80% of students within ASSISTments and 75% of students within Cognitive Tutor were mastering by the 10th attempt.

In an attempt to reduce the mastery skew effecting Andes2, the threshold for mastery was lowered to require accurate responses on only two consecutive skill items. As shown in Figure 2, although gains in mastery were observed across systems, the

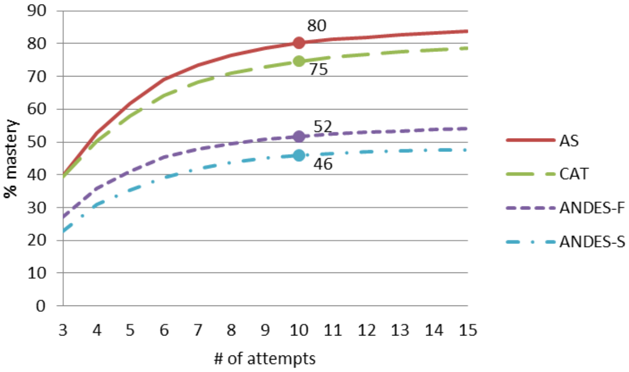


Figure 1. Trends in the percentage of students that master when mastery is defined as 3 consecutive correct responses

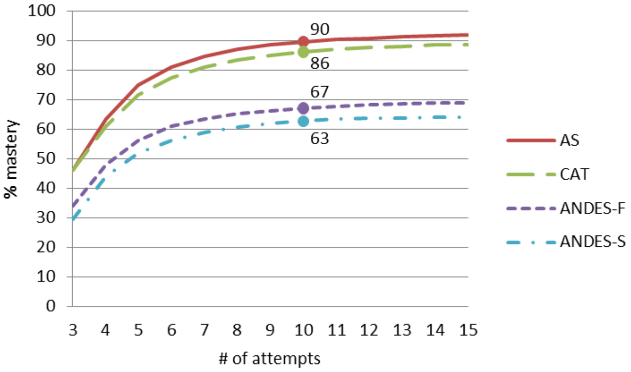


Figure 2. Trends in the percentage of students that master when mastery is defined as 2 consecutive correct responses

percentage of mastery for students working within Andes2 (Fall or Spring) was still below 70%. As it was not logical to reduce the requirement for mastery to accuracy on a single item, Andes2 was simply labeled as a qualitatively different system and it was redacted from amended analyses for mastery speed and wheel-spinning. As such, both original and adjusted averages for these constructs and for the Andes2 system(s) are presented in Table 4.

Variability in Variance Over Time

The final research question guiding the present work was to examine the variability of student and skill variance within systems over time. The longitudinal trends shown in Table 5 are novel in that (to the best of our knowledge) variance for these systems has never been examined longitudinally at such a fine granularity. Findings suggest that the variance explained by models tailored to correctness metrics varies widely across systems, proportional to the distance from predictions of chance accuracy (50% correctness).

On average, most constructs actually grow more difficult to model in both ASSISTments and Cognitive Tutor with each passing year. For instance, when modeling next item correctness in ASSISTments, researchers were able to explain 16% of the variance using both student and skill variables in the 2009-2010 academic year, but this value dropped to only 10% explained in 2013-2014. Similar trends exist for models predicting first item correctness (dropping 11%), for models predicting mastery speed (dropping 5%), and for models predicting wheel-spinning (dropping 12%). Specifically, drops in variance explained could be largely attributed to student across constructs, as shown in Table 5. Although trends were flipped for Cognitive Tutor, in that more variance could be attributed to skill than to student, the longitudinal decline remains. For both systems, the data suggests that something within system modernization has made learner modeling more difficult.

Despite discovering potential issues with using the Andes2 datasets for traditional learner modeling, longitudinal data for both semesters is presented in Table 5 for reference. The system showed no clear longitudinal trends in the variability of student or skill variance across constructs.

Table 5. Longitudinal trends of variance explained (R^2) by Student, Skill, and Student + Skill across systems and constructs

	Student				Skill				Student + Skill			
	FI	*NI*	*MS*	*WS*	*FI*	*NI*	*MS*	*WS*	*FI*	*NI*	*MS*	*WS*
AS 2009-2010	0.22	0.13	0.22	0.25	0.11	0.05	0.10	0.06	0.32	0.16	0.30	0.32
AS 2010-2011	0.17	0.11	0.19	0.18	0.09	0.05	0.10	0.06	0.25	0.15	0.28	0.28
AS 2011-2012	0.17	0.13	0.19	0.19	0.07	0.04	0.08	0.05	0.23	0.16	0.27	0.25
AS 2012-2013	0.14	0.08	0.17	0.15	0.08	0.04	0.09	0.05	0.21	0.11	0.25	0.21
AS 2013-2014	0.14	0.07	0.17	0.13	0.08	0.04	0.09	0.05	0.21	0.10	0.25	0.20
CAT 2005-2006	0.08	0.04	0.13	0.10	0.18	0.12	0.14	0.19	0.26	0.17	0.27	0.30
CAT 2006-2007	0.05	0.03	0.08	0.07	0.19	0.09	0.19	0.13	0.24	0.14	0.26	0.21
ANDES Fall 2005	0.06	0.03	0.06	0.07	0.19	0.06	0.27	0.13	0.25	0.09	0.32	0.22
ANDES Fall 2006	0.11	0.08	0.10	0.17	0.17	0.07	0.24	0.13	0.28	0.15	0.32	0.34
ANDES Fall 2007	0.15	0.14	0.13	0.22	0.19	0.13	0.26	0.26	0.33	0.27	0.34	0.47
ANDES Fall 2008	0.09	0.11	0.10	0.20	0.26	0.15	0.44	0.36	0.34	0.25	0.50	0.54
ANDES Fall 2009	0.16	0.10	0.13	0.21	0.21	0.13	0.27	0.24	0.36	0.23	0.38	0.46
ANDES Spring 2005	0.08	0.05	0.08	0.09	0.14	0.07	0.24	0.11	0.22	0.12	0.30	0.21
ANDES Spring 2006	0.16	0.12	0.14	0.20	0.14	0.05	0.26	0.13	0.29	0.17	0.37	0.34
ANDES Spring 2007	0.26	0.23	0.24	0.39	0.11	0.06	0.26	0.19	0.36	0.27	0.41	0.53
ANDES Spring 2008	0.16	0.11	0.23	0.32	0.20	0.08	0.48	0.28	0.33	0.17	0.56	0.55
ANDES Spring 2009	0.18	0.17	0.17	0.29	0.12	0.06	0.27	0.18	0.30	0.21	0.38	0.44

Note. *FI* = First Item Correctness, *NI* = Next Item Correctness, *MS* = Mastery Speed, *WS* = Wheel-Spinning. Values for mastery speed and wheel-spinning within Andes2 are shaded to remind readers that this system was found to be qualitatively different than ASSISTments and Cognitive Tutor, and as such, these values may carry less reliability.

DISCUSSION

Much like Brahe and Kepler discovering errors in planetary orbits via side-by-side astronomical charts [14], the present results revealed potential errors in the focus of the Educational Data Mining community via side-by-side learner models. The trends observed for next item correctness (e.g., a construct that explains minimal variance in models, that has decreased in power longitudinally within systems, and in which attribution to student or skill is heavily system dependent) suggest that researchers in the field are putting the majority of their focus in an overly complex portion of the space that may not reveal as much about student achievement and skill mastery as other constructs.

The present work sought to determine the proportion of variance attributed to students and skills when looking across systems and constructs. Findings suggested that across systems and constructs, between 4% and 24% of variance could be attributed to student (without corrections for Andes2, M = 0.13, SD = 0.03), and between 4% and 30% of variance could be attributed to skill (without corrections, M = 0.15, SD = 0.05). When not considering Andes2 due to its qualitative differences as a system, student variance in ASSISTments and Cognitive Tutor ranged from 4% to 19% (M = 0.12, SD = 0.04) and skill variance also ranged from 4% to 19% (M = 0.12, SD = 0.03). Further, when looking over time, trends were observed in the variability of student and skill variance for constructs and systems. Findings suggested that learner models are highly sensitive to the system, dataset, and construct being modeled, with different systems and constructs resulting in different trends.

The systems considered herein were chosen for their popularity within the EDM community. ASSISTments and Cognitive Tutor, systems producing some of the most mined datasets, actually appear to behave very differently. Across constructs, student was more valuable when modeling ASSISTments data, while skill was more valuable when modeling Cognitive Tutor data. These extremes in the attribution of variance suggest that learner models treating these systems as equivalents may not be appropriate, as results will look similar on average but the observed effects may be attributed to very different causes.

The set of constructs examined herein was also chosen to address useful issues within the field of EDM. The goal of this work was to promote the importance of modeling a broad range of constructs, extending the field's vocabulary beyond next item correctness. Further, little focus has fallen on comparing learner models across systems [5]. Results from the present work suggest that this approach is critical for understanding the implications of learner modeling from a broader perspective.

When questioning why differences in variance attribution were observed, a few potential causes can be hypothesized. It is possible that skills were poor predictors in ASSISTments due to greater variance in the population of student users (i.e., perhaps students bring a greater range of preparation, knowledge, and behavior). The apparent lack of predictive ability in comparison to Cognitive Tutor may also be due to the fact that skills and specific knowledge components have much more variability, spanning grades and mathematics domains while Cognitive Tutor is limited to Algebra 1 components. It is also possible that the issues inherent to skill-based learner modeling in ASSISTments can be linked to the tagging of knowledge components or to errors in the skill structure itself, although this structure is continuously revised for accuracy through research into prerequisite skills [1]. Still, it is also possible to turn the tables and ask why student was a poor predictor within Cognitive Tutor. It is possible that the student population using Cognitive Tutor to practice Algebra, especially within the restricted data made available through the PSLC DataShop [19] was more homogenous in preparation, ability, and behavior. Cognitive Tutor is presented to students as part of an entire curriculum [18; 20] and is limited to a single mathematics domain.

Not surprisingly, middle school mathematics is qualitatively different than introductory college physics, as confirmed by the difficulty in modeling constructs within Andes2. These systems carry similar knowledge components or skills, but are driven by very different instructional objectives. Middle school mathematics is repetitious, requiring students to practice skills multiple times and offering a clear depiction of learning (i.e., through learning curves). In comparison, physics is far more granular, with knowledge components that correspond to smaller steps within complex problem solving. Students solving physics problems experience less repetition in specific skill practice, making it more difficult to model when learning has occurred by considering student or skill.

A touch of clairvoyance into the future of EDM would suggest that the future of learner modeling will likely look more like the trends observed within ASSISTments, as the platform more closely resembles material from a Massive Open Online Course. These platforms have broader and more loosely defined skills, where students are not constrained to a fixed curriculum and may access lessons at will, as shown by work that has already investigated the application of knowledge tracing to MOOCs [16]. As the field progresses, learner models should be developed cautiously, explained within their context, and presented within a broader perspective of implications.

LIMITATIONS & FUTURE WORK

The present work is not without limitation. First and foremost, while three systems and four constructs were considered to examine how student and skill variance are partitioned within learner models, there are certainly many other systems and constructs that have not been considered. Future work should be considered to extend the findings presented here across additional systems, perhaps to include MOOCs and datasets that have not been primed for presentation in the PSLC DataShop [19]. There are also a number of constructs that are of interest to the greater EDM community that are not considered in the present work (e.g., student affect, or other student, class, and school level characteristics like gender, class size, and urbanicity). Future work should investigate variance attributions across more complex constructs of this nature.

Another limiting factor of this work is the validity of the datasets considered herein. While the authors had control over the query and preprocessing necessary for the ASSISTments dataset, less is known about the steps that established datasets retrieved from the PSLC DataShop [19]. Specifically, the Cognitive Tutor Algebra dataset was promoted for the specific purpose of the KDD Cup [12], a data mining challenge focused on specific predictors and outcomes. Thus, it is possible that the dataset was cleaned in a manner to best suit the needs of data miners with particular goals, which may have led to some of the trends in student and skill variance observed between systems.

Additionally, model-fitting procedures have the capacity to influence the results observed, and while measures were taken to produce valid and reliable results, it is possible that our approach had room for error. Other approaches to partitioning the variance within learner models may result in slightly different outcomes.

CONTRIBUTIONS

The present work offers a novel contribution to the Educational Data Mining community in the form of a cross platform comparison of student and skill variance attributions within learner models predicting first item correctness, next item correctness, mastery speed, and wheel-spinning. This work revealed that much of the field has been focusing on a complex and potentially impractical area in learner modeling – next item correctness. Student characteristics are less helpful in predicting this construct, but may be more practical in predicting other, less sensitive constructs. Further, it revealed that variance in some of the most frequently mined datasets can be system and construct specific, and as such, that broad claims about the generalization of particular learner models should be made with caution.

ACKNOWLEDGMENTS

We acknowledge funding from multiple NSF grants (1440753, 1252297, 1109483, 1316736, 1535428, 1031398), the U.S. Dept. of Ed. (R305A120125, R305C100024, P200A120238), and ONR. The opinions, findings and conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of our funders.

REFERENCES

[1] Adjei, S.A. & Heffernan, N.T. 2015. Improving learning maps using an adaptive testing system: PLACEments. In Conati, et al. (eds), Proc of the 17th Int Conf on AIED. 517-520.

[2] Anderson, J.R., Corbett, A.T., Koedinger, K.R., & Pelletier, R. 1995. Cognitive tutor: Lesson learned. The journal of the learning sciences. 4 (2): 167–207.

[3] Beck, J.E. & Chang, K. 2007. Identifiability: A Fundamental Problem of Student Modeling. In Conati, et al. (eds.) Proc of the 11th Int Conf on User Modeling. 4511: 137-146.

[4] Beck, J.E. & Gong, Y. 2013. Wheel-Spinning: Students Who Fail to Master a Skill. In Lane, et al. (eds.) Proc of the 16th Int Conf on AIED. 431-440.

[5] Beck, J.E. & Xiong, X. 2013. Limits to Accuracy: How Well Can We Do at Student Modeling? In D'Mello, et al. (eds.) Proc of the 6th Int Conf on EDM. 4-11.

[6] Botelho, A., Wan, A., & Heffernan, N. 2015. The Prediction of Student First Response Using Prerequisite Skills. In Kiczales, et al. (eds.) Proc of the 2nd ACM Conf on L@S. 39-45.

[7] Carnegie Learning. 2016. Cognitive Tutor Software. Carnegie Learning, Inc. Retrieved from https://www.carnegielearning.com/learning-solutions/software/cognitive-tutor/

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

[9] Desmarais, M.C. & Baker, R.S.J.d. 2011. A Review of Recent Advances in Learner and Skill Modeling in Intelligent Learning Environments. User Modeling and User-Adapted Interaction. 22 (1-2): 9-38.

[10] Gong, Y. & Beck, J.E. 2015. Towards Detecting Wheel-Spinning: Future Failure in Mastery Learning. In Kiczales, Russell, & Woolf (eds.) Proc of the 2nd ACM Conf on L@S. 67-74.

[11] Heffernan, N. & Heffernan, C. 2014. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. International Journal of AIED. 24 (4): 470-497.

[12] KDD Cup. 2010. Rules of the KDD Cup 2010: Educational Data Mining Challenge. PSLC DataShop. Retreived from https://pslcdatashop.web.cmu.edu/KDDCup/rules.jsp

[13] Kelly, K., Wang, Y., Thompson, T., & Heffernan, N. 2015. Defining Mastery: Knowledge Tracing Versus N-Consecutive Correct Responses. In Santos, et al., (eds.) Proceedings of the 8th Int Conf on EDM.

[14] Kusukawa, S. 1999. Astronomical Tables. University of Cambridge, Department of History and Philosophy of Science. Retrieved: www.hps.cam.ac.uk/starry/tables.html

[15] Mood, A.M. 1971. Partitioning Variance in Multiple Regression Analyses as a Tool for Developing Learning Models. American Educational Research Journal. 8 (2): 191-202.

[16] Pardos, Z.A., Bergner, Y., Seaton, D., Pritchard, D. 2013. Adapting Bayesian Knowledge Tracing to a Massive Open Online Course in edX. In D'Mello, et al. (eds.) Proc of the 6th Int Conf on EDM. 137-144.

[17] Pardos, Z.A. & Heffernan, N.T. 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Bra, et al. (eds.) Proc of the 18th Int Conf on UMAP. 255-266.

[18] Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. 2007. Cognitive Tutor: Applied research in mathematics education. Psychonomic Bulletin & Review. 14 (2): 249-255.

[19] Stamper, J.C., Koedinger, K.R., Baker, R.S.J.d., Skogsholm, A., Leber, B., Demi, S., Yu, S., & Spencer, D. 2011. DataShop: A Data Repository and Analysis Service for the Learning Science Community. In Biswas et al. (eds.) Proc of the 15th Int Conf on AIED.

[20] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. 2005. The Andes Physics Tutoring System: Lessons Learned. International Journal of AIED. 15 (3): 147-204.

[21] Author 1. 2016. Data and code for "Students vs. Skills: Explaining the Variance in Learner Modeling" available at http://tiny.cc/StudentSkillEDM2016

[22] Xiong, X., Li, S., & Beck, J.E. 2013. Will You Get It Right Next Week: Predict Delayed Performance in Enhanced ITS Mastery Cycle. In Proc of the 26th Int FLAIRS Conference.

[23] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. 2013. Individualized Bayesian Knowledge Tracing Models. In Lane, et al. (eds.) Proc of the 16th Int Conf on AIED. 171-180.

[24] Wan, H. & Beck, J.B. 2015. Considering the influence of prerequisite performance on wheel spinning. In Santos, et al., (eds.) Proc of the 8th Int Conf on EDM.

The Opportunity Count Model: A Flexible Approach to Modeling Student Performance

Yan Wang, Korinn Ostrow, Seth Adjei, Neil Heffernan
Worcester Polytechnic Institute
Worcester, MA 01609
{ywang14, ksostrow, saadjei, nth} @wpi.edu

This section has been published at:

Wang, Y., Ostrow, K., Adjei, S. & Heffernan, N. (2016). The Opportunity Count Model: A Flexible Approach to Modeling Student Performance. *In the Proceedings of 3rd Annual ACM Conference on Learning at Scale.* ACM.

ABSTRACT

Rich features can be exploited to better model student performance when predicting next problem correctness (NPC) within intelligent tutoring systems. Yet these features may differ significantly in availability and importance when considering opportunity count (OC), or the number of problems experienced within a skill or knowledge component. Inspired by such intuition, the present study examines the Opportunity Count Model (OCM), a unique approach to student modeling in which separate models are built for differing OCs rather than creating a blanket model to encompass all OCs. Random Forest (RF) is used to establish iterations of the OCM by considering rich features within logged tutor data. Model strength is then tested against standard Knowledge Tracing. Results suggest that prediction of next problem correctness is improved through the OCM approach for lower OCs, and applying different modeling techniques at different phase of students' practice would be plausible. Also, feature variation among OCs justifies our proposal to build OCM.

Author Keywords

Random Forest; Opportunity Count; Student Modeling; Next Problem Correctness; Intelligent Tutoring System; Knowledge Tracing

ACM Classification Keywords

I.6.5. Simulation and modeling: Model Development; J.1. Administrative data processing; K.3.0. Computers and education: General.

INTRODUCTION

Since its creation, Knowledge Tracing (KT) [3] has played a critical role in the intelligent tutoring system (ITS) community for its use in modeling student knowledge and performance. Although it has shown high prediction accuracy, KT overlooks the rich features that are common to many ITSs, such as response time and hint usage. A variety of rich features are easily obtained by data mining the log files of these systems, and as research has shown, these features can be exploited to improve student modeling [4,5,8,9,11]. Specifically, González-Brenes et.al. presented a general method for making use of rich features via dynamic Bayesian Networks, thereby compensating for the limitations of KT [5]. In contrast, Wang and Heffernan [11] established a maximum likelihood tabling method termed the "Assistance" Model, which considered a student's hint and attempt usage to better predict performance. Although this model did not outperform KT, ensembling the two models proved beneficial. Research by Duong, Zhu, Wang and Heffernan [4] considered action sequences in the prediction of next problem correctness, enhancing prediction accuracy over KT. Other feature based methods that have proven successful include Performance Factors Analysis [9], which applies logistic regression to a compounding record of correct and incorrect problem responses in order to predict next problem correctness, and a Random Forest approach by Pardos & Heffernan [8] that examined the significance of numerous rich features in modeling student performance.

Despite the fact that rich features have been shown to enhance student modeling, little focus has been given to the critical significance of opportunity count (OC), or the compounded sequence of skill or knowledge component opportunities within a student's learning experience. It seems intuitive that the availability and importance of rich features within logged data can vary based on opportunity count: different features hold significance for a student on her third opportunity than those important for a student on her seventh opportunity. It may be possible to reduce the noise inherent to low OCs (i.e., the initial parameters used in KT are more critical to prediction when OCs are low) by establishing flexible models that consider opportunity count alongside rich features.

The present study investigates the significance of opportunity count when establishing student models using rich features. We propose building separate models for differing OCs by using a Random Forest approach to determine fluctuations in the importance of rich features

OC	Student	Skill	Correct	Attempts	FRT (ms)	H Used	H Total	FA
1	34	102	1	1	30230	0	2	0
2	34	102	1	1	23432	0	3	0
3	34	102	0	2	32363	1	2	1
4	34	102	1	1	25465	0	2	0
1	56	102	0	1	15201	0	1	2

Table 1. Sample Data

across a dataset stratified by OC. Random Forest, introduced by Leo Breiman, is a proven method for making predictions based on a variety of features [2]. The method trains regression trees based on decision splits made from a random subset of data features. The resulting output offers a prediction model based on an ensemble of regression trees. This method also succinctly defines the degree of feature importance within a model, as measured by out-of-bag error [10].

The Opportunity Count Model proposed here examines the potential flexibility of student modeling when considering opportunity count and rich features inherent to intelligent tutoring systems. We seek to answer the following research questions:

1. Can the accuracy of models predicting next problem correctness be enhanced by establishing separate models for differing opportunity counts when considering rich features?

2. Is there variation of feature importance among different OCs?

DATASET

The current study examines flexible OC modeling using a dataset comprised of student data logged between September 2012 and August 2013 within ASSISTments, an intelligent tutoring system with a primary focus on mathematics content [6]. The log files used in the present study originated solely from Skill Builders, a type of problem set unique to ASSISTments in which students must correctly solve three (by default setting) consecutive problems on a skill in order to complete or 'master' their assignment. Problems are randomly assigned from a large pool of skill content to reduce the likelihood of cheating. For each problem, students are provided correctness feedback along with hints or scaffolding problems that act as tutoring strategies to deter students from getting stuck within the assignment. Hints are provided upon the student's request, while scaffolding problems are presented automatically when an incorrect answer is entered, or upon the student's request. A series of hints offers assistance that grows increasingly specific, until ultimately providing students with the correct answer (i.e., the 'Bottom Out Hint'). Alternatively, scaffolding problems are used to provide worked examples or to break a problem down into steps as a guide for problem solving. A detailed log is kept for each problem with regard to student actions, including answers, attempt count, hint requests, and scaffold usage. Students are not able to skip problems within the problem set, and must answer a problem correctly or arrive at the Bottom Out Hint before moving on to the next problem. Thus, Skill Builders offer the unique opportunity to investigate opportunity count within differing skills in a mastery-learning environment.

The dataset used in the present study only included information logged for main problems. Thus, scaffolding problems were excluded from analysis as they carry a high probability of student accuracy based on their nature. Further, ASSISTments Skill Builders can include problems with a variety of problem types including 'Fill-In,' where the student must answer an exact answer, 'Algebra,' where the student can enter any mathematically equivalent answer, and 'Multiple Choice' in which students must select an answer from a range of possible solutions. Skill Builder problem sets employing Multiple Choice problems were excluded from the present study due to their disproportionate ease and the potential for correct guessing. Additionally, Skill Builders with less than 1000 logged problems were excluded from analysis. Following all exclusions, the resulting dataset contained details for 85,862 problems logged by 3,210 unique students spanning 70 unique skills.

An abbreviated version of the logged data is presented in Table 1, displaying only the information pertinent to feature generation. The full dataset, including all logged data, can be accessed here. Within the sample data in Table 1, each row represents a problem logged for one student at a specific opportunity to practice the skill. A binary score is logged for each problem, along with an attempt count, the student's first response time (in milliseconds), the number of hints used, the number of hints available per problem, and the student's first action on the problem. For instance, the first row represents student 34's first opportunity on skill 102. The student answered the problem correctly in one attempt without the use of hints.

While a value of 1 in this column signifies that the student answered the problem correctly without assistance, a value of 0 may signify an incorrect first attempt or an immediate request for assistance from hints or scaffolding. First response time represents the duration of time in milliseconds from when a problem is started to the first logged action. As shown in Table 1, first action can include an attempt at answering the problem (0), a hint request (1), or a scaffolding request (2).

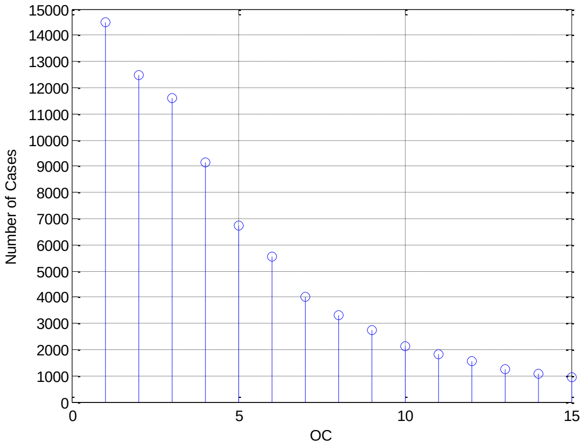


Figure 1. Number of Cases for Differing OCs.

As previously noted, Skill Builders require three correct consecutive answers for skill mastery. Thus, high performing students are likely to have minimal OCs within a skill, while struggling students are likely to have higher OCs within a skill. As OC increases, data points grow scarcer as students master (or fail to master) the skill. Figure 1 depicts this trend for OCs within the dataset. For example, there were approximately 12,000 cases of students experiencing three OCs for a skill, but only about 7,000 cases of students reaching five OCs for a skill. It should also be noted that Skill Builder problem sets carry a daily limit, or a preset number of problems that a student can attempt in one day. By default, the daily limit is set to ten problems. If a student exceeds the daily limit prior to correctly solving three consecutive problems, the problem set is effectively locked until the next day and the student is told to consult with her teacher. Therefore, it might be less accurate to make predictions for OC's greater than ten.

METHODS

Feature Generation and Organization

In order to apply RF to build prediction models, it was first necessary to modify the original data set by generating new features. The first generated feature combined original data for hints used and total hints available to establish the percentage of hints used at each OC. As different problems carry different hint totals, percentage of hints used offers a better understanding of student performance across problems. Next, first response times were groomed to remove outliers that are larger than 400ms (less than 1% of the problems logged were removed in this process) and to simplify the time structure to 10 second increments. We felt that it was unnecessary for time to be measured with such precision and as RF prefers discretized data, this binning process would help to avoid excess node splitting without much information loss. Additionally, a feature called 'historical accuracy' was generated to track a student's percentage of correctness across all prior OCs within a skill. Finally, as an organizational measure, all percentages in the modified dataset were discretized by units of 20% to simplify RF. For example, if historical accuracy was 65%, it was discretized to 60%, while if percentage of hints used was 75%, it was discretized to 80%. A sample of the resulting dataset is presented in Table 2.

In order to generate predictions for next problem performance, RF reads in features based on the organization of training data. We propose two organization methods for the features depicted in Table 2, with results presented for both methods.

Organization Method 1.

The first method of feature organization employs the structure depicted in Table 2. Columns, read left to right, serve as successive features or predictors for RF. Each row or problem serves as a case, and the predicted value is correctness on the next problem.

Organization Method 2.

This organization method sought to amend potential data loss observed in Organization Method 1 due to the

OC	Student	Skill	Correct	Attempts	FRT (10s)	% Hints	FA	Hist. Acc.
1	34	102	1	1	3	0	0	0
2	34	102	1	1	2	0	0	100
3	34	102	0	2	3	60	1	100
4	34	102	1	1	3	0	0	60
1	56	102	0	1	2	0	2	0

Table 2. Sample Data Following the Organization Method 1

OC	Student	Skill	Correct	Attempts	FRT (10s)	% Hints	FA	Corr$_{-1}$	Corr$_{-2}$	Att$_{-1}$	Att$_{-2}$	FRT$_{-1}$	FRT$_{-2}$	%H$_{-1}$	%H$_{-2}$	FA$_{-1}$	FA$_{-2}$	HA$_{-2}$
1	34	102	1	1	3	0	0	X	X	X	X	X	X	X	X	X	X	X
2	34	102	1	1	2	0	0	1	X	1	X	3	X	0	X	0	X	X
3	34	102	0	2	3	60	1	1	1	1	1	2	3	0	0	0	0	0
4	34	102	1	1	3	0	0	0	1	2	1	3	2	60	0	1	0	100
1	56	102	0	1	2	0	2	X	X	X	X	X	X	X	X	X	X	X

Table 3. Sample Data Following the Organization Method 2

consideration of only historical accuracy. Thus, a more detailed historical record is kept by implementing each feature at the current OC, as well as at OC-1 and OC-2. “Historical accuracy” of only OC-2 was included, for the information stored in historical accuracy of OC-1 and OC but not in that of OC-2 is covered by “Correct”. Each level of historical data is stored within each problem or case. A segmented display of this organizational method is provided in Table 3.

Table 2 and Table 3 provide a visual justification for establishing the Opportunity Count Model. Notice the imbalance of information pertaining to performance history observed in each organization method. Within method 1, historical accuracy, which reflects student knowledge of the skill, is not consistently available, with data lost for measures of initial knowledge on the first OC. Within organization method 2, the loss of data across OCs is more critical, with features showing potential inconsistent importance and reliability as predictors of future performance.

Random Forest

This paper used MATLAB’s implementation of RF (TreeBagger) to build student models and make predictions of student performance [7]. The dataset was divided into training and test segments, and 100 regression trees were developed using the training set. In this process, subsets of the training data were repeatedly sampled with replacement to construct trees. Along with these trees, TreeBagger also provided measures of out-of-bag error and feature importance:

Out-of-bag Error [10].

A subset of the training set is left out when building each tree, thereby leaving a portion of data “out of the bag.” After a tree is built, the out of bag subset moves through the tree and arrives at a prediction for the tree. The root mean square of pre-diction errors (RMSE) for all out-of-bag cases becomes known as the out-of-bag error.

Feature Importance [10].

When assessing the importance of a feature, m, the values of m in the out-of-bag cases are randomly permuted. A secondary measure of out-of-bag error is then calculated based on the permuted data. The difference between this secondary out-of-bag error and the original out-of-bag error for m, is regarded as the importance of feature m. The larger the difference in error, the more important role the feature plays in prediction. Negative importance values suggest a feature that is useless or even harmful in prediction.

As RF progress through the decision tree building process, subsets of features are chosen randomly to establish node splits. The number of features, n, in this subset can be limited to make enhance predictive accuracy. For the current study, a wide range of values was explored, ultimately using n with minimum out-of-bag error to drive RF in the test set.

For OCM model, we will run and test RF for each OC, with sub dataset of that OC. Code used in this paper can be accessed here.

KT

For KT, we used the Bayes Net Toolbox for Matlab. [1]

RESULTS

RF was run using both data organization methods to examine prediction accuracy and feature importance for the Traditional Model (TM), a single model for all OCs, and for the Opportunity Count Model (OCM), our proposed flexible approach in which separate models are built for different OCs. Within each organization method, five fold

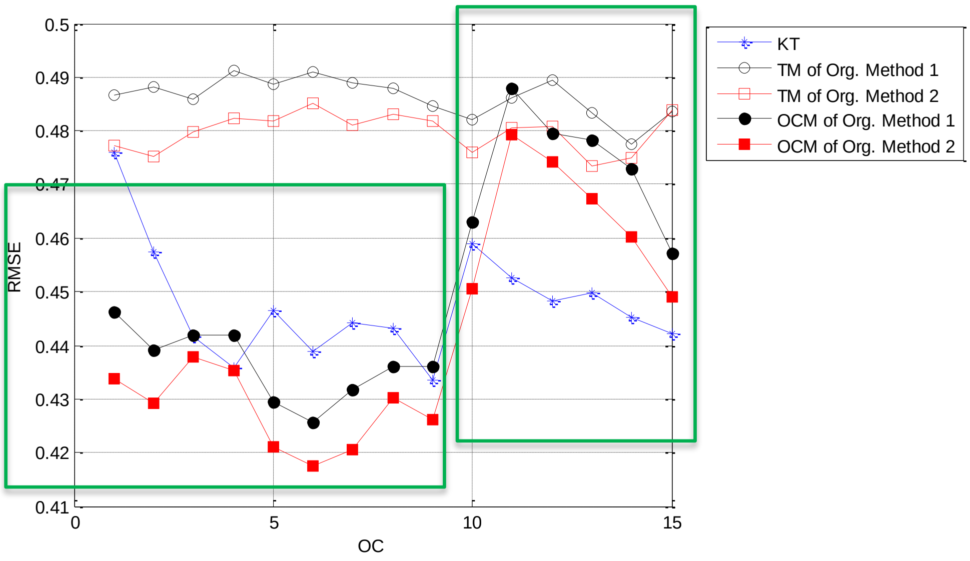


Figure. 2. Prediction Accuracy of Models. (The point at ith OC shows prediction accuracy of i+1th correctness)

cross validation was used to establish RF models. Standard Knowledge Tracing (KT) was also performed for comparison.

Prediction Accuracy

These four models were then used to make predictions of student next problem correctness within the test set. Root mean square error (RMSE) was calculated for each prediction within each fold, and ultimately averaged across all five folds. Figure 2 shows the RMSE of model predictions for next problem correctness at various OCs. Values at the 5th OC represent error in predicting correctness on the 6th OC. Prediction errors for KT are included for comparison.

Feature Importance

Feature importance, designated by the difference in out-of-bag error, was calculated for features within the Traditional Model and the Opportunity Count Model for both data organization methods. Importance was calculated for each feature within each fold, and ultimately averaged across all five folds. Table 4 presents feature importance for each model when using organization method 1, while Table 5 presents feature importance for each model when using organization method 2.

DISCUSSION

There is dramatic decrease at prediction performance for almost all models after 10th OC. Please note that in Figure 2, the 10th OC point represents the prediction accuracy of 11th OC correctness. We believe that the main reason for this decrease in performance is caused by the data set. In ASSISTments, most skill builders have daily limit of 10 OCs, which means students will stop practice after 10 OCs, and data of 11OCs (and later) came from some days later.

OC	Skill	Correct	Attempts	FRT (10s)	% Hints	FA	Hist. Acc.
TM							
All	13.1	2.88	1.89	4.70	2.54	2.19	3.91
OCM							
1	6.21	0.83	0.81	1.16	0.81	0.76	0.00
2	7.35	0.79	0.22	1.29	0.74	0.48	1.79
3	6.99	1.08	0.08	1.38	0.68	0.56	1.89
4	6.67	0.78	0.49	1.20	0.39	0.65	1.57
5	6.11	0.71	0.48	0.98	0.46	0.8	0.51
11	1.29	-0.07	-0.02	0.07	-0.07	0.2	0.44

Table 2. Feature Importance in Data Organization Method 1

OC	Skill	Correct	Attempts	FRT (10s)	% Hints	FA
TM						
All	2.96	-0.98	1	0.41	0.93	-0.28
OCM						
1	2.51	0.24	0.31	0.45	0.33	0.34
2	3.95	0.28	-0.06	0.55	0.23	-0.02
3	3.97	0.46	0.13	0.41	0	0.01
4	3.68	0.33	0.14	0.61	-0.06	0.17
5	3.8	0.23	0.15	0.49	0.06	0.12
11	0.68	0.08	-0.04	-0.19	-0.26	-0.17

OC_{-1}	$Correct_{-1}$	$Attempts_{-1}$	FRT $(10s)_{-1}$	% $Hints_{-1}$	FA_{-1}
TM					
All	-0.01	-1.59	-3.43	-3.73	-1.84
OCM					
1	0	0	0	0	0
2	0.87	0.11	0.3	0.18	0.07
3	0.65	0.11	0.28	0.12	-0.04
4	0.33	0.21	0.26	-0.04	0.12
5	0.1	0.06	0.27	-0.06	0.01
11	0.24	-0.1	0.09	-0.13	-0.19

OC_{-2}	$Correct_{-2}$	$Attempts_{-2}$	FRT $(10s)_{-2}$	% $Hints_{-2}$	FA_{-2}	Hist. $Acc._{-2}$
TM						
All	-2.12	-4.65	-6.23	-2.14	-1.18	-5.09
OCM						
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0.79	0.06	0.08	-0.23	-0.14	0
4	0.63	0.21	0.28	-0.04	-0.27	0.5
5	0.12	-0.17	0.25	-0.28	-0.02	0.17
11	-0.12	0.06	-0.05	-0.23	-0.15	-0.02

Table 3. Feature Importance in Data Organization Method 2 for Traditional Model

The discontinuity between 10th and 11th OC weakens the performance of models that rely on previous OCs to predict NPC. On the other hand, the data size decreased for modeling as OC increases, as shown in Figure 1. KT is based on Hidden Markov Model (HMM), which learns parameters more and more accurately as it encounters more data along OCs. Therefore, data size decrease will not harm performance of KT dramatically. However, OCM is building models at each OC, depending data size at each OC. The model accuracy depends highly on the data size at each OC. This may explain that KT performs better than OCM at later OCs.

The two traditional models have very bad prediction accuracy along all OCs. This is not surprising. Traditional models built one model for all OCs. On one hand, they have much fewer parameters (or freedom degree). On the other hand, they don’t consider the variance of features’ importance and availability at different OCs.

Within 10 OCs, the best OCM performs always better than KT, especially at the 1st and 2nd OCs (predicting 2nd and 3rd correctness). At the very early OCs, KT does not have access to many OCs, and has not learned good parameters for HMM. On the contrary, RF exploits more features that KT, and learns from more previous information to build a better model.

Also, using organization method 2, the model is always better than using organization method 1, whether for TM or OCM. This can be explained by the fact that organization method 2 provides more detailed information, rather than one aggregated feature from previous OCs. Thanks to the self-cross-validation mechanism within RF, we don’t need to worry about overfitting when using a lot of features.

By comparing different models, results suggest using different modeling techniques at different phase of students’ practice.

Further, findings suggest that feature importance varies as OC changes, supporting the proposed approach to student modeling when employing rich features. The results presented in Tables 4 and 5 revealed that regardless of organization method, feature importance could differ considerably with increases in OC. For example, when considering an OC of 2 using organization method 1, aside from the importance of skill identification, the most relevant features within the model were first response time and historical accuracy. However, when considering an OC of 5, first response time still dominated, but historical accuracy was not as important. When observing the same OC models using organization method 2, the most relevant feature within the model for an OC of 2 was previous problem correctness. For an OC of 5, features gain complexity and first response time became most important, aside from skill. It should also be noted, that features with negative values for importance, or those that potentially hinder modeling, differ across OCs. Within the TM, most features that consider historical elements performance using organization method 2 actually hurt the modeling process, as observed in the model’s low predictive accuracy. This is likely due to the fact that these features rely heavily on past OCs, making the model suffer a lot from information loss. On the contrary, within the OCM most of these features appear helpful when modeling low OCs. Thanks to these useful features, OCM is able to outperform KT at low OCs. The more interesting point of feature variation is to find the important factor at different phase of learning. However, since this paper focuses on the prediction accuracy of student modeling, and incorporates a lot of features, it is hard to find a clear pattern of learning. But in future work, we can use a simpler model to detect what features are important at different learning phase and why.

CONTRIBUTION

The present study revealed that the predictive accuracy of models is strongly linked to the organization of a dataset and oscillations in feature availability and importance within differing OCs. The OCM, proposed as a flexible approach to student modeling, was observed to be more successful than traditional modeling methods when considering OCs below 10. Also, different modeling techniques shine at different phase of students’ practice.

ACKNOWLEDGMENTS

We acknowledge funding from NSF (grant #’s 1316736, 1252297, 1109483, 1031398, 0742503), ONR’s “STEM Grand Challenges,” and IES (grant #’sR305A120125, R305C100024).

REFERENCES

[1] Bnt, https://code.google.com/p/bnt/

[2] Breiman, L. Random forests. Machine learning, 45, no. 1 (2001), 5-32.

[3] Corbett, A. T., and Anderson, J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction 4, no. 4 (1994), 253-278.

[4] Duong, H. D., Zhu, L., Wang, Y. and Heffernan, N. A Prediction Model Uses the Sequence of Attempts and Hints to Better Predict Knowledge: Better to Attempt the Problem First, Rather Than Ask for a Hint. Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013), 316-317.

[5] González-Brenes, J. P., Huang, Y. and Brusilovsky, P. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014).

[6] Heffernan, N. and Heffernan, C. 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, no. 4 (2014), 470-497.

[7] MathWorks, http://www.mathworks.com/help/stats/treebagger.html.

[8] Pardos, Z. A. and Heffernan, N. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. Journal of Machine Learning Research W & CP (2010).

[9] Pavlik Jr, P. I., Cen, H. and Koedinger, K. R. Performance Factors Analysis--A New Alternative to Knowledge Tracing. In the Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling. IOS Press Amsterdam (2009), 531-538.

[10] Random Forests, https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.html.

[11] Wang, Y., and Heffernan, N. The" Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. In FLAIRS Conference (2011).

Enhancing the Efficiency and Reliability of Group Differentiation through Partial Credit

Yan Wang, Korinn Ostrow, Joseph Beck, Neil Heffernan
Worcester Polytechnic Institute
Worcester, MA 01609
{ywang14, ksostrow, josephbeck, nth} @wpi.edu

This section has been published at:

Wang, Y., Ostrow, K., Beck, J. & Heffernan, N. (2016). Enhancing the Efficiency and Reliability of Group Differentiation through Partial Credit. *In the Proceedings of 6rd International Conference on Learning Analytics & Knowledge*.

ABSTRACT

The focus of the learning analytics community bridges the gap between controlled educational research and data mining. Online learning platforms can be used to conduct randomized controlled trials to assist in the development of interventions that increase learning gains; datasets from such research can act as a treasure trove for inquisitive data miners. The present work employs a data mining approach on randomized controlled trial data from ASSISTments, a popular online learning platform, to assess the benefits of incorporating additional student performance data when attempting to differentiate between two user groups. Through a resampling technique, we show that partial credit, defined as an algorithmic combination of binary correctness, hint usage, and attempt count, can benefit assessment and group differentiation. Partial credit reduces sample sizes required to reliably differentiate between groups that are known to differ by 58%, and reduces sample sizes required to reliably differentiate between less distinct groups by 9%.

Categories and Subject Descriptors

K: Applications to Education. K.3: Computers and Education. I.6 Simulation and Modeling.

General Terms

Measurement, Experimentation, Reliability.

Keywords

Partial Credit, Group Differentiation, Resampling with Replacement, Randomized Controlled Trial, Data Mining.

INTRODUCTION

The learning analytics and educational data mining communities have established a variety of well-vetted models to predict student knowledge and trace performance both within and across knowledge components (i.e., skills). The gold standard for student modeling, Knowledge Tracing (KT), has maintained its reign for almost a quarter-century despite relying on a rudimentary sequence of correct and incorrect responses to estimate the probability of student knowledge [2]. Attempts to enrich this approach have included supplemental estimates of prior knowledge to individualize predictions to each student [9], supplemental estimates of item difficulty to individualize to each problem [10], and the implementation of flexible correctness via consideration of hint usage and attempt count [12, 13, 7]. Despite these excursions, popular learning systems, including the Cognitive Tutor series, still largely rely on traditional KT to inform mastery learning [4].

In parallel, enthusiastic support has been growing for the use of randomized controlled trials embedded within online learning platforms to investigate best practices and enhance the user experience. Randomized controlled trials are the soundest approach to social science, allowing researchers to postulate causal relationships between independent and dependent variables. Within the realm of education, experimental design has historically been longitudinal, with formal pre- and post-tests, highly controlled curricula, and vast sample populations required for class-level or even school-level randomization. However, the expanding popularity of online learning platforms used for classwork and homework offers researchers an opportunity to gather data more efficiently, with fewer logistic constraints, and requiring smaller samples due to random assignment at the student-level.

The present work employs data mining methodologies on randomized controlled trial data from ASSISTments, a popular online learning platform, to assess the benefits of incorporating additional student performance data when attempting to differentiate between two user groups. The platform, created in 2002, now supports over 50,000 users around the world, providing students with immediate feedback and enhancing assessment for teachers [3]. The ASSISTments platform is an easily accessible shared tool for educational research that offers the unique opportunity to bridge the gap between the analysis of randomized controlled trials and more traditional data mining. Considering student performance variables for the purpose of group differentiation is arguably a worthy venture for both realms.

Many learning platforms assess student performance using standard binary correctness (i.e., a student's accuracy on her first solution attempt). Instead, we argue for a combination of features that better define the learning process: initial accuracy, feedback usage, and attempts required for success. The present work suggests that such features can be combined to establish a partial credit metric to enhance analytic efficiency when attempting to differentiate between two user groups (i.e., experimental conditions). It is not surprising that a more robust view of student performance can alter a researcher's ability to pinpoint the effectiveness of an intervention. Modeling numerous features per data point requires fewer data points to arrive at distinct conclusions (i.e., posttests could simultaneously be shortened and yet made more robust for both students and researchers). Previous work has also suggested that infusing controlled assessment with learning opportunities (i.e., providing feedback or allowing multiple attempts) directly benefits robust student learning [1]. However, many researchers hesitate when considering the allowance of these features within posttests. As such, the present work seeks to validate the allowance of 'partial credit' within randomized controlled trial posttests.

Although ASSISTments employs binary scoring, feedback usage and attempts required for success can be considered in the algorithmic calculation of partial credit scores. Recent research within ASSISTments has examined the potential benefits of partial credit scoring for student modeling [7] and has validated partial credit penalizations using an extensive grid search of possible scoring procedures [6]. We extend this work by asking: Does partial credit scoring enhance the efficiency with which significant differences can be detected between groups of students within a randomized controlled trial? We define 'enhanced efficiency' as a reduction in the sample size required to reliably observe significant differences between groups (akin to enhancing power, or reducing Type II error).

DATASET

The dataset is comprised of log files from a previously published randomized controlled trial on the effects of interleaving skill content within a brief homework assignment [8]. The original study was conducted with a group of participating teachers from a suburban middle school in Massachusetts. Researchers worked with teachers to select content for three skills (A, B, C). A practice session comprised of twelve questions (four per skill) was presented to students in one of two possible linear presentations: blocked or interleaved. Students randomly assigned to the blocked condition received questions grouped by skill (A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4, C_1, C_2, C_3, C_4), while those randomly assigned to the interleaved condition received the same questions in a mixed skill pattern (A_1, A_2, B_1, B_2, C_1, C_2, A_3, B_3, C_3, B_4, C_4, A_4). All students partook in a follow-up assignment containing three questions (A_5, B_5, C_5) as a delayed posttest. The posttest was presented with tutoring in the form of on-demand hint messages and students were allowed multiple attempts to achieve accuracy.

The original work presented an Analysis of Covariance (ANCOVA) on the average posttest performance of 146 students (*n* Blocked = 60, *n* Interleaved = 86) based on binary scoring. Results only trended toward significance across the full sample, but split file analyses revealed significant learning gains for low skill students who had received the interleaved assignment. In a parallel analysis, average hint usage and attempt counts at posttest were considered through a Multivariate Analysis of Covariance (MANCOVA), with results suggesting a significant multivariate effect driven by a reduction in posttest hint usage for students in the interleaved condition. These results inspired the present work. Binary scoring alone could not consistently allow for reliable group differentiation until controlling for student skill level.

Additionally, robust value was added via consideration of posttest variables that define partial credit in the present work. How would results have differed if the authors of the original work had considered algorithmic partial credit scoring?

METHODOLOGY

To examine the potential for using partial credit as a metric to more efficiently differentiate between groups, the dataset was processed using a definition of partial credit scoring previously validated within ASSISTments. Past research on modeling student performance within ASSISTments has revealed that certain definitions of partial credit significantly outperform others when attempting to predict next problem performance [6]. The algorithm presented in Figure 1, originally defined in [7], has been proven as an effective definition in the context of modeling student performance [7]. This algorithm establishes a score categorization based on logged information regarding the student's performance: the number of attempts required to reach an accurate response (attempt), the number of hints requested (hint_count), and whether or not the student was provided the answer through the bottom out hint (bottom_hint). A version of this algorithm was recently implemented within the ASSISTments platform.

After passing the dataset through the algorithm presented in Figure 1, the resulting file contained categorical partial credit scores (0, 0.3, 0.6, 0.7, 0.8, 1.0) for each students' performance on each problem in the practice and posttest sessions. Students could still earn full credit in the traditional sense (i.e., answering correctly on the first attempt), but only lost full credit if they made more than five attempts or were provided the answer through the bottom out hint. An example of the processed data, with variables from the original file as well as the resulting penalizations and partial credit scores, is presented in Table 1. The processed dataset has been stripped of student identifiers and is available at [11] for reference.

When considering user groups, this dataset offered two clear opportunities for group differentiation: experimental condition and discretized student performance level. The latter metric defines students as either high performing or low performing

```
IF attempt = 1 AND correct = 1 AND hint_count = 0
    THEN 1
ELSIF attempt < 3 AND hint_count = 0
    THEN .8
ELSIF (attempt <= 3 AND hint_count=0)
OR (hint_count = 1 AND bottom_hint != 1)
    THEN .7
ELSIF (attempt < 5 AND bottom_hint != 1)
OR (hint_count > 1 AND bottom_hint != 1)
    THEN .3
ELSE 0
```

Figure 1. Partial credit algorithm originally defined in [7]

Table 1. Randomized controlled trial data with partial credit algorithm employed

Student	Condition	Problem	Binary	Hints	Bottom Out	Attempts	Penalization	Partial Credit Score
Student 1	Interleaved	A_1	0	1	0	2	0.3	0.7
Student 1	Interleaved	B_1	0	0	0	2	0.2	0.8
Student 1	Interleaved	C_1	1	0	0	1	0.0	1.0
Student 2	Blocked	A_1	0	3	1	3	1.0	0.0
Student 2	Blocked	A_2	0	0	0	3	0.3	0.7
Student 2	Blocked	A_3	0	1	0	4	0.7	0.3

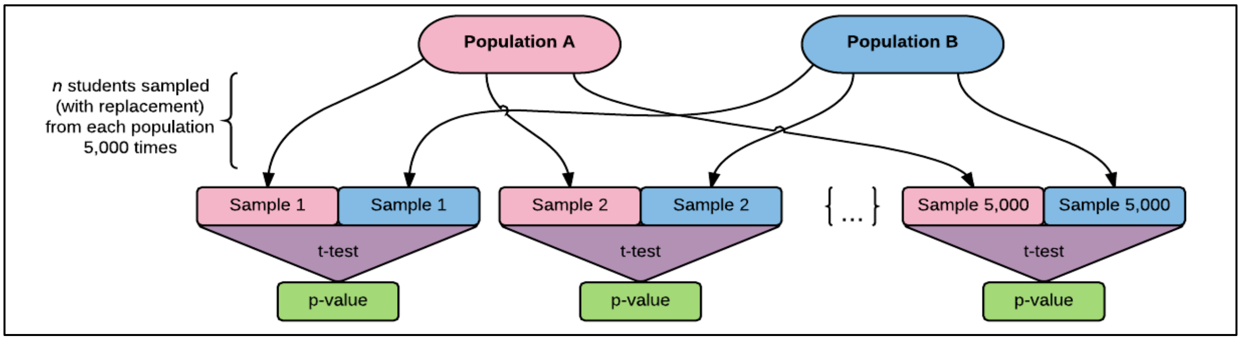


Figure 2. The resampling process used to create samples of *n* students from each population. Each set of samples was used in a t-test and significance values were recorded. This process was repeated 5,000 times for each group of *n* students.

based on a measure of prior knowledge calculated using the ASSISTments database. Prior knowledge is established by considering the average accuracy (in the binary sense) of all problems that a student has ever solved within ASSISTments. A median split can then be applied to this metric within a dataset to discretize groups of generally 'high performing' and generally 'low performing' students. In previous research, these groups have been found to exhibit significantly different performance, with low performing students logging reliably lower accuracy, more hints, and more attempts [8]. Thus, while observing differentiation between experimental conditions is subject to the success of the intervention, grouping students by skill level offers an obvious differentiation to test the efficacy of partial credit.

The full sample (146 students) was used to test differentiation between student performance levels. Equivalent samples of students were randomly selected from each performance level in single student increments (i.e., 5 students, 6 students, 7 students, etc.) For each set of equivalent samples of size *n*, an independent samples t-test was performed to compare the difference in partial credit scores between Sample 1 (a subset, *n*, of high performing students) and Sample 2 (a subset, *n*, of low performing students). A p-value denoting level of significance was recorded. This process was repeated to examine differences between Sample 1 and Sample 2 when considering binary scoring. These 'trials' were repeated 5,000 times per sampling increment. This process is depicted visually in Figure 2. For both partial and binary credit, sets of resulting p-values were then analyzed to determine the percentage of trials in which significant differences were observed between samples ($p < .05$). Findings were graphed for a visual comparison of the two scoring methods. Analyses and mappings were conducted using MATLAB [5] via code available for further consideration at [11].

This procedure was also used to differentiate between students based on experimental condition: blocked or interleaved. As the original work suggested that experimental condition only significantly altered achievement in low performing students, the present analysis considers only this subset of the original sample. Resampling with replacement was then used to establish artificial groups as large as desired. Please note that resampling is not employed in the present work to draw conclusions regarding the strength of a particular subsample or condition. The sole purpose of our analysis is to show that partial credit scoring can be used to reduce the sample sizes required to reliably differentiate between groups.

RESULTS

Results suggest that partial credit is exceptionally efficient in differentiating between distinct groups. Table 2 presents the differences in average correctness, hint usage, and attempt count observed when students are discretized into high and low performance levels - two groups that we know to be quite discernible and are therefore used here to validate our approach. Figure 3 depicts the percentage of samples in which significant differences were observed between these two groups. As these groups show obvious distinctions, both binary and partial credit scoring allow for 100% reliability of group differentiation with samples of fewer than 60 students. However, it should be noted that partial credit (red/dashed line) requires consistently smaller samples and attains reliability far more efficiently than binary scoring (blue/solid line). The resampling procedure suggested that

Table 2. Means and SDs for average correctness, hints, and attempts across performance levels

Group	Correctness	Hints	Attempts
Low Performing	0.54 (0.28)	0.72 (0.69)	2.05 (1.11)
High Performing	0.75 (0.22)	0.08 (0.21)	1.40 (0.43)

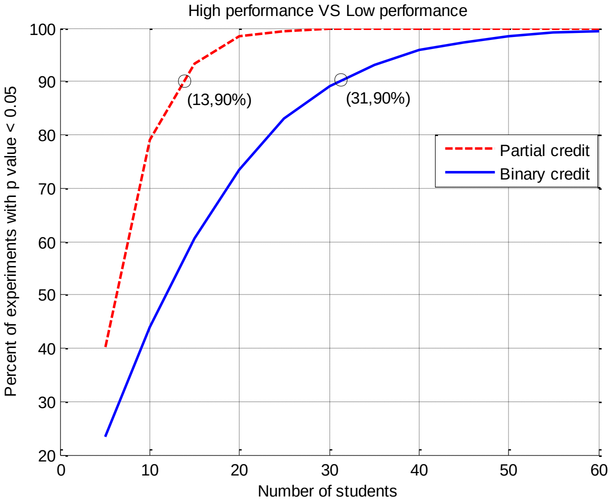


Figure 3. Significant differentiation in Performance Levels using Binary Scoring and Partial Credit Scoring. In groups with a known significant difference, differentiation is more efficient using partial credit. Sample size required for significant differentiation in 90% of trials is reduced by 58%.

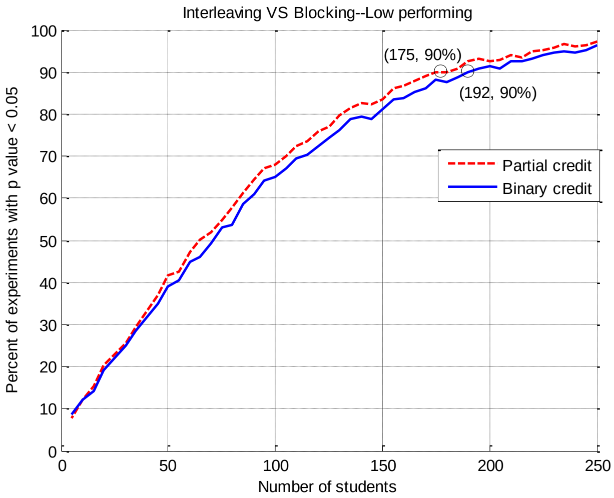


Figure 4. Significant differentiation in Condition using Binary Scoring and Partial Credit Scoring. In groups with a less substantial difference, differentiation is still almost always more efficient using partial credit. Sample size required for significant differentiation in 90% of trials is reduced by 9%.

Table 3. Means and SDs for average correctness, hints, and attempts across conditions for low performing students

Condition	Correctness	Hints	Attempts
Blocked	0.48 (0.25)	0.89 (0.67)	1.98 (0.58)
Interleaved	0.56 (0.29)	0.62 (0.67)	2.16 (1.37)

when using partial credit, equivalent groups of 13 students offer enough power to observe significant differences between performance levels in 90% of trials, while equivalent groups of 31 students were required when using binary scoring. Thus, within this context, using partial credit allowed sample sizes to be reduced by 58% while still obtaining the same result.

Although significant differences between experimental conditions within low performing students were more difficult to discern, as limited by the strength of the intervention, partial credit continued to offer more robust group differentiation when considering these user groups, as depicted in Figure 4. An analysis of means for the variables that combine to form partial credit revealed that low performing students in the interleaved condition were more accurate on average at posttest with fewer hints, as displayed in Table 3. Resampling suggested that when using partial credit, equivalent groups of 175 students offer enough power to observe significant differences between performance levels in 90% of trials, while equivalent groups of 192 students were required when using binary scoring. Thus, within this context, using partial credit allowed sample sizes to be reduced by 9% while obtaining the same result.

METHOD VALIDATION

When smaller equivalent sample sizes are required to differentiate between groups, Type II error is reduced for consistent sample sizes across scoring metrics. Before celebrating this finding, it is necessary to evaluate whether partial credit scoring in turn increases Type I error.

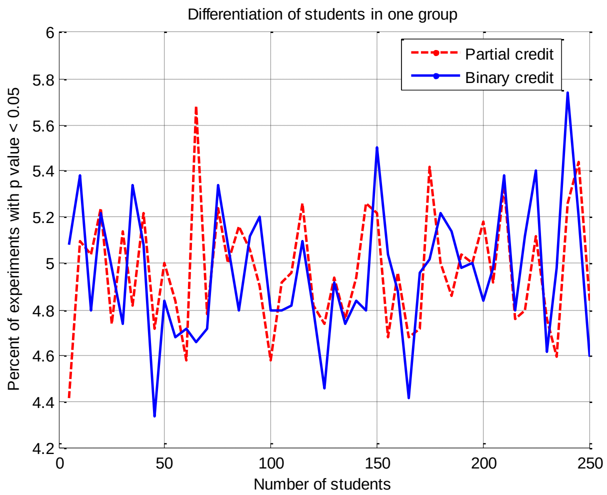


Figure 5. Type I error when resampling students from a solitary population using Binary Scoring and Partial Credit Scoring. Measures show roughly similar trends, suggesting that while partial credit allows for more robust group differentiation, it does not significantly impact Type I error.

If no actual difference exists between two groups and we maintain a threshold of $p < .05$ in determining a significant difference, the Type I error rate, or alpha, should be 5%. In order to determine whether partial credit has reduced Type II error simply by increasing Type I error, we simulated a null experiment with our dataset. The full sample population (146 students) was subjected to the resampling (with replacement) process, without predefining students as having high or low performance or as belonging to a particular experimental condition. Thus, for every sample increment, *n*, Sample 1 and Sample 2 were randomly selected from the full population (establishing samples that were not distinctly different). An independent samples t-test was conducted to analyze the difference in partial credit scores between subsamples. This 'trial' was repeated 5,000 times, with p-values recorded for each trial. Complimentary trials were conducted using binary correctness. The percentage of trials resulting in significantly different subsamples is charted in Figure 5. Both measures show roughly similar trends, with approximately 5% of trials resulting in significant findings. This finding suggests that while partial credit allows for more robust group differentiation, it does not significantly influence Type I error.

DISCUSSION & FUTURE WORK

The present work sought to examine whether partial credit scoring could be used to enhance the efficiency of group differentiation within a previously published randomized controlled trial. Results confirmed our expectations, suggesting that partial credit is a more robust measure of student performance that increases the reliability of group differentiation and reduces the sample size required to observe significant differences (or, enhances power).

Partial credit scoring held merit for differentiating both between student performance levels and between experimental conditions. The lack of strength in the latter finding may be correlated with the efficacy of the intervention itself; differentiation based on a learning intervention should not be expected to be as robust as differentiation based on a mathematically established dichotomy. Still, trends in reliability for both scoring metrics follow the standards of a power analysis: if sample sizes in the original work had been larger, the intervention would have proven reliably significant.

It should be noted that while we observed consistent positive effects for partial credit, it *is* mathematically possible for the metric to underperform binary scoring. When using t-test comparisons, smaller p-values are obtained as t-statistics increase. T-statistics are inflated when mean differences between groups are large while variance within groups is low. Mathematically, the use of partial credit reduces within group variance while increasing the mean for each group. With this increase in means, it would be possible for binary scoring to outperform partial credit in a heavily skewed dataset.

A potential limitation of this approach can be found in the balance between enhancing group differentiation by adding measures of student performance and overfitting student performance. One could argue that to most efficiently differentiate between groups, all available student data could be collapsed into a partial credit metric, perhaps using a regression model. While this would likely result in better differentiation, the overly robust definition of 'partial credit' would fail to generalize to other online learning platforms, or possibly even to other content or user populations within the ASSISTments platform. Future work should consider the pros and cons of supplementing partial credit scoring with additional measures of student performance.

Another potential limitation of this work is that students' habits within the ASSISTments tutor are normative to those of a binary system; the majority of students understand that they will lose all credit if they request tutoring feedback or make more than one attempt. Thus, any definition of partial credit that uses a data mining approach to work backwards toward group differentiation should be considered potentially skewed. As partial credit was recently implemented within ASSISTments, future work should consider how the real-time effects of partial credit scoring impact the power of randomized controlled trials.

Future research should also consider how our partial credit approach contends with latent group differentiation, in an attempt to outperform modeling techniques like Knowledge Tracing. Even if latent, when two groups are qualitatively different (i.e., learned vs. unlearned, denoting skill mastery within KT) our method may be feasible to observe patterns leading to more reliable group differentiation. Future work should examine this paradigm, and consider the generalizability of using partial credit scoring within the context of other platforms and domains.

CONTRIBUTION

The work presented herein is novel in that it sought to bridge the gap between educational research and data mining by applying post hoc mining methods to the results of a previously published randomized controlled trial. Results suggested a substantial benefit of considering partial credit scoring within online learning platforms: increased efficiency in group differentiation which translates to increased power and reduced Type II error. Our findings further confirm the notion that allowing students to learn during assessment is beneficial to students and researchers alike. Student performance metrics that are typically lost on traditional posttests can actually improve data analysis. Further, our results suggest that by using robust measures of student performance, the number of items or opportunities analyzed need not be large to result in significant group differentiation, offering evidence for short, minimally invasive assessments. These findings translate to real world implications: significant outcomes can be observed with smaller samples and with fewer overall data points, reducing the many of the costs and constraints of experimental research.

ACKNOWLEDGMENTS

We acknowledge funding from multiple NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DRL-1031398), the U.S. Dept. of Ed. (IES R305A120125, R305C100024, GAANN), ONR, and the Gates Foundation.

REFERENCES

[1] Attali, Y. & Powers, D. (2010). Immediate feedback and opportunity to revise answers to open-end questions. Educational and Psychological Measures, 70 (1), 22-35.

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

[3] 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 of AIED, 24(4), 470-497.

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

[5] MATLAB version R.2013.a (2013). Natick, Massachusetts: MathWorks, Inc. Accessible at www.mathworks.com

[6] Ostrow, K., Donnelly, C., & Heffernan, N. (2015). Optimizing Partial Credit Algorithms to Predict Student Performance. In Santos, et al. (eds.) Proc of the 8th Int Conf on EDM, 404-407.

[7] Ostrow, K., Donnelly, C., Adjei, S., & Heffernan, N. (2015). Improving Student Modeling Through Partial Credit and Problem Difficulty. In Russell, D.M., Woolf, B., & Kiczales, G. (eds.) Proc of the 2nd ACM Conf on L@S, 11-20.

[8] Ostrow, K., Heffernan, N., Heffernan, C., Peterson, Z. (2015). Blocking vs. Interleaving: Examining Single-Session Effects within Middle School Math Homework. In Conati, Heffernan, Mitrovic, & Verdejo (eds.) Proc of the 17th Int Conf on AIED. Springer International, 388-347.

[9] Pardos, Z.A. & Heffernan, N.T. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In De Bra, Kobsa, & Chin (eds.) Proc of the 18th Int Conf on UMAP, 255-266.

[10] Pardos, Z.A., & Heffernan, N.T. (2011). KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Joseph A. Konstan et al. (Eds.), Proc of the 19th Int Conf on UMAP, 243-254.

[11] Wang, Y. (2015). Data and Code for Enhancing the Efficiency and Reliability of Group Differentiation through Partial Credit: http://tiny.cc/LAK2016-Resampling

[12] Wang, Y. & Heffernan, N.T. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. In Proc of the 24th Int FLAIRS Conf.

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

Partial Credit Revisited: Enhancing the Efficiency and Reliability of Group Differentiation at Scale

Yan Wang, Korinn Ostrow, Neil Heffernan
Worcester Polytechnic Institute
Worcester, MA 01609
{ywang14, ksostrow, nth} @wpi.edu

This section has been submitted to:

Wang, Y., Ostrow, K. & Heffernan, N. (2016). Partial Credit Revisited: Enhancing the Efficiency and Reliability of Group Differentiation at Scale. *The 9th International Conference on Educational Data Mining*. ACM.

ABSTRACT

Partial credit scoring is an assessment technique commonly used by teachers in authentic learning environments to measure student knowledge. Conversely, some of the most popular learner models rely on the binary correctness of skill items to predict student skill mastery. The present work seeks to push this paradigm by extending previous research on the benefits of partial credit for group differentiation. Datasets from ASSISTments and Cognitive Tutor are used to assess the implications of this approach at scale. Within twelve skills (six per platform), a resampling approach is used to conduct 5,000 trials per increment of *n* students to determine the size of equivalent samples required to reach a threshold in which 90% of trials report significant differences between high and low performing students (a ground truth difference). Results suggest that in eleven out of twelve skills, partial credit offered more efficient group differentiation. Applications of this approach to learner modeling and implications for the EDM community are discussed.

Keywords

Partial Credit, Group Differentiation, Resampling, Skill Builders, ASSISTments, Cognitive Tutor.

INTRODUCTION

Partial credit scoring is an assessment technique commonly used by teachers in authentic learning environments to measure student knowledge. The approach provides a softer and generally more accurate assessment of skill knowledge than binary scoring, which argues that students either know (100%; 1) or do not know (0%; 0) a skill item. Previous work promoting the use of partial credit within online learning platforms [8; 9] and within Educational Data Mining (EDM) practices [19; 4] has shown that researchers can gain a more robust understanding of student knowledge by looking beyond binary correctness when using skill items to predict mastery.

This observation, while somewhat obvious, still has the potential to impact traditional EDM approaches to learner modeling. Some of the most popular modeling techniques rely on the binary correctness of skill items to predict when a student will learn or 'mastered' a skill. For instance, Bayesian Knowledge Tracing (BKT), still regarded as a gold standard in student modeling after more than twenty years, relies on four parameters per skill item to predict the moment of learning [3]. Students begin working on a skill with some level of prior knowledge ($P(L_0)$), and within each item exist probabilities that they may get the item incorrect although they know the skill (*slip*, P(S)), that they may accurately answer the item although they do not know the skill (*guess*, P(G)), and that they learn from the item (P(T)) [3]. In recent years, researchers have strived to enhance the predictions produced by BKT by accounting for more robust student measures including personalized predictions of prior knowledge [11; 20], item difficulty [12], and partial credit scoring [17; 18; 9].

Although many of these individualized BKT models have proven successful, standard BKT is still employed in well-known tutoring systems and data mining endeavors. The Cognitive Tutor series uses knowledge tracing to track students' skill progress [4], and its creators discussed the approach in their landmark 'Lessons Learned,' noting that the field should seek to predict skill mastery by first gauging mastery at the item level [1]. BKT within Cognitive Tutor is tailored to individual students to track learning, rather than to the corpus of users as observed in many instances of the approach [1].

Cognitive Tutor, and other Intelligent Tutoring Systems, record binary accuracy scores as students complete items within skills. However, additional data can be extracted from tutor logs to algorithmically calculate partial credit scores for these items through data mining. This practice was used in [19] to show an increase in the efficiency and reliability with which significantly different conditions from a randomized controlled trial conducted within ASSISTments could be observed. The original work also presented proof of concept of the more substantial benefits of partial credit scoring by exploring the efficiency with which discretized student performance levels (i.e., high performing vs. low performing) could reliably be detected using both binary and partial credit scoring approaches [10]. Partial credit was consistently more efficient, requiring smaller sample sizes to detect ground truth differences. The present work seeks to extend these previous accounts of the benefits of partial credit for group differentiation by using datasets from ASSISTments and Cognitive Tutor to assess the implications of this approach at scale.

Specifically, the present work seeks to examine the efficacy and reliability of group differentiation through partial credit across platforms and at scale. Further, a data mining approach is used to explore how the definition used to employ partial credit effects the magnitude of its benefit to group differentiation across platforms.

DATASETS

Datasets from two popular systems were collected to examine the potential benefits of using partial credit in data mining endeavors. The following subsections detail those platforms and the datasets considered herein.

ASSISTments

ASSISTments is a popular online learning platform for K-12 mathematics, with a primary focus on skills at the middle school level. The platform provides assistance to more than 50,000 student users around the world, while simultaneously serving as a powerful assessment tool for teachers [5]. Teachers have the capacity to assign a variety of problem sets for classwork and homework, and often use ASSISTments to collect and grade bookwork while allowing students the benefits of immediate feedback.

Mastery learning based assignments called 'Skill Builders' are the most common type of assignment within ASSISTments. These problem sets are mapped to the Common Core State Standards [5] for clear organization and high accessibility. Skill Builders require students to complete a series of problems randomly selected from a skill pool until meeting a predefined threshold for skill mastery. The default for this threshold requires that students accurately answer three consecutive skill items. The dataset considered herein includes tutor log files from six of the most highly assigned Skill Builders within ASSISTments.

While working through a Skill Builder, students are able to access tutoring in the form of hints and scaffolding problems. For the current analysis, Skill Builders containing scaffolding problems were not considered in an attempt to purify opportunity count. Because scaffolding problems offer worked examples and guided direction by breaking a main problem down into sub-steps, answers for these questions tend to be skewed toward accuracy and can cloud the predictive ability of student models.

The analyses presented herein represent problem level averages across each student's first three skill opportunities (i.e., skill items solved). This approach was taken in an attempt to prove that group differentiation could be accomplished more efficiently through partial credit even when items are limited. Thus, prior to analysis, the dataset was also cleaned to remove students that answered fewer than three items within each skill.

Details pertaining to the six Skill Builders that comprise this dataset are presented in Table 1. The log files used in this analysis were accrued between September 2009 and December 2014 through regular student use of the system. Within the dataset, items were originally scored using binary correctness on the student's first action or attempt. For each item, the log files also contained details pertaining to the tutoring usage and attempts made by each student. For each of the six Skill Builders, an estimate of difficulty was calculated by considering the average accuracy of all students for all items within the skill. Lower values of this metric were considered a higher difficulty, given the inverse nature of examining accuracy. Within Table 1, the Skill Builders are presented from most difficult (Equation Solving with More than Two Steps) to least difficult (Scientific Notation). Multiple skills with varying difficulty were considered in an attempt to assess whether these factors moderate the benefits of partial credit.

Table 1. Details pertaining to ASSISTments skills

Skill Topic	Grade	Students	Difficulty*
Equation Solving (2 Steps +)	8	5,269	0.57
Greatest Common Factor	6	5,169	0.58
Distributive Property	7	5,693	0.63
Mult. Fractions/Mixed #s	5	4,719	0.74
+/- Integers	7	6,314	0.80
Scientific Notation	8	6,502	0.81

*Difficulty is represented by the average accuracy of all students on all problems within the skill.

Cognitive Tutor - Algebra 1

Cognitive Tutor is a series of broad reaching tutoring systems for students in grades 9-12 distributed by Carnegie Learning [2]. These tutors are built around the ACT-R theory of cognition, enlisting humanistic problem solving techniques to compare automated solution steps against student solutions and provide immediate feedback and assistance as necessary [1; 13]. Cognitive Tutors are distributed as a portion of broader curriculum reform, with courses available in multiple mathematics domains [15; 2]. Teachers generally adapt Cognitive Tutor and assign content for classwork or homework in alignment with other Carnegie Learning materials.

Table 2. Details pertaining to Cognitive Tutor skills

Skill Topic	Students	Difficulty*
Expressions, Negative Slopes	263	0.34
Combine Like Terms	264	0.62
Find X, Positive Slopes	268	0.65
Labeling Axes	263	0.67
Consolidate Var w/ Coeff	266	0.85
Consolidate Var w/o Coeff	263	0.90

*Difficulty is represented by the average accuracy of all students on all problems within the skill. Grade is not accessible within Cognitive Tutor data, but all skills fall within the domain of Algebra 1.

The Cognitive Tutor dataset used in the present work is composed of data from the Algebra 1 Course and was promoted as part of the Knowledge Discovery and Data Mining (KDD) Cup dataset in 2010 [6]. The full dataset, as retrieved from the PSLC DataShop [14], spans on academic year (2005-2006), with over 880K skill items completed by 559 students working within 106 Algebra skills. The present investigation focuses on data from the six most highly populated Knowledge Components, or skills. Details pertaining to these six skills are presented in Table 2.

All items within the Cognitive Tutor dataset carried binary scores for student performance, based on the student's first action or attempt. The dataset also included information about the tutoring experienced by students while working through skill items. In order to mirror the ASSISTments dataset as closely as possible for equivalent analyses, the Cognitive Tutor dataset was further cleaned to remove students that answered fewer than three items within each skill, and estimates of difficulty for each of the six skills were calculated based on the average accuracy of all students for all items within the skill. Within Table 2, skills are presented from most difficult (Expression, Negative Slopes) to least difficult (Consolidate Variables without Coefficients).

METHODS

As the present work serves to extend previous research on the efficiency and reliability of partial credit in the context of group differentiation, the methodology presented herein was adapted from previous work and presents much of the same terminology [19]. The following subsections highlight the three primary steps required to evaluate partial credit in the context of group differentiation.

Defining Partial Credit

Previous work vetted partial credit as a method to efficiently and reliably differentiate between groups of students when running randomized controlled trials to examine the efficacy of learning interventions within ASSISTments [19]. Building upon that work, the present analysis relies on the same definition of partial credit, presented algorithmically in Figure 1 (originally sourced from [9]). For each skill item, this algorithm considers the student's binary credit score alongside the first action they take when tackling the item (first_action), the number of attempts required to solve the item (attempt), the number of hints required to solve the item (hint_count), and a binary flag showing whether or not the student was given the answer through a bottom out hint (bottom_hint).

By running the algorithm presented in Figure 1 on both datasets, categorical partial credit scores (0, 0.3, 0.7, 0.8, 1.0) were amended to each skill item for each student. Using this approach, students lost credit primarily through the use of multiple hints or attempts. Full credit was only redacted for a skill item if the student used more than five attempts or requested the answer. When implemented in the platform, this goal of this method would be to allow students to access hint tutoring without suffering full penalization. Examples of this algorithmic calculation are presented for both ASSISTments and Cognitive Tutor data in Table 3. Full versions of the modified datasets have been stripped of student identifiers and made available at [16] for further reference.

Discretizing Student Performance

Within both datasets, students can be discretized as either high performing or low performing based on variables constructed to estimate of prior knowledge. Significantly different performance has been observed between these groups, with low performing students exhibiting reliably lower accuracy and higher hint and attempt use [10]. Using this type of known skill dichotomy offers a ground truth to test the strength of partial credit against binary scoring when differentiating between groups. Further, this metric's success in previous work [19] reinforced its use when scaling up the examination of partial credit.

```
IF attempt = 1 AND correct = 1 AND hint_count = 0
    THEN 1
ELSIF attempt < 3 AND hint_count = 0
    THEN .8
ELSIF (attempt <= 3 AND hint_count=0)
OR (hint_count = 1 AND bottom_hint != 1)
    THEN .7
ELSIF (attempt < 5 AND bottom_hint != 1)
OR (hint_count > 1 AND bottom_hint != 1)
    THEN .3
ELSE 0
```

Figure 1. Partial credit algorithm originally defined in [9]

Within ASSISTments, a student's "prior knowledge" is established by considering the average accuracy of all items (across skills) ever solved by that student. This variable is available in all ASSISTments data reports. Within Cognitive Tutor, a similar variable was calculated by averaging a student's accuracy across all available content with timestamps prior to beginning a particular skill. It is possible that this metric was a more reliable account of prior knowledge within Cognitive Tutor, as knowledge components, or skills in the system all pertain to Algebra I. Based on these prior knowledge metrics, samples were divided into high and low performing students using a median split, and students were flagged as generally high performing or low performing, as shown in Table 3.

Resampling with Replacement

After defining partial credit and discretizing students by performance level, the datasets were primed for examining the efficiency of partial credit in comparison to binary scoring through a rigorous resampling procedure. To conduct resampling, equivalently sized groups of students were randomly sampled (with replacement) from the discretized performance levels in increments of five students (i.e., 5 students, 10 students, 15 students, etc.). The replacement procedure allowed equivalent sample sizes to extend beyond the actual number of students available in the dataset to examine the simulated efficacy of partial credit within larger samples as necessary.

After each equivalent sampling, an independent samples t-test was conducted to compare the difference in partial credit scores between performance levels. A second independent samples t-test was conducted to compare the difference in binary credit scores between performance levels. Resulting p-values were recorded for each test, concluding a single "trial." "Trials" were repeated 5,000 times per sampling increment. Essentially, this produced a list of 5,000 p-values per metric, per equivalent sampling increment. P-values were then analyzed to determine the percentage of trials in which differences between student performance levels were observed to be significant ($p < .05$). Findings for each metric were graphed for comparison across all twelve skills (six from each system), and are presented in Section 4, Figure 2. All analyses and mappings were conducted using MATLAB [7] via code that has been made available at [16].

Table 3. An excerpt merged from both ASSISTments and Cognitive Tutor datasets to exemplify algorithmic partial credit scoring

Student/System	*Performance*	*Skill*	*Opportunity*	*Binary*	*Hints*	*Attempts*	*Answer*	*Partial Credit Score*
1-ASM	High	Distributive Property	1	0	1	2	0	0.7
1-ASM	High	Distributive Property	2	1	0	2	0	0.8
1-ASM	High	Distributive Property	3	1	0	1	0	1.0
2-ASM	Low	Scientific Notation	1	0	2	3	0	0.3
1-COG	Low	Combine Like Terms	1	0	3	4	1	0.0
1-COG	Low	Combine Like Terms	2	0	0	3	0	0.7
2-COG	High	Labeling Axes	1	0	1	2	0	0.7
2-COG	High	Labeling Axes	2	1	0	1	0	1.0

Note. ASM = ASSISTments, COG = Cognitive Tutor. Performance = Discretized student performance level. Opportunity = Sequential count of skill items experienced. Binary = Original binary score. Hints, Attempts, and Answer flag = student performance metrics for use in calculating partial credit.

RESULTS

Table 4. Means & SDs for correctness (C), hints (H), and attempts (A) across performance levels in ASSISTments

Skill Topic	*C*	*H*	*A*
Equation Solving (2 Steps +)			
High	0.65 (0.33)	0.63 (0.83)	1.82 (3.80)
Low	0.49 (0.37)	1.13 (1.05)	2.04 (2.46)
Greatest Common Factor			
High	0.65 (0.30)	0.42 (0.68)	1.95 (6.24)
Low	0.50 (0.33)	0.94 (0.95)	2.56 (3.16)
Distributive Property			
High	0.71 (0.31)	0.47 (0.80)	1.77 (2.93)
Low	0.55 (0.35)	0.93 (1.04)	2.14 (4.08)
Mult. Fractions/Mixed #s			
High	0.82 (0.25)	0.22 (0.50)	1.72 (10.22)
Low	0.66 (0.32)	0.67 (0.89)	1.91 (2.96)
+/- Integers			
High	0.87 (0.22)	0.08 (0.30)	1.24 (0.56)
Low	0.73 (0.31)	0.26 (0.62)	1.66 (2.27)
Scientific Notation			
High	0.86 (0.23)	0.13 (0.40)	1.33 (1.01)
Low	0.75 (0.30)	0.35 (0.71)	1.83 (6.36)

ASSISTments

Considering the ASSISTments dataset, results suggested that partial credit consistently offered more efficient group differentiation. For each skill topic, an analysis of means was performed to compare average correctness, hint usage, and attempt count within the first three items experienced by each student, depicting distinct trends between discretized performance levels, as show in Table 4. The set of graphs in the left half of Figure 2 depict the percentage of samples in which significant differences ($p < .05$) were observed between performance levels for each skill topic. The graphs are presented from most difficult skill on the top left, to least difficult on the bottom right. For all graphs, red lines denote partial credit and blue lines denote binary scoring.

Within each skill topic, partial credit consistently outperformed binary scoring across sampling increments. The magnitude of this benefit was differential across sets, but did not appear to be correlated with skill difficulty. Benefit magnitude was determined by calculating the reduction in the size of equivalent samples required for significant group differentiation in 90% of Trials. This threshold is pinpointed in the graphs within Figure 2, and presented in detail in Table 5. Within ASSISTments data, partial credit allowed reliable group differentiation to be attained with significantly fewer students regardless of skill topic. The average reduction across skill topics from binary scoring to partial credit was 23%, with a standard deviation of 8.8%.

Table 5. Group size at which 90% of samples result in significant differentiation ($p < .05$) for ASSISTments skills

	Group Size		Reduction
Skill Topic	*Partial*	*Binary*	*Binary to Partial*
Equation Solving (2 Steps +)	75	95	21%
Greatest Common Factor	55	90	39%
Distributive Property	85	100	15%
Mult. Fractions/Mixed #s	55	75	27%
+/- Integers	70	85	18%
Scientific Notation	115	140	18%

Note. A paired samples t-test of group sizes suggested that observed sample reductions were significant, p < .05.

Table 6. Means & SDs for correctness (C), hints (H), and attempts (A) across performance levels in Cognitive Tutor

Skill Topic	*C*	*H*	*A*
Expressions, Negative Slopes			
High	0.42 (0.31)	1.10 (1.36)	2.41 (1.27)
Low	0.26 (0.30)	2.01 (1.85)	2.91 (1.72)
Combine Like Terms			
High	0.72 (0.30)	0.18 (0.53)	3.64 (3.26)
Low	0.53 (0.35)	0.46 (1.09)	5.05 (4.49)
Find X, Positive Slopes			
High	0.72 (0.27)	0.46 (1.10)	1.93 (2.21)
Low	0.58 (0.28)	1.35 (1.97)	2.57 (1.95)
Labeling Axes			
High	0.69 (0.30)	0.17 (0.49)	1.38 (0.51)
Low	0.65 (0.32)	0.38 (0.98)	1.45 (0.63)
Consolidate Var w/ Coeff			
High	0.88 (0.22)	0.08 (0.29)	1.18 (0.35)
Low	0.81 (0.25)	0.23 (0.55)	1.30 (0.55)
Consolidate Var w/o Coeff			
High	0.92 (0.20)	0.05 (0.20)	1.09 (0.34)
Low	0.88 (0.26)	0.10 (0.32)	1.12 (0.31)

Cognitive Tutor

A mirrored analysis was conducted for the Cognitive Tutor dataset. Results suggested that in five out of six skills, partial credit offered more efficient group differentiation. Means analyses for average correctness, hint usage, and attempt count within the first three items experienced by each student within each skill again depicted highly discretized performance levels, as shown in Table 6. The set of graphs in the right half of Figure 2 depict the percentage of samples in which significant differences (p < .05) were observed between performance levels for each skill topic. Again, the graphs are presented from most difficult skill on the top left, to least difficult on the bottom right, and red lines denote partial credit while blue lines denote binary scoring.

Partial credit failed to outperform binary credit in one skill, “Combine Like Terms.” Within this skill, binary credit impressively outperformed partial credit, reaching reliable group differentiation with equivalent samples of 55 students, while partial credit required equivalent samples of 235 students (a 327% increase in sample size). In all other skills, the magnitude of the benefit provided by partial credit did not clearly correlate with skill difficulty. The magnitude of this benefit is pinpointed in Figure 2, and presented in detail in Table 7. Considering the five skills in which differentiation benefited from partial credit, average reduction across skill topics from binary scoring to partial credit was 42%, with a standard deviation of 21.6%.

Table 7. Group size at which 90% of samples result in significant differentiation (p < .05) for Cognitive Tutor skills

	Group Size		Reduction
Skill Topic	*Partial*	*Binary*	*Binary to Partial*
Expressions, Negative Slopes	70	85	18%
Combine Like Terms	235	55	-327%
Find X, Positive Slopes	60	80	25%
Labeling Axes	350	1090	68%
Consolidate Var w/ Coeff	110	275	60%
Consolidate Var w/o Coeff	350	575	39%

Note. A paired samples t-test of group sizes suggested that observed sample reductions were significant, p < .05.

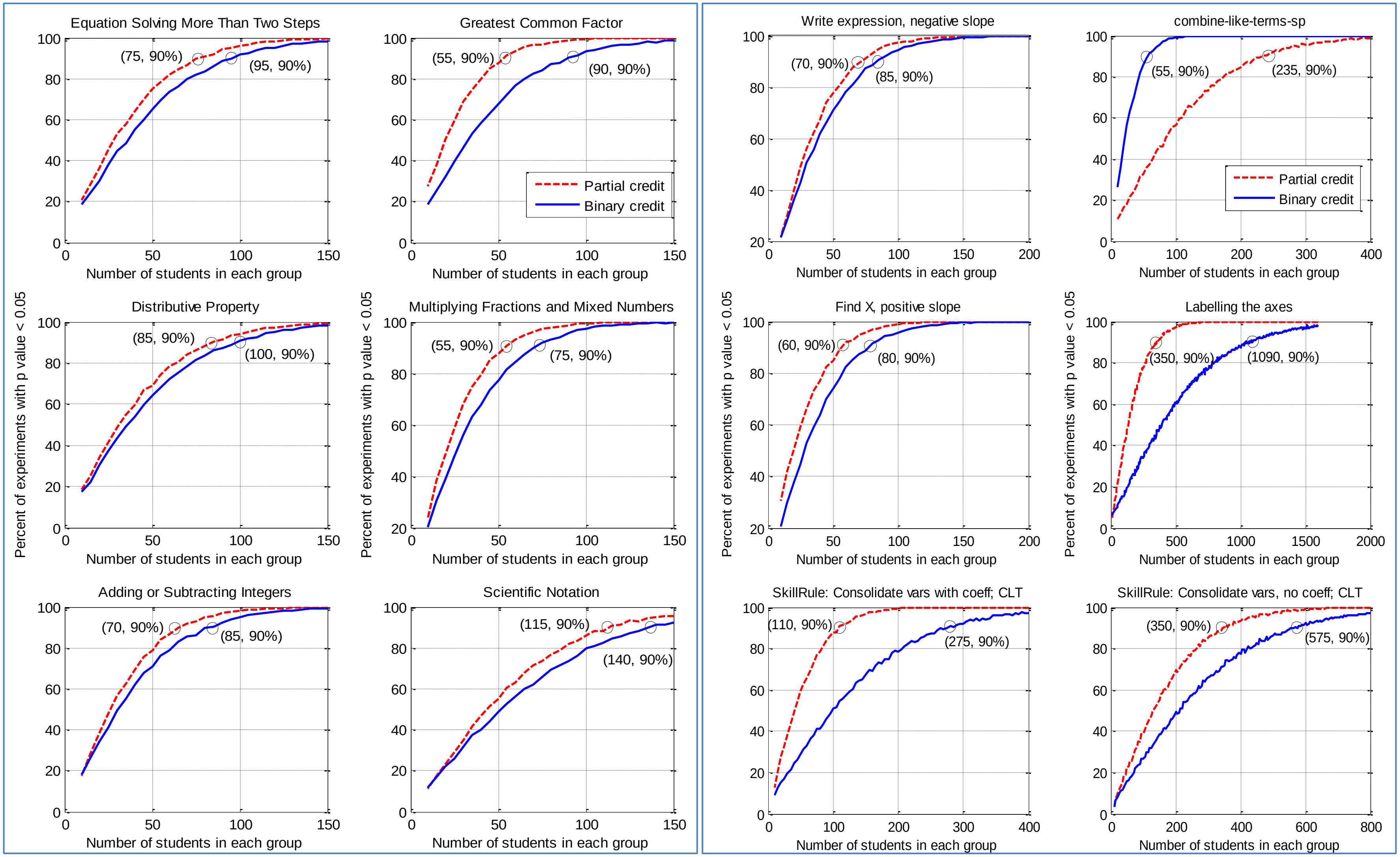


Figure 2. Significant differentiation in student performance level across six ASSISTments skills (Left) and six Cognitive Tutor skills (Right) using binary scoring (Blue) and partial credit (Red). Considering groups that should be significantly different (with an effect possibly mediated by skill difficulty), differentiation is more efficient using partial credit in 11/12 trials. Amongst successful trials, sample size required for significant differentiation in 90% of trials was reduced by between 15-39% within ASSISTments data (M = 23.0 SD = 8.8), and between 18-68% in Cognitive Tutor data (M = 42.0, SD = 21.6). Binary credit was found to be more successful at differentiating between groups within one trial of Cognitive Tutor data, for “Combining Like Terms” (Top Right).

METHOD VALIDATION

1.1 Validity of Partial Credit Metric

The partial credit algorithm used in the present work and derived in [9] was developed by two of the leading members of the ASSISTments team that are experienced math teachers and domain experts with strong knowledge of how students interact with the rich tutoring features of ASSISTments. The use of partial credit makes sense to most teachers and has been suggested as a more robust measure of student learning in previous work [8]. From an expert's point of view, partial credit scoring is logical and sound. Does it follow that the approach is also beneficial to data mining endeavors? As data miners commonly predict student knowledge without actually knowing ground truth, this question is difficult to answer directly.

As binary credit is the most commonly accepted metric in learner modeling, it is possible to compose "ground truth" for each student by averaging binary credit predictions. It is then possible to compare partial credit and binary scoring in relation to this "ground truth" when predicting student knowledge.

An analysis testing the validity of partial credit as a metric was conducted on students that had completed at least 30 ASSISTments Skill Builders and likewise, on students that had completed at least 30 Cognitive Tutor skills (both from the originally sourced datasets). For students that completed more than 30 Skill Builders or Cognitive Tutor skills, 30 were randomly selected from that student's logged data. As with earlier trials, only the first three skill items were considered within each Skill Builder or Cognitive Tutor skill. The resulting ASSISTments dataset included 2,206 students participating in at least 30 Skill Builders, while the resulting Cognitive Tutor dataset include 327 students participating in at least 30 skills. These datasets are available at [16] for additional reference. Binary credit was collected from a random selection of 15 Skill Builders and 15 Cognitive Tutor skills to represent "ground truth" knowledge. Then, partial and binary credit were tested and compared within the remaining 15 Skill Builders and 15 Cognitive Tutor skills in an attempt to predict ground truth. This process was conducted using five-fold cross validation. Prediction accuracy across the five folds was averaged to establish an overall prediction accuracy.

As it is difficult to attain robust trends from a single run of this procedure, the process was repeated 100 times. Results for average R^2 and RMSE of predictions are presented in Figure 3. Not surprisingly, as the number of sampled Skill Builders (Left) or Cognitive Tutor skills (Right) increased, prediction accuracy increased. Partial credit and binary credit showed similar predictive capacity within ASSISTments data. When sampling few Skill Builders, partial credit had slightly better capacity for prediction. As more Skill Builders were sampled, the capacity for prediction of binary credit increased. These trends were reasonable, as "ground truth" was defined as the average of binary predictions. However, it is clear that the predictive capacity of partial credit was less powerful in the Cognitive Tutor dataset. This finding provides theoretical support for the notion that partial credit definitions may be system specific and may not generalize well to other platforms, especially when predicting across a large number of skills.

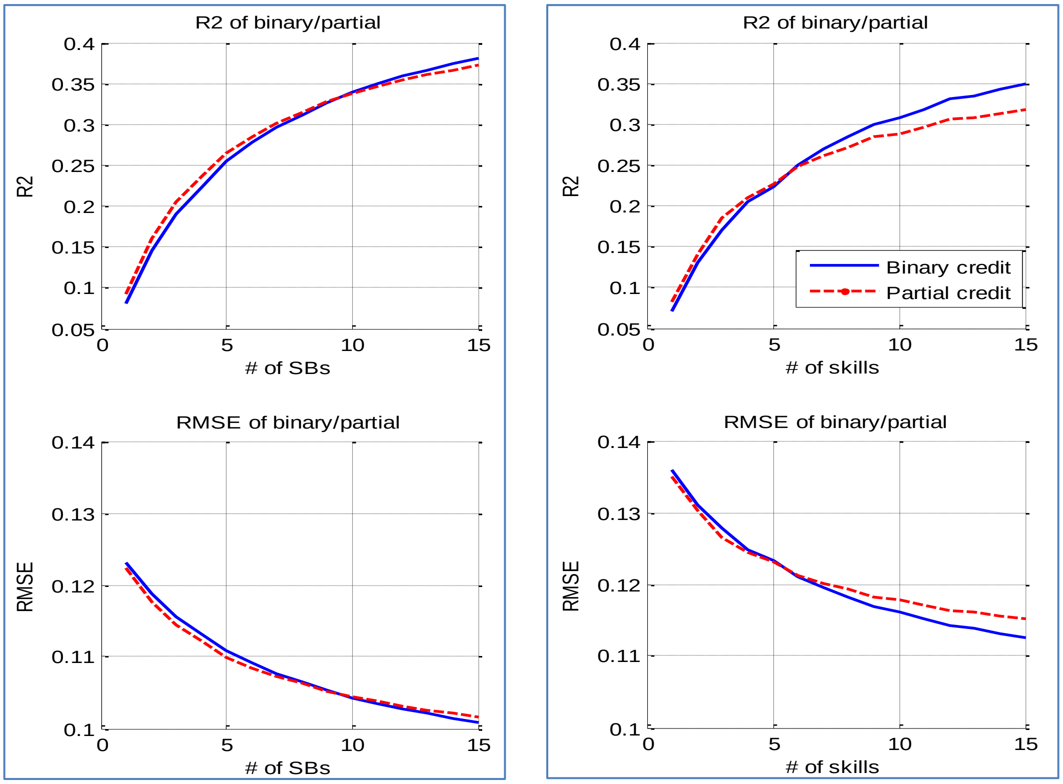


Figure 3. R^2 (top) and RMSE (bottom) of predictions of student knowledge considered using different numbers of ASSISTments Skill Builders (Left) and Cognitive Tutor skills (Right) through a resampling (with replacement) process.

Type I vs. Type II Error Tradeoff

To mirror the validation check in previous work [19], a final analysis was conducted to verify that the observed reduction in Type II error made possible by partial credit (i.e., smaller sample sizes required to differentiate between discretized groups) was not linked to an increase in Type I error. When group differences are not actually significant, maintaining a significance threshold of p < .05, Type I error should equal 5% (i.e., the alpha value). To verify this concept for each skill, null trials were simulated by randomly selecting students (disregarding performance level) to establish homogenous groups of students with no expected significant difference. P-values were collected from 5,000 trials for each scoring metric, following the resampling methodology presented in Section 3.3.

Example skills from ASSISTments and Cognitive Tutor are shown with the percentage of trials claiming significantly different samples charted in Figure 4. The metrics show similar and nondescript noise around the alpha value, suggesting that while partial credit allows for more efficient group differentiation, it does not significantly inflate Type I error, as observed in previous work [19].

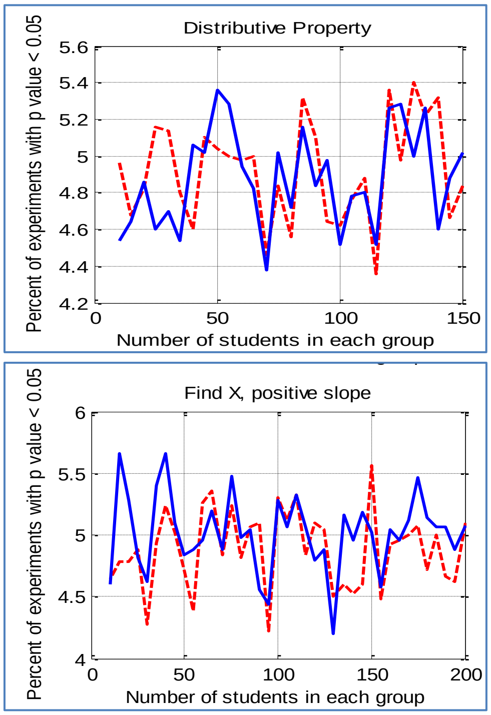


Figure 4. Type I error within an ASSISTments skill (Top) and a Cognitive Tutor skill (Bottom) using Binary Scoring (Blue) and Partial Credit (Red). These two measures show natural noise around the alpha value, α = 0.05, suggesting that while partial credit typically allows for more robust group differentiation, it does not significantly influence Type I error.

DISCUSSION

This work sought to extend previous research on the efficiency and reliability of partial credit when used for group differentiation [19]. Using datasets from ASSISTments and Cognitive Tutor – Algebra, algorithmically defined partial credit was compared to traditional binary scoring when detecting significant differences between discretized groups of student performance levels. A resampling method was used to determine the sample sizes required to reach a threshold at which 90% of trials would report high performing and low performing students was significantly different (p < .05). This method was employed across six skills per platform in an attempt to determine if the magnitude of observed benefits for partial credit scoring was correlated with skill difficulty. In eleven out of twelve trials, partial credit proved more efficient than binary scoring, requiring smaller samples to reach reliably significant group differentiation. These findings were mediated by skill content but did not appear to be directly linked to the difficulty of skills.

It is possible that although partial credit scoring allowed for more efficient group differentiation in the majority of cases, the algorithm behind the metric could be improved to enhance the magnitude of this effect even further. For instance, previous work has shown that while definitions of attempt penalization within partial credit algorithms are more sensitive than definitions of hint penalizations [8], attempts do not necessarily help to significantly differentiate between groups [10]. Thus, the variables that combine algorithmically to form partial credit may be critical to the scoring process while not as important in practice. Further, it should be noted that the definition of partial credit presented herein was originally conceived for data mining within ASSISTments dataset. Generalizability to Cognitive Tutor data was not perfect, with the metric showing success in only five out of six skills. This may suggest that definitions of partial credit are somewhat system specific and should be tweaked to adequately suit other systems.

It is also important to note that regardless of scoring metric, the threshold for reliable group differentiation was achieved in all skills with 182.3 students on average (SD = 231.3), using only the first three data points for each student. While this is likely due in part to the distinct nature of groups split by performance level, it also speaks to the validity of using fewer items enriched with assessment variables in situations like posttests.

In experimental data such as that investigated in [19], partial credit scoring has clear potential to reduce the cost of running randomized controlled trials. However, the present work suggests that the benefits of partial credit extend to the EDM community. At scale, partial credit could be used to reduce the processing time required for building individualized learner models that attempt to predict student performance or proficiency. In any realm, minimization of the number of items required to observe significant effects translates to saved money and saved time.

LIMITATIONS & FUTURE WORK

As touched on in [19], a known limitation of this work is that there are mathematically possible situations in which partial credit can underperform binary scoring. This is another possible explanation for why binary credit was more efficient at differentiating between student performance levels in the context of the Cognitive Tutor skill “Combine Like Terms.” T-tests result in greater significance when homogenous groups have large mean differences. As partial credit makes groups appear more homogenous by reducing within group variance while simultaneously adjusting group means, higher efficiency in group differentiation may be attained by binary scores in skewed datasets. For instance, Table 8 examines two examples in which between-group (A & B) comparisons of scoring metrics are assessed using independent samples t-tests. Example 1 looks quite similar to the findings for eleven out of twelve skills in the present work, while Example 2 reveals a scenario

Table 8. The potential for Partial Credit to outperform Binary Scoring (Example 1) and the reverse (Example 2)

Example 1				Example 2			
Binary Scoring		Partial Credit		Binary Scoring		Partial Credit	
A	B	A	B	A	B	A	B
1	1	1	1	1	1	1	1
1	0	1	0.1	1	0	1	0.6
1	1	1	1	1	1	1	1
0	0	0.5	0.3	0	0	0.2	0.3
0	0	0.8	0.2	0	0	0.5	0.2
0	0	0.3	0.8	0	0	0.3	0.8
t = 0.53 p = 0.60		t = 0.96 p = 0.36		t = 0.53 p = 0.60		t = 0.08 p = 0.94	

much like that for the twelfth skill in which binary scoring outperforms partial credit, resulting in a lower p-value.

As noted in the Discussion, it is also possible that algorithmically defined partial credit may be highly system specific and may not generalize with strong validity. Future work should examine the sensitivity of such definitions and how generalizability can be improved. Future work should also assess potential avenues for using group differentiation within learner models to predict student mastery (i.e., groups that will reach mastery vs. those that will not). Implications for learner modeling suggest that the resampling approach presented herein could be used for successful latent group differentiation, which may enhance or even outperform techniques like Knowledge Tracing

CONTRIBUTIONS

The work presented herein extended previous research detailing the benefits of partial credit scoring within Intelligent Tutoring Systems and online learning platforms, in the context of enhanced efficiency and reliability when differentiating between user groups. This work extended a previously established resampling approach to consider group differentiation using partial credit in broader skill contexts and across platforms. It is possible that this approach could be applied to EDM practices to reduce sample sizes or the number of items required to build learner models that reliably detect skill mastery.

ACKNOWLEDGMENTS

We acknowledge funding from multiple NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736 & DRL-1031398), the U.S. Department of Education (IES R305A120125 & R305C100024 and GAANN), the ONR, and the Gates Foundation. Thanks to S.O. & L.P.B.O.

REFERENCES

[1] Anderson, J.R., Corbett, A.T., Koedinger, K.R., & Pelletier, R. 1995. Cognitive tutor: Lesson learned. The journal of the learning sciences. 4 (2): 167–207.

[2] Carnegie Learning. 2016. Cognitive Tutor Software. Carnegie Learning, Inc. Retrieved from https://www.carnegielearning.com/learning-solutions/software/cognitive-tutor/

[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] Desmarais, M.C. & Baker, R.S.J.d. 2011. A Review of Recent Advances in Learner and Skill Modeling in Intelligent Learning Environments. User Modeling and User-Adapted Interaction. 22 (1-2): 9-38.

[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. International Journal of AIED. 24 (4): 470-497.

[6] KDD Cup. 2010. Rules of the KDD Cup 2010: Educational Data Mining Challenge. PSLC DataShop. Retrieved from https://pslcdatashop.web.cmu.edu/KDDCup/rules.jsp

[7] MATLAB version R.2013.a (2013). Natick, Massachusetts: The MathWorks, Inc. Accessible at www.mathworks.com

[8] Ostrow, K., Donnelly, C., & Heffernan, N. (2015). Optimizing Partial Credit Algorithms to Predict Student Performance. In Santos, et al. (eds.) Proc of the 8th Int Conf on EDM. 404-407.

[9] Ostrow, K., Donnelly, C., Adjei, S., & Heffernan, N. (2015). Improving Student Modeling Through Partial Credit and Problem Difficulty. In Russell, et al. (eds.) Proc of the 2nd ACM Conf on L@S. 11-20.

[10] Ostrow, K., Heffernan, N., Heffernan, C., Peterson, Z. (2015). Blocking vs. Interleaving: Examining Single-Session Effects within Middle School Math Homework. In Conati, et al. (eds.) Proc of the 17th Int Conf on AIED. 388-347.

[11] Pardos, Z.A. & Heffernan, N.T. 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Bra, et al. (eds.) Proc of the 18th Int Conf on UMAP. 255-266.

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

[13] Ritter, S., Anderson, J.R., Koedinger, K.R., & Corbett, A. 2007. Cognitive Tutor: Applied research in mathematics education. Psychonomic Bulletin & Review. 14 (2): 249-255.

[14] Stamper, J.C., Koedinger, K.R., Baker, R.S.J.d., Skogsholm, A., Leber, B., Demi, S., Yu, S., & Spencer, D. 2011. DataShop: A Data Repository and Analysis Service for the Learning Science Community. In Biswas et al. (eds.) Proc of the 15th Int Conf on AIED.

[15] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. 2005. The Andes Physics Tutoring System:

Lessons Learned. International Journal of AIED. 15 (3): 147-204.

[16] Author1. (2015). Data and Code for ‘Partial Credit Revisited: Enhancing the Efficiency and Reliability of Group Differentiation at Scale.” Accessed from http://tiny.cc/EDM2016-PartialCredit

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

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

[19] Wang, Y., Ostrow, K., Beck, J., & Heffernan, N. In Press. Enhancing the Efficiency and Reliability of Group Differentiation through Partial Credit. To be included in Proc of the 6th Int Conf on LAK (2016).

[20] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. 2013. Individualized Bayesian Knowledge Tracing Models. In Lane, et al. (eds.) Proc of the 16th Int Conf on AIED. 171-180.

How Long Must We Spin Our Wheels?
Analysis of Student Time and Classifier Inaccuracy

Yue Gong
Facebook
Facebook 1 Hacker Way
Menlo Park, CA, 94025
yuegong@fb.com

Yan Wang, Joseph Beck
Worcester Polytechnic Institute
Worcester, MA 01609
{ywang14, josephbeck} @wpi.edu

This section has been submitted to:

Gong, Y., Wang, Y., & Beck, J. (2016). How Long Must We Spin Our Wheels? Analysis of Student Time and Classifier Inaccuracy. *The 9th International Conference on Educational Data Mining.* ACM.

ABSTRACT

Wheel-spinning is the phenomenon where students, in spite of repeated practice, make no progress towards mastering a skill. Prior research has shown that a considerable number of students can get stuck in the mastery learning cycle--unable to master the skill despite the affordances of the educational software. In such situations, the tutor's promise of "infinite practice" via mastery learning becomes more a curse than a blessing. Prior research on wheel spinning overlooks two aspects: how much time is spent wheel spinning and the problem of imbalanced data. This work provides an estimate of the amount of time students spend wheel spinning. A first-cut approximation is that 24% of student time in the ASSISTments system is spent wheel spinning. However, the data used to train the wheel spinning model were imbalanced, resulting in a bias in the model's predictions causing it to undercount wheel spinning. We identify this misprediction as an issue for model extrapolation as a general issue within EDM, provide an algebraic workaround to modify the detector's predictions to better accord to reality, and show that students spend approximately 28% of their time wheel spinning in ASSISTments.

Keywords

Wheel-spinning; Precision; Recall; Intelligent Tutoring Systems

INTRODUCTION

Mastery learning has been implemented and applied in intelligent tutoring systems (ITS) in a variety of contexts. One common foundation builds on the ACT-R theory, which assumes that procedural knowledge of a skill can be acquired through repeated problem solving of what is initially declarative knowledge, causing it to compile into production rules for a procedural representation [1]. The rationale of mastery learning is also well supported by the theory of "learning-by-doing," which refers to the capability of learners to improve their efficiency by regularly repeating the same type of action via practice [2]. The use of mastery learning is driven by the desire to provide students efficient practice, by avoiding giving them too many problems to solve, which could waste valuable learning time [3] and possibly jeopardize student motivation to learn, but simultaneously ensuring there are not too few practice problems, which might leave students poorly prepared for learning future content [4] due to the lack of mastery.

An application of mastery learning is that students are presented as many problems as needed to master the skill. Consequently, the system keeps giving the student more problems to practice in the hope that he might utilize these new opportunities to master the skill. The student however could keep failing to learn the skill, which triggers the system to present even more problems to the student. Thus, the student can possibly become trapped in the mastery learning cycle if he fails to achieve mastery. We term this phenomenon "wheel-spinning", analogous to a car stuck in mud or snow; its wheels are spinning rapidly and there is the illusion of progress, but it is not going anywhere. Similarly, the tutor is presenting students with many problems to solve and there is the appearance of productive work, but the students are not making progress towards mastery.

Prior work [5] introduced the concept of wheel spinning, which describes the phenomena that students can not master a skill in a timely manner. Using data from two ITS called the Cognitive Algebra Tutor [13] and ASSISTments [14], they analyzed the severity of wheel-spinning, and build a logistic model to predict students wheel spinning. In general, the model provided good prediction accuracy with an AUC of 0.88 [6]. However, since the model was trained based on imbalanced data (most students master a skill rather than wheel spinning), the model has high false negative rate, which means wheel spinning cases are relatively more likely to be mispredicted as mastery cases. Therefore, when we apply this model to indeterminate cases (which we can not label wheel spinning or mastery based on the given data), the estimated rate of wheel spinning is likely an undercount. This paper addresses the undercount, and further estimates how much time students spend wheel spinning.

DATA SET

In this paper, we used the similar data set used in [6] from ASSISTments. ASSISTments is a web-based computer tutor, primarily used for middle-school math education (approximate ages 12 to 15). This data set contains information from 5997 students chosen at random, who used ASSISTments during the time period of September 2010 to July 2011. The students completed a total of 208,328 math problems during this time period. These students were primarily from the northeast United States. We have student self-reported ages, and 75% of the students asserted they were 12 to 15 years of age on January 1, 2011. Since the students spread across a wide range of grades, they solved problems including a large range of skills as well. The problems cover 190 math skills, such as Equation-Solving-More-

Than-Two-Steps, Area-Irregular-Figure, etc. Since we have access to the ASSISTments system's database, we can reach fine-grained information, such as every action the student made while he was solving the problem. This allows us to analyze the relationship between wheel-spinning and non-productive "learning" behaviors induced by these fine-grained data.

This work retains the initial definition of wheel spinning [5] of failing to master a skill within 10 practice opportunities. We define mastery as getting three problems correct in a row. This threshold of mastery is rather low, and so these results are a lower bound on wheel spinning. Some students practiced fewer than 10 problems without reaching mastery. It is not obvious whether these students would master the skill or not, and we categorize them as "indeterminate." Table 1 shows the number of student-skill pairs in each category.

Note that a student could wheel spin on adding fractions but master multiplying decimals. Therefore, we speak of wheel spinning or mastering a particular skill by a student. Thus, when characterizing the amount of wheel spinning, our analysis is in terms of student-skill pairs.

Table 4. Breakdown of student performance by mastery type

Category	Mastery	Indeterminate	Wheel-spin
Number of student-skill pairs	25449 (55.6%)	17528 (38.3%)	2810 (6.1%)

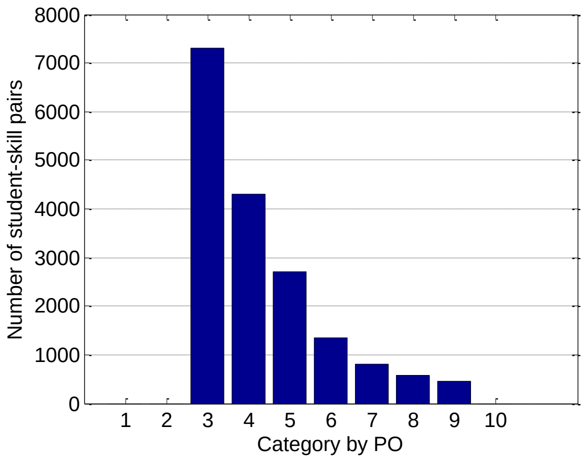


Figure 3. Number of indeterminate student-skill pairs at each PO

Since wheel spinning is trivial to predict for cases where we can observe either wheel spinning or mastery, we are more interested in the distribution of indeterminate cases. Figure 1 shows frequencies of student-skill pairs at a certain number of practice opportunities of indeterminate cases. Clearly, the larger the PO is, the fewer observations we have. Students in the indeterminate group tend to have fewer PO; the majority of students did no more than 5 problems. It is interesting that students seem to give up relatively rapidly on a problem set.

Overall, there is a large imbalance of more mastery cases than wheel spinning cases. However, this imbalance interacts with the number of practice opportunities (PO) a student has had on a skill, as shown in Figure 2. The number of student-skill pairs considered wheel-spinning does not change with PO, since by definition a student must reach PO 10 in order to be categorized as wheel spinning. The reason the number of wheel spinning cases is constant is that when we observe a sequence as either wheel spinning or mastery, we label all PO in the sequences with that label. Since 10 PO are required for wheel spinning, all 10 bins have the same quantity. However, students can master a skill after 3 PO. Therefore, the number of student-skill pairs still working towards mastery decreases rapidly as PO increases.

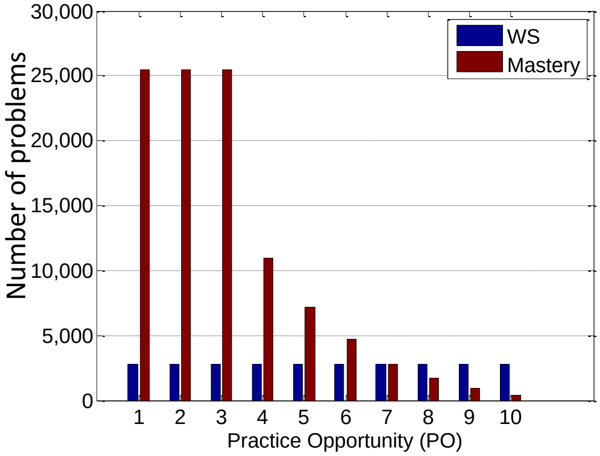


Figure 4. Number of wheel spinning and mastery problems at each PO

REVISIT THE WHEEL SPINNING PREDICTIVE MODEL

Model Performance Metrics

In this paper, we reused the model provided in [7]. The model is trained based on determinate cases (mastery and wheel-spinning cases), and then it is applied to indeterminate cases to make predictions and estimate the rate of wheel spinning. The model was trained using three fold cross validation. This model has strong performance statistics on the test set of unseen students: R2 of 0.4 and AUC of 0.88. However, its precision and recall are reasonable but less strong: 0.76 and 0.53, respectively. We now develop an argument to show as a consequence of the precision and recall statistics, the predictive model undercounts the amount of wheel spinning on the indeterminate cases.

Evaluation of the Model with Precision and Recall

In a classification model, the precision of a model, P, is the number of true positives, TP, divided by the total number of cases predicted as positive, PP. A model's recall, R, is the number of true positives divided by the total number of cases that are actually positive, +. As a consequence, we have the formulas

P = TP / PP (1)

R = TP / + (2)

A model's precision is how selective it is. When it predicts the category will occur, how often is it right? Recall measures how comprehensive a classifier is. Of the actual cases, how many can it detect? Clearly, there is trade off between precision and recall. A classifier could be very cautious and only make a positive prediction when it was very certain, resulting in a high precision but low recall. Conversely, a classifier could categorize everything as an instance of the category, achieving perfect recall but (presumably) low precision. The precision and recall of wheel spinning and mastery are shown in Table 2.

Table 5. Precision and recall for Mastery and wheel spinning

Category	Mastery	Wheel spinning
Precision	88.3%	75.6%
Recall	95.3%	52.5%

This model has a high precision and recall for predicting mastery. However, the precision and recall of wheel spinning is relatively low. Wheel spinning's precision of 75.6% means that about one out of four of the cases that is predicted as wheel spinning is actually mastery. Recall of 52.5% means that the model can only capture successfully about half of the WS cases.

The low recall of WS is not surprising if we look at the distribution of data set shown in Table 1. Mastery cases occupy a large portion. Under such a circumstance, it is understandable that the model tends to predict cases as mastery to reduce the prediction error—the goal of the model fitting process.

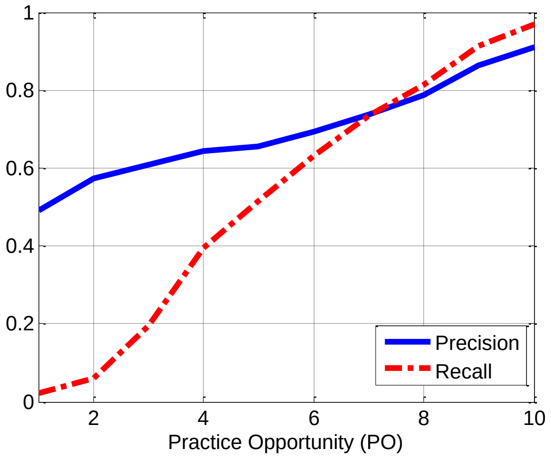


Figure 5. Precision and recall for Wheel Spinning prediction

More specifically, we analyzed precision and recall at different POs. Figure 3 and Figure 4 show the precision and recall of wheel-spinning and mastery of the wheel spinning prediction model, both disaggregated by PO. Interestingly, precision and recall both improve for problems in the wheel spinning category as the model observes the student making more practice opportunities on the skill. This explanation makes intuitive sense: as the model acquires more data, it is better able to detect when a student will wheel spin. Interestingly, precision and recall of the Mastery category both decrease with additional observations of the student performing the skill. At first, this situation seems paradoxical, until one consider the distribution of Mastery vs. Wheel Spinning in Figure 2. Initially, Mastery is the majority class. Its relative advantage begins to slip after PO 3, and by PO 7 it has achieved numerical parity with Wheel Spinning. After PO 7, Wheel Spinning is the majority class. As Mastery becomes less and less dominant in the data set, its predictive accuracy decreases.

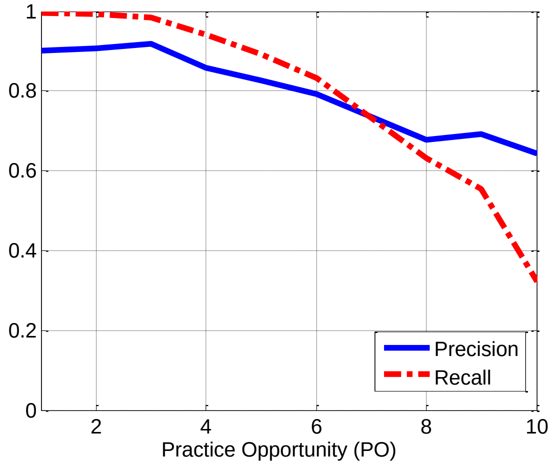


Figure 6. Precision and recall for Mastery prediction

Implications of imbalances in classifier accuracy

Consider the relationship between the precision and recall and the number of true positives. From a standpoint of precision, the number predictions made multiplied by the precision is equal to the number of correct predictions. That is:

P * **PP** = **TP** (3)

Conversely, we can define the number of true positives using recall. Specifically, the number of actual occurrences of a category, multiplied by the model's recall, provides the number correct predictions of that category. That is:

R * + = **TP** (4)

Since equations 3 and 4 both have the number of true positives on their right-hand side, we can set them equal to each other:

R * + = **P** * **PP** (5)

Dividing both sides by R and rearranging we get:

+ = **PP** * (**P** / **R**) (6)

In other words, the number of positive examples in a data set is equal to the number of predicted positives, multiplied by the precision over recall. A few points of discussion. First, it may seem conceptually odd to need to compute the number of positive examples in a data set, as it is normally countable directly from the data. However, for our problem we have a large number of indeterminate cases where we are unable to observe what their true label would be, and we need to infer it. More broadly, applying behavioral classifiers outside of the labeled training data encounters this same problem: how many instances are there really in the data set? Such a situation would arise when attempting to apply a model trained on one system to a second system. The second observation is that the (P/R) term in Equation 6 can be thought of as a normalizing constant for reweighting the data. The number of instances predicted to be positive is adjusted by P/R. Sometimes this adjustment will increase the number of instances and other times it will decrease the number of instances. In either case, *this adjusted number of instances is a better estimate of the number of positive examples in the data than the number of predicted positives from the classifier*.

An intuitive way to reweight the prediction results is to directly use the precision and recall ratios shown in **Table 5** to compute the P/R ratio. However, we have additional information in that we know the relative counts of Wheel Spinning and Mastery change dramatically with PO. Therefore, rather than applying a global reweighting term of 0.756/0.525 for Wheel Spinning and 0.883/0.953 for Mastery, we instead create more fine-grained reweightings based on PO.

Figure 5 shows the P/R ratio for both categories broken down by PO. Note that for a low number of PO, the P/R ratio for wheel spinning is noticeably higher than 1. In other words, early on in the sequence many wheel spinning cases are miscategorized as Mastery by the classifier, and there is a systematic undercount in the number of Wheel Spinning students. In contrast, the Mastery category has a P/R ratio of approximately 1.0 throughout its range, only rising noticeably above 1.0 on PO 9 and 10. Thus, Mastery cases are undercounted late in the sequence of problem solving.

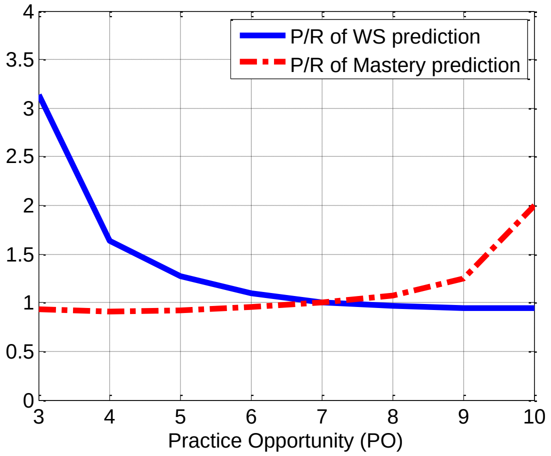


Figure 7. Ratio of precision/recall for Wheel Spinning and Mastery

REANALYSIS OF PREDICTION RESULTS FOR COMPUTING AMOUNT OF WHEEL SPINNING

We now turn our attention to first reestimating past results using the reweighted data. Then we focus on estimating the time spent wheel spinning using both the straightforward approach of using the classifier results as-is (i.e., the PP value) vs. using the reweighted PP * (P/R) value.

Estimating amount of mastery

By applying the predictive model to indeterminate cases, we can get predicted category of these cases. Since the ratio of precision and recall is not very large for Mastery prediction, the modification of those predictions is generally a small decrement. However, for Wheel Spinning predictions, the P/R ratio is generally higher than 1, causing an increase in the number of predicted cases of Wheel Spinning.

Figure 6 shows the number of indeterminate student-skill pairs predicted to result in Mastery. For each PO, the bar on the left represents the number of cases that will result in Mastery originally predicted by the model. The bar on the right for each PO represents the adjusted count by reweighting each student-skill pair by its corresponding P/R ratio. For problems at PO 3, the P/R ratio for Wheel Spinning predictions was over 3, so those cases are weighted 3 times as heavily. For Mastery problems, the P/R ratio was just under 1.0, so those counts are relatively unchanged. As a result of this reweighting, there is a noticeable drop in the estimated number of indeterminate students who will master the skill after 3 PO.

For PO3 through PO6, the reweighting is pessimistic and causes more student-skill pairs to be categorized as Wheel Spinning than the model predicts on its own. At PO 7, both categories have a P/R ratio of approximately 1.0, so the counts are (roughly) unchanged. For PO 8 and 9, since the P/R ratio of Mastery is larger than for Wheel Spinning, we see an increase in the expected number of students who Master the skill relative to the model's predictions.

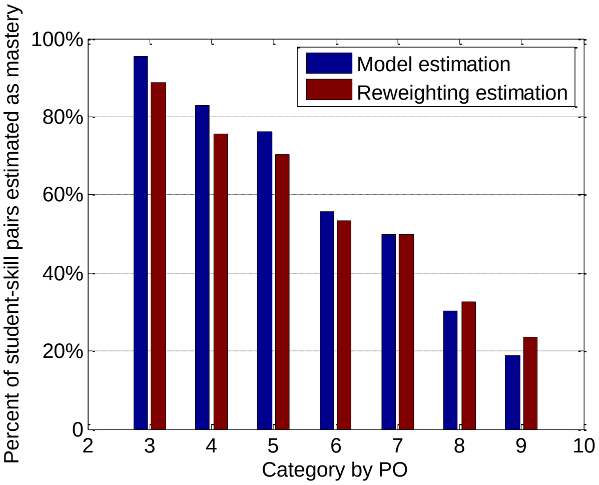


Figure 8. Original and reweighted proportions of student-skill pairs predicted as resulting in mastery

We now compute the cumulative percent of students who will master a skill, by assigning the indeterminate student-skill pairs to either Mastery or Wheel Spinning. Figure 7 shows the result of this process. The upper and lower lines are optimistic and pessimistic assumptions of student performance, and provide an absolute upper- and lower-bound on the percentage of student-skill pairs that will result in mastery. The upper-bound on mastery assumes all indeterminate students will master the skill. The lower-bound assumes all students will wheel spin. Our goal is to better estimate mastery within that range of possible values. The solid green line in the middle of the graph is the result of applying the model's predictions to the indeterminate data points (identical to the analysis in [6]). The dashed red line represents using the same model predictions, but reweighting them according to the P/R ratio provided in Figure 5. For example, if an indeterminate case was predicted as resulting in Wheel Spinning, we would count that as approximately 1.6 observations of Wheel Spinning, as that is the P/R ratio for that category for that number of practice opportunities. Overall, there is not a large change in the expected proportion of students-skill pairs reaching mastery. There is a slight decrease of 2% absolute in the expected amount of mastery, with about 16% (shown in Figure 7) of student-skill pairs expected to exhibit Wheel Spinning.

As another illustration of the impact of weighting the model's output, Table 3 shows the impact on the number of indeterminate cases counted as mastery or as wheel spinning. Note that no student-skill pair actually receives a different prediction as a result of the modification, the counts in the table change strictly as a result of reweighting the counts by P/R. Although we are able to obtain more accurate counts, we are not able to more accurately predict any individual case as Wheel Spinning or Mastery. Note that the percentage of mastery in Table 3 (75%) differs from Figure 7 (84%) since Figure 7 refers to wheel spinning, mastery, and indeterminate cases, while Table 3 zooms in and considers only the indeterminate cases.

Table 6. Estimated number (percent) of indeterminate student-skill pairs predicted as each category

Category	Mastery	WS
Estimation	14028 (80%)	3500 (20%)
Modified Estimation	13086 (75%)	4442 (25%)

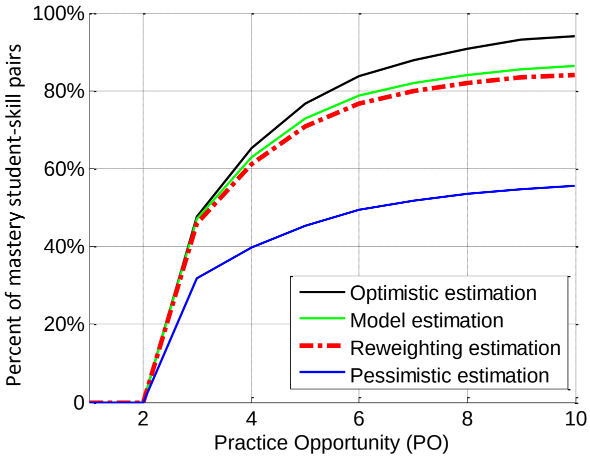


Figure 9. Cumulative percent of student mastering the skill by PO. Model estimate and reweighted estimate.

Estimating time spent wheel spinning

Our final analysis is to estimate the amount of time students spend in the wheel spinning state. First, we examined how long students spent solving a problem. Figure 8 shows the average number of seconds students spent on a problem, broken down by category (observed mastery, observed wheel spinning, or indeterminate), and plotted by PO. Several trends are evident. First, problems solved in skills where the student will wheel spin take approximately 25% longer to solve than problems solved in skills that the student eventually masters. The other observation is that there is a sharp drop in time to solve a problem from PO 1 to PO 2, presumably due to memory effects as students swap into working memory [7] the necessary procedures for solving problems of this type. After PO2, there is a slight decreasing trend in time spent per problem across indeterminate, mastery, and wheel spinning student-skill pairs. This interaction of time and PO illustrates the importance of using a P/R ratio conditioned by PO, as shown in Figure 5, as early values of PO, where the P/R ratio is greatest, take the greatest amount of time to solve.

The other thing to note is that wheel spinning students spend much longer on skills than students who master. First, wheel spinning students spend more time per problem (Figure 8). Second, wheel spinning students attempt many more problems on a skill than students who master it. Observed wheel spinning requires 10 observations. So we should expect the time spent wheel spinning to be substantially higher than the 16%, which the percent of student-skill pairs observed to exhibit wheel spinning.

To compute time spent wheel spinning, we treated student-skill pairs that resulted in either wheel spinning or mastery as time spent in the respective state. For indeterminate sequences, we compute the probability of Wheel Spinning according to the model for the last problem in the sequence. Presumably the final PO has the most information, and provides the best estimate of whether the student will wheel spin or not. We then use the P/R reweighting term for the final PO to reweight time spent in all of the problems for this student-skill pair. This approach maximizes information used in making the prediction, and uses the P/R ratio that is associated with that model's prediction. So if a student reaches PO 6 and is predicted to wheel spin, we use a ratio of approximately 1.1 (from Figure 5) to reweight the time spent in all 6 POs, and do not artificially inflate the time by using the P/R ratio from PO 1 through 5 for this student-skill pair.

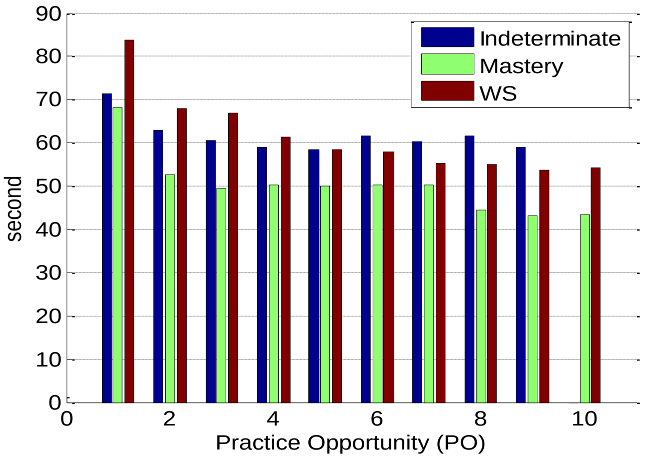


Figure 10. Average time spent on a problem

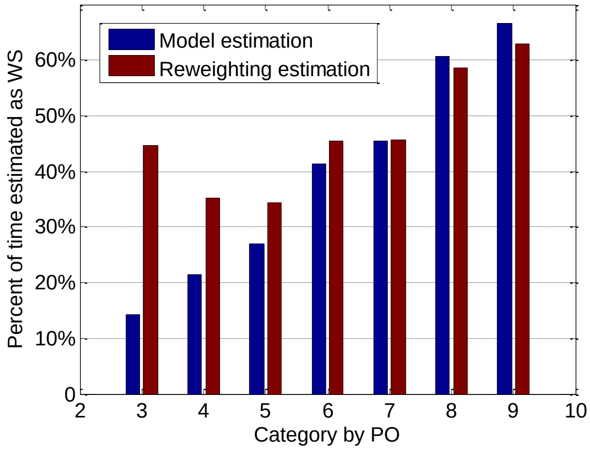


Figure 11. Estimated time spent wheel spinning for indeterminate cases

Figure 9 provides the amount of time students spend wheel spinning, broken down by PO. At PO 3, the reweighting results in a sharp increase in the amount of time estimated as spent wheel spinning. As the P/R ratio becomes closer to 1, the reweighted counts and model predictions become more similar to each other.

Table 7. Estimated time (in number of hours and as a percentage) spent Mastering and Wheel Spinning

Category	Optimistic	Model estimated	Reweighting data	Pessimistic
Mastery	2239	2035	1921	1481

	(84%)	(76%)	(72%)	(56%)
WS	422 (16%)	626 (24%)	740 (28%)	1181 (44%)

After reweighting the predicted amount of time spent Wheel Spinning or Mastering for each student-skill pair, we computed the total amount of time spent wheel spinning. Table 4 shows the time spent wheel spinning and mastering for our data set. Using the model's predictions as-is, we get that students spent 626 hours wheel spinning, or 24% of their time. Reweighting the data results in that amount increasing to 740 hours, or 28% of their time. Finding that over 600 hours of student time was wasted over a year is not a comforting thought. Using the reweighted estimate, over one-quarter of student time is spent in the wheel spinning state. This value is not a small number, and should be a focus of attention for improving the tutor.

CONTRIBUTIONS

This paper makes contributions to understanding wheel spinning and more broadly to the field of educational data mining. Within the context of wheel spinning, this paper extends prior work on estimating the amount of wheel spinning [6]. Given the prevalence and breadth of wheel spinning, approximately 26% in the Cognitive Algebra Tutor, 16% student-skill pairs in ASSISTments, and over one-third in a study of the cognitive tutor on a non-WEIRD population [8], efforts to better understand wheel spinning can have a broader impact than on other constructs commonly studied which are typically observed on many fewer students. Prior research [5, 6, 7] examined the total number of student-skill pairs that exhibit wheel spinning. Such analysis is informative, but neglects to consider the amount of time student spend spinning their wheels in the mastery learning cycle. The amount of time is particularly relevant given that problems where students are wheel spinning take somewhat longer to complete. Furthermore, students perform more problems in wheel spinning sequences than in sequences that end in mastery. Consequently, students in ASSISTments wheel spin on 16% of problem sequences, but spend 28% of their time in the wheel spinning state. The 28% would be even worse, except that some students who are likely to wheel spin stop doing the tutor's exercises and give up on the problem set. Realizing that much student time is being wasted by a commonly used computer tutor is surprising, and such analysis of time is rarely done, with a few exceptions [8].

The second contribution this paper makes is refining the understanding of a classifier for wheel spinning, and by extension, other classifiers used in educational data mining. The precision, recall, and AUC of the previously published predictive model of wheel spinning are quite good. However, looking at the performance in detail indicates there are systematic biases in its predictions, which should lead us to be cautious in interpreting its results.

The final contribution of this paper is in an interesting approach of correcting for imbalanced data in a classifier. The classifier is doing a good job for its role: minimize its prediction error, possibly extended with an asymmetric loss function to penalize certain types of mistakes more heavily. The classifier's job is not to make the most accurate extrapolation at a coarse grain size by correctly estimating the total number of times a certain behavior occurs. As a result, when a classifier is used to extrapolate to a new dataset and estimate the rate of occurrence of a phenomenon, there is a mismatch between that mission and its goal. As a simple example, for a problem with 99% positive examples, a very accurate classifier would categorize all examples as positive. It would not, however, be useful for extrapolating population statistics as it would claim that 100% of the data were positive examples when we know that is not true. Although we know the classifier is overpredicting the majority class, we are not sure *which specific instances* are being overcounted.

This work provides a means for reweighting the data to cause the classifier to better-align its predictions with known counts in the data. We are able to perform this reweighting by taking advantage of the relationship between precision, recall, and the known base rates. In addition, we leverage the strong relation between practice opportunity and precision/recall. Consequently, we are able to make better predictions about collections of data points, and better allocate student time between wheel spinning and mastery states. However, this algebraic trick does not allow to modify our prediction about any specific student-skill pair and increase the classification accuracy of the detector. This apparent conundrum, and separation of the roles of behavioral models into predictions of individuals and categorizing large numbers of trials is a contribution to the field of educational data mining[1]: simply extrapolating model predictions can lead to erroneous claims about the amount of a behavior or the time spent in that behavior. In fairness, the change for this study was moderate in scope: the amount of time spent wheel spinning is approximately 28% of total time rather than 24%. However, for detectors with weaker performance metrics, this difference could be much larger.

FUTURE WORK AND CONCLUSIONS

The most obvious line of future work is the creation of a stronger classifier for wheel spinning, as well as for other detectors of learner behavior and affect. The wheel spinning detector has strong performance metrics (on test-set data): AUC of 0.88, R2 of 0.4, precision of 0.76 and recall of 0.53 [6]. In spite of those solid metrics, there is a notable problem with extrapolation due to the skew between precision and recall. A naïve approach would be to simply alter the loss function [10] to balance precision and recall. However, this approach would reduce the predictive accuracy of the model, its sine qua non. Also, some algorithms in AI domain also provides possible solutions [11, 12], but those approaches modify the classifier's predictions, so there is a loss in accuracy of predictions. On the other hand, semi-supervised learning is also a technique we would like to try in the future. [15]

The second area is to analyze whether student characters that influence wheel-spinning between determinate cases and indeterminate cases are similar. In this paper, we assume that the model built on determinate cases also applies to indeterminate cases. However, whether this assumption holds should be validated. In the future, more data (previous information) about students in both determinate cases and indeterminate cases should be gathered, and analyzed for comparison of similarity between the two groups.

[1] We suspect we are not the first to reweight our data in this manner, but none of us are experts in information retrieval. The second author of the paper developed the idea independently while thinking about the classifier's performance metrics, and the first author developed an explanation for this paper and did a quick literature search to no avail. We would appreciate any pointers to the literature of making use of this approach to enable a model to better extrapolate.

The third area of research is to reduce the amount of time spent wheel spinning. Wheel spinning consumes a large amount of student time, typically in a block spent working on a particular topic. Beyond being ineffective for learning, it is presumably disengaging for learners as well. The problem is that most obvious interventions have been tried, as ITS designers attempt to construct systems from which students can learn. Analysis of how much wheel spinning could be reduced by ensuring students understood their prerequisite skills reveals a modest decrease [10]. Thus, there is a need for effective strategies for reducing wheel spinning. One possible strategy is a strong detector capable of quickly detecting that a student is likely to wheel spin, and simply stop providing her/him problems on the topic. This creation of an escape mechanism from the mastery learning cycle would reduce time spent wheel spinning, and couple with instruction by a human teacher or tutor, could possibly be an effective intervention.

The fourth area of future work is further thinking about the different uses of predictive models. This work examines two: predicting individual cases and extrapolating the model to an aggregate group, and identifies an issue with undercounting the minority class for analyzing the impact of a behavior. Are there other crucial differences between these two uses beyond the one noted in this paper? Is there a third type of use of models that has different properties entirely?

In conclusion, this paper extends what is known about wheel spinning. We have found that students spend approximately 28% of their time in a wheel spinning state. More interesting is how we calculated this number: reweighting the data to modify the impact of the model’s predictions. Thus, this paper not only extends our understanding of the common and detrimental behavior of wheel spinning, but improves our methodological sophistication for understanding behavioral detectors.

ACKNOWLEDGMENTS

We acknowledge funding from NSF (# 1440753, 1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024) grant for ASSISTments and support of the author.

REFERENCES

[1] Anderson, J.R., et al. 1995. Cognitive tutors: Lessons learned. The Journal of the Learning Sciences (April, 1995), 167-207.

[2] Wikipedia. Learning-by-doing (economics). Available from: http://en.wikipedia.org/wiki/Learning-by-doing_(economics) (March, 2016).

[3] Cen, H., Koedinger, K. and Junker. B. 2007. Is More Practice Necessary? - Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. Frontiers in Artificial Intelligence and Applications, 158 (2007), 511.

[4] Baker, R.S., Gowda, S. and Corbett, A. 2011. Automatically Detecting a Students Preparation for Future Learning: Help Use is Key. In Proceedings of the International Conference on Educational Data Mining (Eindhoven, the Netherlands, July 06 - 08, 2011). EDM'11. ACM, New York, NY, 179-188.

[5] Gong, Y. and Beck, J. 2015. Towards Detecting Wheel-Spinning: Future Failure in Mastery Learning, in Proceedings of Conference on Learning @ Scale (Vancouver, Canada, March 14 - 18, 2015). L@S'15. ACM. New York, NY, 67 – 74.

[6] Beck, J. and M.M.T. 2014. Rodrigo, Understanding Wheel Spinning in the Context of Affective Factors, in Intelligent Tutoring Systems (2014), 162-167.

[7] Beck, J.E. and Gong, Y. 2013. Wheel-Spinning: Students Who Fail to Master a Skill, in Proceedings of International Conference on Artificial Intelligence in Education (Memphis, USA, July 09 – 13, 2013). AIED'13. 431-440.

[8] Mostow, J., et al. 2002. A la recherche du temps perdu, or as time goes by: Where does the time go in a Reading Tutor that listens? In Proceedings of the International Conference on Intelligent Tutoring Systems (Biarritz, France, 2002). ITS'2002. 383 – 390.

[9] Mitchell, T. 1997. Machine Learning. McGraw-Hill. 432.

[10] Wan, H. and Beck, J. B. 2015. Considering the influence of prerequisite performance on wheel spinning. In Proceedings of the International Conference on Educational Data Mining (Madrid, Spain, July 26 - 29, 2015). EDM'15. ACM, New York, NY.

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16 (2002), 321-357.

[12] Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. 2003. SMOTEBoost: Improving prediction of the minority class in boosting. Knowledge Discovery in Databases: PKDD (Sep, 2003), 107-119.

[13] Koedinger, K. R., & Corbett, A. T. 2006. Cognitive tutors: Technology bringing learning science to the classroom. The Cambridge Handbook of the Learning Sciences. Cambridge University Press, Cambridge, MA.

[14] Heffernan, N. T., & Heffernan, C. L. 2014. The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education, 24 (Dec, 2014), 470-497.

[15] Zhu, X., & Goldberg, A. B. 2009. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3 (Jun, 2009), 1-13.

Defining Mastery: Knowledge Tracing Versus N-Consecutive Correct Responses

Kim Kelly, Yan Wang, Tamisha Thompson, Neil Heffernan
Worcester Polytechnic Institute
Worcester, MA 01609
{kkelly, ywang14, tsthompson, nth} @wpi.edu

This section has been published at:

Kelly, K., Wang, Y., Thompson, T., & Heffernan, N. (2015). Defining Mastery: Knowledge Tracing Versus N-Consecutive Correct Responses. *In the proceedings of the 8th International Conference on Educational Data Mining.*ACM*.*

ABSTRACT

Knowledge tracing (KT) is well known for its ability to predict student knowledge. However, some intelligent tutoring systems use a threshold of consecutive correct responses (NCCR) to determine student mastery, and therefore individualize the amount of practice provided to students. The present work uses a data set provided by ASSISTments, an intelligent tutoring system, to determine the accuracy of these methods in detecting mastery. Study I explores mastery as measured by next problem correctness. While KT appears to provide a more stringent threshold for detecting mastery, NCCR is more accurate. An incremental efficiency analysis reveals that a threshold of 3 consecutive correct responses provides adequate practice, especially for students who reach the threshold without making an error. Study II uses a randomized- controlled trial to explore the efficacy of various NCCR thresholds to detect mastery, as defined by performance on a transfer question. Results indicate that higher thresholds of NCCR lead to more accurate predictions of performance on a transfer question than lower thresholds of NCCR or KT.

Keywords

Intelligent Tutoring System, Knowledge Tracing, Mastery Learning.

INTRODUCTION

Intelligent tutoring systems are known for their ability to personalize the learning experience for students. One way that learning is individualized is by providing just the right amount of practice to meet the student's needs. Determining the correct amount of practice is critical because over-practice might bore students and take an un-necessarily long time, while under-practice might not provide enough opportunities for a student to learn a skill. To determine the correct amount of practice, systems must identify the point in time when students have learned the skill, otherwise referred to as reaching mastery. To predict this latent variable, mastery, systems must rely on student performance.

Defining mastery may vary between systems. One measure of mastery includes next problem correctness, another is performance on a transfer question, and yet another is performance on a delayed retention test. Some systems rely on knowledge tracing (KT), others use a predetermined number of consecutive correct responses (NCCR). In each case, mastery status is used by the system to determine the end of an assignment.

KT is known to be highly accurate at predicting next problem correctness [4]. By providing the probability that the student is in the learned state, knowledge tracing can also be used to predict, or detect mastery. Fancsali, Nixon and Ritter [4] examined the prevalence of two types of errors introduced when using various KT thresholds to establish mastery. False positives occur when a student without knowledge has been judged mastered, and false negatives occur when a student receives additional practice despite having the knowledge. Different mastery thresholds will affect the relative frequency of these errors. It was determined that using a probability of being in the learned state of 95% is a conservative trade-off between over-practice (false negatives) and avoiding premature mastery judgment (false positives).

One disadvantage to KT is that it requires a substantial amount of data to learn parameters and to fit a model. Therefore for new skills typical parameters would have to be used, or an alternative is needed until enough data is collected to fit KT. Additionally, KT may not be particularly effective in the first few attempts when student data is limited as it is very susceptible to initial parameter values. Therefore a more naïve approach that is equally accurate may be more appropriate. Finally, KT can produce several sets of parameters that equally fit the data. However, interpreting these parameters is not always meaningful. Beck [2] refers to this as the identifiability problem. Calculating the probability that the student is in the learned state ("probability of learned") is particularly vulnerable as high guess or slip parameters may impact this value.

There are some systems that use a predetermined number of consecutive correct responses (NCCR) to detect mastery. Early on, Khan Academy [5, 9] used ten correct as the criteria for assignment completion. Recently [6] this has been reduced to five questions plus a combination of item difficulty and spaced repetition. Another well-known system, ASSISTments [7], uses three-right-in-a-row as the default setting for assignment completion and then additional spaced practice (ARRS). However, teachers can adjust this setting as desired. Prior research suggests that three-right-in-a-row may be an accurate threshold to detect mastery if that threshold is met early in a problem set [2]. Beck found that when the threshold is met later in the sequence, students are often unsuccessful on a delayed reassessment. This suggests that a blanket default setting across all problem sets and all sequences may be flawed.

One disadvantage to a consecutive correct threshold is that the "slips" as defined by KT have significant impacts on practice opportunities. A slip is defined as an incorrect response by a student who is predicted to be in the learned state. Typical slip parameters in KT are between 0% and 10% [1]. This suggests that on average, 5% of students who in fact know a skill will answer a question incorrectly. When using NCCR to determine mastery, students who slip are penalized heavily, requiring them to complete additional unnecessary practice.

Accurately predicting or detecting mastery status is critical to intelligent tutoring systems, because the amount of practice provided to students depends on this. An overly cautious prediction will lead to unnecessary practice (false negatives), while less strict criteria will not provide enough (false positives). We are investigating whether additional attempts, due to a higher mastery threshold, will lead to increased accuracy in detecting mastery while not increasing false negatives. False negatives are challenging to detect using performance data. However, considering the amount of additional practice required for different NCCR thresholds will shed some light on the impact of false negatives. For example, one system that requires 10 correct-in-a-row (10-CCR), might be able to identify mastery with 95% accuracy, while another that requires 5 correct-in-a-row (5-CCR) reaches 80% accuracy. If the 10-CCR requires students to complete on average 8 questions more than the 5-CCR, we must consider whether that degree of accuracy is worth the time spent by students.

Therefore, Study I of the present study leverages data generated by an intelligent tutoring system to explore the ability of NCCR and KT to detect mastery. Mastery will be measured by next problem correctness. Additionally, an incremental efficiency analysis will also be presented that sheds light on the number of additional questions students must answer to reach a given threshold.

Next problem correctness is arguably a weak measure of mastery as slips are possible. A measure of more robust learning is performance on a transfer task [10]. Therefore, in Study II, a randomized-controlled trial was conducted to compare the accuracy of different potential thresholds of number of consecutive correct responses. This data was then used to further explore KT predictions, compared to NCCR in an attempt to determine which method should be used in intelligent tutoring systems who rely on mastery to determine amount of practice.

2. METHODOLOGY (Study I)

ASSISTments is an intelligent tutoring system that is widely used by students, predominantly in elementary and middle school, and relies on NCCR. The focus problem sets are considered skill builders, which are created to provide individualized practice to students. Specifically, students must continue to complete problems until a set number of consecutive problems are answered correctly. Presumably, the threshold has been selected because the system is predicting that the student has mastered the skill and no longer needs practice. Next problem correctness provides a measure of accuracy of this mastery determination. It is important to note ASSISTments provides an optional automatic reassessment of skills with spaced practice to better detect mastery. However, this data was not available and therefore was not considered for the present study.

Problem logs generated during the 2012-2013 school year using ASSISTments were used for the current study. From the original data set, we selected problem sets with the mastery setting as 5-CCR. Using a threshold setting of five consecutive questions allows us to analyze student responses on the fourth and fifth questions to explore the accuracy of 3-CCR. We also limited the problem sets selected to those with at least 50 problem logs to ensure enough data to fit KT. This resulted in data from 395 students who completed 25 problem sets, generating 5,928 rows of data. NCCR is attached to assignment, therefore we care about the number of student-assignment pairs when it comes to an NCCR relative analysis. If a student completed more than one assignment, they were used multiple times. In this data set, the number of student-assignment pairs is 698. The data set can be accessed online [13].

To examine NCCR, strings of student responses were analyzed specifically looking at the two actions immediately following the first string of three consecutive correct responses. In consideration of Beck's [2] findings, that students who reach the mastery threshold late in a problem set often fail a delayed retention test, students who answered the first three questions correctly were separated from those who completed at least one question incorrectly before answering three correct in a row. We calculated the percent of students falling into each of the four response combinations (see Figure 1): fourth question incorrect and fifth question incorrect (A), fourth question incorrect and fifth question correct (B), fourth question correct & fifth question incorrect (C), and fourth question correct and fifth question correct (D).

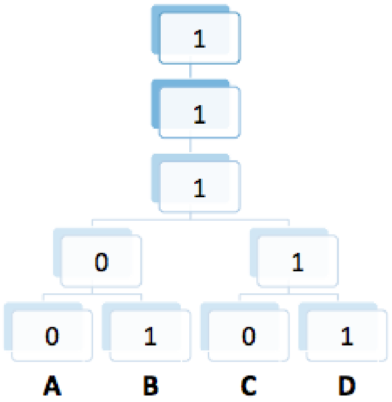


Figure 1. Potential student response combinations.

To explore KT, we fit a model, and for each skill we generated the four parameters of *guess*, *slip*, *learn* and *prior*. These parameters were then used to calculate the probability that the student was in the learned state at each student action step. Looking specifically at the first time the student answered three consecutive questions correctly, we calculated the percentage of students who had a probability of being in the learned state (at least 95%) at the third, fourth and fifth action. Again, students were separated into those who answered the first three questions correctly and those who answered at least one question incorrectly prior to the three consecutive correct responses. It is important to note that with as few as 50 problem logs for some skills, some of the KT parameters may be compromised.

RESULTS (Study 1)

NCCR

In examining the fourth and fifth action after a string of three consecutive correct responses, the percentage of students with each response combination was calculated. For all 285 student-assignment pairs who correctly answered the first three consecutive questions correctly (see Table 1), 80.0 % also answered the fourth and fifth questions correctly. This could be interpreted as the accuracy measure of a threshold of three consecutive correct responses as it seems to confirm the mastery classification according to NCCR. 18.2% of students answered either the fourth or fifth question correctly but the other incorrectly. Assuming NCCR is accurate at detecting mastery, these students could be considered to have slipped on that question. However, these students could also represent the false positive error rate, meaning they were considered mastered by the threshold of three, yet did not learn the skill as demonstrated by their incorrect response. The 1.8 % of students who answered both the fourth and fifth questions incorrectly could represent the error rate of NCCR as they suggest an inaccurate detection of mastery by NCCR.

Table 8. For students who answered three consecutive questions correctly without an error, percentages of students with each response combination on the fourth and fifth action are presented.

3 Consecutive No Errors		Fourth Question	
Fifth Question		Incorrect	Correct
	Incorrect	1.8% (5)	9.8% (24)
	Correct	8.4% (28)	80.0% (228)

For all 309 student-assignment pairs who had at least one incorrect response in the current assignment prior to obtaining three right-in-a-row, the percentages of answer combinations on the fourth and fifth questions were also computed (see Table 2.) While 75.1% of students answered both the fourth and fifth questions correctly, 4.2% answered them both incorrectly. This suggests that for students who did not answer the first three questions correctly, this threshold may be too lenient as 25% of students were classified as mastered went on to answer at least 1 question incorrectly. 20.8% answered the fourth or fifth question correctly, but the other question incorrectly. Again, this could be considered a “slip” assuming NCCR is an accurate detection of mastery, or might indicate a false positive error in NCCR.

Again, we purposely used problem sets with a mastery threshold of five so that the fourth and fifth actions after three consecutive correct responses could be analyzed and serve as a measure of accuracy of NCCR. Using two questions provides a more robust measure of next problem correctness. However, this presents a challenge in interpreting the classification of students who answered one of the two questions incorrectly. We can conclude that for 3-CCR, at least 75% of students are correctly identified as mastering this skill, and that this percentage is slightly lower for those students who made at least one error prior to answering three consecutive questions correctly. This percentage might be higher depending on how we interpret the approximately 20% of students who made one error after reaching the 3-CCR threshold.

Table 9. For students who answered three consecutive questions correctly AFTER at least one error, percentages of students with each response combination on the fourth and fifth action are presented.

3 Consecutive with Errors		Fourth Question	
Fifth Question		Incorrect	Correct
	Incorrect	4.2% (13)	10.4% (32)
	Correct	10.4% (32)	75.1% (232)

KT

The learned parameters from fitting KT were used to compute the probability of the student being in the learned state for each student action. We fit a set of parameters for each of the 25 problem sets ensuring that these parameters are reasonable. In looking at only the first string of three consecutive correct responses, the KT probability for the third, fourth and fifth action were analyzed. To determine how KT as a mastery threshold compares to NCCR, the percentage of students who had a KT prediction of 95% or higher was calculated. Again, students who answered the first three questions correctly were separated from those who answered at least one question incorrectly.

For students who answered the first three questions correctly, 88.9% had at least a 95% probability of being in the known state according to KT. This increased to 96.1% after the fourth question was answered correctly and to 100% after the fifth question (see Table 3).

Table 10. KT prediction for students who answered three consecutive questions correctly without an error.

Number of consecutive questions correct	3	4	5
Percentage of Students with KT prediction at least 95%	88.9%	98.1%	100%

For students who had at least one incorrect response before getting three correct in a row, only 63.2% of students had at least a 95% probability of being in the known state according to KT (see Table 4). This increased to 89.6% after the fourth question and to 96.8% after the fifth.

Table 11. KT prediction for students with three consecutive questions correct AFTER at least one error.

Number of consecutive questions correct	3	4	5
Percentage of Students with KT prediction at least 95%	63.2%	89.6%	96.8%

This suggests that NCCR, at all thresholds, is more lenient than KT, as not all students who met the NCCR threshold met the KT threshold. This difference is particularly pronounced for students who made at least one error, potentially lending support to the findings in Beck [2] that students who reach three correct in a row later in a problem set may not have mastered the skill.

However, it is necessary to determine the accuracy of these KT predictions. To understand how we measure accuracy of detecting mastery see Table 5. We consider students in the "threshold met/correct" cell or "threshold not met/incorrect" cell to be accurately identified as mastered. We calculated the percentage of students who were labeled mastered, according to reaching the threshold, who also answered the next question correctly, and those who did not reach the threshold, and answered the next question incorrectly. We recognize that students in the "threshold met/correct" cell may represent false negatives in that students may have had to complete additional questions beyond the moment they acquired the knowledge.

Table 5. Defining how NCCR's accuracy is measured.

	Mastery Status		
Performance		Threshold Met	Threshold Not Met
	Correct	• Accurate	• False Negatives
	Incorrect	• False Positives	• Accurate

To assess the accuracy of KT, we identified mastery status by computing the student's probability of being in the learned state after their third consecutive correct responses. We examined student performance on the fourth question based on this mastery status. For students who answered three consecutive questions correctly, without an error, KT accurately identified mastery 82% of the time (see table 6), which is consistent with the performance of 3-CCR.

Table 6. Accuracy of KT detecting mastery for students who answered three consecutive questions correctly without an error. (n=287)

	Threshold Met **(>95%)**	**Threshold Not Met (<95%)**
Next Question Correct	80.5% (231)	9.4% (27)
Next Question Incorrect	8.4% (24)	1.7% (5)

For students who made at least one error before reaching 3 correct in a row, accuracy drops to 66% and false negatives increase by 18% (Table 7). This suggests that while KT is more stringent than 3-CCR, it is less accurate due to the increase in students who are unable to reach the threshold, yet seem to have learned the skill.

Table 7. Accuracy of KT detecting mastery for students who answered three consecutive questions AFTER at least one error. (n=324)

	Threshold Met **(>95%)**	**Threshold Not Met <95%**
Next Question Correct	57.7%(187)	27.8%(90)
Next Question Incorrect	5.9%(19)	8.6%(28)

3.2 Incremental Efficiency Analysis

This preliminary data suggests that perhaps a higher threshold of consecutive correct responses might yield a more accurate detection of mastery. However, we must consider the amount of time required to reach such a threshold, and the potential introduction of false negatives. We used the same data set from above, in which students are required to reach 5CCR. As mentioned earlier, if a student completed more than one assignment, they were used multiple times. The table below (Table 8) shows the distribution of the maximum N-CCR thresholds reached by 698 students. The number (percent) of students who maxed out at each NCCR threshold and the average (standard deviation) number of items completed for students at that threshold are presented. For example, 542 students reached the 5-CCR threshold; 43 students reached 4-CCR (failing to answer five consecutive questions correctly); 67 students failed to reach more than 2-CCR.

Table 8: Distribution of students across the maximum N-CCR thresholds met and the number (std) of questions completed to reach that threshold.

NCCR	**5CCR**	**4CCR**	**3CCR**	**<3CCR**
Number (Percent) of students	542 (77.7%)	43 (6.2%)	46 (6.6%)	67 (9.6%)
Average number (std) of questions	7.7 (4.2%)	13.6 (9.0%)	11.3 (7.0%)	8.8 (5.7%)

Generally, students who failed to reach the 5-CCR threshold completed more questions. This is potentially an indication of wheel spinning [2] and could be used to detect this undesirable behavior.

For the majority, students who finally reached 5CCR, it is important to know whether the threshold is so high that we have introduced false negatives. In other words, were students forced to practice beyond the moment when they learned the skill?

Using the data generated from the students reaching the 5-CCR threshold, we can determine how many additional questions were required to reach each incremental threshold, one through five. This will provide insight into the tradeoff between potential increased mastery detection and time consumption, as measured by number of questions completed. Similarly to the analysis above, we divided students into two groups: students who answered three consecutive questions correctly without an error, and students who answered three consecutive questions correctly after at least one error.

Table 9 shows the distribution of students across the number of questions necessary to move through each threshold of NCCR. For example, after answering three questions correctly without an error, 249 students needed one additional question to reach 4-CCR, however 18 students needed five questions to reach 4-CCR, resulting in a response sequence of: 1,1,1,0,1,1,1,1.

The numbers that are identified in blue represent students, who reach the N-CCR threshold, made only one error and then reach the N+1-CCR threshold. It is very likely that these students slipped at the N+1th problem. For students who need more additional items (noted as red), it is very likely that they in fact need the extra practice. Therefore, for students who reach 3-CCR without an error, it seems that about 6% students are wasting time with a higher threshold, while about only 2% will benefit from the additional practice. In the contrast, for students who reach 3-CCR after making at least one error, about 3% to 4% of students will waste time while 4% will actually benefit from the additional practice required by the higher threshold.

Table 9. Distribution of students across the number of questions necessary to move through each threshold of NCCR.

Number (percent) of students		3CRR without an error (n = 272)		3CRR with at least one error (=270)	
		3CRR	4CRR	3CRR	4CRR
Number of additional questions needed for students to reach the next threshold of NCCR	1	249 (91.5%)	248 (91.2%)	249 (92.2%)	253 (93.7%)
	5	18 (6.6%)	\	10 (3.7%)	\
	6	5 (1.8%)	18 (6.6%)	11 (4.1%)	7 (2.6%)
	>6		6 (2.2%)		10 (3.7%)

This suggests that for students who reach the lower threshold without errors, a higher threshold is unnecessary and will waste student's time. However, for students who do not initially reach the 3-CCR, a higher threshold is more appropriate to provide sufficient practice to in fact learn the skill.

METHODOLOGY (Study II)

The results of the initial study suggested that for detecting mastery, as defined by next problem correctness, because of KT's more stringent threshold than 3-CCR, it's accuracy suffers due to false negatives. We proposed a second study to examine the potential improvement to accuracy higher threshold of NCCR might provide. Additionally, other measures of mastery should be considered. To determine the efficacy of NCCR with different thresholds and KT at detecting mastery, a randomized-controlled trial was conducted. A post-test was used to measure next problem correctness and a transfer question was included to provide an additional measure of mastery.

Seventy-seven students in a seventh grade math class participated in the experiment in ASSISTments as part of their math class. Students answered questions from two topics (order of operations and ratios), which were counterbalanced for order and NCCR threshold (3-CCR and 5-CCR). Students were randomly assigned in one of four conditions (see Table 10 for distribution of students).

The randomization into conditions and the percent of students who completed the assignment by condition was not even. However, if we ignore the order of the topics and collapse conditions A with D and B with C, the percent of students by NCCR is even. Specifically, the percent of students who completed Order of Operations with 3-CCR is 48% and the percent of students who completed Ratios with 3-CCR is also 48%. Students were given different amounts of time in class to work on the assignment. Therefore, only students who had enough time to complete both topics were included in the analysis n=37).

Table 10: Distribution of students among the four conditions.

Condition	Students Assigned	Students Completed
A. Order of Operations 3-CCR then Ratios 5-CCR	18	11
B. Order of Operations 5-CCR then Ratios 3-CCR	15	8
C. Ratios 3-CCR then Order of Operations 5-CCR	29	13
D. Ratios 5-CCR then Order of Operations 3-CCR	15	5

End of problem-correctness feedback and hints upon request were available for every question. For each topic, once students met the given threshold, they were immediately given a post-test that consisted of two morphologically similar questions and one transfer question. We recognized that the morphologically similar questions were providing additional practice beyond the set threshold, therefore the post-test questions were assigned in a random order. Additionally, to provide data to compute a partial credit score on the post-test, correctness feedback and hints were provided.

To ensure that students would reach both post-tests, the post-test was administered following the 14th question, even if the threshold was not achieved. This also allows us to detect students who were not labeled mastered yet who were successful on the post-test and/or transfer question, serving as an indication of false negatives.

RESULTS (Study II)

NCCR

An initial analysis of the data revealed that while both topics were balanced in terms of overall difficulty, (paired t-test p=0.33), post test scores for Ratios (62%) was slightly lower than Order of Operations (68%). However, performance on the transfer question was significantly lower for Ratios (27%) than Order of Operations (70%) (paired t-test p<0.001). Using a partial credit score [11] for the post test that accounts for number of hints and attempts used, also failed to show any differences in learning (paired t-test p=0.33). Order appears to have a slight effect for Order of Operations (t-test p=0.02), completing it first lead to slightly higher post test scores (m=81%) compared to second (57%). However this effect was not found for Ratios (t-test p=0.86).

We assume that if students complete more problems, due to a higher threshold of mastery, they should learn more. To assess if students do in fact learn more when completing an assignment with a higher threshold of consecutive correct responses, post-test scores for 3-CCR were compared to 5-CCR. A paired t-test revealed that post-test performance on the topic with 5-CCR (66%) was not significantly higher than post-test performance on the topic with 3-CCR (63%) despite having answered more questions (p=0.890). When completing the topic with a threshold of 5, students completed on average 11 questions (sd=3.6) whereas the topic with a threshold of 3 resulted in an average of seven questions completed (sd=3.5). This suggests that a higher threshold of NCCR does not lead to improved learning despite the additional practice that it requires. However, it is important to note that not all students reached the set threshold for mastery and we capped practice attempts at 14. This means that students in both conditions, who did not master the skill, received the same amount of practice.

For 3-CCR, 57% of students were accurately identified as mastered or not mastered. However, 43% of students were identified mastered, yet failed to answer the transfer question correctly (Table 11). This suggests that 3-CCR has a higher rate of false positives. Interestingly, 3-CCR was more accurate for the topic Order of Operations (88%) than the topic Ratios (33%).

Table 11. Student performance on transfer question based on 3-CCR threshold.

Percent(Number) of students	Threshold Met	Threshold Not Met
Transfer Correct	46%(17)	0%
Transfer Incorrect	43%(16)	11%(4)

For 5-CCR, 73% of students were accurately identified as mastered or not mastered (Table 12). Unlike with 3-CCR, this accuracy persists across topics, Order of Operations (76%) and Ratios (69%). However, 8% of students who were unable to meet the threshold were able to answer the transfer item correctly. As expected, a higher mastery threshold introduces false negatives, which were not present in 3-CCR. Specifically, three students were subjected to additional practice that did not appear to be necessary. Of the 14 students who did not meet the higher threshold of 5-CCR, 13 were able to meet the 3-CCR. Of those 13, 62% (n=8) failed to answer the transfer question correctly. This provides further confirmation that 5-CCR is more accurate at detecting mastery, as defined as performance on a transfer question, than 3-CCR.

Table 12. Student performance on transfer question based on 5-CCR threshold.

Percent(Number) of students	Threshold Met	Threshold Not Met
Transfer Correct	43%(16)	8%(3)
Transfer Incorrect	19%(7)	30%(11)

Unlike in Study I, we did not separate students who met the threshold without an error, from those who made at least one error. The sample size was too small and more than 85% of students in both conditions made at least one error.

KT

To fit our model with knowledge tracing, we set the initial guess rate as 0.05 to avoid degenerate models. Other parameters were set randomly. We fit KT repeatedly eight times for each topic, and chose the non-degenerate learned parameters, which best fit the data (i.e. with maximum log likelihood). Then we used the chosen parameters as initials to fit KT one more time to build the prediction models, which were used to predict performance of students in the experiment. The learned parameters for Order of Operations are: prior=0.862, learn=0.137, forget=0.000, guess=0.305, and slip=0.205. The learned parameters for Ratios are: prior=0.805, learn 0.144, forget=0.000, guess=0.329, slip=0.205.

The parameters were used to calculate the probability that a student was in the learned state after each question. This value was used to determine mastery. Students with a probability greater than 95% were considered mastered. Results of a paired t-test indicate that the average KT probability of learned for Order of Operations (m=94%, sd=20) was significantly higher than for Ratios (m=90%, sd=23) (p=0.04, effect size 0.19).

To determine the effect of additional practice on KT predictions, the probability the student was in the learned state for the topic with an NCCR threshold of 3 was compared to that of the same student for the topic with an NCCR threshold of 5. Results indicate that the additional practice required by a higher threshold does not increase the probability that a student will be in the learned state. When the threshold for number of correct responses was three, students had an average probability of being the learned state of 91.8% (sd-21.9), yet for a threshold of five, the average probability was 92.0% (sd-21.5). Similar to the findings of NCCR, this suggests that additional practice created by a higher mastery threshold of consecutive correct responses does not in fact lead to increases in learning.

To determine the accuracy of KT at detecting mastery, as measured by performance on the transfer question, we calculated the percent of students who had at least a 95% probability of being in the learned state who also answered the transfer question correctly and those who were less than 95% who answered the question incorrectly. Results indicate that KT accurately detected mastery 54% of the time (Table 13). This is comparable to the accuracy of three right-in-a-row used by the NCCR method, but lower than the accuracy of 5-CCR.

Table 13. Student performance on the transfer question based on KT's 95% threshold.

Percent(Number) of students*	Threshold Met	Threshold Not Met
Transfer Correct	42%(31)	7%(5)
Transfer Incorrect	39%(29)	12%(9)

*Each student is counted twice for they worked on 2 skills.

When looking at the topics separately, KT was accurate 64% of the time for Order of Operations but only 43% of the time for Ratios. KT predications differed slightly when accounting for the difference in question completion rates due to the higher NCCR threshold used to design the assignment. Specifically, for sections that required 3-CCR, KT's accuracy was 60% and for 5-CCR KT's accuracy was 54%. This suggests that KT is more accurate at detecting performance on the next problem than performance on a transfer question.

DISCUSSION

Detecting mastery is essential when personalizing the amount of practice a student receives when working in an intelligent tutoring system. However, mastery can be measured in different ways, which affects the accuracy of the mastery detection method.

When detecting mastery, as measured by next problem correctness, it appears that despite its simplicity, 3-CCR is a highly effective method for detecting mastery. The incremental efficiency analysis revealed that, for students who do eventually reach a 5-CCR threshold, 3-CCR is most likely sufficient, as any error made beyond that is most likely a slip and over 90% of students will go on to answer the next two questions correctly.

There is a slight suggestion that for students who do not immediately reach the 3-CCR threshold, the additional practice required by a higher NCCR threshold might be beneficial. The results of the randomized-controlled trial support this finding, as a higher threshold did not lead to increased performance on a post-test. KT's more stringent threshold reduces its accuracy by introducing false negatives, leading to unnecessary additional practice for many students.

When detecting mastery, as measured by performance on a transfer question, 5-CCR appears to be more accurate than 3-CCR or KT. Both struggled to accurately detect mastery for a challenging question. Specifically, many students were able to reach the threshold, yet were not able to answer the transfer question correctly. On-the-other-hand, with 5-CCR, very few students who were able to reach the threshold answered the transfer question incorrectly. This suggests that for detecting robust learning, a higher NCCR threshold is superior. However, there were a handful of students who were unable to reach the threshold who did answer the transfer question correctly. This confirms that with an increased threshold comes an increased frequency in false negatives. Yet the frequency of these false negatives is lower with 5-CCR than with KT.

As a measure to reduce the amount of unnecessary practice a student may receive, we capped the number of questions students were given to fourteen. Using this method, a system can comfortably use a higher mastery threshold to more accurately identify students who have or have not mastered a skill, without subjecting students to endless practice. Students who reach this cap, prior to mastering the skill, could be given an alternate intervention. This strategy could then be used to reduce wheel spinning.

This analysis began when the first author, a school teacher using ASSISTments, became frustrated with the handful of students who were reaching the 3-CCR threshold yet not retaining the skill. Upon learning about the more sophisticated KT, it was suggested that ITS, like ASSISTments, would be more effective using this method to determine mastery. However, the results do not support this. The naive three-correct-in-a-row method for detecting mastery, seems to predict next problem correctness well and a higher NCCR threshold is superior to KT when predicting performance on transfer items.

IMPLICATIONS AND FUTURE RESEARCH

It is necessary to consider the purpose, and therefore importance, of mastery detection in an intelligent tutoring system. Many systems use mastery to determine the number of questions students will complete for that topic. If students will be exposed to delayed/spaced practice for that skill, then accuracy of mastery detection is less critical because students are guaranteed to have additional opportunities to demonstrate mastery. If mastery means that students will no longer have exposure to that skill, then accurately detecting mastery is essential and worth additional student practice to ensure that accuracy.

Results from these studies suggest that if accuracy is important, NCCR with a higher threshold, such as five, is preferable. Not only does it decrease the likelihood of guessing, it was also shown to more accurately predict performance on a transfer question. However, both Study I and II suggest that 3-CCR is a reasonable method for detecting mastery because higher thresholds did not lead to improved learning, as measured by a post-test, nor did it differ from KT when predicting performance on a transfer question. The results from the incremental efficiency analysis justify the exploration of an adaptive NCCR threshold. For students who do not make an error, 3-CCR could be used as the threshold. As soon as students make an error, a higher threshold, such as 5-CCR, could be imposed.

While we were able to manipulate different thresholds of NCCR, we were not able to manipulate KT. It would be interesting to explore the accuracy of KT when it is used to determine assignment completion, instead of being applied to the data later. Specifically, does using KT to detect mastery lead to improved learning when compared to NCCR?

Finally, we know that 5-CCR leads to increased accuracy in predicting retention question performance. This suggests that 5-CCR leads to more robust learning. This hypothesis should be explored further using other measures of robust learning, including performance on delayed retention tests [10].

ACKNOWLEDGMENTS

We thank multiple NSF grant (ACI-1440753, DRL-1252297,DRL-1109483,DRL-1316736 & DRL-1031398) , the US Dept of Ed's (IES R305A120125 & R305C100024 and GAANN), the ONR, and the Gates Foundation.

REFERENCES

[1] Baker, R.S.J.d, Corbett, A., Aleven, V. (2008). More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. *Carnegie Mellon University Research Showcase @CMU*.

[2] Beck, J., Gong, Y. (2013). Wheel-Spinning: Students Who Fail to Master a Skill. In *Proceedings of the 16th Artificial Intelligence in Education*, Lane, H.C., Yacef, K., Mostow, J., & Pavlik, P. (Eds.) 431-440.

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

[4] Fanscali, Stephen E., Nixon, Tristan, & Ritter, Stephen (2013). Optimal and worst-case performance of mastery learning assessment with bayesian knowledge tracing. *Proceedings of the 6th International Conference on Educational Data Mining.* D'Mello, S., Calvo, R., Olney, A. (Eds). 35-42.

[5] Faus, M. (2014). Improving Khan Academy's student knowledge model for better predictions. *MattFaus.com* [web log]. Retrieved October, 2014, from http://mattfaus.com/2014/05/improving-khan-academys-student-knowledge-model-for-better-predictions/

[6] Faus, M. (2014). Khan Academy Mastery Mechanics. *MattFaus.com* [web log]. Retrieved October, 2014, from http://mattfaus.com/2014/07/khan-academy-mastery-mechanics/

[7] Feng, M., Heffernan, N. T., Koedinger, K. R.: Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19, 243-266 (2009)

[9] Hu, D. (2011). How Khan Academy is using Machine Learning to Assess Student Mastery. *David-Hu.com* [web log]. Retrieved October, 2014, from http://david-

hu.com/2011/11/02/how-khan-academy-is-using-machine-learning-to-assess-student-mastery.html

[10] Koedinger, K., Aleven, V. (2007). Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. Educational Psychology Review 19:239-264.

[11] Ostrow, K., Donnelly, C., Adjei, S., & Heffernan, N. (In Press). Improving Student Modeling Through Partial Credit and Problem Difficulty. To be included in the Proceedings of the 2nd Annual Meeting of the ACM Conference on Learning at Scale.

[13] Data set and code. https://drive.google.com/folderview?id=0B5Nb7T9qsMmHMFdBdmNVZk5LeFE&usp=drive_web

Improving Students’ Long-Term Retention Performance: A Study on Personalized Retention Schedules

Xiaolu Xiong, Yan Wang, Joseph Beck
Worcester Polytechnic Institute
Worcester, MA 01609
{ xxiong, ywang14, josephbeck } @wpi.edu

This section has been published at:

Xiong, X., Wang, Y., & Beck, J. (2015). Improving students' long-term retention performance: a study on personalized retention schedules. *In Proceedings of the 5th International Conference on Learning Analytics & Knowledge*.

ABSTRACT

Traditional practices of spacing and expanding retrieval practices have typically fixed their spacing intervals to one or few predefined schedules [5, 7]. Few have explored the advantages of using personalized expanding intervals and scheduling systems to adapt to the knowledge levels and learning patterns of individual students. In this work, we are concerned with estimating the effects of personalized expanding intervals on improving students’ long-term mastery level of skills. We developed a Personalized Adaptive Scheduling System (PASS) in ASSISTments’ retention and relearning workflow. After implementing the PASS, we conducted a study to investigate the impact of personalized scheduling on long-term retention by comparing results from 97 classes in the summer of 2013 and 2014. We observed that students in PASS outperformed students in traditional scheduling systems on long-term retention performance (p = 0.0002), and that in particular, students with medium level of knowledge demonstrated reliable improvement (p = 0.0209) with an effect size of 0.27. In addition, the data we gathered from this study also helped to expose a few issues we have with the new system. These results suggest personalized knowledge retrieval schedules are more effective than fixed schedules and we should continue our future work on examining approaches to optimize PASS.

Categories and Subject Descriptors

H.4 Information Systems Applications; K.3.1 Computer Uses in Education; J.4 Social and Behavioral Sciences

General Terms

Algorithms, Measurement, Performance, Design, Theory.

Keywords

Knowledge retention, retrieval practice, spacing effect, intelligent tutoring system, personalization

INTRODUCTION

Automatic Reassessment and Relearning System

Based on a robust memory phenomenon known as the *spacing effect* [4], expanding retrieval practice is often regarded as a superior technique for promoting long-term retention relative to equally spaced retrieval practice [3, 8]. Expanding retrieval practice works by, after the student learns a skill, having the student perform the skill at gradually increasing spacing intervals between successful retrieval attempts. Research has shown that spacing practice has a cumulative effect so that each time an item is practiced it receives an increment of strength [10]. This effect is specifically crucial to subjects such as mathematics: we are more concerned with students’ capability to recall the knowledge that they acquired over a long period of time. What is more, the ability to retain a skill long-term is one of the three indicators of robust learning [2].

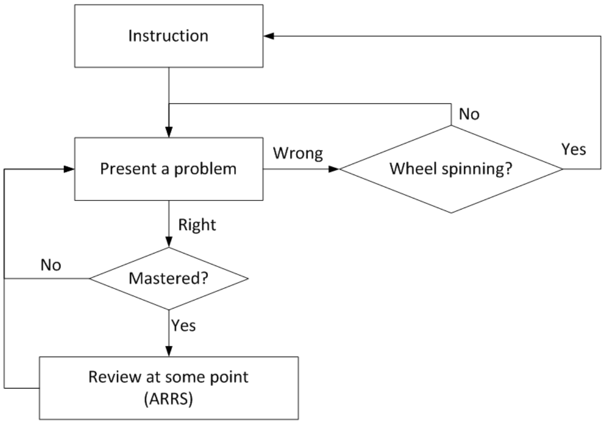


Figure 12. The enhanced ITS mastery learning cycle

Inspired by the importance of long-term retention and the design of the enhanced ITS mastery cycle in **Figure 12** proposed by Wang and Beck [11], we developed and deployed a system called the Automatic Reassessment and Relearning System (ARRS) [13] to make decisions about when to review skills that students have mastered in ASSISTments, a non-profit, web-based tutoring system. ARRS is an implementation of expanding retrieval in the ITS environment. Unlike most ITS systems in which the tutoring stops if the student masters a given skill, ARRS assumes that if a student masters a skill with three correct responses in a row, such mastery is not necessarily an indication of long-term retention. Therefore, ARRS will present the student with retention tests on the same skill at expanding intervals spread across a schedule of at least 3 months. The default setting of the ARRS scheduling system uses a spacing interval of 7-14-28-56, and this indicates that each skill requires 4 level tests: the first level of retention tests takes place 7 days after the initial mastery; the second level of retention tests 14 days after successfully passing the first retention test, and so on. If a student answers incorrectly in one of these retention tests, ASSISTments will give him an opportunity to relearn this skill before redoing the same level of test.

Table 12. Retention performance by mastery speed and retention interval from pilot study

Retention test delay	# tests	% correctness
Mastery speed 3 – 4		
1 day	1186	84.4%
4 days	1169	82.2%
7 days	1171	81.7%
14 days	1233	81.2%
Mastery speed 5 – 7		
1 day	467	77.9%
4 days	432	76.2%
7 days	362	77.1%
14 days	420	73.1%
Mastery speed > 7		
1 day	280	67.5%
4 days	320	62.8%
7 days	267	59.6%
14 days	243	54.8%

In our previous studies [13, 14] of modeling student retention performance, we found that the number of problems required achieving mastery, which we referred to as the *mastery speed*, is an extremely important feature for predicting students' retention performance. We observed that, in general, the slower the mastery speed, the lower the probability that the student can answer the problems in the retention test correctly. Students who mastered a skill in 3 or 4 problems had approximately an 82% chance of responding correctly on the first retention test, while students who took over 7 attempts to master a skill only had a 62% chance [13]. Based on these results, we conclude that students with different mastery speeds have different retention patterns, so we began searching for the optimal retrieval schedules for different levels of student knowledge.

In order to find the optimal retention schedule for students and the best way to boost their performance in long-term mathematics learning, we conducted a pilot study by setting up four different interval schedules (1 day, 4 days, 7 days, and 14 days) and examined the impact on retention performance by comparing results across different groups of students. The results are shown in **Table 12** and [12]. We saw a consistent decrease in retention performance with the longer retention intervals across in all students, no matter if they fell into the high mastery level, medium mastery level or low mastery level category. The results from Table 1 also demonstrated a main effect of mastery speed on retention performance: students with slower mastery speed had lower performance than students with a faster mastery speed; this statement is true even when we compared a 1-day performance of students with a mastery speed of over 7 (67.5% correct) speed versus a 14-day performance of students with a mastery speed of 3 or 4 (81.2% correct). A sizeable and interesting effect is that students with slower mastery speeds had bigger decreases in retention performance as retention intervals lengthened. For example, a high mastery level student had a decrease of 3.2% between 1 day tests and 14 days tests but the retention performance of low mastery level students dropped 12.7%. These results suggest retention intervals probably should vary, rather than be fixed, based on the student's knowledge of the skill.

Personalized Adaptive Scheduling System

Although ARRS helps students review knowledge after a time period, it neither knows a student's knowledge level, nor does it have the mechanism to change the retention schedule based on a particular student's performance. Here we formed a hypothesis that we can improve students' long-term retention levels by adaptively assigning students with gradually expanding and spacing intervals over time and we proposed to design and develop such a system, called Personalized Adaptive Scheduling System (PASS), as shown in **Figure 13**. PASS enables ARRS to schedule retention tests for students based on their knowledge levels. In the spring of 2014, we enhanced the traditional ARRS with the PASS and deployed it in ASSISTments.

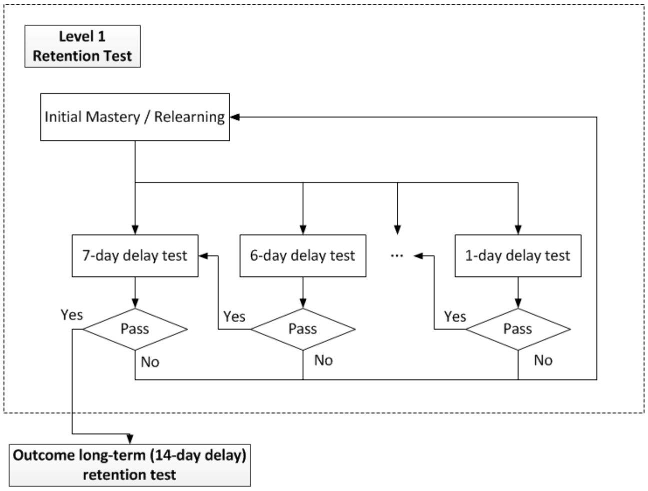


Figure 13. Design of Personalized Adaptive Scheduling System (PASS)

The current workflow of PASS aims to improve students' long-term retention performance by setting up personalized retention test schedules based on their knowledge levels. Here we rely on the mastery speed of a skill as an estimate of the student's knowledge and, consequently, predictor of retention performance. We retained the ARRS design of 4 expanding intervals of retention tests for each skill; however, PASS alters how the first interval behaves. When a student finishes initially learning a skill, we use his mastery speed to decide when to assign his first level 1 retention test. The mapping between mastery speed and retention delay intervals of the level 1 test is shown in **Table 13**. When a student passes the first test, PASS will schedule another test with a 1-day longer delay. Once the student passes the 7-day test, he is promoted to level 2 with a delay of 14 days. From that point on the intervals are the same as in the ARRS system. Note that mastery speed can be extracted from both students' initial learning and relearning processes. Therefore, when a student fails a retention test, a relearning assignment will be assigned to the student immediately. How quickly the student relearns this assignment will be used to set the interval for his next test. The mechanism of level 2 to level 4 tests is simpler. When a student fails a retention test, the retention delay will be reduced to the previous level (e.g., from 56 days to 28 days). It will be increased to the next level if the student passes the delayed retention test.

Table 13. Mapping between mastery speed and level 1 retention delays

Mastery Speed	Retention Delay
3	7
4	6
5	5
6	4
7	3
> 7	1

Here is an example of a student working with PASS in ASSISTments. Let's assume he needed 4 attempts to achieve three correct responses in a row in an initial learning assignment, so his mastery speed on this skill was 4. PASS then scheduled the first level 1 retention test for him to complete 6 days after the initial mastery. 6 days later, the student passed the retention test and PASS scheduled a 7-day retention test. Then a week later, the student passed the 7-day retention test and moved to the level 2 retention tests.

A STUDY ON IMPACT OF PERSONALIZED EXPANDING RETENTION INTERVALS

After the deployment of PASS in ASSISTments, several key issues were revealed that needed to be explored in order to realize the potential benefits of personalized expanding retention intervals and scheduling for students. We first conducted a study in ASSISTments to compare the new PASS with the traditional ARRS without PASS. In addition, this study explored the influence of personalized scheduling on students' long-term performance, student learning patterns and how they interact with the ASSISTments.

There were several objectives for this study. A central goal was to investigate potential long-term retention performance improvement to the benefit of personalized spacing schedules. We enabled PASS for all classes that were using ARRS on May 15, 2014; we expected students in these classes might be assigned homework during the next few months and thereby become the participants in the study. We ended this study on September 1, 2014 and found that 2,052 students from 40 classes were using PASS in the summer of 2014. Teachers of these classes assigned 93 different homework assignments to their students. Since traditional ARRS had been deployed in ASSISTments for over two years and a lot of data have been accumulated in the system, we extracted previous summer's ARRS-enabled classes that used the same assignments as the historical control group. 2,541 students from 57 classes in the summer of 2013 were qualified to act as historical control group.

During these two summer periods, students consistently received mathematics problem sets as homework assignments from their teachers. Once they answered three consecutive questions correctly in a problem set, students in the PASS condition would be given retention tests based on their mastery speed. If a student answered a retention test correctly, he was then given another retention test with a longer delay until he passed the level 1 test with a 7-day delay. On the other hand, students in traditional ARRS condition got 7-day delay retention tests after the mastery and went on with the 14-day tests if they answered the 7-day tests correctly. In this study, we defined how students performed on the 14-day retention tests (14 days after passing the level 1 test and at least 21 days after the initial mastery learning) as the outcome *long-term retention tests*. It is important to note that students usually receive several homework assignments and they may perform differently in these assignments, which means a student would have multiple tests that should be accounted for in the long-term performance. However, it is also possible that students do not complete assignments. Specifically, if a student has not finished the outcome retention test of a homework assignment by the end of this study, we cannot take this record into account.

RESULTS AND ANALYSIS

Retention test completion rate was calculated based on the number of homework assignments that had outcome tests answered divided by the total number of homework assignments. Days spent is the time interval between the start time of level 1 retention tests and the start time of outcome tests in days. Test count accounts for how many level 1 retention tests a student has to answer before this student can proceed to outcome tests. Students' long-term performance was calculated as the ratio of number of questions answered correctly in outcome tests to number of all questions answered in outcome tests.

Retention Test Completion Rate, Day Spent and Test Count

At the end of this study, the first result we noticed was that a lot of homework assignments in both groups did not have the records for associated outcome tests. In other words, a lot of students did not reach the 14-day retention tests. In the traditional ARRS condition, a total of 8404 homework assignments had been assigned to students but only 1,558 (18.5%) of these assignments had 14-days retention tests answered. When looking at the PASS condition, the retention test completion rate was even lower, only 1,029 (13.6%) of total 7,589 homework assignments had outcome tests answered. In one sense these low completion rates could result from the fact these homework and retention tests were assigned to students during the summer vacation so that perhaps many students did not treat these assignments seriously. The data also indicated the difference in the completion rates of the two conditions were statistically significant ($p < 0.0001$). We hypothesized that this was due to the fact that students in the PASS condition took more tests in order to pass the 7-day delay tests. Remember, some medium- and low-knowledge students had to pass a number of shorter-delay tests to even reach the 7-day and then 14-day retention tests. To address this hypothesis, we investigated how many days were needed to reach the 14-day test from the beginning of level 1 retention tests. The data was grouped by the three identified mastery speed bins to represent high-, medium- and low-knowledge students on their homework assignments

Table 14. Average day spent of each knowledge level by conditions

Initial mastery performance	ARRS	PASS	*p*-value
Mastery Speed	16.80	18.96	0.0002

3 - 4			
Mastery Speed 5 - 7	17.67	33.24	0.0001
Mastery Speed > 7	17.34	32.33	0.0001

Table 14 describes the differences in average days spent between ARRS and PASS conditions. The minimum possible delay is 14 days, achievable for ARRS students who answer the 7-day test correctly, and then take their ARRS test when it is immediately available. Students who failed the first ARRS test would have to take one or more additional 7-day tests until they responded correctly and could be promoted to the 14-day test. For the PASS condition, 14 days is a lower bound only for those students with an initial mastery speed of 3, as slower mastery speeds would require multiple first-level tests before being promoted to the 14-day interval. As expected, students in the PASS condition spent more time in the practices of level 1 retention tests; especially for medium- and low-knowledge students who spent nearly two more weeks in the process of passing the 7-day delay tests relative to ARRS students. **Table 15** demonstrates that students in the PASS condition had more tests to answer by showing the average test count of the two conditions therefore it took them more days to reach 14-day tests.

Table 15. Average test count of each knowledge level by conditions

Initial mastery performance	ARRS	PASS	*p*-value
Mastery Speed 3 - 4	1.34	1.21	0.0003
Mastery Speed 5 - 7	1.44	3.25	0.0001
Mastery Speed > 7	1.59	3.69	0.0001

Long-Term Retention Performance

After it was observed that PASS made students take more practice in the retention tests, we became more curious about the impact of PASS on long-term retention performance. It is important to emphasize that students were balanced with respect to proficiency in the ARRS and PASS conditions given their close homework performance level: 71.0% correct versus 71.2%. An initial analysis on long-term retention performance across all students showed the PASS condition (83.4%) outperformed the ARRS condition (77.2%) with a reliable but small improvement (p = 0.0002, effect size = 0.15). When considering the performance changes in different knowledge level of students, we again grouped the data by three identified mastery speed bins; then we examined students' long-term retention performance with p-values and effect sizes.

Table 16. Long-term (14-day) retention performance comparison and sample size (in parenthesis)

Initial mastery performance	ARRS	PASS	*p*-value	Effect size
Mastery Speed 3 – 4	81.79% (978)	83.91% (889)	0.2266	0.06
Mastery Speed 5 – 7	73.08% (327)	84.53% (97)	0.0209	0.27
Mastery Speed > 7	64.82% (253)	70.59% (51)	0.4301	0.12

The comparison of long-term retention performance shows that all three groups of students in the PASS condition outperformed those in the ARRS condition, although the improvements were not all statistically significant; only students with medium-knowledge on skills performed reliably better with an effect size of 0.27. For students with high knowledge on skills, the benefit of using PASS was limited; this suggests that solely relying on 7-day delay tests is sufficient for this population. A previous study [12] also suggested that high-knowledge students have high resistance against forgetting. On the other hand, providing low-knowledge students with more spaced retention tests and relearning assignments did not stop the decay of retention even after these students had approximately 3 additional relearning assignments on the same skill, and we only noticed a small effect size (0.12) improvement on the retention performance. Because PASS employs a higher stand of mastery and retention, thus few low-knowledge students reached outcome tests; we in fact noticed that only 51 tests had been completed, so this also prevented us from achieving a higher effect size in PASS condition. Another notable result was when we compared **Table 16** vertically: we could see that PASS helped to close the performance gap between different groups of students. In fact, in the PASS condition, the long-term performance of medium-knowledge students even outperformed the high-knowledge students. Of course, the small sample size tells us we need more studies to validate this result.

CONTRIBUTIONS, FUTURE WORK AND CONCLUSIONS

The paper makes three contributions. First, the work behind this paper designed and deployed a personalized expanding interval scheduling system that utilizes spacing effect in the field. Through the participation of thousands of students, we carried out a study to test the idea of assigning students with different delays of retention tests to help them better retain skills. As the first study on this system, the paper explores the path of improving ITS to help students achieve robust learning via personalized expanding retrieval practices. The second contribution of this paper is a validation of the hypothesis that students' long-term performance can be improved by giving them tests that are well spaced out and scheduled appropriately, before gradually expanding the spacing between these tests. Most importantly, this study demonstrates the importance of individualization in scheduling retention tests, as it shows that students with medium knowledge can match up their long-term performance with high-knowledge students by using PASS. The third contribution of this paper is the confirmation of concept of finding the optimal retention interval by using mastery speed as a measurement of students' knowledge level. By using mastery speed to group students, we can distinguish different learning and retention patterns among students with different knowledge levels. In the process of work, we have noticed that there has been other work on retention, such as the personalized spaced review system [6]; however, this work focuses on fact retrieval and is able to make far stronger assumptions of when students are exposed to content. Our work examines a procedural skill, in a classroom context where we cannot be sure what material teachers cover in class and we are not aware of all homework assignments, thus we cannot be sure when students last saw a skill.

This PASS and its implementation in ASSISTments have been introduced to the field for just a few months, so we are still at the initial phase of study. Our goal is to find the optimal spacing schedules for students and the best way to boost their performance in long-term mathematics learning. There are many further problems that we are interested in: What should we do to help low-knowledge students, considering the improvement we saw in the study was so small, particularly given the increased amount of practice they received? From the data we collected, it was obvious that there were some areas that required improvement. For example, we simulated a scenario to improve the retention performance of low-knowledge students to match up to the performance level of high-knowledge students (83.91%) and also improve completion rates to the level of ARRS condition so we could collect 228 data points. Given these optimistic assumptions, there intervention would have an effect size of 0.45. Thus, in this scenario, achieving a medium effect size (0.5) is not feasible. What is the fundamental cause of mistakes? Lack of effort or interest on the student's part, or a genuine lack of knowledge [1]? How can we increase the completion rate? Most importantly, how can we solve the optimization problem to balance time cost and performance improvement [9]? Is there a better way than just assigning high-frequency retention tests to students?

This paper presents the initial study of using the personalized adaptive scheduling system to explore a solution to the optimal spacing schedule problem. With the experiment data we collected, we are excited to see that the PASS can help to improve long-term retention performance across all three groups of students and become the backbone of future development for promoting student robust learning.

ACKNOWLEDGMENTS

We acknowledge funding for ASSISTments from NSF (# 1440753, 1316736, 1252297, 1109483, 1031398, and 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024) grants.

REFERENCES

[1] Anderson, J. R. (2014). Rules of the mind. Psychology Press.

[2] Baker, R. S., Gowda, S. M., Corbett, A. T., & Ocumpaugh, J. (2012, January). Towards automatically detecting whether student learning is shallow. In Intelligent Tutoring Systems (pp. 444-453). Springer Berlin Heidelberg.

[3] Crowder, R. G. (1976). Principles of learning and memory.

[4] Hintzman, D. L. (1974). Theoretical implications of the spacing effect.

[5] Logan, J. M., & Balota, D. A. (2008). Expanded vs. equal interval spaced retrieval practice: Exploring different schedules of spacing and retention interval in younger and older adults. Aging, Neuropsychology, and Cognition, 15(3), 257-280.

[6] Lindsey, R. V., Shroyer, J. D., Pashler, H., & Mozer, M. C. (2014). Improving students' long-term knowledge retention through personalized review. Psychological science, 25(3), 639-647.

[7] Karpicke, J. D., & Roediger III, H. L. (2007). Expanding retrieval practice promotes short-term retention, but equally spaced retrieval enhances long-term retention. Journal of Experimental Psychology: Learning, Memory, and Cognition, 33(4), 704.

[8] Melton, A. W. (1967). Repetition and retrieval from memory. Science (New York, NY), 158(3800), 532-532.

[9] Pavlik, P. I., & Anderson, J. R. (2008). Using a model to compute the optimal schedule of practice. Journal of Experimental Psychology: Applied, 14(2), 101.

[10] Thalheimer, W. (2006). Spacing learning events over time: What the research says.

[11] Wang, Y., & Beck, J. E. (2012). Using Student Modeling to Estimate Student Knowledge Retention. International Educational Data Mining Society.

[12] Xiong, X., & Beck, J. E. (2014). A Study of Exploring Different Schedules of Spacing and Retrieval Interval on Mathematics Skills in ITS Environment. In Intelligent Tutoring Systems (pp. 504-509). Springer International Publishing.

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

[14] Xiong, X., Adjei, S. A., & Heffernan, N. T. (2014) Improving Retention Performance Prediction with Prerequisite Skill Features. The 7th International Conference on Educational Data Mining.

The Assessment of Learning Infrastructure (ALI): The Theory, Practice, and Scalability of Automated Assessment

Korinn S. Ostrow, Doug Selent, Yan Wang,
Eric G. Van Inwegen, Neil T. Heffernan
Worcester Polytechnic Institute
Worcester, MA 01609
{ksostrow, dseslent, ywang14,
egvaninwegen, nth} @wpi.edu

Joseph Jay Williams
VPAL Research
Harvard University
Cambridge, MA 02138
joseph_jay_williams@harvard.edu

This section has been published at:

Ostrow, K., Selent, D., Wang, Y., Van Inwegen, E., Heffernan, N. & Williams, J. (2016). The Assessment of Learning Infrastructure (ALI): The Theory, Practice, and Scalability of Automated Assessment. *In the Proceedings of 6rd International Conference on Learning Analytics & Knowledge*.

ABSTRACT

Researchers invested in K-12 education struggle not just to enhance pedagogy, curriculum, and student engagement, but also to harness the power of technology in ways that will optimize learning. Online learning platforms offer a powerful environment for educational research at scale. The present work details the creation of an automated system designed to provide researchers with insights regarding data logged from randomized controlled experiments conducted within the ASSISTments TestBed. The Assessment of Learning Infrastructure (ALI) builds upon existing technologies to foster a symbiotic relationship beneficial to students, researchers, the platform and its content, and the learning analytics community. ALI is a sophisticated automated reporting system that provides an overview of sample distributions and basic analyses for researchers to consider when assessing their data. ALI's benefits can also be felt at scale through analyses that crosscut multiple studies to drive iterative platform improvements while promoting personalized learning.

Categories and Subject Descriptors

K: Applications to Education. K.3: Computers and Education. I.2.2: Automatic Programming. G.3: Probability and Statistics.

General Terms

Measurement, Documentation, Experimentation, Standardization.

Keywords

Assessment of Learning Infrastructure, Automated Analysis, Randomized Controlled Experiments at Scale, The ASSISTments TestBed, Universal Data Reporting, Tools for Learning Analytics.

INTRODUCTION

An immense community of researchers, educators, and administrators seeks to enhance the effectiveness of educational practices. Those invested in K-12 education struggle not just to enhance pedagogy, curriculum, and student engagement, but also to harness the power of technology in ways that will optimize learning. Researchers often fall back on observational studies or turn to data mining large longitudinal datasets due to the difficulties inherent to conducting student-level randomized controlled experiments (RCEs) in authentic learning environments. Software for sharing educational data has driven tremendous progress in educational research and best practices. For instance, the Pittsburgh Science of Learning Center's DataShop [8], funded by the National Science Foundation, provides an extensive database of educational datasets for post hoc data mining and analysis. However, the pace and power of educational research would increase drastically if researchers had easier access to environments in which they could design, implement, and analyze hypothesis driven experiments. The RCE remains the "gold standard" in determining causal relationships and was referred to when the U.S. Department of Education advocated for K-12 schools to apply basic findings from cognitive science to improve educational practices [16]. Without the assistance of scalable technologies, it has been difficult for researchers to answer the call to conduct RCEs within authentic academic settings [6] due to the high cost of establishing and maintaining sample populations, the complications inherent to randomization at the teacher-level (i.e., vast samples are required), and the often invasive curriculum restrictions necessary to establish sound controls.

When designed with flexibility and collaboration in mind, online learning platforms offer a unique and scalable approach to educational research and data analysis. Users of online learning platforms (i.e., students and teachers) create hundreds of thousands of data points each day, with databases of rich learner information growing exponentially as platforms gain popularity and validity as powerful learning aids. Beyond achievement measures, these systems provide opportunities to collect information including (but not limited to) behavior and affect [2, 17], learning interventions within content or feedback [14, 15], and interactions between skill domains that help guide curriculum development [1]. Through flexibility in content design, manipulation, and delivery, researchers are able to tap into the elements that drive effective learning within authentic K-12 classroom environments. When content can be manipulated to include parallel assignments, fashioned as conditions within RCEs, researchers are able to determine best practices and work toward personalized learning. Further, designing these environments with the open, collaborative, and perhaps even competitive design of RCEs in mind can strengthen internal validity and promote open source data reporting for review and replication of findings upon publication [11]. By allowing data scientists, educational researchers, and K-12 educators to work collaboratively within online learning platforms, all are empowered to dynamically evaluate and improve the effectiveness of the platform and its content while fostering growth in learner analytics.

2.1 Research in the ASSISTments TestBed

ASSISTments is a unique online learning platform that was designed with educational research as one of its primary goals [5]. The platform is used for both classwork and homework by over 50,000 users around the world, and provides students with immediate feedback and rich tutorial strategies and teachers with powerful assessment through a variety of reports that pinpoint where students are struggling and empower data driven teaching [5]. Recent funding from the NSF has allowed ASSISTments to promote educational research at scale through the development of the ASSISTments TestBed (www.ASSISTmentsTestBed.org). External researchers can use the TestBed to embed studies within ASSISTments content and non-invasively tap into our user population at virtually no cost and in a fraction of the time previously required to run experiments within K-12 environments.

The process of conducting an RCE within the TestBed typically involves researchers modifying preexisting certified content to include treatment interventions and student-level random assignment. The latter feature makes the TestBed a unique and robust tool for conducting research; rather than delivering the same treatment condition to all students within a particular class, students in the same class will be randomly assigned to different conditions while participating in the same assignment (i.e., content, feedback, or delivery may vary from student to student). The library of certified ASSISTments content consists primarily of middle and high school mathematics skills, with content organized and tagged by Common Core State Standard [10]. However, this library has grown to include content in physics, chemistry, and electronics, and researchers are able to develop their own content for experimentation in other domains.

Figure 1 depicts a simple study design implemented within the ASSISTments TestBed. Inclusion of a student in this type of study is dependent on her ability to access video content (note that many schools block video servers like YouTube). When the student begins her assignment, she must first pass a "Video Check," or a standard problem that serves as password protection to study participation. If the student can access video, she enters the 'password' provided in the short clip as her answer, and her correct response serves as the "Then" in an "If-Then" routing structure. If the student enters anything other than the password as a response, she is provided a default assignment without video content and is not considered a study participant. While this process attempts to control for technical issues, it does not demand the fidelity of study participants (i.e., we cannot currently track viewing statistics for embedded videos). Upon being routed into the study depicted in Figure 1, students are randomly assigned into one of two conditions using a "Choose Condition" routing structure. Note that although two conditions are presented here for simplicity, the system is able to compare any number of conditions. The platforms approach to random assignment will be discussed further in Section 3.1.2.

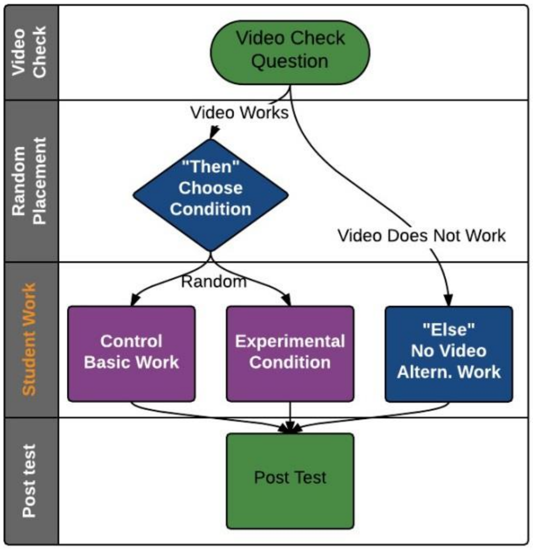


Figure 1. A simple research design that can be built using the ASSISTments TestBed to compare learning interventions.

In the present example, there are three possible paths that a student may follow as she progresses through her assignment (the specific trace of these paths will become important in the automated reporting and analysis of student performance presented in Section 3). For each student, regardless of path, ASSISTments logs substantial data detailing performance as the student progresses through the assignment. This data includes binary measures of problem accuracy (i.e., a correct or incorrect first response), the students first action (i.e., an attempt vs. requesting tutoring), the number of attempts per problem, the number of feedback interactions per problem (i.e., hints requested or scaffolds seen), whether or not the student saw the bottom out hint (i.e., the correct answer, provided to keep the student from getting stuck within the assignment), and start and end times for each problem. For researchers with a fine-toothed comb, ASSISTments can also provide logged information at the action level, detailing each step taken within a problem. ASSISTments is also able to track user information that is ultimately helpful to researchers, including data on the students performance in the system prior to their inclusion in a study, student characteristics (i.e., gender, age), and additional variables at the class and school levels. Through use of the TestBed, this information is consolidated, anonymized, and provided to researchers through unified reports (depicted in Section 3.1.1) to enhance the ease with which RCEs are conducted at scale.

2.2 Utility of Automated Data-Preprocessing

With students accessing experiments naturally in authentic learning environments, sample populations increase as a function of time. For instance, within three months of deploying a study within ASSISTments, a researcher may accrue 740 participants. This process does not require direct interaction between researcher and teachers, although some researchers choose to work directly with local classrooms to establish stronger controls. As external researchers are unfamiliar with the ASSISTments database and the inner workings of the platform, universal data reporting and preprocessing techniques were designed to ease the hurdle of interpreting system output. Without preprocessing, a researcher analyzing data from the study depicted in Figure 1 would need to use raw data to decipher whether students should be included in analyses, what condition each student experienced, details pertaining to each students experience within that condition (i.e., how many problems were completed, their content, and all associated performance data), and how each student performed at posttest. While such rich information is helpful in analyzing a study, providing researchers with a surplus of data necessitates larger and more complex datasets that must still meet ease of use requirements. Although different researchers focus on different information (as it applies to their particular hypotheses), an infrastructure for data preprocessing, restructuring, and reporting was necessary to bring ASSISTments to the next level as a shared scientific instrument for educational research.

In the following sections we discuss the creation of an automated reporting and analysis system built to provide researchers with data logged from RCEs conducted within the ASSISTments TestBed. The Assessment of Learning Infrastructure (ALI) builds upon existing technology to foster a symbiotic relationship beneficial to students, researchers, the platform and its content, and the science of learning. Evolving from a universal data logging and retrieval tool, ALI is quickly becoming a sophisticated system for automated analysis, offering researchers an overview of their sample population and conducting a selection of analyses for consideration when assessing data. The benefits of ALI can also be felt at scale, with analyses spanning content to drive platform improvements with the long-term goal of personalizing learning.

ALI IN THEORY

The Assessment of Learning Infrastructure is an automated research assistant that, while not meant to replace the researcher, is meant to lighten the load of working with large data files output from RCEs conducted within the ASSISTments TestBed. ALI alerts the researcher to new data, presents that data in a meaningful way, tentatively examines effects observed between conditions, and flags potential threats to validity. On a weekly basis, as well as on demand, ALI consults all logged information pertaining to a study and conducts preliminary analyses on student participation and performance (described further in Section 3). The potential benefits of automated reporting and analysis are broad; in the next four sections we briefly discuss how ALI's success will affect ASSISTments and its users, researchers and the Testbed, and the greater learning analytics community.

2.3 Benefits to ASSISTments Users

ALI's work at scale will help to guide the development of stronger learning interventions and, eventually, drive personalized learning within ASSISTments. Research conducted within the TestBed is unique in that while researchers are able to alter content and deliver versatile interventions as previously exemplified in Figure 1, such manipulations are not invasive. Study participation and student performance within an assignment is passively logged. A student may notice that some of her assignments include video feedback or have extra survey questions while others do not, but she is not informed that she is participating in an RCE. A primary goal driving the TestBed's ability to implement RCEs within ASSISTments is the provision of normal instructional practice and interventions that do not compromise learning.

ALI is also beneficial to teachers, as the infrastructure is able to separate rich study information from daily assessment data. Teachers are responsible for assigning content within ASSISTments to their students. Although it seems as though research designs created in the TestBed would complicate daily assessment, class and student reports have been designed such that teachers are provided pertinent information in a clean and concise manner. This low profile approach to conducting research maintains a highly participatory subject pool. Teachers wishing to conduct action research within their classes may do so by working with the TestBed as well, although most prefer to use day-to-day reports to guide their teaching practices rather than large automated data files.

2.4 Benefits to the Researcher

For those conducting RCEs within the ASSISTments TestBed, ALI plays the role of research assistant. The infrastructure intelligently communicates with researchers when new data is available for analysis and provides an overview of the sample distribution across conditions to signify the power of current analyses. Although researchers will undoubtedly run their own in depth analyses, standard high-level analyses can be automated to save time and reduce monotony. For example, ALI's ability to trace a student's path through an assignment allows the infrastructure to infer what condition the student experienced. This allows ALI to test for differential attrition rates across conditions and notify the researcher of apparent selection biases. This simple analysis can serve as a beneficial warning against analyzing posttest results due to potential threats to internal validity. Combined with the data preprocessing and sophisticated reporting that ALI's analytics are built upon, these notifications are often enough to save researchers from hours of wasted labor.

2.5 Benefits to the Platform

When considered at scale, ALI's capabilities for data reporting and analysis contribute to the enhancement of the ASSISTments platform by supporting practical improvements to content and feedback without interrupting student learning. As researchers collaborate and compete to design interventions within the ASSISTments TestBed, it will grow increasingly possible to evaluate interventions at scale, both across skills and longitudinally within students. Ideally, the best version of content and delivery observed (to date) for a particular skill would be delivered to students as the control condition in new RCEs. Through this approach, each study offers the potential for iterative improvement as experiments are launched and re-launched, capturing key features of design-based educational research methodology [3]. Such improvements additionally benefit users through the predicted outcome of enhanced learning gains and researchers through the rapid succession and enhanced validity of positive findings.

ALI's ability to analyze at scale will also help the ASSISTments team to quickly isolate and remove ineffective interventions. It is our goal that in the near future, ALI will conduct robust analyses across multiple studies while considering student, class, and school level characteristics. Roughly speaking, ALI will allow ASSISTments to personalize learning by better understanding why certain educational practices and interventions work for certain students but not for others.

Benefits for Learning Analytics

How can ALI and the promotion of infrastructures like ALI within other learning platforms benefit the learning analytics community? At its very core, ALI answers the general call of learning analytics, in that the infrastructure "emphasizes measurement and data collection as activities that institutions need to undertake and understand, and focuses on the analysis and reporting of the data" [20]. A strong focus on providing universal measures of learning garnered from authentic learning environments will strengthen the validity of findings from a broad range of interventions that seek to isolate best practices in education.

Further, much attention in the broader scientific and psychological research communities has recently befallen the general inability to replicate research findings [7, 11]. The same is likely true for educational research, with little emphasis placed on data accountability. Perhaps the best outlet for promoting open data, the Pittsburgh Science of Learning Center's Data Shop [8] takes a number of steps in the right direction with regard to shared datasets that promote open, replicable, and sound science. ALI builds upon the PSLC's model of open data reporting by establishing stable, timestamped links to every data analysis report ever provided to a researcher throughout the duration of their work within ASSISTments. Researchers are asked to cite the report from which they draw data for final analyses and publication (explained further in Section 3.1.5). References to these reports will also drastically increase the availability of preprocessed and anonymized educational datasets for researchers wishing to mine big data without designing specific interventions.

In some ways, ALI is also an extension of industry track research focused on learning analytics; companies like Google and Microsoft increasingly implement large-scale experimentation in online learning environments to consider reporting metrics and analytic methods that meet practical goals rooted in scientifically sound evidence [9]. If infrastructures like ALI were incorporated into other learning platforms, similar large-scale experimentation could easily be promoted for its importance to learning analytics.

ALI IN PRACTICE

The Assessment of Learning Infrastructure has grown considerably over the past year. ALI began as a robust SQL query to the ASSISTments database to retrieve unified information across multiple studies and to present it to researchers in a single format. Ease of use requirements, communication considerations, and feedback from external researchers has helped ALI to grow beyond data preprocessing and reporting into a tool for learning analytics at scale. The following sections discuss how ALI has evolved and provides examples of the infrastructure's current capabilities in reporting, analyzing, and communicating data from RCEs conducted within the ASSISTments TestBed.

ALI's Current Capabilities

Data Reporting at Scale

When a researcher submits a study to the ASSISTments TestBed, details about the study and the researcher's contact information are entered into ALI's study repository. Although researchers can request immediate data analysis reports on demand, ALI defaults to a weekly inspection of each study in the database and makes a decision regarding whether or not to process a data analysis report for the researcher. This decision is based on measured increases in sample size. Due to common curricula structures, certain skills are only used at specific times of year and thus, an assignment with an embedded study may be highly popular during the Fall term but not the Spring term. When ALI inspects the study's logged data, at least three new participants since the last ALI communication are required to trigger a new data report.

As teachers using ASSISTments are able to make copies of assignments and alter their content, ALI is also able to detect when teachers have assigned a copy of a study. ALI is sophisticated enough to recognize when a copy is identical to the original study and include data associated with the copy in each report. If a copy of the study has been altered (i.e., problems were removed or sections were changed), ALI does not report data associated with the copy. This ensures that researchers receive all data associated with their experiment without corrupt data.

Once ALI has determined that new data is available, several robust SQL queries are run on the ASSISTments database. Three major queries are used to a) retrieve student data detailing student, class, and school level characteristics for each student recorded prior to random assignment (see Table 1; field definitions are beyond the scope of this paper but are available in our glossary at [13] for additional reference), b) retrieve problem level data (see Table 3), and c) detect the problem set structure (i.e., the paths depicted in Figure 1) for each student with logged data. These three queries provide ALI with the information necessary to establish reports and conduct automated analysis. By working closely with researchers throughout the development of ALI, we have designed four different universal data representations in an attempt to meet dynamic research needs. Subsets of data exemplifying each type of report are provided below. Table 2 shows fields typical to the Action Level file. This file offers the finest granularity of data logged by ASSISTments as a student works through an assignment. Each row provides information pertaining to a single step within a problem (i.e., when the problem is initiated, or when the student asks for a hint). A subset of the Problem Level file is depicted in Table 3. This file provides the same data as that found in the Action Level file, but the granularity has increased. Each row provides information pertaining to a single problem, with actions collapsed across columns. Student Level files, as depicted in Table 4, offer the coarsest granularity of data reporting. In this type of file, each row provides information pertaining to the entire assignment for a single student. For each feature or action, problem information is presented across columns in the order in which the student experienced the assignment, with the number of columns for each feature extrapolated to the maximum number of problems experienced by any student in the file. An alternative version of Student Level data is also provided in which each student assignment is represented by a series of rows, each representing a feature for problems displayed across columns (akin to a pivot table of the file described in Table 4). Full examples of each data file are available at [13] for further consideration. Links to each data file are gathered and presented to the researcher in a single, organized communication, depicted in Figure 2 and discussed further in Section 3.1.5.

When preprocessing is complete and all data files have been compiled, ALI sends analytic commands to Rserve, an extension to the R programming language that allows for other applications to call R functions via a TCP/IP connection [19]. The ASSISTments team created a client side API to interact with Rserve, allowing ALI to send requests to R. Because Rserve is not multithreaded, several instances of Rserve run on separate ports on the ALI server. The server is designed to recycle existing connections, with a connection pool equal to the maximum number of threads used by ALI. This allows several data

Table 1. A theorized subset of student historical data. Each row contains student, teacher, and school characteristics linked to a particular student, using information sourced prior to random assignment.

Student	Class ID	Grade	School ID	Guessed Gender	Birth Year	Prior HW Completion %	Prior Class HW Completion %	Normalized HW Mastery Speed
A	1007475	8	5597	Male	2001	0.83	0.88	0.33
B	1180278	8	5597	Male	2001	0.76	0.88	0.03
C	1180278	8	5597	Male	2001	0.76	0.88	0.03
D	1322778	7	2342	Female	2002	0.95	0.97	-0.39

Table 2. A theorized subset of an action level data file. Each row represents a single action within a single problem as experienced by a student. This is the finest granularity of data reported by ALI.

Student	Problem ID	Sub-Problem ID	Order	Action Type	Timestamp	Answer	Correctness
A	PRAUVJS	806533	1	Start	08/26/15 15:25:26	--	--
A	PRAUVJS	806533	2	Hint	08/26/15 15:25:52	--	--
A	PRAUVJS	806533	3	Answer	08/26/15 15:26:40	18.2	TRUE
A	PRAUVJS	806533	4	End	08/26/15 15:26:42	--	--
A	PRAVKJX	833840	1	Start	08/26/15 15:26:43	--	--

Table 3. A theorized subset of a problem level data file. Each row contains all the information linked to a single problem as experienced by a student. This is a popular form of data for student modeling and analytics.

Student	Assignment ID	Problem ID	Correct	Answer	Hints	Attempts	Start Time	End Time
A	1007475	PRAUVJS	1	18.2	0	1	08/26/15 15:25:26	08/26/15 15:26:42
A	1007475	PRAVKJX	1	14.3	0	1	08/26/15 15:26:43	08/26/15 15:27:45
A	1007475	PRAVKHT	1	6.4	0	1	08/26/15 15:27:50	08/26/15 15:28:47
B	1180278	PRAUVJX	0	22.8	2	3	08/26/15 17:14:22	08/26/15 17:15:42
B	1180278	PAVKGZ	0	7.2	0	2	08/26/15 17:15:43	08/26/15 17:17:31

Table 4. A theorized subset of a student level data file. Each row contains all information linked to a single student's experience of the problem set. Assignment information is presented across columns in the order in which the student experienced problems.

Student	Assignment ID	Late	Mastered	Correct Q1	Correct Q2	Correct Q3	Answer Q1	Answer Q2	Answer Q3
A	1007475	1	1	1	1	1	18.2	14.3	6.4
B	1180278	0	0	0	0	1	17	14.1	6.4
C	1180278	1	0	0	1	--	24.6	14.3	--
D	1322778	0	1	1	1	1	18.2	14.3	6.4

analysis reports to occur simultaneously, all using different Rserve connections. This approach lowers the turnaround time when a researcher actively requests data. It also keeps weekly reporting as efficient as possible, as all datasets in ALI's study repository are assessed weekly for potential reporting.

Smart Structures

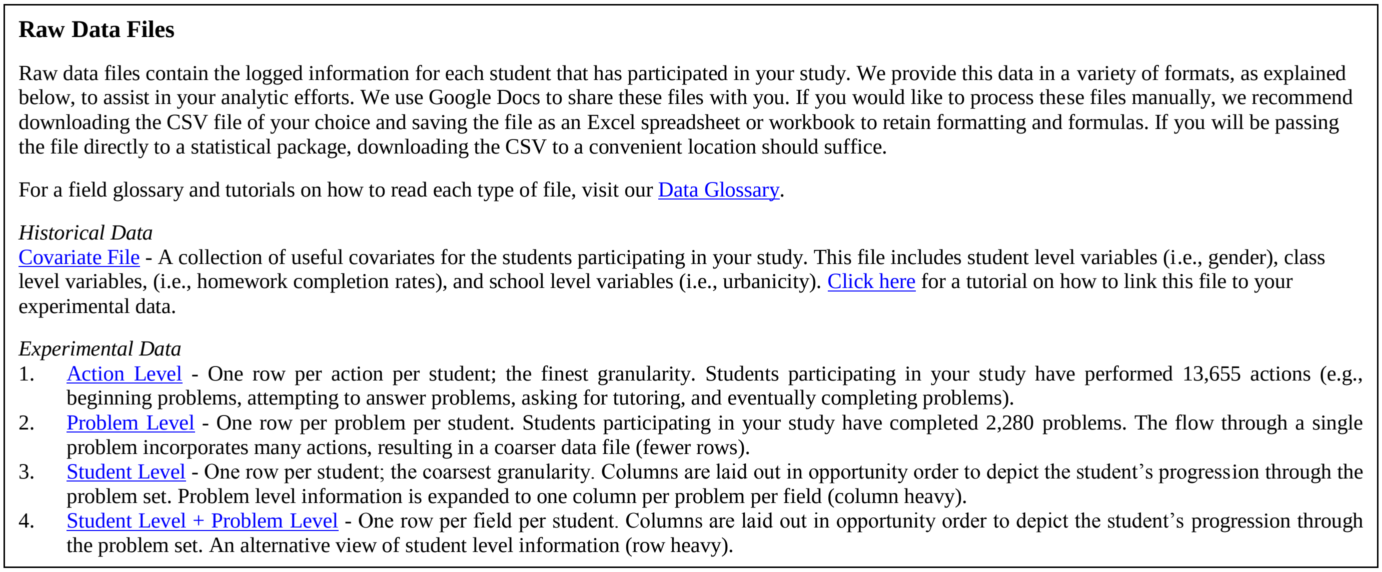
In order to determine *what* to analyze, ALI must first process the structure of a study and trace each student's path through the assignment (as previously discussed in relation to Figure 1). As ALI parses the assignment's structure, the infrastructure is able to make intelligent decisions upon meeting certain section types within the design. This is accomplished by recursively generating the assignment's reported structure into tree form. Within the Problem Level data file presented in Table 3, each problem is labeled with a path, similar to that used when traversing a set of folders within an operating system. ALI steps through each problem path for each student to establish an intuitive structure of the study and to cluster students by condition.

RCEs within the ASSISTments TestBed are designed by taking advantage of a variety of section types offered by the platform. The "If-Then" routing discussed in Section 1.2 was an example of a section type. When ALI observes an If-Then structure that issues a routing standard like a "Video Check," the infrastructure intelligently conducts its analyses on students assigned to the study and disregards students routed to alternative content. Similarly, studies often employ parallel experimental and control conditions delivered using a section type referred to as a "Choose Condition." This section type is used to drive random assignment. The "Choose Condition" depicted in Figure 1 included two parallel conditions: an assignment with video content and a control assignment with traditional text content. Currently, in order for ALI to recognize an assignment as a research study, a "Choose Condition" must be present when mapping the assignment's structure. ALI then assesses logged data within each condition and considers any section immediately following these conditions as a subsequent posttest (see Figure 1). Using this information, ALI is able to aggregate statistics and perform a selection of simple analyses across problems and students.

It is important to note that research designs within the ASSISTments TestBed can grow far more complex than the simple structure presented herein. When assignments include nested section types and multiple "If-Then" routing standards, ALI currently has difficulty interpreting condition and isolating posttest content. In its current form, ALI is only meant to assist researchers with the analysis of common design patterns. Future work, discussed in Section 5, will expand ALI's ability to intelligently parse studies using tagging rules set forth by the researcher.

Selection Bias

After establishing a study's structure and sample distribution, ALI is able to assess assignment completion rates across conditions and alert researchers to potential threats to internal validity due to selection bias. ALI records the observed number of students in each condition that began the assignment, and considers logged assignment end times to consider the proportion of students that ultimately completed the assignment. The observed distribution is then compared to the expected distribution of proportional attrition in a normal sample. A Chi-squared analysis is used to determine if the observed distribution of attrition significantly differs from the expected distribution. ALI then flags conditions that have a reliably different attrition rate and alerts the researcher of a potential threat to internal validity. Without considering differential attrition across conditions, an analysis of posttest performance may inaccurately suggest the significant effect of a particular condition that was actually driven by the disproportionate loss of weaker students. This simple analysis, presented to researchers as shown in Figure 3, may help even the most seasoned experts to accurately assess their sample. It is important to note that while ALI provides this warning, the infrastructure still releases all data to the researcher and never prohibits the researcher from further analysis. The goal of ALI's selection bias assessment is not to impede or prevent analysis, but rather to advocate sound analytic practices.



Raw Data Files

Raw data files contain the logged information for each student that has participated in your study. We provide this data in a variety of formats, as explained below, to assist in your analytic efforts. We use Google Docs to share these files with you. If you would like to process these files manually, we recommend downloading the CSV file of your choice and saving the file as an Excel spreadsheet or workbook to retain formatting and formulas. If you will be passing the file directly to a statistical package, downloading the CSV to a convenient location should suffice.

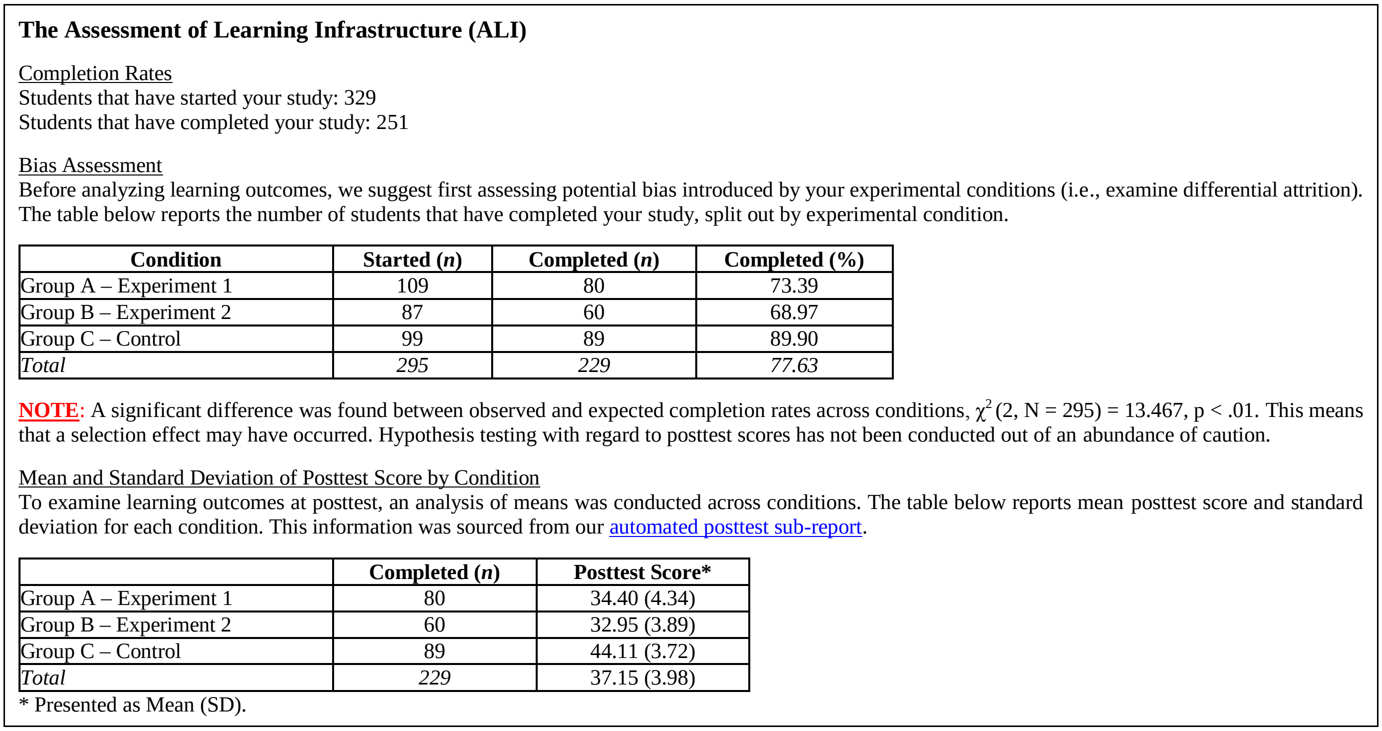
For a field glossary and tutorials on how to read each type of file, visit our Data Glossary.

Historical Data
Covariate File - A collection of useful covariates for the students participating in your study. This file includes student level variables (i.e., gender), class level variables, (i.e., homework completion rates), and school level variables (i.e., urbanicity). Click here for a tutorial on how to link this file to your experimental data.

Experimental Data

1. Action Level - One row per action per student; the finest granularity. Students participating in your study have performed 13,655 actions (e.g., beginning problems, attempting to answer problems, asking for tutoring, and eventually completing problems).
2. Problem Level - One row per problem per student. Students participating in your study have completed 2,280 problems. The flow through a single problem incorporates many actions, resulting in a coarser data file (fewer rows).
3. Student Level - One row per student; the coarsest granularity. Columns are laid out in opportunity order to depict the student's progression through the problem set. Problem level information is expanded to one column per problem per field (column heavy).
4. Student Level + Problem Level - One row per field per student. Columns are laid out in opportunity order to depict the student's progression through the problem set. An alternative view of student level information (row heavy).

Figure 2. A thoroughly developed universal reporting of logged data from students participating in RCEs. Each file presented here is discussed further, including depictions of file subsets, in Section 3.1.1.



The Assessment of Learning Infrastructure (ALI)

Completion Rates
Students that have started your study: 329
Students that have completed your study: 251

Bias Assessment
Before analyzing learning outcomes, we suggest first assessing potential bias introduced by your experimental conditions (i.e., examine differential attrition). The table below reports the number of students that have completed your study, split out by experimental condition.

Condition	Started (*n*)	Completed (*n*)	Completed (%)
Group A – Experiment 1	109	80	73.39
Group B – Experiment 2	87	60	68.97
Group C – Control	99	89	89.90
Total	*295*	*229*	*77.63*

NOTE: A significant difference was found between observed and expected completion rates across conditions, χ^2 (2, N = 295) = 13.467, p < .01. This means that a selection effect may have occurred. Hypothesis testing with regard to posttest scores has not been conducted out of an abundance of caution.

Mean and Standard Deviation of Posttest Score by Condition
To examine learning outcomes at posttest, an analysis of means was conducted across conditions. The table below reports mean posttest score and standard deviation for each condition. This information was sourced from our automated posttest sub-report.

	Completed (*n*)	Posttest Score*
Group A – Experiment 1	80	34.40 (4.34)
Group B – Experiment 2	60	32.95 (3.89)
Group C – Control	89	44.11 (3.72)
Total	*229*	37.15 (3.98)

* Presented as Mean (SD).

Figure 3. Current ALI analytic reporting. Available analyses include a Chi-squared test comparing the observed and expected sample distributions, simple hypothesis testing, and an analysis of means on posttest performance between conditions. Note that these analyses are currently driven by the structure of the assignment as parsed by ALI from Problem Level data. Future work includes allowing researchers to tag their study with items of interest to automate analysis with greater sophistication.

Simple Hypothesis Testing

After conducting a selection bias assessment, ALI progresses to a set of simple hypothesis tests with regard to posttest performance. If ALI detects a posttest section when parsing an assignment's structure, the infrastructure compares performance across conditions by referring to the previously aggregated group distributions. ALI approaches posttest analysis much like a researcher would: if only two conditions are detected within the study, ALI conducts a t-test, while if more than two conditions are detected, ALI conducts an Analysis of Variance (ANOVA). ALI currently has the API to support simple univariate and multivariate analyses including ANOVA, ANCOVA, MANOVA, and MANCOVA. ALI stores all input parameters for a given statistical test in a single object. The parameters are extracted from this object and transformed into the appropriate R function calls through the Rserve API communication. Results are accumulated and presented to the researcher alongside an analysis of means, as shown in Figure 3, allowing the researcher to observe the direction of the reported effect. Note that in the present example, ANOVA results are not presented to the researcher out of an abundance of caution due to ALI's detection of a potential selection bias. Our goal in restricting this information is strictly in the promotion of sound scientific inquiry. It should also be noted that covariates are not presently considered in ALI's hypothesis testing. Future work will control for student, class, and school level characteristics sourced from the historical student data file (see Table 1) by using ANCOVA or MANCOVA approaches in an attempt to explain additional variance in learning outcomes.

Data Storage and Researcher Output

When ALI's automated analysis is complete, ALI stores all data files and analytic output on Google Drive in archival quality. This data cannot be altered but can be downloaded by anyone. For active studies, copyright protection will be placed on new data analysis reports for one year from the study's initial run date. This means that researchers will have a full calendar year to publish on their findings before their data becomes freely available to the public.

ALI communicates to researchers via email, providing a link to a stable URL for a Google Doc housing that week's data analysis report. The Doc contains links to all raw data files, as shown in Figure 2, and provides automated analysis as depicted in Figure 3. The creation of this Google Doc is automated, based on an HTML template file that uses custom tagging conventions to insert variables with dynamic text or data. Using this method, the same report can be generated multiple times or across multiple assignments with changes to only the pertinent information. This allows for customized reporting based on the results of ALI's analysis. The Google Doc report also provides researchers with links to additional resources including a glossary explaining features of the data and video tutorials on how to understand each file type (available at [13]).

When researchers are ready to publish findings, a condition of working with the ASSISTments TestBed requires that they include a reference in their work to the stable record from which they sourced the data files used for final analyses. This approach allows reviewers and secondary researchers to gain access to raw study data, thereby encouraging replication and open science [11]. In addition to the raw data, secondary researchers will also be able to use these references to access ALI's analytic report, including all automated analyses.

ANALYSIS AT SCALE

Although ALI's analytic structure is still somewhat rudimentary, considered at scale, comparisons of findings from multiple studies can offer substantial insights for the ASSISTments platform and in more general terms, for the learning analytics community. By simultaneously examining attrition outcomes across studies it becomes possible to make claims about the quality of interventions that crosscut multiple skills. As ALI's analytical capabilities increase, analysis at scale will grow even more powerful.

As a proof of concept of the potential benefits of automated analysis at scale, ALI was run across a special dataset including 25 studies that are currently running within ASSISTments. This file was created for another sophisticated approach to modeling student performance across multiple studies [18], but serves as a perfect example of ALI's capabilities at scale. In the spirit of open data, this file is available for reference at [12]. The studies in this file were selected from a group of 126 studies currently running within the ASSISTments platform based on the following criteria:

- Studies selected contained at least 50 students within each condition that completed the assignment.
- Studies selected were designed within Skill Builders, a mastery learning based assignment that considers predefined thresholds for student completion (i.e. by default, to complete the assignment the student must solve three consecutive problems accurately).

As most of the studies in this file were built prior to the implementation of automated path-logging (which drives ALI's ability to read in the structure of the study and infer a condition for each student), condition was manually traced and logged for each student based on his or her observed problem sequence. A number of these studies were also built before the availability of If-Then routing and subsequent checks for internal validity (i.e., the "Video Check" explained in connection to Figure 1). As such, it is difficult to tell if students experienced technical difficulties during the course of a condition. To analyze this dataset using all of the capabilities that ALI has with recently designed studies, we manually notated flags regarding the observed fidelity of conditions. This flagging also included whether students 'tested out' of the condition experience (i.e., if a student was assigned to a condition in which the treatment was presented through feedback but answered the first 3 consecutive problems accurately, they did not ultimately experience the treatment). As only three of the studies in this file contained valid posttest information, we only present ALI's selection bias assessment for consideration at scale (see Table 5).

The 25 studies presented in Table 5 span a variety of investigations including: assessing the effect of various types of video tutoring (i.e., pencasts, teacher recorded instruction, online resources) compared to traditional text-based tutoring across multiple designs (i.e., using scaffolding, using hints, as an intervention to wheel-spinning [2], or provided based on student choice), investigating the manipulation of content (i.e., interspersing learning with humor through comics in content or feedback, asking students to gauge their confidence in solving problem content, and altering student mindset (as inspired by [4]), and challenging cognitive principles (i.e., mental representations, and alterations in the consistency of math equations). Assignment names, as presented in Table 5, are tagged with the grade level and domain of the skill content as defined by Common Core State Standards [10]. Despite differences in domain and experimentation, ALI is able to provide a sense of condition quality across studies at scale.

The results of the simple Chi-squared analyses in Table 5 may not seem significant at first, but are actually quite insightful at scale. In studies with two conditions, experiment vs. control (20 comparable sets of the 25 shown in Table 5), the control groups showed less attrition in 15, while the experimental groups showed less attrition in only five. On its own, this comparison suggests that experimental conditions correlate with higher attrition rates. However, this attrition is only significantly different than that of a normally distributed sample in five studies ($p < .05$), with experimental conditions showing significantly more attrition than expected in four studies, and control conditions showing significantly more attrition than expected in only a single study.

At scale, these analyses can help researchers and developers determine which interventions are effectively retaining students,

Table 5. ALI's Bias Assessment at Scale - Observed Distributions and Chi-Squared Analyses Across 25 Problem Sets

Problem Set by Condition	*Started (n)*	*Completed (n)*	*Completed (%)*	*df*	χ^2	*p*
Multiplying Mixed Numbers 5.NF.B.4a	**775**	**466**	**60.13**	**1**	**5.30**	**0.021***
Control	403	258	64.02			
Experiment	372	208	55.91			
Understanding Vocabulary About Circles G-C.A.2	**695**	**674**	**96.98**	**1**	**4.87**	**0.027***
Control	330	325	98.48			
Experiment	365	349	95.62			
Equivalent Expression 6.EE.B.4	**273**	**240**	**87.91**	**1**	**0.39**	**0.532**
Control	138	123	89.13			
Experiment	135	117	86.67			
Writing Inequalities from Situations 6.EE.B8	**627**	**539**	**85.96**	**1**	**2.21**	**0.138**
Control	338	297	87.87			
Experiment	289	242	83.74			
Dividing Mixed Numbers 6.NS.A.1	**1864**	**1285**	**68.94**	**1**	**0.99**	**0.321**
Control	943	660	69.99			
Experiment	921	625	67.86			
Finding Expected Value SS.MD.B.5	**457**	**337**	**73.74**	**1**	**0.06**	**0.802**
Control	224	164	73.21			
Experiment	233	173	74.25			
Conditional Probability SS-CP.A.3	**515**	**366**	**71.07**	**1**	**0.70**	**0.401**
Control	281	204	72.60			
Experiment	234	162	69.23			
Permutations and Combinations SS-CP.B.2	**540**	**456**	**84.44**	**1**	**0.00**	**0.958**
Control	265	224	84.53			
Experiment	275	232	84.36			
Basic Logarithm Manipulation F-BF.B.5	**136**	**121**	**88.97**	**1**	**0.21**	**0.645**
Control	62	56	90.32			
Experiment	74	65	87.84			
Properties of Exponents 8.EE.A.1	**545**	**435**	**79.82**	**1**	**0.24**	**0.626**
Control	264	213	80.68			
Experiment	281	222	79.00			
Intermediate Logarithm Manipulation F-BF.B.5	**205**	**169**	**82.44**	**1**	**8.44**	**0.004****
Control	102	92	90.20			
Experiment	103	77	74.76			
Solving ab^{ct} = d LE.A.4a	**147**	**122**	**82.99**	**1**	**0.01**	**0.914**
Control	72	60	83.33			
Experiment	75	62	82.67			
Finding Inverse Functions F-BF.B.4	**301**	**143**	**47.51**	**1**	**3.32**	**0.068†**
Control	145	61	42.07			
Experiment	156	82	52.56			
Composition of Functions F-BF.A.1c	**219**	**173**	**79.00**	**1**	**0.86**	**0.354**
Control	118	96	81.36			
Experiment	101	77	76.24			
Sequences F-BF.A.2	**382**	**241**	**63.09**	**1**	**0.20**	**0.658**
Control	198	127	64.14			
Experiment	184	114	61.96			
Comparing Values - Multiplying by Fractions 5.NF.B.5a	**129**	**121**	**93.80**	**1**	**1.59**	**0.208**
Control	69	63	91.30			
Experiment	60	58	96.67			
Converting Radians to Degrees F-TF.A.1	**245**	**226**	**92.24**	**1**	**0.23**	**0.631**
Control	129	120	93.02			
Experiment	116	106	91.38			
Trigonometric Ratios G-SRT.C.8	**307**	**266**	**86.64**	**1**	**0.91**	**0.341**
Control	141	125	88.65			
Experiment	166	141	84.94			
Pythagorean Theorem – Finding the Hypotenuse 8.G.B.7	**447**	**349**	**78.08**	**1**	**6.40**	**0.011***
Control	237	174	73.42			
Experiment	210	175	83.33			
Solving 1-Step Equations 7.EE.B.4a	**928**	**818**	**88.15**	**1**	**0.01**	**0.934**
Control	459	405	88.24			
Experiment	469	413	88.06			
Prime Factorization 6.NS.B.4	**1238**	**1058**	**85.46**	**2**	**0.97**	**0.616**
Control	430	369	85.81			
Experiment 1	399	345	86.47			
Experiment 2	409	344	84.11			
Order of Operations (No Exponents) 7.NS.A.3	**1231**	**1172**	**95.21**	**2**	**4.50**	**0.105**
Group A - Consistent/Neutral	597	574	96.15			
Group B - Inconsistent	300	287	95.67			
Group C - Mixed	334	311	93.11			

Note. †p < .10, *p < .05, **p < .01. *df* = Degrees of Freedom.

Table 5. ALI's Bias Assessment at Scale - *Continued*

Problem Set by Condition	*Started (n)*	*Completed (n)*	*Completed (%)*	*df*	χ^2	*p*
Multiplying Simple Fractions 5.NF.B.4a	**598**	**559**	**93.48**	**3**	**1.54**	**0.673**
Group A – No Choice + Text	142	131	92.25			
Group B – Choice + Text	222	211	95.05			
Group C – Choice + Video	76	71	93.42			
Group D – No Choice + Video	158	146	92.41			
Rotations 8.G.A.3	**306**	**186**	**60.78**	**1**	**0.82**	**0.365**
Experiment 1	145	92	63.45			
Experiment 2	161	94	58.39			
Reflections 8.G.A.3	**239**	**171**	**71.55**	**1**	**0.17**	**0.680**
Experiment 1	125	88	70.40			
Experiment 2	114	83	72.81			

Note. †p < .10, *p < .05, **p < .01. *df* = Degrees of Freedom.

and more importantly, critical design issues that drive students away. As many of these 25 studies were designed prior to the implementation of internal validity checks (i.e., assessing a student's technical abilities with video content), we believe that the analyses in Table 5 suggest higher attrition in experimental conditions because certain students were assigned to content that they had difficulty accessing. This finding would not likely hold true when considering studies run more recently, suggesting the importance of the recent implementation of If-Then routing. Future work with ALI at scale will help to confirm this hypothesis. Usability is a concern within any online learning system, and providing students with access to default assignments when they cannot access enriched content is a safe practice.

It is also important to consider the percentage of students excluded from analysis prior to the assessments presented in Table 5. Within all sets, an average of 22.85% of students did not actually experience condition and were removed from the sample prior to analysis. Students that fail to experience interventions implemented within feedback (due to mastery or performance at ceiling) provide valuable information to researchers regarding the raw (inflated) sample size required to achieve statistical power. Certain elements of a study's design, including the content domain (i.e., some topics are easier than others and students require less feedback on average), and the type of feedback provided (i.e., on demand feedback requires a larger raw population than feedback provided automatically upon the student's incorrect response), can have a significant impact on the raw sample size required to attain enough treated students to reliably detect effects. RCEs that consider interventions implemented strictly within problem content have fewer issues with regard to raw sample sizes as all students experience the intervention regardless of performance, easing potential issues surrounding intent-to-treat analyses.

Finally, analyzing the selection effects inherent to multiple assignments simultaneously allows ASSISTments to evolve more rapidly, providing benefits to users, researchers, and the learning analytics community. As the experimental conditions in Table 5 exhibited only 1.5% greater attrition on average than control conditions, it is possible that the benefits of these experimental interventions may still outweigh the increase in attrition. Additional data mining would be necessary to determine a standard at which the potential for emphasized learning gains within an experimental condition no longer outweighed the potential for increased attrition. However, regularly conducting this type of broad scale analysis across assignments could quickly isolate studies with conditions considered extremely detrimental, and the condition could be discontinued in order to limit the intervention's negative impact on students. ALI's automated analysis makes the process of intervention validation dramatically more efficient and robust. From these findings, and from future, more powerful iterations of ALI's at-scale capabilities, ASSISTments will be able to deliver rapid iterations of interventions with the goal of optimizing students' interactions with the system through enhanced usability and strengthened content and delivery methods.

LIMITATIONS & FUTURE WORK

As ALI is constantly evolving and gaining new capabilities, the version of the infrastructure presented here carries a number of limitations. As made apparent by the complex methods applied to consider ALI's effects at scale, the infrastructure is currently only able to recognize studies with logged path information. The implementation of path logging occurred in March 2015, and ALI is only able to reliably analyze studies that were created after this implementation. This limitation is compounded by ALI's inferences of the study design and posttest items. As studies within the ASSISTments TestBed can be designed using a number of complex, nested structures, ALI's current decisions about study designs are not exceptionally intelligent. A serious limitation of the work presented herein is that the infrastructure is currently only able to reliably recognize and analyze study designs with simple structures (i.e., "If-Then" routing, a single "Choose Condition," and a clear cut posttest section that directly follows an intervention).

While these limitations influence ALI's significance for the learning analytics community, they can easily be resolved through future work. One of our current focuses is the implementation of a tagging system that will allow researchers to identify pertinent sections of a study prior to its distribution. Using unified naming structures for the design of assignment sections within the building process (e.g., [experiment], [control], [posttest]), researchers will essentially be able to tell ALI exactly how to approach analysis. This will allow ALI to provide customized analysis and, potentially, refined data files that are preprocessed according to the researcher's distinct needs. Tagging will also allow for analyses that collapse similar treatment groups (i.e., experimental group 1 and experimental group 2 could both be tagged with [experiment] to denote that ALI should collapse these conditions), that isolate unconventional posttest problems (i.e., problems falling within a section that does not immediately follow a "Choose Condition"), and that assess growth models of student performance (i.e., by measuring pre- to posttest gains, or through more complex hierarchical models).

Future work for the ALI team also includes defining a powerful list of student, class, and school level variables for use as covariates in statistical analyses. Variables that have already been established include measures of each student's prior performance within ASSISTments, measures of their completion rate on classwork and homework assignments, and normalized values that compare the student's performance and attrition against that of their class. As such, future iterations of ALI's at-scale capabilities will also be able to control for particular student characteristics in order to assess the true variance established by experimental interventions. Additional content is also being built into ASSISTments and made available in the TestBed to collect self-report measures from students for use as possible covariates. Rich covariates will provide ALI with the ability to examine the effects of experimental interventions across groups while controlling for substantial variance, making automated analyses far more robust.

CONTRIBUTION

The learning analytics community will benefit greatly from the Assessment of Learning Infrastructure (ALI) and the promotion of similar infrastructures for other online learning platforms. Currently, very few learning technologies serve as scientific tools for researchers to conduct and communicate the findings of sound educational research at scale. By allowing researchers to conduct research within authentic learning environments through classwork and homework completed within online learning platforms, it is possible to collect rich log files that can be reported in universal formats and analyzed using automated processes. As a community, a strong focus on providing universal measures and analyses from these platforms will strengthen the validity of findings from a broad range of interventions that seek to isolate best practices in education. The broad dissemination of vast anonymized educational datasets will also propel the field toward more transparent, replicable, and reputable scientific practice, improving learning analytics for all.

ACKNOWLEDGMENTS

We acknowledge funding from multiple NSF grants (1440753, 1252297, 1109483, 1316736, 1535428, 1031398), the U.S. Dept. of Ed. (R305A120125, R305C100024, P200A120238), and ONR. We also thank our colleagues Yang Lu and Anthony Botelho for their roles in designing ALI. Thanks to S.O. & L.P.B.O.

REFERENCES

[1] Adjei, S.A. & Heffernan, N.T. (2015). Improving Learning Maps Using an Adaptive Testing System: PLACEments. In Conati, Heffernan, Mitrovic, & Verdejo (eds.) Proc of the 17th Int Conf on AIED. Springer, 517-520.

[2] Beck, J.E. & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. In Lane, Yacef, Mostow & Pavlik (eds.) Proc of the 16th Int Conf on AIED. Springer-Verlag, 431-440.

[3] Brown, A.L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. J of Learning Sciences, 2(2), 141-178.

[4] Dweck, C.S., Chiu, C., & Hong, Y. (1995). Implicit theories and their role in judgments and reactions: A world from two perspectives. Psychological Inquiry, 6(4), 267-285.

[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 of AIED, 24(4), 470-497.

[6] Institute of Education Sciences. (2003). Identifying and Implementing Educational Practices Supported by Rigorous Evidence: A User Friendly Guide. U.S. Dept of Ed. Washington, D.C.

[7] Ioannidis J.P.A. (2005). Why Most Published Research Findings Are False. PLoS Med 2(8): e124.

[8] Koedinger, K.R., Baker, R.S., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the EDM community: The PSLC DataShop. Handbook of EDM, 43.

[9] Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide. Data mining and knowledge discovery, 18(1), 140-181.

[10] National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). Common Core State Standards. Washington, DC: Authors.

[11] Open Sci Collab. (2015). Estimating the reproducibility of psychological science. Science, 349 (6251).

[12] Ostrow, K. (2015). Data for "The Assessment of Learning Infrastructure (ALI): The Theory, Practice, and Scalability of Automated Assessment." Accessed from http://tiny.cc/LAK2016-ALI

[13] Ostrow, K. & Heffernan, C. (2014). How to Create Controlled Experiments in ASSISTments. Retrieved from https://sites.google.com/site/assistmentstestbed/

[14] Ostrow, K.S. & Heffernan, N.T. (2014). Testing the Multimedia Principle in the Real World: A Comparison of Video vs. Text Feedback in Authentic Middle School Math Assignments. In Stamper, et al. (eds.) Proc of the 7th Int Conf on EDM, 296-299.

[15] Ostrow, K., Heffernan, N., Heffernan, C., & Peterson, Z. (2015). Blocking vs. Interleaving: Examining Single-Session Effects within Middle School Math Homework. In Conati, Heffernan, Mitrovic, & Verdejo (eds.) Proc of the 17th Int Conf on AIED. Springer, 388-347.

[16] Pashler, H., Rohrer, D., Cepeda, N. & Carpenter, S.K. (2007). Enhancing learning and retarding forgetting: Choices and consequences. Psychonomic Bulletin & Review. 14 (2), 187-193.

[17] San Pedro, M., Baker, R., Gowda, S., & Heffernan, N. (2013). Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In Lane, Yacef, Mostow & Pavlik (eds.) Proc of the 16th Int Conf on AIED. Springer-Verlag, 41-50.

[18] Selent, D., Patikorn, T., Heffernan, N. (Under Review). ASSISTments Dataset from Multiple Randomized Controlled Experiments. Submitted to the 3rd Annual ACM Conference on L@S.

[19] Urbanek, S. (2003). Rserve—a fast way to provide R functionality to Applications. In Hornik, Leisch, & Zeileis, Proc of the 3rd Int Workshop on DSC, ISSN 1609-395X. http://rosuda.org/rserve.

[20] U.S. Department of Education, Office of Educational Technology. (2012). Enhancing teaching and learning through educational data mining and learning analytics: An Issue Brief. Washington, DC.

Acknowledgement

Besides acknowledgement we did in individual sections above, I would like to show great gratitude for my advisor Neil Heffernan, and my reader Joseph Beck, for their support and advice. Also, many thanks to my beloved family, friends, colleagues in ASSISTments team, and great faculty of CS department!