2014-04-25

Boredom and student modeling in intelligent tutoring systems

William J. Hawkins
Worcester Polytechnic Institute

BOREDOM AND STUDENT MODELING IN INTELLIGENT TUTORING SYSTEMS

by

William Joseph Hawkins

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 2014

APPROVED:

_____	_____
Dr. Neil T. Heffernan, Advisor	Dr. Sonia Chernova, Thesis Reader

_____

Dr. Craig Wills, Head of Department

Abstract

Over the past couple decades, intelligent tutoring systems (ITSs) have become popular in education. ITSs are effective at helping students learn (VanLehn, 2011; Razzaq, Mendicino & Heffernan, 2008; Koedinger et al, 1997) and help researchers understand how students learn. Such research has included modeling how students learn (Corbett & Anderson, 1995), the effectiveness of help given within an ITS (Beck et al, 2008), the difficulty of different problems (Pardos & Heffernan, 2011), and predicting long-term outcomes like college attendance (San Pedro et al, 2013a), among many other studies.

While most studies have focused on ITSs from a cognitive perspective, a growing number of researchers are paying attention to the motivational and affective aspects of tutoring, which have been recognized as important components of human tutoring (Lepper et al, 1993). Recent work has shown that student affect within an ITS can be detected, even without physical sensors or cameras (D'Mello et al, 2008; Conati & Maclaren, 2009; Sabourin et al, 2011; San Pedro et al, 2013b).

Initial studies with these sensor-less affect detectors have shown that certain problematic affective states, such as boredom, confusion and frustration, are prevalent within ITSs (Baker et al, 2010b). Boredom in particular has been linked to negative learning outcomes (Pekrun et al, 2010; Farmer & Sundberg, 1986) and long-term disengagement (Farrell, 1988). Therefore, reducing or responding effectively to these affective states within ITSs may improve both short- and long-term learning outcomes.

This work is an initial attempt to determine what causes boredom in ITSs. First, we determine which is more responsible for boredom in ITSs: the content in the system, or the students themselves. Based on the findings of that analysis, we conduct a randomized controlled trial to determine the effects of monotony on student boredom. In addition to the work on boredom, we also perform analyses that concern student modeling, specifically how to improve Knowledge Tracing (Corbett & Anderson, 1995), a popular student model used extensively in real systems like the Cognitive Tutors (Koedinger et al, 1997) and in educational research.

Acknowledgements

I would like to thank my advisors, Neil T. Heffernan and Ryan S.J.d. Baker, for the time and effort they dedicated to advising me on the work contained here and in general.

I would like to thank Neil and Cristina Heffernan for creating the ASSISTments system and for helping me understand it better. Thanks to Cristina for help with constructing and running the randomized controlled trial.

I would like to thank the ASSISTments lab for helping me understand the system and providing me with the data I needed.

For applying the affect detectors to data sets and helping me understand how they work, I would like to thank Supreeth Gowda and Maria San Pedro.

For coauthoring one of the published papers that comprise this thesis and giving advice on other parts of it, I would like to thank Yutao Wang.

I would like to thank Lisa Rossi for proofreading Chapter 2 of this thesis before it was published.

I would like to thank Professors Joseph Beck and Ivon Arroyo and all the students that attended the weekly Educational Data Mining research meetings for sharing their expertise and giving valuable feedback on my work. Thanks to Xiaolu Xiong for organizing these meetings.

This work was funded by Graduate Assistance in Areas of National Need (GAANN) and Partnership in Math and Science Education (PIMSE) fellowships, a CAREER grant, the National Science Foundation (#1316736, #1252297, #1109483, #1031398, #0742503), ONR's 'STEM Grand Challenges,' IES (# R305A120125 & R305C100024) and grant #OPP1048577 from the Bill & Melinda Gates Foundation.

Many other funders of the ASSISTments system are listed at http://www.webcitation.org/5ym157Yfr.

Table of Contents

Table of Figures

Table of Tables

Chapter 1: Introduction

Over the past couple decades, intelligent tutoring systems (ITSs) have become popular in education. Used in a variety of environments with various age groups for various purposes, ITSs give students individual attention they do not get in the classroom, and feedback they do not get from traditional pencil-and-paper assignments. Several studies have shown the effectiveness of ITS (VanLehn, 2011; Razzaq, Mendicino & Heffernan, 2008; Koedinger et al, 1997).

In addition to helping students, ITSs have helped researchers learn about how students learn. Such research has included modeling how students learn (Corbett & Anderson, 1995), the effectiveness of help given within an ITS (Beck et al, 2008), the difficulty of different problems (Pardos & Heffernan, 2011), and predicting long-term outcomes like college attendance (San Pedro et al, 2013a), among many other studies.

While most studies have focused on ITSs from a cognitive perspective, a growing number of researchers are paying attention to the motivational and affective aspects of tutoring, which have been recognized as important components of human tutoring (Lepper et al, 1993). Recent work has shown that student affect within an ITS can be detected, even without physical sensors or cameras (D'Mello et al, 2008; Conati & Maclaren, 2009; Sabourin et al, 2011; San Pedro et al, 2013b). This is important because it allows for affect detection to be much more scalable, given the high cost and short lifespans of physical sensors in school environments.

Initial studies with these sensor-less affect detectors have shown that certain problematic affective states, such as boredom, confusion and frustration, are prevalent within ITSs (Baker et al, 2010b). Boredom in particular has been linked to negative learning outcomes (Pekrun et al, 2010; Farmer & Sundberg, 1986) and long-term disengagement (Farrell, 1988). Therefore, reducing or responding effectively to these affective states within ITSs may improve both short- and long-term learning outcomes.

This work is an initial attempt to determine what causes boredom in ITSs. The second chapter is a published analysis that sets out to determine which is more responsible for boredom in ITSs: the content in the system, or the students themselves. Based on the findings of that analysis, the unpublished work in the third chapter involves a randomized controlled trial whose purpose is to determine the effects of monotony on student boredom. In addition to the work on boredom, the fourth and fifth chapters of this work are published conference papers that both concern student modeling, specifically how to improve Knowledge Tracing (Corbett & Anderson, 1995), a popular student model used extensively in real systems like the Cognitive Tutors (Koedinger et al, 1997) and in educational research. Chapters 2, 4 and 5 appear in this thesis as they did when they were originally published, except for formatting and table and figure numbers for the sake of consistency throughout the document, and the acknowledgements and references for each have been pooled together in the respective sections of this document.

All of the work presented here was done using the ASSISTments system (Feng, Heffernan & Koedinger, 2009), a free web-based platform used primarily for middle- and high-school mathematics. Developed at Worcester Polytechnic Institute in collaboration with Carnegie Mellon University, ASSISTments was used by approximately 60,000 students during the 2013-2014 school year.

All code and data used in the work presented here is available online (Hawkins, 2014).

Chapter 2: Determining the main cause of boredom in intelligent tutoring systems

Boredom is unpleasant, and has been repeatedly shown to be associated with poor performance and long-term disengagement in educational contexts. Boredom is prevalent within a range of online learning environments, has been shown to correlate negatively with learning in those environments, and often precedes disengaged behaviors such as off-task behavior and gaming the system. Therefore, it is important to identify the causes of boredom in these environments. In psychology research, there is ongoing debate about the degree to which individual students are prone to boredom ("trait" explanations) or the degree to which boredom is driven by state-based factors, such as the design of the learning environment. In this study, we apply an unobtrusive computational detector of student boredom to log data from an intelligent tutoring system to determine whether state or trait factors better predict the prevalence of boredom in students using that system. Knowing which type of factor better predicts boredom in a specific system can help us to narrow down further research on why boredom occurs and what steps should be taken to mitigate boredom's negative effects.

This chapter was published at the following venue:
Hawkins, W., Baker, R.S.J.d., Heffernan, N.T. (2013) Which is more responsible for boredom in intelligent tutoring systems: students (trait) or problems (state)? *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, Geneva, Switzerland, pp. 618-623.

I. Introduction

Over the past several decades, there has been a considerable degree of interest regarding student boredom during learning. Many agree that boredom is an unpleasant or negative experience (Harris, 2000; Pekrun et al, 2010) but propose different potential causes and effects of the emotion, and disagree about its impact and how to respond to it (Belton & Priyadharshini, 2007).

According to Belton and Priyadharshini's survey of boredom research (Belton & Priyadharshini, 2007), most psychological research concerned with the causes of boredom has posited on a dichotomy of two possible causes that has variously been referred to as "responsive" vs. "chronic," "agitated" vs. "apathetic," "dispositional" vs. "situational," and most commonly "state" vs. "trait".

The "state" construct of boredom describes boredom as being caused by a specific situation or experience that is objectively boring, where external stimulus is lacking (Belton & Priyadharshini, 2007). A specific situation or experience can lack stimulation for a variety of reasons (cf. Vodanovich, 2003; Larson & Richards, 1991; Belton & Priyadharshini, 2007; Harris, 2000; Pekrun et al, 2010; Mikulas, 1993).

Alternatively, state boredom may be caused by temporary aspects of the student, for example fatigue. In a 2000 study conducted with 170 American university students, 17% identified

fatigue as an indicator that they were bored, 8% identified it as a cause of boredom, and 15% reported that sleeping was one way they coped with boredom (Harris, 2000). Therefore, the causes of state boredom can be thought of in terms of the current state of the person, as well as the nature of the activity.

Another viewpoint is that boredom is caused by student traits (Belton & Priyadharshini, 2007), where certain individuals are more prone to boredom than others. Various studies have used the Boredom Proneness Scale (Farmer & Sundberg, 1986), a questionnaire measure, to assess the susceptibility of different individuals to boredom, and have found links between this boredom "trait" and other personality characteristics (Farmer & Sundberg, 1986), as well as to various negative and destructive behaviors (Harris, 2000; Vodanovich, 2003).

Boredom in education has been studied in terms of state and trait constructs (Larson & Richards, 1991), as it has in other contexts such as in the workplace (Belton & Priyadharshini, 2007). Within education, state explanations for boredom have blamed schools and the design of classroom activities, arguing that boredom is caused by meaningless or repetitive tasks (Reid, 1986), overly abstract activities (Condry, 1978), and tasks being too challenging (Cullingford, 2002) or not challenging enough (Moneta & Csikszentmihalyi, 1996). However, Larson and Richards also found that some students associated boredom with fatigue, and that boredom co-occurred with tiredness and drowsiness (Larson & Richards, 1991).

On the trait side of the debate, the dispositions that students bring to school have been blamed for the boredom they experience there (Gjesme, 1977; Farrell, 1988). Additionally, learning goals, perceived level of control, and the relative value a student places in a skill or activity have all been argued to be associated with boredom in educational settings (Pekrun et al, 2010).

To determine the relative effects of state and trait boredom among middle school students, Larson and Richards (Larson & Richards, 1991) measured boredom experienced by students over the course of a week using randomly timed surveys. They found that boredom was prevalent both in and out of school, and that it depended more on the individual student than on the subject or the activity. They found that boredom students experienced in and out of school was highly correlated ($r = 0.68$). However, both subject and activity also had a substantial influence on boredom in this research.

Understanding why students become bored is important, as boredom has been shown to be associated with poorer learning (Pekrun et al, 2010; Farmer & Sundberg, 1986) as well as long-term disengagement (Farrell, 1988). Within intelligent tutoring systems (ITSs), it has been shown that boredom is one of the most persistent affective-cognitive states (Baker et al, 2010b), and that it leads to gaming the system, or "attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material" (Baker et al, 2010b). Gaming the system has also been linked to poorer learning, both in the short-term (Baker et al, 2004; Gong et al, 2010) and in the long-term (Baker et al, 2004).

Boredom has also been shown to lead to off-task behavior in ITSs (Baker et al, 2011), which is also associated with poorer learning (Karweit & Slavin, 1982) and long-term disengagement (Finn, 1989). However, Baker et al.'s research (2011) suggested that off-task behavior can relieve boredom and allow the student to refocus on their work with a lower probability of experiencing boredom later. Regardless, both gaming the system and off-task behavior take up considerable time (Baker et al, 2004), giving students less time to use ITSs constructively and learn from them. Given the increasing use of ITSs within education, the effects boredom has within ITSs and on learning in general, it is important to study the causes of boredom in ITSs.

In order to study boredom adequately in ITSs, it is important to have a broadly applicable method of assessing the presence and intensity of boredom. While many studies and systems have used physical sensors to detect boredom (Arroyo et al, 2009; D'Mello & Graesser, 2010), these approaches can be difficult to scale within large numbers of classrooms due to issues such as internet bandwidth, cost of sensors, and breakage under classroom conditions.

Due to the restricted applicability of models built using physical sensors, researchers have recently worked to develop affect detectors based solely on log files for various platforms, including AutoTutor (D'Mello et al, 2008), Prime Climb (Conati & Maclaren, 2009), Crystal Island (Sabourin et al, 2011), and ASSISTments (San Pedro et al, 2013b). As these types of detectors rely only on log data, they can be applied to large amounts of data, allowing datasets to be labeled for use in exploratory analyses.

Given the need to study boredom in ITSs and the availability of broadly applicable log-based boredom detectors, this work uses results from a real-time log-based boredom detector (San Pedro et al, 2013b) to determine whether boredom in ITSs is caused more by state or trait characteristics. Specifically, linear regression models are constructed to determine whether state or trait factors better predict boredom, as done previously by others for studying whether gaming the system is better predicted by state or trait factors (Baker, 2007; Muldner et al, 2011).

In addition to state vs. trait analysis, models incorporating proxies for fatigue are fit. Due to previously reported associations between fatigue and boredom (Larson & Richards, 1991; Harris, 2000) and the finding that off-task behavior in ITSs helps relieve boredom (Baker et al, 2011), we hypothesize that measures of boredom and fatigue within this study will be positively correlated.

Once it is determined which construct better predicts boredom in ITSs, it can be studied closer to determine what steps should be taken to respond to it. If boredom is better predicted as a state variable, then the content and interface of the system should be studied further and improved to reduce boredom. If on the other hand boredom is better predicted as a trait variable, then student characteristics should be studied further to determine how to change the system specifically for students prone to boredom. By narrowing down the type of approaches that are likely to address boredom, we can move towards reducing its prevalence and impacts in ITSs, with the goal of improving both student engagement and learning.

Section II introduces the dataset and methods used in this work. Section III presents the results of the analyses, and Section IV concludes with discussion and possible directions for future work.

II. Methods

A. Tutor and Data

For this work, data from the ASSISTments intelligent tutoring system (Feng, Heffernan & Koedinger, 2009) was used. ASSISTments is a web-based ITS used primarily by middle- and high-school students. In the 2012-2013 school year, it is being used by 40,000 students, mostly in the Northeastern USA, around once a week. While using ASSISTments, students are assessed based on their performance within the system, which is reported back to teachers. Additionally, students are assisted while working through problem sets in three main ways: hint messages, which progress from high-level hints to a "bottom-out hint" containing the answer to the problem; feedback given when the student gives incorrect answers; and scaffolding, where the system breaks a problem down into sub-problems. An example of a student working through a problem in ASSISTments is shown in Figure 1.

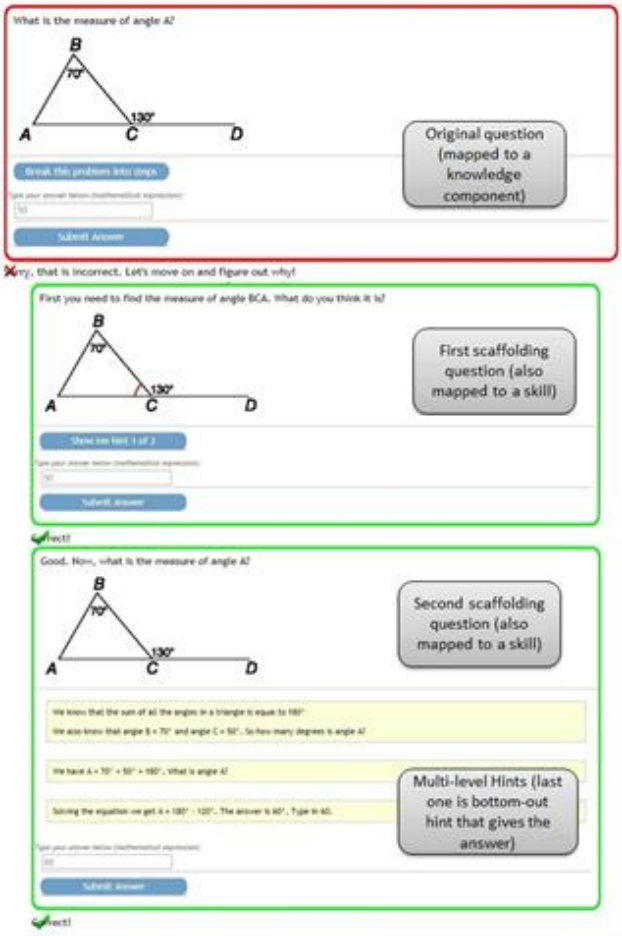

Figure 1: Example of a student working through a problem in ASSISTments, from top to bottom. The student first answers the question incorrectly, resulting in feedback. The problem is then scaffolded, and the student answers the first

scaffolding question correctly. Finally, the student clicks through all the hint messages of the second scaffolding question, reaching the "bottom-out hint," which contains the answer the student types in to solve the problem.

Within this study, data previously collected was used in analysis. The data was collected from ASSISTments data logs from September 2004 to May 2005 for 724 students from four central Massachusetts middle schools, consisting of 107,382 problem attempts. This data set was chosen because the affect detector had already been applied to it and it had been used in a previous study (e.g., San Pedro et al, 2013b). Each problem attempt includes the ID of the student that made the attempt, the problem ID, the relevant "skill" being tested by the problem (e.g., multiplication, area, equation-solving, etc.), and the "type" of problem or method of answering the question (e.g., multiple choice, fill-in, etc.). There were 10 different types of problems and 70 skills represented.

Additionally, each problem attempt was labeled with a real-valued confidence level between 0 and 1 that boredom was present. Confidences can be interpreted as the detector's estimate of the probability that the student was bored at a specific time. These values were computed using a real-time boredom detector, discussed in full detail in (San Pedro et al, 2013b); in brief, this detector was developed by synchronizing thousands of field observations of boredom (with inter-rater reliability over 0.6) with log files, and using data mining to infer the human codes. The mean boredom confidence across all problem attempts was 0.2469 (SD = 0.1293). Confidences were used instead of binary predictions of boredom, in order to leverage the detector's ability to distinguish cases it is unsure of – for instance, claiming that a case with 51% certainty is identical to a case with 100% certainty, but is fundamentally different from a case with 49% certainty, throws out considerable information and increases the noise in the data set.

B. Modeling Method

To determine whether boredom can be modeled better as a state or trait construct, we fit linear regression models to the data to predict the confidence of the boredom detector. One model is trained using only the problem ID from each problem attempt as a predictor, while the other uses only the student ID from each problem attempt as a predictor. The state theory hypothesizes that the difference between problems will predict much of the variance in student boredom, while the trait theory hypothesizes that the difference between students will predict much of the variance in boredom. The R^2 and Bayesian Information Criterion (BiC'; Raftery, 1995) values of the predictions made by these models can then be used to assess which construct is a better predictor of boredom. Additionally, a third model is fit with both problem ID and student ID as predictors.

Similar procedures to that described above have been performed for gaming the system in order to assess whether it is better viewed as a state or trait construct (Baker, 2007; Gong et al, 2010; Muldner et al, 2011). Baker's analysis found that lessons predicted gaming better than students, which was contradicted by the findings of the other two groups (Gong et al, 2010; Muldner et al, 2011); further unpublished analysis conducted by two of these research groups working together suggests that this may be due to differences in the operational definition of gaming used in the different studies. In line with Baker's method, we use IDs rather than the average confidence of

affect detector results; however, we analyze state-level prediction at the problem-level as in (Muldner et al, 2011), rather than analyzing at a coarser grain-size.

Finally, a proxy for fatigue is computed and added to the data. In this study, the proxy for fatigue is operationalized as the number of minutes that passed since the student last took a break of a certain number of minutes. We hypothesize that fatigue will be a successful predictor due to previous findings of relationships between fatigue and boredom (Harris, 2000; Larson & Richards, 1991) and the relationship between boredom and off-task behavior, in which off-task behavior appears to relieve boredom (Baker et al, 2011). We hypothesize that the longer a student goes without an opportunity to relieve their boredom, the higher their boredom will be. A number of linear regression models are built using only this fatigue statistic for different time durations that constitute a break. The best of these fatigue attributes is then combined with predictors from the other models described above and tested.

III. Analyses

A. State vs. Trait

The first research goal was to determine whether state (problems) or trait (students) was a better predictor of boredom. For this analysis, two linear regression models were fit: one that used only problem ID as a predictor, and one that only used student ID. The target attribute for both was the detector's real-valued confidence that boredom was present. These models were evaluated using two measures: R^2, and the Bayesian Information Criterion (BiC'; Raftery, 1995), which calculates the degree to which a model's predictions are better than what would be expected solely from the number of parameters used. Lower values of BiC' are better, and a difference of six or more between the BiC' values of two models is considered equivalent to being statistically significant at the $p < 0.05$ level (Raftery, 1995). The R^2 values were computed using five-fold cross-validation, where the folds were stratified both by problem ID and student ID. The same folds were used for all models, which were built using Matlab's LinearModel class. The BiC' values were computed over the entire dataset. The results are shown in Table 1.

Table 1: Model fitting results for state vs. trait. R^2 calculated using five-fold cross-validation, BiC' calculated over the entire dataset

Model	R^2	BiC'
Baseline	0.0000	0.00
Problem ID	0.0516	-13,002
Student ID	0.0061	2,845
Both	0.0818	-11,012

As Table 1 shows, the model based only on problem IDs is significantly more predictive than that based only on student IDs, suggesting that the incidence of boredom is more dependent on the problem being attempted rather than on the student attempting it. Additionally, the student model does worse than the Baseline model (which predicted the mode of boredom confidences, 0.1273,

for all problem attempts) judging by its large positive BiC' value. A third model that used both student IDs and problem IDs as predictors achieved a higher R^2 value, but did worse for the number of parameters it used compared with the model that only used problem IDs, based on their respective BiC' values.

B. Fatigue

When predicting boredom, it may be helpful to include other factors, such as the current state of the student. One key aspect of the student is whether the student is fatigued. To test this hypothesis, a proxy for the construct of fatigue was added to each problem attempt in the dataset. Fatigue was calculated in two ways, producing models we refer to as MFatigue and PFatigue.

MFatigue was calculated as the number of minutes that had passed since the student had taken a break of a certain number of minutes. For example, the attribute "MFatigue(60)" (M stands for minutes) is defined as the number of minutes that have elapsed since the student in question last took a break of 60 minutes or more.

PFatigue was calculated as the number of problems completed by the student since last having a break of a certain number of minutes. We hypothesized that the monotony, and therefore boredom, that the students experienced would increase with the number of problems they completed. For example, the attribute "PFatigue (60)" is defined as the number of problems a given student has completed since last having a break of 60 minutes or more.

A number of linear regression models were built using each fatigue statistic for different amounts of minutes that constituted a break. This was done since it was uncertain how long of a break was necessary to "reset" a student's level of fatigue. The results were computed using five-fold cross-validation with the same five folds used for the state and trait models. The results are shown in Table 2.

Table 2: Model fitting results for fatigue. Models including both fatigue and other factors given at bottom.

Model	R^2	BiC'
MFatigue(60)	0.0000	-354
MFatigue(30)	0.0000	-537
MFatigue(15)	0.0000	-938
MFatigue(10)	0.0002	-1,469
MFatigue(5)	0.0006	-2,136
PFatigue(60)	0.0013	-2,784
PFatigue(30)	0.0016	-3,093
PFatigue(15)	0.0029	-3,914
PFatigue(10)	0.0039	-4,495
PFatigue(5)	0.0038	-4,473
Problem ID, PFatigue(10)	0.0644	-15,836
Problem ID, Student ID, PFatigue(10)	0.0878	-12,229

Both the problem-based and the minute-based fatigue models were better than the baseline model and the student/trait model. However, the problem-based fatigue models were much more predictive than their minute-based counterparts. Using 5 or 10 minutes as the duration of break needed to reset the student's fatigue produced the best models for predicting boredom from fatigue. The best combination of problem IDs with a fatigue attribute used a break of 10 minutes for resetting fatigue, and significantly improved upon the model that only predicted boredom from the problem ID. This combined (problem ID, fatigue) model was the most predictive model found in these analyses, with a difference in BiC' of almost 3,000 from the next best model (problem ID). Similarly, adding PFatigue(10) to the (problem ID, student ID) model also improved performance, though this model's performance still fell short of the problem ID model (or the problem ID, PFatigue(10) model).

Having found that our hypothesis about fatigue being a strong predictor of boredom was correct, we next set out to determine if our hypothesis about higher fatigue leading to higher boredom was correct. However, it turned out that "fatigue" was instead negatively correlated with boredom; e.g., the longer it had been since the student took a break, the less bored they were. Examining the linear regression models based on fatigue revealed that fatigue had a negative coefficient, and correlation analysis showed that all calculations of fatigue were negatively correlated with boredom. The strongest of these correlations was for PFatigue(10), where r = -0.20. It should be noted this correlation was calculated over the entire dataset, and therefore does not correspond to the cross-validated R^2 value reported for the PFatigue(10) linear regression model in Table 2.

C. Type and Skill

Since problems were found to be more predictive than students, we focused on state explanations of boredom for the remainder of our analyses. Additionally, it has been found that boredom is associated with the difficulty of tasks within achievement settings (Pekrun et al, 2010; San Pedro et al, 2013b). Therefore, as a preliminary analysis of which specific problem features cause boredom in ASSISTments, the association between the type and skill of problems, as defined above, and the level of boredom experienced on them was studied.

Linear regression models were tested using type and skill, using five-fold cross-validation and the same folds used for all previous models. As was done for problems and students, separate models were fit for type and skill, and a third model combining the two was also fit. The results are shown in Table 3.

Table 3: Model fitting results for type and skill

Model	R^2	BiC'
Type	0.0039	-4,290
Skill	0.0131	-7,063
Type and Skill	0.0184	-8,530
Type, Skill, PFatigue(10)	0.0294	-11,491

Both the skill and type models outperformed the student ID model, but performed more poorly than the problem ID model. In general, the results also indicate that the skill of a problem is more predictive than its type. Combining type and skill into one model further improved performance, but still not to the same level as the problem ID model. Additionally, PFatigue(10) proved useful again as adding it to the (type, skill) model significantly improved performance – but again, still not to the level of the problem model. Therefore, skill and type seem important, but it appears there are other important problem features missing from this analysis due to the differences in predictive power of the type and skill model and the problem ID model. A full study of which problem features lead to boredom (cf. Doddannara et al, 2013) is warranted.

IV. Discussion and Future Work

Contrary to previous work (Larson & Richards, 1991), we found that state (individual problems) was more predictive of boredom than trait (individual students) in an intelligent tutoring system, ASSISTments. Two possible explanations for this are the different methodologies employed by the two studies, and the different contexts in which boredom was studied.

First, boredom in this work is measured by an unobtrusive computational detector, whereas the prior study measured it using questionnaires and interviews (Larson & Richards, 1991); these differences in measurement may change the results in multiple ways. For instance, potential differences in student comfort and attitudes in self-reporting their boredom could drive the appearance of a student-level effect within self-report methodologies. Second, this work studied boredom at the individual problem level within an intelligent tutoring system, whereas the prior study looked at boredom at a much higher level. The prior study looked at variations in boredom across different subjects like mathematics and English; across different activities within the school environment (such as listening to a teacher or student, reading, and taking a test or quiz); and across school, home, and public environments (Larson & Richards, 1991).

Since it appears that boredom is caused more by individual problems than by students, at least in the context of ASSISTments, future work should focus on identifying which features of problems are the most responsible for boredom. Identifying such features will help inform the design of problems in the future to reduce boredom, increase learning rates, and reduce long-term disengagement.

In this work, two features of problems were considered: the skill (multiplication, equation-solving, etc.) and the type of problems (multiple choice, fill-in, etc.). The linear regression model that considers these features together achieves a paltry R^2 of 0.0184, leaving a significant portion of the R^2 achieved by the problem ID model (0.0516) unexplained. Future work should explore what other properties of problems contribute to boredom.

The level of knowledge a student possesses in a given skill has been shown to explain some of the variance in boredom, though one study found boredom to be higher among highly skilled students (Larson & Richards, 1991) while a set of five other studies have found the opposite relationship

(Pekrun et al, 2010). Therefore, the relative difficulty of a problem or skill may help predict boredom in an intelligent tutoring system context. Looking beyond the aforementioned high-level problem features, a process for determining relevant features similar to what was done by Doddannara et al. (2013) may be relevant and useful for following up the results seen here.

Additionally, we found that modeling a proxy for student "fatigue" can add predictive power to the problem model. The best operationalization of fatigue was the number of problems completed since the student last had a break of 10 minutes or more. However, somewhat surprisingly, this measure we constructed as a proxy for fatigue was negatively correlated with boredom. From the results and above analysis, it appears there is merit in considering attributes like fatigue that are not specific to problems (i.e., not uniquely identified by problem ID), but that describe the "session" (what a student completes in a single sequence of activity) or the student's current state. It may be worth considering further attributes of this nature, such as how many times the student has seen the current skill, or how the current problem relates to previous problems in the problem set in terms of their other attributes, such as difficulty, skill or type.

In this work, we have shown that (at least within the ASSISTments ITS) boredom is better explained by problems (state) than students (trait). We did initial research into the specific components of state boredom within an ITS by fitting models to attributes of problems (type, skill) and the session (fatigue), both of which performed better than the baseline model at predicting boredom. Future work should consider additional factors such as problem difficulty, student knowledge (which is specific to individual students but changes dramatically over time, and which is more consistent with state than trait), and the amount of previous practice the student has had, among other state features.

Another valuable area of work is to follow up this work with replications in other online learning environments, and using alternate operationalizations of boredom (as in the follow-ups to (Baker, 2007) by Gong et al. (2010) and Muldner et al. (2011) for gaming). By better understanding the factors that influence whether a student becomes bored while using an ITS, we may be able to develop more emotionally-sensitive learning systems that lead to better learning and higher long-term engagement.

Chapter 3: Determining the effect of monotony on boredom in intelligent tutoring systems

I. Introduction

The purpose of this study is to determine whether monotonous content in an intelligent tutoring system (ITS) causes student boredom. We define monotonous content as problem sets whose problems are very similar to each other in terms of the skill(s) and steps required to complete them, the cover story used to give context to the problem, images used, and any other details that could make them seemingly indistinguishable from each other as a student works through them.

A retrospective analysis using machine-learned affect detectors (San Pedro et al, 2013b) conducted over ASSISTments (Feng, Heffernan & Koedinger, 2009) data from the 2009-10 school year showed that one of the most boring skill builders (a problem set where a student must answer three consecutive questions correctly to complete) in the system was one that helped students learn how to read histograms. This problem set had the same cover story for every question (light bulb lifespans), used four similar histogram images, and asked two similar types of questions about the data: how many light bulbs had lifespans less than or equal to a certain number of hours, and how many had lifespans greater than or equal to a certain number of hours. An example of one of these questions is shown in Figure 2. We believe the high degree of similarity among the problems of this problem set, or its monotony, is the reason it was among one of the most boring skill builders in ASSISTments.

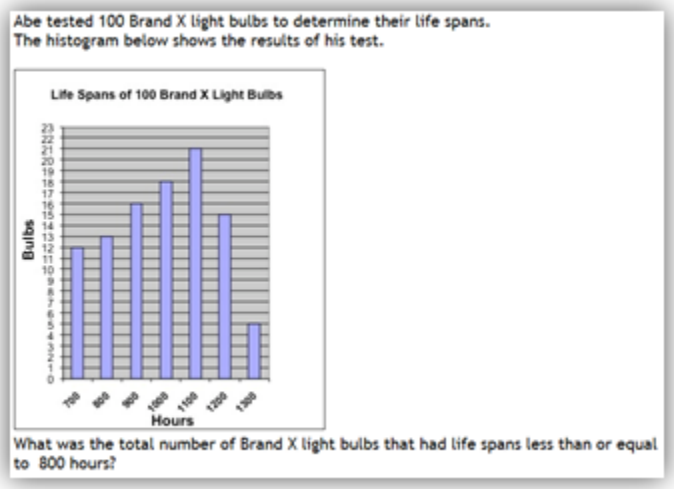

Figure 2: Example of a "light bulb" question in ASSISTments.

II. Methods

To test this hypothesis, we conducted a randomized controlled trial in ASSISTments. We designed a problem set focused on histograms that, when assigned to students, randomly placed

them into one of two types of conditions: *control*, where students saw questions that were very similar to each other, and *experimental*, where students saw a mixture of different types of questions. In order to make the experimental condition possible, we created additional histogram questions that were similar in terms of the questions asked and the hint messages provided on the original questions, but that had different cover stories and used different images. Including the original questions, this gave us four unique cover stories, with four histogram images associated with each of them. Figure 3 shows an example of a problem whose cover story is about the heights of trees in the Redwood Forest. The other two cover stories added to the problem set were about the running speeds of cheetahs, and the distances that students were able to throw softballs in gym class.

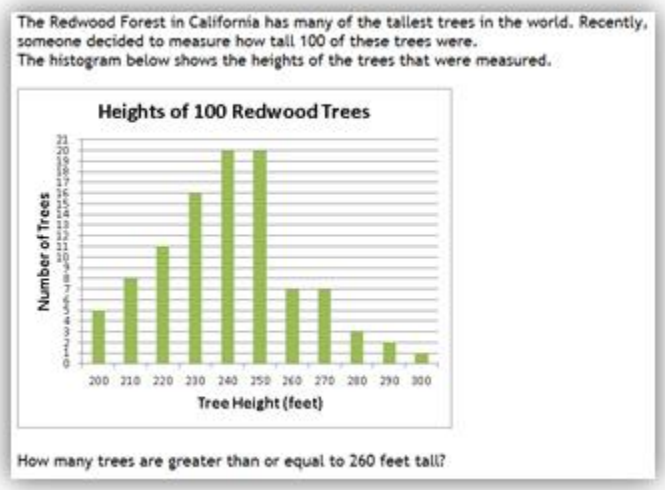

Figure 3: Example of a histogram question whose cover story is about heights of trees in California's Redwood Forest.

While only one mixed condition is necessary, several control conditions are necessary to isolate monotony as the reason we see any potential effects. It could be the case that some questions are more engaging than others due to the nature of their cover stories, which could lead students who see those questions in the mixed condition to be less bored than those who only see one of the less engaging questions. This would lead to the erroneous conclusion that reducing monotony reduces boredom, when instead the observed effect was actually caused by students seeing engaging questions. To combat this, we had one control condition per cover story, where students in a given control condition only saw questions with the cover story that corresponds to that condition. Having a separate control condition for each cover story allows us to see if any of the questions are inherently more engaging than the others, helping us to separate that from any effect that reducing monotony may have on boredom.

Finally, we noticed that a more recent version of the 2009-10 histograms problem set still had only one cover story, but also only used one image instead of four. Therefore, we added a fifth control condition where students only see light bulb questions that all share the same image. This

gave us six conditions total, where half of the students were randomly assigned to one of the five control conditions, and the other half were assigned to the experimental condition. The diagram in Figure 4 shows what students assigned to different conditions may have seen in terms of cover stories and images.

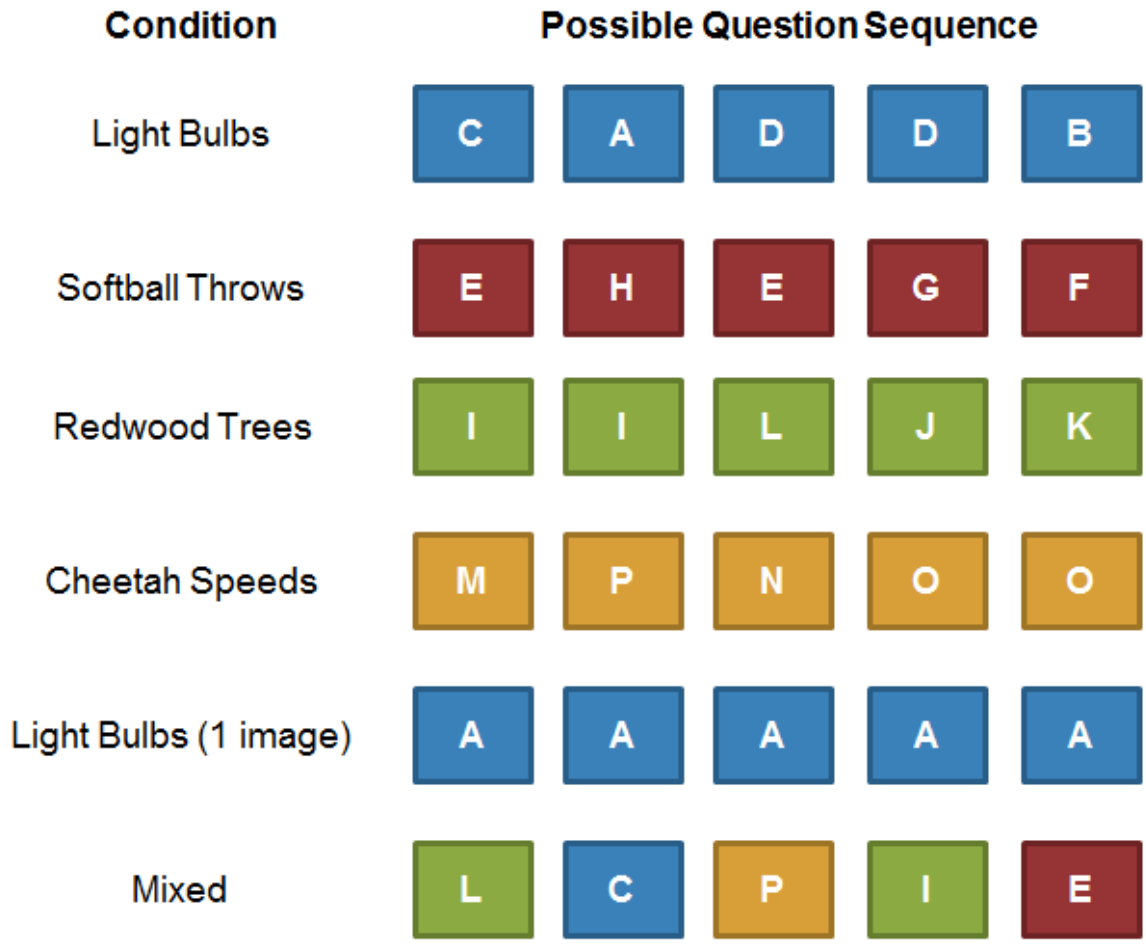

Figure 4: Possible questions sequences for students in each condition.

In Figure 4, the name of the condition appears in the left column, while a possible sequence of questions a student in that condition may see appears to the right, where each question is represented by a colored box containing a letter. The color of the box indicates the cover story of the question, and the letter inside the box indicates the histogram image used. The first five conditions in the figure are control conditions as described above, which means their questions all have the same cover story (and therefore, the same color in the figure). The fifth condition has the same cover story as the first, but uses only one unique image instead of four (therefore, the letters inside the boxes must all be the same). Finally, the last condition, which is the experimental "Mixed" condition, may include questions with any of the four cover stories and

any of the four associated histogram images. Therefore, its boxes may have different colors and letters from each other.

Since the questions in our problem set can be interpreted as histograms or bar graphs, we made two copies of the problem set: one for sixth grade (when histograms are covered, according to the Common Core State Standards) and one for third grade (when bar graphs are covered). Therefore, it is possible we could see an interaction effect between grade level and monotony regarding boredom.

In order to assess the impact of monotony on boredom, we applied the affect detectors developed for ASSISTments in (San Pedro et al, 2013b). These include detectors for boredom, concentration, confusion and frustration. Therefore, in addition to boredom, we also analyzed the data to determine if monotony had any effect on any of the other three affective states.

We ran the experiment in January and February 2014, during which time 125 students worked on the problem set. However, we only wanted to analyze students who had never completed the original histograms problem set before. This left us with 66 students: 30 had been assigned to one of the control conditions, and 36 to the experimental condition.

III. Results

We conducted three types of analyses: *overall*, where we simply looked at the average incidence of each affective state in the control and experimental groups, *previous performance*, where we split the analysis based on students' previous performance in ASSISTments, and *previous affect*, where we split the analysis based on students' previous average inferred level of each affective state. Due to the low number of students that participated in the experiment (who had not previously completed a histograms problem set in ASSISTments), we did not analyze the control conditions separately, nor did we analyze the third grade and sixth grade problem sets separately.

For each of the analyses reported here, all differences in means were tested with two-tailed t tests to compute p-values. Storey's (2002) false discovery rate procedure was then used to compute q-values from these p-values. None of the computed q-values were significant, though a few of the original p-values were marginally significant.

Overall

None of the differences in overall means between the control and experimental groups were significant at the 0.05 level, though the difference between average frustration in the control and experimental conditions is marginally significant before the FDR adjustment ($p = 0.0640$, $t(64) = -1.8848$). These means are reported in Table 4.

Table 4: Average confidence of each affective state for each condition, first averaged within each student, then averaged across all students.

	Boredom	**Concentration**	**Confusion**	**Frustration**

Control	0.2752	0.7669	0.2661	0.3737
Experimental	0.2778	0.7669	0.2838	0.3998

Split by Previous Performance

For each class that participated, we split the students in each class into two groups: low-performance and high-performance, based on class rank. We did this by computing the percentage of questions that each student answered correctly in ASSISTments during the entire school year, not counting the current experiment, and then assigned the top 50% of the students in each class to the high-performance group, and the bottom 50% of each class to the low-performance group.

Once we split the students into these two groups, we performed the same analysis we did above, but did so for each group of students separately. The results are shown in Table 5.

Table 5: Average confidence of each affective state split by condition and previous performance.

	Boredom		**Concentration**		**Confusion**		**Frustration**	
	Low	*High*	*Low*	*High*	*Low*	*High*	*Low*	*High*
Control	0.2542	0.3067	0.7669	0.7669	0.2482	0.2930	0.3757	0.3706
Experimental	0.2161	0.3330	0.7669	0.7669	0.3609	0.2148	0.4098	0.3908

None of these differences were significant. There was just one marginally significant difference, which was between low- and high-knowledge students in the experimental condition in terms of boredom before the FDR adjustment ($p = 0.0602$, $t(34) = -1.9439$).

Split by Previous Affect

Similar to the previous analysis, in this analysis, students in each class were split into two groups by class rank, but this time by how often they exhibited each of the four affective states. Therefore, there were four different orderings, one for each state. The results are shown in Table 6.

Table 6: Average confidence of each affective state split by condition and previous performance.

	Boredom		**Concentration**		**Confusion**		**Frustration**	
	Low	*High*	*Low*	*High*	*Low*	*High*	*Low*	*High*
Control	0.2333	0.3475	0.7669	0.7669	0.2707	0.2601	0.3679	0.3812
Experimental	0.2397	0.3083	0.7669	0.7669	0.3145	0.2496	0.3967	0.4025

None of the differences here were significant either, only the difference in frustration across conditions for those with a history of low frustration was marginally significant before the FDR adjustment ($p = 0.0869$, $t(32) = -1.7662$).

IV. Discussion and Future Work

Although it appears that there may have been some effects with marginal reliability, these are most likely statistical artifacts according to Storey's FDR adjustment (Storey, 2002). Therefore, it is difficult to make any conclusive claims about the outcome of the study.

It is possible that some or all of the observed marginal effects (before the FDR adjustment) are real, but not detectable with the small number of students in the study. For example, it is possible that seeing multiple cover stories is more frustrating than seeing just one, possibly because it forces the students to read each new problem statement rather than just being able to get into a rhythm and extract the important information quickly, which is easier if the problems all look the same. Additionally, seeing multiple cover stories could widen the gap in boredom between low- and high-achieving students. Low-achieving students are more likely not to know the new material and actually need to learn from it, and therefore seeing the new skill applied in multiple contexts as they are learning it may be more engaging. On the other hand, high-achieving students may already know the new skill and just want to get through it, making variable irrelevant cover story details a distraction to them. However, all of this is conjecture without a larger study to confirm it.

Another possible reason for there being no reliable effects is that there was a "ceiling effect" – the vast majority of the students in both conditions finished the problem set in three problems, which is the minimum (students needed to answer three consecutive questions correctly on the same day to complete the assignment). Therefore, students may simply have not had to spend enough time in the problem set for there to be a detectable effect on their affective states.

Therefore, the next logical step would be to run a larger study with more difficult skills to determine whether monotony affects students' affective states. More difficult skills will help avoid the "ceiling effect," and more students will make effects more detectable, if in fact there are any.

Chapter 4: Using tabling methods and ensembling to improve student performance prediction

In the field of educational data mining, there are competing methods for predicting student performance. One involves building complex models, such as Bayesian networks with Knowledge Tracing (KT), or using logistic regression with Performance Factors Analysis (PFA). However, Wang and Heffernan showed that a raw data approach can be applied successfully to educational data mining with their results from what they called the Assistance Model (AM), which takes the number of attempts and hints required to answer the previous question correctly into account, which KT and PFA ignore. We extend their work by introducing a general framework for using raw data to predict student performance, and explore a new way of making predictions within this framework, called the Assistance Progress Model (APM). APM makes predictions based on the relationship between the assistance used on the two previous problems. KT, AM and APM are evaluated and compared to one another, as are multiple methods of ensembling them together. Finally, we discuss the importance of reporting multiple accuracy measures when evaluating student models.

This chapter was published at the following venue:
Hawkins, W., Heffernan, N.T., Wang, Y., Baker, R.S.J.d. (2013) Extending the Assistance Model: Analyzing the Use of Assistance over Time. *Proceedings of the 6th International Conference on Educational Data Mining*, Memphis, TN, pp. 59-66.

I. Introduction

Understanding and modeling student behavior is important for intelligent tutoring systems (ITS) to provide assistance to students and help them learn. For nearly two decades, Knowledge Tracing (KT) (Corbett & Anderson, 1995) and various extensions to it (Pardos & Heffernan, 2010a; Wang & Heffernan, 2012; Xu & Mostow, 2012) have been used to model student knowledge as a latent using Bayesian networks, as well as to predict student performance. Other models used to predict student performance include Performance Factors Analysis (PFA) (Pavlik, Cen & Koedinger, 2009) and Item Response Theory (Johns, Mahadevan & Woolf, 2006). However, these models do not take assistance information into account. In most systems, questions in which hints are requested are marked as wrong, and students are usually required to answer a question correctly before moving on to the next one. Therefore, the number of hints and attempts used by a student to answer a question correctly is likely valuable information.

Previous work has shown that using assistance information helps predict scores on the Massachusetts Comprehensive Assessment Systems math test (Feng & Heffernan, 2010), can help predict learning gains (Arroyo et al, 2010), and can be more predictive than binary performance (Wang, Heffernan & Beck, 2010). Recently, it has been shown that using simple probabilities derived from the data based on the amount of assistance used, an approach called the Assistance Model (AM), can improve predictions of performance when ensembled with KT (Wang & Heffernan, 2011).

This work continues research in the area of using assistance information to help predict performance in three ways:

1. Specifying a framework for building "tabling methods" from the data, a generalization of AM

2. Experimenting with a new model within this framework called the Assistance Progress Model (APM), which makes predictions based on the relationship between the assistance used on the previous two problems

3. Experimenting with new ways of ensembling these models to achieve better predictions

Additionally, the importance of reporting multiple accuracy measures when evaluating student models is discussed, as well as why three of the most commonly reported measures (mean absolute error (MAE), root mean squared error (RMSE) and area under the ROC curve (AUC)) do not always agree on which model makes the most accurate predictions.

Section II describes the tutoring system and dataset used. Section III describes the methodology: the models and ensembling methods used, the tabling method framework, and the procedure for evaluating the models. Section IV presents the results, followed by discussion and possible directions for future work in Section V.

II. Data

The data used here was the same used in (Wang & Heffernan, 2011), which introduced AM. This dataset comes from ASSISTments, a freely available web-based tutoring system for 4^{th} through 10^{th} grade mathematics.

While working on a problem within ASSISTments, a student can receive assistance in two ways: by requesting a hint, or by entering an incorrect answer, as shown in Figure 5.

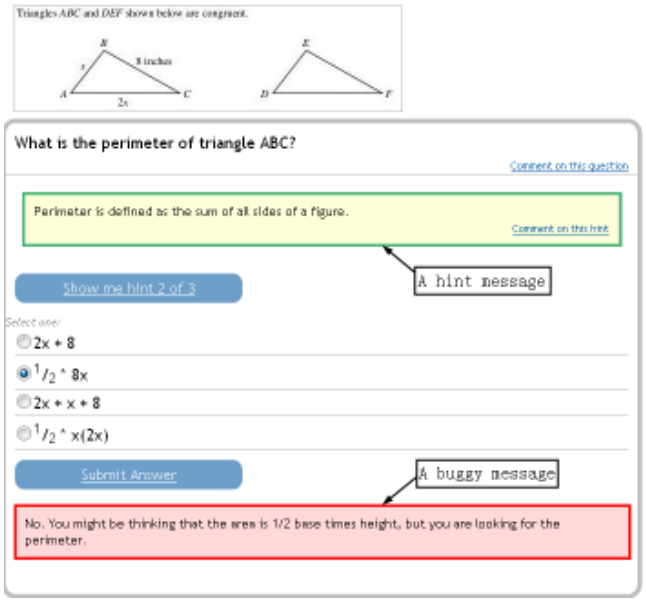

Figure 5: Examples of assistance within ASSISTments (from Wang and Heffernan, 2011)

The dataset comes from four Mastery Learning classes conducted in 2009, where students worked on problem sets until achieving some criterion, usually specified as answering three questions in a row correctly. The questions in these problem sets were generated randomly from templates, with the difficulty of each question assumed to be the same as all other questions generated from the same template. No problem selection algorithm was used to select the next question.

Two hundred 12-14 year old 8th grade students participated in these classes, generating 17,776 problem logs from 93 problem sets. However, due to the nature of the models studied in this paper, data from two of these students could not be used since these two students never answered more than one question within the same problem set.

Since two of the models cannot be used to predict performance on the first question of a problem set, as they rely on assistance usage on previous problems, these models were not trained or evaluated on the first question answered by a student on a given problem set. This reduced the dataset for these models to 12,099 problem logs. KT models were still trained using the entire dataset, but only evaluated on the 12,099 logs they had in common with the other models.

III. Methods

This section begins by giving an overview of KT, then introduces a framework for building data-driven student models called "tabling methods," and describes two such methods: AM and APM. Next, the approaches used to ensemble these individual models together are briefly discussed. Finally, the procedure and measures used to evaluate all models are discussed.

Knowledge Tracing

KT is a well-studied student model introduced in (Corbett & Anderson, 1995) that keeps track over time of the probability that a student has mastered a given skill, given their past performance as evidence. The probability that a skill for a given student is in the "known" (vs. the "unknown") state can then be used to predict future performance.

Constructing KT models involves learning four parameters:

1. Initial Knowledge (L_0) – the probability the student has mastered the skill before attempting the first question
2. Learn Rate (T) – the probability the student will have mastered the skill after attempting a given question if they have not mastered the skill already, independent of performance
3. Guess Rate (G) – the probability the student will answer correctly despite not having mastered the skill
4. Slip Rate (S) – the probability the student will answer incorrectly despite having mastered the skill

KT models can be represented as static, "unrolled" Bayesian networks, as shown in Figure 6. The level of knowledge K_m at time step m influences performance on question Q_m. Initial knowledge influences K_0, while knowledge at time step m-1 influences knowledge at time step m for $m > 0$. The learned T, G and S parameters are the same across all practice opportunities, meaning that the conditional probability tables (CPTs) for all nodes K_m where $m > 0$ have the same values, and the CPTs for all Q nodes have the same values.

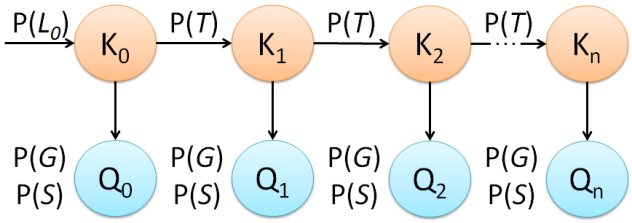

Figure 6: Static Bayesian network representation of Knowledge Tracing

In this work, the Bayes Net Toolbox for Matlab (Murphy, 2001) is used to create separate KT models for each problem set. The parameters for each model are learned using Expectation-Maximization, with initial values of 0.3 for L_0, 0.09 for T, 0.1 for G and 0.09 for S.

Tabling Methods

In previous work (Wang & Heffernan, 2011), a data-driven approach called AM was used to predict performance based on the number of attempts and hints used on the previous problem. This was done by creating a table of probabilities of the student answering the next question correctly on the first attempt without any hints, indexed by the number of attempts and hints used

on the previous problem. These probabilities were computed simply by computing the percentage of questions answered correctly on the first attempt with no hints, parameterized by the number of attempts and hints used on the previous problem.

Then, unseen test data was predicted by using the number of attempts and hints used on the previous problem to do a table lookup. The corresponding probability of getting the next question correct in the table was assigned as the prediction.

In this work, we present a generalization of this approach that serves as a framework for data-driven approaches for student modeling. The general procedure is as follows:

1. Create a table based on one or more attributes of the training data.
2. Compute the probability of answering a question correctly for each combination of values of the attributes selected in Step 1, and insert these probabilities into the proper cells in the table.
3. For each previously unseen test case, do a table lookup based on the attributes of the test case to obtain the probability (over the training data) of the student answering the question correctly.
4. Assign the retrieved probability as the prediction for the test case.

The attributes selected in Step 1 can be anything available or computable from the data, such as the number of hints and attempts used on the previous problem as AM does, or the correctness of the previous problem, the time taken, the type of skill, etc. These attributes could also represent which bin an instance falls into, where bins are constructed by splitting up students and/or problems based on some criteria.

Cells may need to be added to the table when values for one or more of the attributes are not available, depending on the nature of the attributes. If there are not enough data points for certain cells, it may help to simply combine them with others. Finally, depending on the nature of the selected attributes and the data, it may be useful to split certain cells based on some criterion.

In this work, two data-driven approaches that follow this framework are explored: the Assistance Model, as described by Heffernan and Wang and further described below, and the Assistance Progress Model (APM), which constructs a table based on the relationships between hints and attempts used on the previous two problems.

Assistance Model

As described previously, AM consists of a table of probabilities of a student answering a question correctly based on the number of attempts and the percentage of available hints used on the previous problem of the same skill. Attempts are broken into three bins: 1, (1, 6] and (6, ∞),

while the percentage of hints is broken into four: 0, (0, 50], (50, 100) and 100. The AM table constructed from the entire dataset is shown in Table 7.

Table 7: AM table for entire dataset

	Attempts			
Hint %		**1**	**(1, 6]**	**(6, ∞)**
	0	0.778	0.594	0.480
	(0, 50]	0.560	0.623	0.444
	(50, 100)	0.328	0.461	0.444
	100	0.264	0.348	0.374

For instance, according to Table 7, when students answered correctly on the first attempt with no hints, they answered the next question correctly 77.8% of the time. On the other hand, if they required over six attempts and used all of the hints available, they answered the next question correctly only 37.4% of the time.

According to Table 7, when attempts are held constant, the general trend is that as hint usage increases, the probability that the student will answer the next question correctly decreases. This makes sense since hints are more likely to be used by students with lower knowledge of the skill.

When hints are held constant, different patterns occur with respect to the number of attempts used. When no hints are used, the probability of answering the next question correctly decreases as the number of attempts increases. This relationship is reversed when all hints are used. Finally, if just some of the hints are used, making a few attempts (between 2 and 6, inclusive) helps more than making one attempt, but making many attempts (> 6) decreases the probability of answering the next question correctly.

The pattern for no hints can be explained as more attempts required being indicative of lower student knowledge. For all hints being used, more attempts may indicate the student is attempting to learn rather than just requesting hints until the answer is given to them. Using some of the hints suggests the student has not mastered the skill, but has some knowledge of it and is attempting to learn. The relationship between making one attempt and making a few attempts can be explained by the more attempts the student makes, the more they learn, to a point. The use of excessive amounts of attempts probably indicates the student is not learning, despite using some of the hints.

The highest probability in the table, 0.778, corresponds to the case where the previous question was answered correctly. This is unsurprising since in this case, the student likely has mastered the skill. The lowest probability, 0.264, corresponds to making only one attempt while requesting all of the hints. This corresponds to the case where the student requests hints until the answer is given to them. This could be caused by the student simply not understanding the skill, or by the student "gaming the system," or "attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material" (Baker, 2007). In either case, not much learning takes place.

In (Wang & Heffernan, 2011), the AM table was constructed using 80% of the data and used to predict the remaining 20%. In this work, all models were evaluated using five-fold cross-validation.

Assistance Progress Model

AM only takes into account the number of attempts and percentage of hints required on the previous question to predict the student's performance on the following question, without considering the progress the student is making over time in terms of attempts and hints used. APM, on the other hand, takes into account the relationships between the attempts and percentage of hints used on the previous two problems to predict performance on the next question.

The initial model looked like Table 8, each entry corresponding to a case where the second of the two previous problems requires a lower, equal or higher number of attempts or percentage of hints than the one before it. The number of data points for each cell appears in parentheses.

Table 8: Initial APM table for the entire dataset

	Hint % Relationship			
Attempts Relationship		<	=	>
	<	0.672 (586)	0.611 (1410)	0.567 (60)
	=	0.649 (248)	0.734 (8309)	0.590 (83)
	>	0.541 (85)	0.552 (1019)	0.512 (299)

However, it was necessary to extend the model to handle the case where there were fewer than two previous questions, so a separate cell was added for this situation (it had been treated as (equal attempts, equal hint %)). Next, it was observed that certain cells had few observations, so these cells were combined. Finally, it was realized that the (equal attempts, equal hint %) cell

combined data of two very different situations: the case where both questions being compared were answered correctly, and the case where they were both answered incorrectly. Therefore, this cell was split into two cells according to correctness. The final APM table, with probabilities taken over the entire dataset, is shown in Table 9. Cells without enough data on their own have been merged, and the (equal attempts, equal hint %) cell has been split in two: the top cell corresponds to the case when both questions are answered correctly, and the bottom to when both are answered incorrectly. The top-left cell contains the probability that questions with fewer than two predecessors will be answered correctly. The number of data points per cell are in parentheses.

Table 9: APM table for the entire dataset

<table>
<tr><td></td><td colspan="4">Hint % Relationship</td></tr>
<tr><td rowspan="5">Attempts Relationship</td><td>0.708 (2722)</td><td><</td><td>=</td><td>></td></tr>
<tr><td><</td><td>0.672 (586)</td><td>0.611 (1410)</td><td rowspan="3">0.580 (143)</td></tr>
<tr><td>=</td><td rowspan="2">0.649 (248)</td><td>0.791 (5028)</td></tr>
<tr><td></td><td>0.352 (559)</td></tr>
<tr><td>></td><td colspan="2">0.551 (1104)</td><td>0.512 (299)</td></tr>
</table>

According to Table 9, when the relationship between attempts is held constant, the general pattern is that the probability of correctness decreases as the relationship between the percentage of hints used worsens. The (equal attempts, equal hint %) cells do not fit this pattern, though this could be because they are split based on correctness. However, the same cell from Table 8 also does not fit the pattern. The same relationship exists between the attempt relationship and probability of answering correctly when the hint % relationship is held constant, again with the exception of the (equal attempts, equal hint %) cells. These patterns are intuitive, as students who are learning the material should require less assistance from one problem to the next and are likelier to answer the next question correctly, whereas those who are not learning will generally require the same amount of assistance or more to proceed, and are less likely to answer the next question correctly without assistance.

The highest probability in the table corresponds to the case where the hints and attempts used are the same for the previous two questions, and both are answered correctly (0.791). The lowest is when they are the same and are both answered incorrectly (0.352). The former result is intuitive since it corresponds to the case where the student answers two questions in a row correctly, the

best situation represented in the table. The latter corresponds to no progress in terms of assistance over the previous two questions, indicating that little if any learning has taken place.

Ensembling Models

As shown in (Wang & Heffernan, 2011), ensembling models can give better results than any individual model on its own. There are two goals in this work regarding ensembled models: improving the predictive power of AM by ensembling it with APM, and improving the predictive power of KT using both AM and APM. Wang and Heffernan already showed that ensembling KT with AM gives better results than KT on its own. It remains to be seen whether including APM will result in further improvements.

In addition to using means and linear regression models, as done in (Wang & Heffernan, 2011), this work also uses decision trees and random forests.

Evaluation

To evaluate the models, three metrics are computed: MAE, RMSE, and AUC. These metrics are computed by obtaining predictions using five-fold cross-validation (using the same partition for each model), then computing each metric per student. Finally, the individual student metrics are averaged across students to obtain the final overall metrics. Computing the average across students for each metric in this way avoids favoring students with more data than others, and avoids statistical independence issues when it comes to computing AUC. For these reasons, Pardos et al used average AUC per student as their accuracy measure in their work in evaluating several student models and various ways of ensembling them (Pardos et al, 2012).

All three of these metrics are reported because they are concerned with different properties of the set of predictions and therefore do not always agree on which model is best. MAE and RMSE are concerned with how close the real-valued predictions are, on average, to their actual binary values. On the other hand, AUC is concerned with how separable the predictions for positive and negative examples are, or how well the model is at predicting binary classes rather than real-valued estimates.

For example, in Table 10, the first two sets of predictions (P1 and P2) achieve AUCs of 1 since both perfectly separate the two classes (0 and 1). However, P2 achieves much better MAE (0.3960) and RMSE (0.6261) values than P1 (0.5940 and 0.7669, respectively). What's more, P3 achieves an AUC of only 0.5, but outperforms both P1 and P2 in terms of RMSE (0.5292) and P1 in terms of MAE (0.4400).

Table 10: Example dataset

Actual Value	P1	P2	P3
0	0	0.99	0.8
1	0.01	1	0.8

1	0.01	1	0.8
0	0	0.99	0.8
1	0.01	1	0.8

Therefore, it is important to report all of these metrics. As shown above, they do not necessarily agree with each other. Additionally, although MAE and RMSE are similar, not even they always agree on the best model, as RMSE punishes larger errors more than MAE does.

IV. Results

In this section, the results for both the individual models and the ensemble models are reported. Given the importance of reporting multiple accuracy measures as discussed in the preceding section, three measures are reported for each model: MAE, RMSE and AUC. Each measure is computed by first computing the measure for each individual student, then averaging across students. The individual student measures are obtained by using predictions made using five-fold cross-validation, where the folds used are identical for every model.

Individual Models

The results of the three individual models are shown in Table 11. As described before, each of the three metrics are measured for each individual student, and then averaged across students.

In addition to the individual models discussed in Section 3, the results for a baseline model (always predicts 1, the majority class) are reported to serve as a baseline for the other models.

Table 11: Results for the individual models

	MAE	**RMSE**	**AUC**
Baseline	0.2510	0.4642	0.5000
AM	0.3657	0.4129	0.5789
APM	0.3844	0.4221	0.5618
KT	0.3358	0.4071	0.6466

Unsurprisingly, KT performs reliably better than AM and APM in MAE ($t(197) = -8.45$, $p < .0001$; $t(197) = -13.55$, $p < .0001$) and AUC ($t(187) = 6.35$, $p < .0001$; $t(187) = 5.97$, $p < .0001$), and at least marginally better in RMSE ($t(197) = -1.75$, $p = .0824$; $t(197) = -4.44$, $p < .0001$), as KT is a full student model, whereas AM and APM do not attempt to model student knowledge and make predictions solely on the basis of table lookups. Additionally, AM outperforms APM in MAE and RMSE ($t(197) = -12.88$, $p < .0001$; $t(197) = -5.61$, $p < .0001$), which is also not

surprising considering that APM does not consider the actual number of attempts or percentage of hints used, only the relationships between them for the previous two questions. APM also has fewer parameters (9) than AM (12). The difference in AUC was not reliable ($t(187) = 1.62$, $p = .1063$).

The baseline model reliably outperforms all other models in terms of MAE ($t(197) = -15.30$, $p < .0001$; $t(197) = -18.36$, $p < .0001$; $t(197) = -10.62$, $p < .0001$), and reliably underperforms all other models in terms of RMSE ($t(197) = 5.87$, $p < .0001$; $t(197) = 4.92$, $p < .0001$; $t(197) = 6.01$, $p < .0001$) and AUC ($t(187) = -9.72$, $p < .0001$; $t(187) = -6.34$, $p < .0001$; $t(187) = -12.80$, $p < .0001$). It makes sense that the baseline performs well in terms of MAE, given that the mean value of the target attribute, the correctness of a question, is 0.6910. RMSE punishes larger differences more than MAE, making the baseline pay more for its wrong predictions of all cases where the student got the question wrong. Finally, since all predictions share the same value, the baseline cannot do any better than chance at separating the data. Therefore, it earns an AUC value of 0.5000.

These drastic differences in performance for the baseline alone across measures highlight the need for reporting multiple accuracy measures when evaluating student models.

Ensembled Models

In this subsection, various ways of ensembling the individual models are evaluated. Since KT was the best performer of the individual models in all three measures by at least marginally reliable margins, the ensembled models here are compared to KT. In the results for each ensemble method, underlined type indicates measures that are reliably worse than those for KT, **boldface** type indicates measures that are reliably better than those for KT, and regular type indicates there is no reliable difference between the measures for KT and the model in question. Statistical significance was determined using two-tailed pairwise t-tests and Benjamini and Hochberg's false discovery rate procedure (1995).

Mean

The first ensembling method involved taking the simple mean of the predictions given by the various models. This was done in five ways: 1) with AM and APM to determine if it outperformed AM and APM on their own; combining KT with 2) AM and 3) APM to determine if either AM or APM improved predictions over using KT on its own; 4) with all three models to determine if it outperformed any of the individual models, and 5) taking the mean of AM and APM first, then taking the mean of those results with KT. The intuition for the last method is that KT performs better than AM, and most likely APM as well. Therefore, taking the mean of AM and APM first gives KT more influence in the final result while still incorporating both AM and APM. The results for these models are shown in Table 12.

Table 12: Results for the mean models

	MAE	**RMSE**	**AUC**
AM, APM	0.3751	0.4137	0.5917
KT, AM	0.3508	**0.4006**	0.6472
KT, APM	0.3601	0.4033	0.6409
KT, AM, APM	0.3620	0.4032	0.6433
KT, (AM, APM)	0.3554	**0.4010**	0.6469

According to the table above, taking the mean of KT and any combination of AM and APM predictions produces results that do as well as or reliably outperform KT in RMSE and AUC but reliably underperform in MAE. There is no reliable difference between the top two performing models, "KT, AM" and "KT, (AM, APM)" except in MAE, where "KT, AM" performs reliably better ($t(197) = -12.88$, $p < .0001$). Therefore, at least when taking means, adding APM to a model that already includes AM and KT does not reliably improve accuracy in any measure.

Additionally, taking the mean of the AM and APM models yields predictions that are comparable in RMSE and AUC, while reliably worse in MAE ($t(197) = 12.88$, $p < .0001$). Therefore, including APM predictions in mean models does not appear to improve predictive accuracy.

Linear Regression

The second ensembling method is linear regression. In this method, the training data for each fold was used to construct AM, APM and KT models. Predictions were then made for each training instance using these models, and then a linear regression model was built using the three individual predictions as predictors, along with the number of attempts and percentage of hints used, and nominal attributes describing the relationship between the attempts and hints used on the previous two problems. This model was then applied to the fold's test data, whose instances were augmented with predictions from the AM, APM and KT models built from the fold's training data.

Linear regression models were built with six different subsets of the aforementioned features:

1. AM – includes the AM prediction as well as the number of attempts and percentage of hints used on the previous problem
2. AM, KT – the AM set, along with the KT prediction
3. AM, APM* – the AM set, along with the two nominal attributes indicating the relationships between the attempts and hints used for the previous two problems

4. AM, APM*, KT – the AM, APM* set, along with the KT prediction
5. AM, APM – the AM, APM* set along with the APM prediction
6. AM, APM, KT – the AM, APM*, KT set along with the APM prediction

The motivation for testing these subsets of attributes is to determine the relative improvements attained by progressively adding more assistance relationship information to the model, both with and without KT. These models are built in Matlab using the LinearModel class. The results for the linear regression models are shown in Table 13.

Table 13: Results for the linear regression models

	MAE	**RMSE**	**AUC**
AM	0.3701	0.4148	0.5770
AM, KT	**0.3338**	**0.4024**	0.6500
AM, APM*	0.3671	0.4127	0.5753
AM, APM*, KT	**0.3319**	**0.4005**	0.6341
AM, APM	0.3647	0.4112	0.5874
AM, APM, KT	**0.3316**	**0.4000**	0.6379

Not surprisingly, models that incorporate KT predictions all outperform their counterparts that lack KT predictions across all three measures. AM and APM together do better than AM, but not when KT is included. The best combination of models for linear regression is AM and KT, as it was for the mean models.

Unlike its corresponding mean model, the linear regression model that combines AM and KT reliably outperforms KT in MAE and RMSE, and is comparable in terms of AUC. This is consistent with the previous finding that combining AM and KT using linear regression outperforms KT (Wang & Heffernan, 2011), though their model did reliably better than KT for all three measures, which were taken over the entire dataset rather than averaged across students.

Decision Trees

Next, decision tree models were built from the results of the three individual models in the same way that the linear regression models were built, with the exception that the minimum number of data points per leaf and the level of pruning were optimized using brute force search per fold by using sub-fold cross-validation. The search varied the pruning level from 0 to 100% of the model in steps of 5%, and varied the minimum data points per leaf from 5 to 50 in steps of 5.

The decision trees were given the same set of attributes as the linear regression models, and were tested using the same six subsets of those attributes as described above for the linear regression models. The decision trees were built in Matlab using classregtree, specifying the method as 'regression'.

The same sub-folds were used for each fold for all decision tree models. The results for these models are reported in Table 14. The model names correspond to the same subsets of attributes used for the linear regression models.

Table 14: Results for the decision tree models

	MAE	**RMSE**	**AUC**
AM	0.3637	0.4119	0.5793
AM, KT	**0.3293**	**0.4009**	0.6385
AM, APM*	0.3586	0.4087	0.5847
AM, APM*, KT	**0.3286**	**0.4008**	0.6358
AM, APM	0.3586	0.4090	0.5860
AM, APM, KT	**0.3290**	**0.4012**	0.6351

As for the linear regression models, the models that include KT predictions perform better than those that did not, across all three accuracy measures. Adding APM* to AM reliably improves accuracy, but there is no difference between this and combining AM and APM. Adding APM features of any kind do not improve models that include KT predictions. As for the linear regression models, the decision tree that performs the best is the one that only includes KT and AM, which reliably outperforms KT in both MAE and RMSE, with no reliable difference in AUC.

Random Forest

The final ensembling method used in this work was Random Forest, which is a collection of decision trees where each individual decision tree was built from a random subset of the attributes and a random subset of the data. In this work, random forests consisted of 1,000 such trees, which were each built randomly from any subset of the attributes and between 10% and 90% of the data. The prediction of the random forest as a whole for a given test instance was the simple mean of the predictions given by each individual tree within the forest. The trees were regression trees and required a minimum of five data points per leaf node. No pruning was done, as varying the pruning levels did not appear to significantly affect the predictive accuracy of the forests for this dataset.

The same set of attributes used for linear regression and decision trees were used in the random forest models, and the same six attribute subsets were tested separately as for the other methods.

With the exception of MAE (many of the predictions were 1, which happens to be the majority class), these models performed worse than the other ensembling methods. This could be due to most of the trees being overfit to the training data, as sub-fold cross-validation with brute force search of optimal pruning parameters was not performed for these trees as it was for the individual decision trees reported on in the previous section.

However, averaging these models with KT produced better results, as shown in Table 15.

Table 15: Results for averaging the KT and random forest models

	MAE	**RMSE**	**AUC**
AM	0.3505	**0.4002**	0.6461
AM, KT	**0.3054**	0.4117	0.6313
AM, APM*	0.3479	**0.3985**	0.6477
AM, APM*, KT	0.3005	**0.4109**	0.6358
AM, APM	0.3485	**0.3990**	0.6468
AM, APM, KT	0.2997	**0.4090**	0.6375

Unlike other ensembling methods, when random forest predictions are averaged with those of KT, progressively more APM data improves accuracy, though not always significantly. Otherwise, adding APM predictions appears to worsen results.

Overall

For the first three ensembling methods, those that included only AM and KT performed the best. However, for random forests, it was the average of KT with the random forest consisting of predictions from all three individual models. Table 16 reproduces these results, with **bold-faced** type indicating values that are reliably better than KT, and underlined type indicating values that are reliably worse. Table 17 reports the p-values of the differences between these models for each accuracy measure, with values indicating reliable differences in **bold-faced** type.

Table 16: Results for the best of each ensembling method

	MAE	**RMSE**	**AUC**
MEAN	0.3508	**0.4006**	0.6472
LR	**0.3338**	**0.4024**	0.6500

TREE	**0.3293**	**0.4009**	0.6385
RF	**0.2997**	0.4090	0.6375

Table 17: Significance tests for the best ensembling methods

	MAE	**RMSE**	**AUC**
MEAN, LR	**0.0000**	0.1659	0.4274
MEAN, TREE	**0.0000**	0.8803	0.1116
MEAN, RF	**0.0000**	**0.0022**	0.1400
LR, TREE	**0.0000**	0.1669	0.0223
LR, RF	**0.0000**	**0.0026**	0.0406
TREE, RF	**0.0000**	**0.0001**	0.8476

From Tables 16 and 17, it appears that either the decision tree or random forest (averaged with KT) models could be considered the best model, depending on which measure is considered the most important. The random forest model is reliably better than the decision tree in terms of MAE, but reliably worse in terms of RMSE.

In general, it appears there is some value in comparing the usage of assistance over the previous two problems, as ensembling APM with AM consistently gives better results than using AM on its own, except when taking means. Despite this, ensemble methods that use only KT and AM perform better than any other model studied in this work, including all of those using APM. One explanation could be that one important thing that APM captures is learning over the previous two questions, which is already modeled in KT. The one exception is when a random forest of all individual models is averaged with KT, which indicates that there is information that APM takes into account that neither AM nor KT considers. Right now, it is not clear which of these ensemble models is best given the disagreement among the metrics. It depends on the relative importance placed on each metric.

V. Discussion and Future Work

In this work, we generalized an existing raw data model, AM, into a framework for predicting student performance by tabling raw data. This framework provides an efficient way for adding new sources of information into existing student models. From there, we developed a new model, APM, which makes predictions based on the relationship between the assistance used on

the previous two problems. Finally, we evaluated these models and KT, and then explored several ways of ensembling these models together.

We found that although APM is not as predictive as AM, combining the two with various ensembling methods produces models that reliably outperform AM on its own. This shows that prediction accuracy can be strengthened by recognizing the progress a student makes in terms of the assistance they use. However, for the most part, the best models studied in this paper were those that only ensembled KT and AM. Adding APM to such models did not improve accuracy, except in the case of random forests averaged with KT. Despite this, it is still evident that there is value in considering student progress in terms of assistance. Perhaps there are better methods of incorporating that information into predictive models that will yield better results.

We also confirmed that ensembles of AM and KT reliably outperform KT, in line with previous work (Wang & Heffernan, 2011). Whereas previous work showed this was the case when computing the measures across all problem logs, this work shows it also holds when the measures are computed as averages across students.

We reported three different accuracy measures to fairly compare models against each other, and argued that reporting multiple measures is necessary since they measure different properties of the predictions and therefore do not always agree on which model is best. We also argued that computing these measures per student, then averaging across students is more reliable than treating all problems as equal since the latter approach favors models that are biased towards students with more data.

Although we found that the ensemble methods perform better than KT at predicting performance, such models are difficult to interpret and therefore may be limited in usefulness. Fitting a KT model for a given skill yields four meaningful parameters that describe the nature of that skill, whereas ensemble methods in this work give models of how to computationally combine predictions from KT and AM to maximize predictive accuracy. Since KT models student knowledge, it can be used to guide an ITS session. KT can also be extended to quantify the effects of help (Beck et al, 2008), gaming (Gong et al, 2010), and individual items (Pardos, Dailey & Heffernan, 2011), among other factors, on learning and performance. It appears the usefulness of the ensemble methods is limited to prediction of the next question, a task that serves as a good measure of the validity of a student model but does not appear to be useful in guiding ITS interaction.

On the other hand, AM and APM are simple to compute and do not suffer from the identifiability problem that KT does (Pardos & Heffernan, 2010b). AM and APM consist of summaries of the raw data rather than inferred parameters. Although not as predictive as KT, AM and APM give interpretable statistics with little chance of overfitting. Additionally, they consider the assistance used, which could indicate the usefulness of a system's help features. Other tabling methods

could be used to study the effects of other aspects of ITS, though likely with lower predictive accuracy than KT due to the limited set of values such methods can use as predictions.

Since the ensemble models outperformed KT but appear to be limited to predicting a student's performance on the next question, finding a way to use such predictions within ITS would be a useful contribution. Question selection could be a possible application (i.e. selecting an easier question if the model predicts the student will answer the next question incorrectly).

Another direction for future work could be determining other useful specializations of the framework we presented for building models from raw data. AM and APM focus on assistance, but other attributes could prove useful. Additionally, this work did not investigate grouping students or problems.

Another future direction could be determining why some models, like APM, can reliably improve a model such as AM when ensembled with it, but not improve results when a third model such as KT is involved. It appears that the information that APM uses is important, but may not be used by APM in the best way possible. Examining the use of assistance over the course of more than just the previous two problems may also prove useful.

Finally, experimenting with other methods of ensembling the models described here and other raw data models within this framework is also worth looking into. Previous work experimented with means and linear regression (Wang & Heffernan, 2011), and this work expanded upon those methods by including decision trees and random forests. However, other work in ensembling student models suggests that neural networks may perform better (Pardos et al, 2012).

Chapter 5: Using a simplified process to fit Knowledge Tracing models

Student modeling is an important component of ITS research because it can help guide the behavior of a running tutor and help researchers understand how students learn. Due to its predictive accuracy, interpretability and ability to infer student knowledge, Corbett & Anderson's Bayesian Knowledge Tracing is one of the most popular student models. However, researchers have discovered problems with some of the most popular methods of fitting it. These problems include: multiple sets of highly dissimilar parameters predicting the data equally well (identifiability), local minima, degenerate parameters, and computational cost during fitting. Some researchers have proposed new fitting procedures to combat these problems, but are more complex and not completely successful at eliminating the problems they set out to prevent. We instead fit parameters by estimating the mostly likely point that each student learned the skill, developing a new method that avoids the above problems while achieving similar predictive accuracy.

This chapter was published at the following venue:
Hawkins, W., Heffernan, N.T., Baker, R.S.J.d. (2014) Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. To appear in *Proceedings of the 12th International Conference on Intelligent Tutoring Systems*, Honolulu, HI.

I. Introduction

Within the field of Intelligent Tutoring Systems (ITSs), student modeling is important because it can help guide interaction between a student and an ITS. By having a model of student knowledge, an ITS can estimate how knowledgeable a student is of various knowledge components (or "skills") over time and use that to determine what the student needs to practice.

However, student modeling is also important to researchers. The parameters learned from BKT can be used to characterize how students learn and to evaluate ITS content. Examples of this include studying the effects of "gaming the system" on learning (Gong et al, 2010) and evaluating hint helpfulness (Beck et al, 2008), among many other studies.

While BKT is popular and useful, researchers have found problems with fitting BKT models. One such problem is identifiability: there may be multiple sets of parameters that fit the data equally well (Beck & Chang, 2007), making interpretation difficult. Additionally, the learned parameters may produce what is called a degenerate model, or a model that fits the data well but violates the assumptions of the approach, generally leading to inappropriate pedagogical decisions if used in a real system (Baker, Corbett & Aleven, 2008).

Two popular fitting methods in the literature, Expectation-Maximization (EM) (Moon, 1996) and brute force grid search, both suffer from identifiability. Additionally, EM can get stuck on local minima, and brute force comes with a high computational cost.

Researchers have attempted to deal with these issues through strategies like limiting the values brute force searching can explore (Baker et al, 2010a), determining which starting values lead to degenerate parameters in EM (Pardos & Heffernan, 2010b), computing Dirichlet priors for each parameter and using these to bias the search (Rai, Gong & Beck, 2009), clustering parameters across similar skills (Ritter et al, 2009), and using machine-learned models to detect two of the parameters (Baker, Corbett & Aleven, 2008).

This work introduces a simple method of estimating BKT parameters that sacrifices the precision of optimization techniques for the efficiency and interpretability of empirical estimation. Briefly, we estimate when students learn skills heuristically, and then use these estimates to help compute the four BKT parameters. Our goal is to efficiently produce accurate, non-degenerate BKT models.

II. Data

For this work, we used data from ASSISTments (Feng, Heffernan & Koedinger, 2009), an ITS used primarily by middle- and high-school students. In this dataset taken from the 2009-10 school year, 1,579 students worked on 61,522 problems from 67 skill-builder problem sets. The skill-builders used had data from at least 10 students, used default mastery settings (three consecutive correct answers to achieve mastery, ending the assignment), and had at least one student achieve mastery. A student's data was only included for a specific skill-builder if they answered at least three questions.

III. Methods

In this work, we developed and analyzed a new fitting procedure for BKT. We begin this section by describing BKT and then introduce our empirical approach to fitting BKT models. Finally, we describe the analyses we performed.

Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (Corbett & Anderson, 1995) is a student model used in ITS research that infers a student's knowledge given their history of responses to problems, which it can use to predict future performance. Typically, a separate BKT model is fit for each skill. It assumes that a given student is always either in the known state or the unknown state for a given skill, with a certain probability of being in each. To calculate the probability that a student knows the skill given their performance history, BKT needs to know four probabilities: $P(L_0)$, the probability a student knows the skill before attempting the first problem; P(T), the probability a student who does not currently know the skill will know it after the next practice opportunity; P(G), the probability a student will answer a question correctly despite not knowing the skill; and P(S), the probability a student will answer a question incorrectly despite knowing the skill.

According to this model, knowledge affects performance (mediated by the guess and slip rates), and knowledge at one time step affects knowledge at the next time step: if a student is in the unknown state at time *t*, then the probability they will be in the known state at time *t+1* is P(T).

Additionally, BKT models typically assume that forgetting does not occur: once a student is in the known state, they stay there.

Computing Knowledge Tracing Using Empirical Probabilities

In this section, we present a new approach to fitting BKT models we call Empirical Probabilities (EP). EP is a two-step process that involves annotating performance data with knowledge, and then using this information to compute the BKT parameters.

Annotating Knowledge

The first step in EP is to annotate performance data for each student within each skill with an estimate of when the student learned the skill. We assume there are only two knowledge states: known (1) and unknown (0), and do not allow for forgetting (a known state can never be followed by an unknown state).

In this work, we use a simple heuristic for determining when a student learns a skill: we choose the knowledge sequence that best matches their performance. This is illustrated by Figure 7. A full description of this heuristic can be found online (Hawkins, 2014).

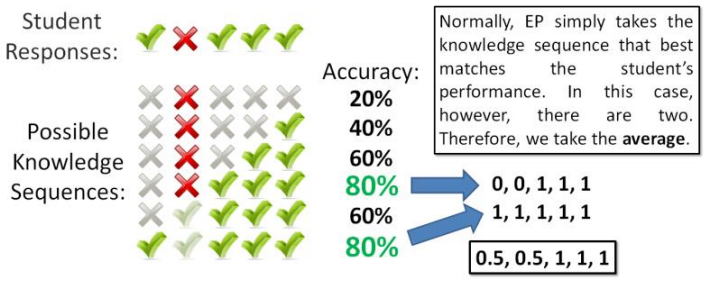

Figure 7: Each of the six possible knowledge sequences are tried for a student's performance history, and in this case, the best two are averaged together to get the final sequence.

Computing the Probabilities

Using the knowledge estimates, we were able to compute each of the four BKT parameters for each skill empirically from the data.

The first of these parameters is $P(L_0)$, the probability that the student knew the skill before interacting with the system. We can empirically estimate this by taking the average value of student knowledge on the first practice opportunity:

$$P(L_0) = \frac{\sum K_0}{|K_0|} \quad (1)$$

Equation (1) is similar to a heuristic in (Pardos & Heffernan, 2010a) for estimating individual student prior knowledge. While that paper used performance to compute a prior for each student as opposed to using knowledge to compute a prior for each skill as we do here, the idea that prior knowledge can be estimated mathematically in this way is similar.

Using K_i and C_i as knowledge and correctness at problem *i*, respectively, the following equations are used to compute the other three BKT parameters:

$$P(T) = \frac{\sum_{i\neq 0}(1-K_{i-1})K_i}{\sum_{i\neq 0}(1-K_{i-1})} \quad (2)$$

$$P(G) = \frac{\sum_i C_i(1-K_i)}{\sum_i(1-K_i)} \quad (3)$$

$$P(S) = \frac{\sum_i(1-C_i)K_i}{\sum_i K_i} \quad (4)$$

Experiments

In this paper, we compare BKT models fit with EM and EP in terms of predictive accuracy, model degeneracy, and training time. Due to space constraints, only the predictive accuracy results are reported here. Results for the other experiments as well as the code and data used in all the experiments are available online (Hawkins, 2014).

To fit EM, we used Murphy's Bayes Net Toolbox for MATLAB (BNT) (Murphy, 2001). For EM, it is necessary to specify a starting point. We chose an initial $P(L_0)$ of 0.5, and set the other three parameters to 0.1. Additionally, we set a maximum of 100 iterations and used the default BNT improvement threshold value of 0.001.

To compute the parameters using EP, we implemented the equations in the previous section in MATLAB using basic functionality. Then, we entered these values into the conditional probability tables of a BKT model constructed with BNT.

IV. Results

First, we examine how predictive each method is of student performance under five-fold student-level cross-validation. We evaluated the methods using mean absolute error (MAE), root mean squared error (RMSE), and A'. These metrics were computed for each student and then used in two-tailed paired t-tests to determine the significance of the differences between the overall means of the two models. The degrees of freedom for the MAE and RMSE significance tests was one less than the number of students, whereas that of the A' significance test was lower due to some students being excluded (students who gave all correct or all incorrect answers for all skills were excluded since A' is undefined in such cases). The values below represent the average of the student metrics. Lower values of MAE and RMSE indicate better performance, whereas the opposite is true of A'. The results are shown in Table 18.

Table 18: Prediction results for the two methods of learning BKT parameters: Expectation Maximization and Empirical Probabilities

Learning Method	MAE	RMSE	A'
EM (BNT)	0.3830	0.4240	0.5909
EP	0.3742	0.4284	0.6145

Although the differences between these metrics are all statistically significant according to two-tailed paired t-tests (MAE: $t(1,578) = 10.88$, RMSE: $t(1,578) = -6.74$, A': $t(1,314) = -7.01$, $p < 0.00001$), the differences are small. Therefore, we believe the two methods are comparable in terms of predicting performance.

We also tested EM and EP in terms of model degeneracy and fitting time. In summary, we found that only EM learned degenerate parameters, and that EP runs significantly faster than EM. The full results are available online (Hawkins, 2014).

V. Conclusions and Future Work

From this work, it appears that a simple estimation of knowledge followed by computing empirical probabilities may be a reasonable approach to estimating BKT parameters. We found that EP had comparable predictive accuracy to that of EM. Additionally, it is mathematically impossible for EP to learn theoretically degenerate guess and slip rates (i.e. above 0.5) (Hawkins, 2014), and it is at least as good as EM at avoiding empirically degenerate parameters, based on tests suggested and used in (Baker, Corbett & Aleven, 2008). We also found it was considerably faster than EM (Hawkins, 2014).

An improvement to EP would be to annotate knowledge more probabilistically. EP makes only binary inferences of knowledge based on predictive performance. For example, EP always considers incorrect responses on the first problem to be made in the unknown state, even though some of these are slips. Therefore, a more probabilistic approach may be able to produce better parameter estimates.

EP could be used as a tractable way to help improve accuracy by incrementally incorporating data into models as it becomes available during a school year. This would improve models for skills with little or no previous data and make use of student and class information. If a skill has little or no previous data, using current school year data may improve estimates of its parameters. Also, it has been shown that incorporating student (Pardos & Heffernan, 2010a) and class (Wang & Beck, 2013) information can improve predictive performance, which cannot be done before the start of a school year.

While EP achieves similar accuracy to EM and appears not to learn degenerate parameters, we did not perform any external validations of the learned parameters for either approach. Such an analysis would help determine how much we can trust EP parameters, especially when they differ from those learned by EM.

References

Arroyo, I., Woolf, B.P., Cooper, D., Burleson, W., Muldner, K., Christopherson, R. (2009) Emotion Sensors Go To School. In Dimitrova, V., Mizoguchi, R., du Boulay, B., Graesser, A.C. (Eds.) *Proceedings of the 14th International Conference on Artificial Intelligence In Education*, Brighton, UK, pp. 17-24.

Arroyo, I., Cooper, D.G., Burleson, W., & Woolf, B.P. (2010). Bayesian Networks and Linear Regression Models of Students' Goals, Moods, and Emotions. In Romero, C., Ventura, S., Pechenizkiy, M., & Baker, R.S.J.d. (Eds.) *Handbook of educational data mining* (pp. 323-338). Boca Raton, FL: CRC Press.

Baker, R.S.J.d. (2007) Is Gaming the System State-or-Trait? Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model. In *Comple On-Line Proceedings of the Workshop on Data Mining for User Modeling, at the 11th International Conference on User Modeling (UM 2007)*, Corfu, Greece, pp. 76-80. Boston, MA: User Modeling Inc.

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

Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.M., Kauffman, L.R., Mitchell, A.P., Giguere, S. (2010) Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. In De Bra, P., Kobsa, A., Chin, D. (Eds.) UMAP 2010. LNCS, vol. 6075/2010, pp. 52-63. Springer-Verlag, Berlin Heidelberg

Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System." In Dykstra-Erickson, E. & Tscheligi, M. (Eds.) *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems*, Vienna, Austria, pp. 383-390.

Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., Graesser, A.C. (2010). Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments. *International Journal of Human-Computer Studies*, *68*(4), 223-241.

Baker, R.S.J.d., Moore, G., Wagner, A., Kalka, J., Karabinos, M., Ashe, C., Yaron, D. (2011) The Dynamics Between Student Affect and Behavior Occuring Outside of Educational Software. In D'Mello, S., Graesser, A., Schuller, B., Martin, J.-C. (Eds.) ACII 2011. LNCS, vol. 6974/2011, pp. 14-24. Springer-Verlag, Berlin Heidelberg

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

Beck, J.E., Chang, K., Mostow, J., Corbett, A. (2008) Does help help? Introducing the Bayesian Evaluation and Assessment methodology. In Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (Eds.) ITS 2008. LNCS, vol. 5091/2008, pp. 383-394. Springer, Berlin Heidelberg.

Belton, T., Priyadharshini, E. (2007). Boredom and schooling: A cross-disciplinary exploration. *Cambridge Journal of Education, 37*(4), 579−595.

Benjamini, Y., Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society, Series B, 57*(1), 289–300.

Conati, C., Maclaren, H. (2009). Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, *19*(3), 267-303.

Condry, J. (1978). The role of incentives in socialization. In Lepper, M. & Greene, D. (Eds.) *The hidden costs of reward: new perspectives on the psychology of human motivation* (pp. 179-192). New York: John Wiley & Sons.

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

Cullingford, C. (2002). *The best years of their lives? Pupils' experience of school.* London: Kogan Page.

D'Mello, S.K., Craig, S.D, Witherspoon, A.W., McDaniel, B.T., Graesser, A.C. (2008). Automatic Detection of Learner's Affect from Conversational Cues. *User Modeling and User-Adapted Interaction*, *18*(1-2), 45-80.

D'Mello, S.K., Graesser, A.C. (2010). Multimodal semiautomated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, *20*(2), 147-187.

Doddannara, L.S., Gowda, S.M., Baker, R.S.J.d, Gowda, S.M., de Carvalho, A.M.J.B. (2013) Exploring the relationships between design, students' affective states, and disengaged behaviors within an ITS. In Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (Eds.) AIED 2013. LNCS, vol. 7926/2013, pp. 31-40. Springer, Berlin Heidelberg.

Farmer, R., Sundberg, N. (1986) Boredom proneness: the development and correlates of a new scale. *Journal of Personality Assessment*, *50*(1), 4–17.

Farrell, E. (1988). Giving Voice to High School Students: Pressure and Boredom, Ya Know What I'm Sayin'? *American Educational Research Journal*, *25*(4), 489-502.

Feng, M., and Heffernan, N.T. (2010) Can We Get Better Assessment From a Tutoring System Compared to Traditional Paper Testing? Can We Have Our Cake (Better Assessment) and Eat It Too (Student Learning During the Test)? In Baker, R.S.J.d, Merceron, A., Pavlik, P.I. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*, Pittsburgh, PA, pp. 41-50.

Feng, M., Heffernan, N.T., Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *User Modeling and User-Adapted Interaction*, *19*(3), 243-266.

Finn, J.D. (1989). Withdrawing From School. *Review of Educational Research*, *59*(2), 117-142.

Gjesme, T. (1977). General Satisfaction and Boredom at School as a Function of the Pupils' Personality Characteristics. *Scandinavian Journal of Educational Research*, *21*(1), 113-46.

Gong, Y., Beck, J., Heffernan, N., Forbes-Summers, E. (2010) The impact of gaming (?) on learning at the fine-grained level. In Aleven, V., Kay, J., Mostow, J. (Eds.) ITS 2010. LNCS, vol. 6094/2010, pp. 194-203. Springer-Verlag, Berlin Heidelberg

Harris, M.B. (2000). Correlates and Characteristics of Boredom Proneness and Boredom. *Journal of Applied Social Psychology*, *30*(3), 576–598.

Hawkins, W.J. (2014). *Publications*. Retrieved from https://sites.google.com/site/whawkins90/publications

Johns, J., Mahadevan, S., Woolf, B. (2006) Estimating student proficiency using an item response theory model. In Ikeda, M., Ashley, K.D., Chan, T. (Eds.) ITS 2006. LNCS, vol. 4053/2006, pp. 473-480. Springer, Berlin Heidelberg

Karweit, N., Slavin, R.E. (1982). Time-On-Task: Issues of Timing, Sampling, and Definition. *Journal of Educational Psychology*, *74*(6), 844-851.

Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A. (1997). Intelligent Tutoring Goes To School in the Big City. *International Journal of Artificial Intelligence in Education*, *8*(1), 30-43.

Larson, R.W., Richards, M.H. (1991). Boredom in the Middle School Years: Blaming Schools versus Blaming Students. *American Journal of Education*, *99*(4), 418-443.

Lepper, M.R., Woolverton, M., Mumme, D.L., & Gurtner, J. (1993). Motivational Techniques of Expert Human Tutors: Lessons for the Design of Computer-Based Tutors. In Lajoie, S.P. & Derry, S.J. (Eds.) *Computers as cognitive tools* (pp. 75-105). Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc.

Mikulas, W. (1993). The essence of boredom. *Psychological Record*, *43*(1), 3–13.

Moneta, G., Csikszentmihalyi, M. (1996). The effect of perceived challenges and skills on the quality of subjective experience. *Journal of Personality*, *64*(2), 275–310.

Moon, T. K. (1996). The expectation–maximization algorithm. *IEEE Signal Process. Mag.*, *13*, 47–60.

Muldner, K., Burleson, W., Van de Sande, B., VanLehn, K. (2011). An analysis of students' gaming behaviors in an intelligent tutoring system: predictors and impacts. *User Modeling and User-Adapted Interaction - Special Issue on Data Mining for Personalized Educational Systems*, *21*(1-2), 99-135.

Murphy, K. The bayes net toolbox for matlab. (2001). *Computing science and statistics*, *33*(2), 1024-1034.

Pardos, Z.A., Dailey, M.D., Heffernan, N.T. (2011). Learning what works in ITS from non-traditional randomized controlled trial data. *International Journal of Artificial Intelligence in Education*, *21*(1), 47-63.

Pardos, Z.A., Gowda, S. M., Baker, R.S.J.d., Heffernan, N. T. (2012). The Sum is Greater than the Parts: Ensembling Models of Student Knowledge in Educational Software. *ACM SIGKDD Explorations*, *13*(2), 37-44.

Pardos, Z.A., Heffernan, N.T. (2011) KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (Eds.) UMAP 2011. LNCS, vol. 6787/2011, pp. 243-254, Springer

Pardos, Z.A., Heffernan, N.T. (2010) Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In De Bra, P., Kobsa, A., Chin, D.N. (Eds.) *Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, Big Island, Hawaii, pp. 255-266.

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

Pavlik, P.I., Cen, H., Koedinger, K. (2009) Performance Factors Analysis – A New Alternative to Knowledge. In Dimitrova, V., Mizoguchi, R., du Boulay, B., Graesser, A.C. (Eds.) *Proceedings of the 14th International Conference on Artificial Intelligence In Education*, Brighton, UK, pp. 531-538.

Pekrun, R., Goetz, T., Daniels, L.M., Stupnisky, R.H., Perry, R.P. (2010). Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion. *Journal of Educational Psychology*, *102*(3), 531-549.

Raftery, A.E. (1995). Bayesian Model Selection in Social Research. *Sociological Methodology*, *25*, 111-163.

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

Razzaq, L., Mendicino, M., Heffernan, N.T. (2008) Comparing classroom problem-solving with no feedback to web-based homework assistance. In Woolf, B., Aimeur, E., Nkambou, R., Lajoie, S. (Eds.) ITS 2008. LNCS, vol. 5091/2008, pp. 426-437. Springer, Berlin Heidelberg

Reid, K. (1986). *Disaffection from school*. London: Methuen.

Ritter, S., Harris, T.K., Nixon, T., Dickison, D., Murray, R.C. (2009) Reducing the Knowledge Tracing Space. In Barnes, T., Desmarais, M., Romero, C., Ventura, S. (Eds.) *Proceedings of the 2nd International Conference on Educational Data Mining*, Cordoba, Spain, pp. 151-160.

Sabourin, J., Rowe, J., Mott, B., Lester, J. (2011) When Off-Task is On-Task: The Affective Role of Off-Task Behavior in Narrative-Centered Learning Environments. In Biswas, G., Bull, S., Kay, J., Mitrovic, A. (Eds.) AIED 2011. LNCS, vol. 6738/2011, pp. 534-536. Springer, Berlin Heidelberg

San Pedro, M., Baker, R.S.J.d., Bowers, A., Heffernan, N.T. (2013) Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In D'Mello, S., Calvo, R., Olney A. (Eds.) *Proceedings of the 6th International Conference on Educational Data Mining*, Memphis, TN, pp. 177-184.

San Pedro, M., Baker, R.S.J.d, Gowda, S.M., Heffernan, N.T. (2013) Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In Lane, H.C., Yacef, K., Mostow, M., Pavlik, P. (Eds.) AIED 2013. LNCS, vol. 7926/2013, pp.41-50. Springer-Verlag, Berlin Heidelberg.

Storey, J.D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society*, *64*(3), 479-498.

VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, *46*(4), 197-221.

Vodanovich, S. (2003). Psychometric measures of boredom: A review of the literature. *Journal of Psychology*, *137*(6), 569–595.

Wang, Y., Beck, J. (2013) Class vs. Student in a Bayesian Network Student Model. In Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (Eds.) AIED 2013. LNCS, vol. 7926/2013, pp. 151-160. Springer-Verlag, Berlin Heidelberg.

Wang Y., Heffernan N. T. (2011) The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. In Murray, R.C., McCarthy, P.M. (Eds.) *Proceedings of the 24th International FLAIRS Conference*, Palm Beach, FL, pp. 549-554.

Wang, Y., Heffernan, N.T. (2012) The Student Skill Model. In Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K.-K. (Eds.) ITS 2012. LNCS, vol. 7315/2012, pp. 399-404. Springer-Verlag, Berlin Heidelberg.

Wang, Y., Heffernan, N.T. and Beck, J.E. (2010) Representing Student Performance with Partial Credit. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*, Pittsburgh, PA, pp. 335-336.

Xu, Y., Mostow, J. (2012) Comparison of methods to trace multiple subskills: Is LR-DBN best? In Yacef, K., Zaiane, O., Hershkovitz, H., Yudelson, M., Stamper, J. (Eds.) *Proceedings of the Fifth International Conference on Educational Data Mining*, Chania, Greece, pp. 41-48.