Sequencing Educational Content in Classrooms using Bayesian Knowledge Tracing

Yossi Ben David
Department of Information Systems Engineering
Ben-Gurion University, Israel
bendavidyossi@gmail.com

Avi Segal
Department of Information Systems Engineering
Ben-Gurion University, Israel
avise@post.bgu.ac.il

Ya'akov (Kobi) Gal
Department of Information Systems Engineering
Ben-Gurion University, Israel
kobig@bgu.ac.il

ABSTRACT

Despite the prevalence of e-learning systems in schools, most of today's systems do not personalize educational data to the individual needs of each student. This paper proposes a new algorithm for sequencing questions to students that is empirically shown to lead to better performance and engagement in real schools when compared to a baseline approach. It is based on using knowledge tracing to model students' skill acquisition over time, and to select questions that advance the student's learning within the range of the student's capabilities, as determined by the model. The algorithm is based on a Bayesian Knowledge Tracing (BKT) model that incorporates partial credit scores, reasoning about multiple attempts to solve problems, and integrating item difficulty. This model is shown to outperform other BKT models that do not reason about (or reason about some but not all) of these features. The model was incorporated into a sequencing algorithm and deployed in two classes in different schools where it was compared to a baseline sequencing algorithm that was designed by pedagogical experts. In both classes, students using the BKT sequencing approach solved more difficult questions and attributed higher performance than did students who used the expert-based approach. Students were also more engaged using the BKT approach, as determined by their interaction time and number of log-ins to the system, as well as their reported opinion. We expect our approach to inform the design of better methods for sequencing and personalizing educational content to students that will meet their individual learning needs.

1. INTRODUCTION

The proliferation of e-learning systems in schools means that educational software is increasingly used by a wide array of learners from different age groups, socio-economic backgrounds and cultures. This creates new opportunities for using computational methods to support students in their learning process.

We focus on e-learning systems that are deployed in K12 schools, consisting of large repositories of questions in a variety of subjects (e.g., math, English, physics). Students can solve exercises and practice their skills in tests. Teachers can prepare assignments to the entire class or to a group of students, assess their students' answers and follow their progress using graphs and reports. In contrast to intelligent tutoring systems, in which hints and content selection are adapted to the students' needs, there is no "off-the-shelf" sequencing mechanism that is available in e-learning systems. For the most part, these systems require teachers to select the questions for students to solve, which demands time and effort, or allow students free choice about which questions to solve, which may result in students picking very easy questions that do not advance their learning. This paper describes a general approach for augmenting e-learning systems that are deployed in schools with automatic tools for sequencing educational content to students.

Our approach follows the mastery learning paradigm [3] in which domain knowledge is divided into a hierarchy of component skills, students learn skills at their own pace, and knowledge of simple skills should be demonstrated before moving on to more difficult questions relating to more complex skills. We model the acquisition of students' knowledge of these skills over time using a computational model, and use this model to personalize educational content to students.

There is significant work on computational models for tracing students' knowledge to determine when skills have been learned. A family of methods called Bayesian Knowledge Tracing (BKT) use probabilistic methods and machine learning to model students' skill acquisition in Intelligent Tutoring Systems. Most uses of the BKT model focus on predicting individual student performance on a given set of problems, or selecting the set of questions that are deemed most suitable for the students' inferred skill level. We focus on a different problem: which question to select next that will best advance the student's knowledge given the student's inferred skill level.

The basic BKT model includes inherent assumptions that limit its applicability in e-learning systems: it assumes that question answers are dichotomous (correct or incorrect), it does not reason about question difficulty and it does not consider the general case where students perform several attempts to solve the same given problem. Extensions to the basic BKT model have been proposed separately, but integrating them together poses additional challenges to the model representation and parameter learning. We present a model-selection approach which receives as input a stu-

LAK '16 April 25-29, 2016, Edinburgh, United Kingdom

ACM ISBN 978-1-4503-2782-4/14/04 ...$15.00.
http://dx.doi.org/10.1145/2883851.2883885.
DOI: 10.1145/1235

dent's response and a set of candidate BKT models, and chooses the best performing BKT model from the training set to use for predicting the student's performance on the question. Our extension to the BKT model is shown to lead to significant improvement from the state of the art BKT models, when compared on two e-learning datasets in mathematics from the literature, that contain millions of responses to questions.

The model described above was integrated into a sequencing algorithm that selects questions to students in a way that is based on their inferred skill mastery. The algorithm receives as input a set of questions for a student and a BKT model, and ranks the questions based on the predicted score given by the model as well as the student's knowledge of the relevant skills. It selects the question that is intended to advance the student's knowledge of the relevant skill, while matching the student's capabilities, so that the question is within the "zone of possible achievement" for the student [21].

We compared the performance of students using this algorithm to that of students using a sequencing approach that was engineered by pedagogical experts and which selects questions in increasing order of difficulty. We compared both approaches empirically in two different schools in Israel using the math component of an e-learning system that includes more than 10,000 questions divided into hundreds of topics. In each school, two classes were chosen in which students exhibited similar performance on a set of predetermined math questions. All students in a class subsequently used the BKT or the expert-based sequencing approach for a period of several weeks, in which they were able to use the system at their leisure at home or in school. Our results showed that in both schools, students using the BKT-Sequencing algorithm solved more harder questions (e.g., higher difficulty levels), performed significantly better on harder questions, and spent more time in the system than did the students using the expert-based approach. In a follow-up survey that was administered in one of the schools, the BKT-Sequence algorithm was reported by students to be more helpful than the expert-based approach.

The contributions of this paper are threefold. First, it extends the Bayesian Knowledge Tracing literature to reason about partial credit scores and number of question retries. Second, it allows to choose the best performing BKT model on new data using model selection. Third, it provides a new algorithm for sequencing questions to students in realtime that was evaluated in two different schools, demonstrating that combining knowledge tracing models in existing e-learning systems can improve students' performance.

2. BACKGROUND AND RELATED WORK

Our work is based on computational models for tracing students' knowledge in e-learning, called Bayesian Knowledge Tracing (BKT). BKT was originally proposed by Corbett and Anderson [6, 7] to model student's evolving knowledge of skills required to solve problems in an intelligent tutoring system for teaching LISP programming. BKT models students' knowledge as a set of binary variables, one per skill, where the skill is either mastered by the student or not. Future implementations of BKT used a Dynamic Bayesian Network to maintain a probability distribution over knowledge of each skill. Observations in the BKT model consist of the correctness of students' responses to questions which

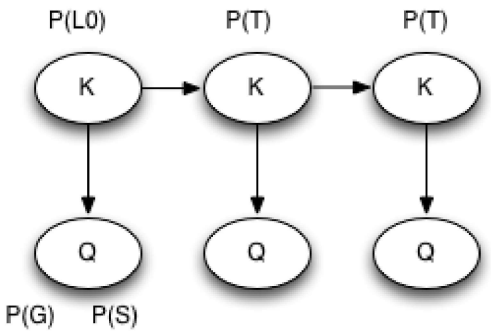

Figure 1: Basic Knowledge Tracing Model

are binary: a student can get a score of 0 or 1 on each problem/step. It updates the distribution over time by observing the correctness of students' responses when applying the relevant skill to a question.

The basic knowledge tracing model is shown in Figure 1. This model is defined by the following four parameters: the probability that a skill L is known prior to answering the first question of this skill, denoted $P(L_0)$; the probability the skill will be learned at each opportunity to use the skill, denoted $P(T)$; the probability the student will guess correctly if the skill is not known, denoted $P(G)$; and the probability of answering the question incorrectly despite knowledge of the skill (slipping), denoted $P(S)$. The variable Q is the observed response to a question by a student. Each of the variables has binary values, true or false. Based on these parameters, inference can be made about the student's knowledge at the n'th opportunity to apply a skill, which is denoted $P(L_n)$. Two basic assumptions of this model are that 1) a student is scored dichotomously (true or false) on each time; 2) a student can be in a known or unknown state for each skill, and that once a skill is known it is not forgotten.

Using the parameters defined above, predicting whether a student's response is correct at time opportunity n is computed using the law of total probability as follows:

$$P(\text{correct}_n) = P(L_n) \cdot (1 - P(S)) + (1 - P(L_n)) \cdot P(G) \quad (1)$$

Similarly, the probability that a response is incorrect at time n is computed below:

$$P(\text{incorrect}_n) = P(L_n) \cdot P(S) + (1 - P(L_n)) \cdot (1 - P(G)) \quad (2)$$

There are two distinct stages to using the knowledge tracing model. The first stage is to fit the model parameters from data. The second stage is using the model to infer the student's knowledge over time given the student's responses. Given an observation of the student's response at time opportunity n (correct or incorrect) the probability $P(L_n)$ that a student knows the skill is calculated using Bayes rule. When a correct response is observed, this probability is as follows:

$$P(L_n \mid \text{correct}_n) = \frac{P(L_n) \cdot (1 - P(S))}{P(\text{correct}_n)} \quad (3)$$

where $P(\text{correct}_n)$ is defined in Equation 1. When an incor-

rect response is observed, this probability is as follows:

$$P(L_n \mid \text{incorrect}_n) = \frac{P(L_n) \cdot P(S)}{P(\text{incorrect}_n)} \tag{4}$$

where $P(\text{incorrect}_n)$ is defined in Equation 2.

Lastly, we show how the student's knowledge of the skill is updated given its interaction with the system. This estimate is the sum of two probabilities: the posterior probability that the student already knew the skill (contingent on the evidence), and the probability that the student did not know the skill, but was able to learn it.

$$\begin{aligned} P(L_n) =& P(L_{n-1} \mid \text{evidence}_{n-1}) + \\ & (1 - P(L_{n-1} \mid \text{evidence}_{n-1})) \cdot P(T) \end{aligned} \tag{5}$$

Many extensions have been proposed to the BKT model over the years. We mention those most relevant to our work, in that they address challenges that arise in the classroom, namely modeling partial credit scores, adapting the model to handling retries and item difficulty, and heuristics for fitting BKT parameters to data. We refer to the survey by Desmarais and Baker [10] for a detailed account of these methods. Wang and Heffernan [20] extended the model to account for partial credit scores, by assigning continuous values to the question node in the model and learning Gaussian distributions over the guess and slip parameters with fixed standard deviations. Ostrow et al. [14] adapt the BKT model to handle partial credit score using a "tabling" method. They maintain a discrete probability distribution table over item correctness that depends on the partial credit scores attributed to the question and its item difficulty.

Pardos and Heffernan [17] introduced the KT-IDEM model which extends the basic BKT model to account for item difficulty. The model fits separate "guess" and "slip" parameters for each item in a skill, and the question node is conditioned on the item node in the network topology. This model has been shown to outperform the standard BKT model on the ASSISTment data set. In subsequent work, Pardos et al. [16] have extended the KT-IDEM model to account for the retries of each question. Using only the first or last response of a student to a question loses important information about how the student learns the skill. They included a "count" node representing the number of tries for each question, and conditioned the "guess" and "slip" parameters on the value of the count nodes. They showed this model, called Count-KT-IDEM, was able to improve performance on a homework data sets from a MOOC course in which multiple attempt behavior for questions was prevalent.

We now turn to the first stage of using the BKT model which is to learn the parameter values of the model from data. The most commonly used approach to fit parameters in the BKT literature uses the Expectation-Maximization (EM) [9] algorithm which is a maximum likelihood approach for training the model in the presence of missing data [15, 17, 16]. Using the EM algorithm for large-scale datasets carries a high computational cost and has been shown to lead to model degeneracy, characterized by extreme parameter values and identifiability issues [8].

An alternative for using EM for large-scale educational datasets includes the Empirical Probabilities approach by Hawkins and Heffernan [11] which is based on a heuristic for estimating at which point the student learned the skill. The skill is assumed to be completely unknown when solving all questions requiring the skill before that point, and completely known when solving all questions beyond that point. A knowledge sequence for the student is the resulting sequence of known and unknown states. This is in accordance with the assumptions of the BKT model, where learned skills are never forgotten. The approach selects the knowledge sequence that best matches the observations consisting of an ordered sequence of the student's responses to questions, called a performance sequence. Let C_i be the student's performance for the question at time i (using value 0 for "incorrect" and value 1 for "correct). Let K be a knowledge sequence and let $K_i \in K$ denote the state for the question at time i (using value 0 for "unknown" and value 1 for "known"). Let K^* be the optimal knowledge sequence that satisfies the following:

$$K^* = \underset{K}{\text{argmin}} \sum_i |C_i - K_i| \tag{6}$$

For example, if the observations relating to a student's performance on a sequence of questions are ("correct, incorrect, incorrect, correct, correct"), then the most likely knowledge sequence that matches up the observation sequence is ("unknown, unknown, unknown, known, known"), since this knowledge sequence matches up with four of the five performances correctly, more than any other possible knowledge sequence does. The BKT parameters are assigned as follows:

$$P(L_0) = \sum_i \frac{K_0}{|K_0|} \tag{7}$$

$$P(T) = \frac{\sigma_{i \neq 0}(1 - K_{i-1}) \cdot K_i}{\sigma_{i \neq 0}(1 - K_{i-1})} \tag{8}$$

$$P(G) = \frac{\sum_i C_i(1 - K_i)}{\sum_i (1 - K_i)} \tag{9}$$

$$P(S) = \frac{\sum_i (1 - C_i) K_i}{\sum_i K_i} \tag{10}$$

Our work also relates to computational models for sequencing educational content to students. Shen et al. [19] used ontologies of competencies to recommend learning paths to students based on competency gap analysis. Huang et al. [13] used Markov chains to find popular sequences over learning objects considering choices of many students and trying to minimize entropy between different learning objects in the sequence. Brusilovsky et al. [4] tracked user navigation and used user direct feedback to recommend paths of additional content to visitors of educational digital libraries. Clement et al. [5] used a Multi-Armed Bandit approach to suggest policies for intelligent tutoring systems which were evaluated in simulation.

Item Response Theory (IRT) [1] is a theory of education measurement that is used to select the most appropriate items for students based on individual ability, as inferred from their past interactions with the system. Unlike knowledge tracing, it does not seek to maintain a model of the student's knowledge. IRT has been used extensively in examination settings such as the SAT, and has recently been applied to on line settings such as Khan Academy.

Among the few works that consider difficulty of questions or other learning objects, Bielikova et al. [2] proposed an Adaptive Learning Framework that recommends sequence of items based on content similarity and knowledge prerequisites. Their field trial shows that using an adaptive system, where the learning object difficulty is an aspect of

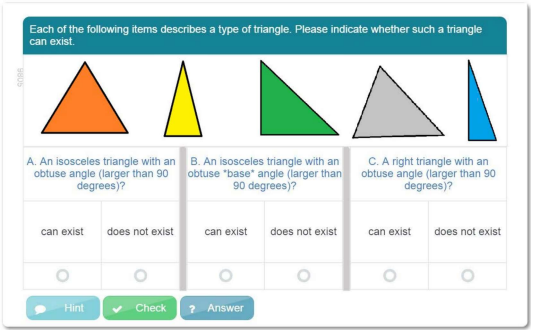

Figure 2: A translation to a question from the K9 system involving identifying triangles.

adaptation, delivers higher learning gain for students comparing to manual recommendations. In their work however the learning object difficulty is not personalized by student basis, as is the case in our work. In another work by Hsieh et al. [12], the authors used fuzzy logic theory to construct an appropriate learning path based on the learners' misconceptions found in a preceding quiz. This process however is not adaptive on an ongoing basis and requires a pre-test as an integral part of the mechanism, unlike our proposed solution.

3. ADAPTING BKT TO THE CLASSROOM

This section describes how we augmented the existing BKT models to capture students' interactions with e-learning systems in the classroom. We begin by describing the K9 e-learning system, which constitutes a formidable part of our empirical analysis. K9 is deployed in over 120 schools in Israel and used by over 10,000 students in grades 1 through 9. It spans questions from diverse subjects, such as English as a foreign language, science education, and mathematics, which was chosen as the focus of our empirical investigation.

Each mathematics question in the K9 database is mapped to one of 256 different topics and sub-topics at increasing levels of specifity (e.g., arithmetic operators, addition, two-digit numbers, etc.), and each question is also labeled by a difficulty level (1 being easiest; 5 being hardest) that was determined by pedagogical experts. Figure 2 shows one of the mathematics questions in the database that requires students to identify different types of triangles. This question was assigned a difficulty level of 3.

We collected about 4 million student responses to over 10,000 unique questions in mathematics spanning four years of use in different schools. Each response is a tuple that includes a question, a unique (anonymized) identifier for the student answering the question, number of retries for this question, and the student's score on the question, which is a positive number in the range of $[0, 1]$.

There are several challenges to using BKT in existing e-learning applications in the classroom. First, the standard BKT model assumes an all-or-nothing score (whether or not the student answered a question correctly). We need to incorporate into the representation that answers may be partially correct. We also need to account for the fact that the question difficulty and the number of retries affect students' skill acquisition for the question. The second challenge is that the computational complexity of the EM algorithm is exponential in the number of variables, and its use in practice is costly in terms of computation time. Both issues has been addressed separately in prior work [11, 20, 16], but as we show, integrating the approaches and using these approaches in the classroom in not straightforward.

In our approach for reasoning about partial credit scores we treat the evidence (the student's score for a question) as a weighting factor that determines the extent to which the student's response was correct. It differs from Wang and Heffernan [20] in that it does not rely on the EM algorithm, and differs from Ostrow et al. [14] in that it does not use predefined methods to determine the posterior distributions. For example, receiving a score of 66% for the question shown in Figure 2 means that 66% of the student's response is correct, while 34% of the student's response is incorrect. Hence we define the posterior probability $P(L_n \mid \text{evidence}_n)$ of learning a skill given the evidence as the sum of posterior probabilities of answering correctly and incorrectly, weighted by the student's score for the question.

$$P(L_n \mid \text{evidence}_n) = \text{score}_n \cdot P(L_n \mid \text{correct}_n) + (1 - \text{score}_n) \cdot P(L_n \mid \text{incorrect}_n) \quad (11)$$

where $P(L_n \mid \text{correct}_n)$ and $P(L_n \mid \text{incorrect}_n)$ are computed in Equations 3 and 4, respectively.

It can be shown that the term on the right-hand-side of this equation yields a number in the range $[0, 1]$. In particular, if the student's score is a perfect score of 0 or 1, then Equation 11 reduces to the one that is used in the classic BKT model. Thus the equation defines a probability distribution over L_n given the evidence at time n.

We addressed the parameter fitting challenge by adapting the Empirical Probabilities method [11] to handle continuous scores. Specifically, for each question i, we replaced the binary valued C_i term in the optimal knowledge sequence definition of Equation 6 with score_i for each question i. (And similarly for computing the guess and slip parameter values of Equations 9 and 10).[1] For example, if a student's performance sequence in K9 is (0.3, 0.8, 1, 0.6, 1) where 1 is the perfect score for that question, the accuracy score for the knowledge sequence "(unknown, known, known, known, known)" and the student's performance sequence will be $(0.3 + 0.2 + 0 + 0.4 + 0) = 0.9$.

4. CANDIDATE MODELS

We incorporated the partial credit score approach in several BKT models from the literature which differ in their representation and number of parameters. We assigned a skill for each topic in the set of 256 topics in the K9 database.

The first model we considered was the basic BKT model shown in Figure 1 (denoted SIMPLE). There were two parameters in this model (for the prior over L_0 and T) and a guess and slip parameter for each skill, totaling 4 parameters per skill.

We also considered several extensions by Pardos et al. [17] to the basic model. One extension assigned guess and slip parameters for different items (denoted KT-IDEM), in order to represent item difficulty and other information that is embedded in the question itself. Although the questions

[1]We ran a pilot study in which we compared several variants of the EM algorithm (Baum-Welch and Gradient Descent) to using the EP heuristic to estimate the parameters for our candidate model.

in the K9 system were labeled by difficulty level, individual students vary in their perceived difficulty of questions. We assigned each item directly to a question, thus the number of parameters for this model was $2 + 2 \cdot \#(\text{num. questions})$ per skill. Another extension assigned separate parameters for different number of question retries (denoted COUNT). We discretized the number of retries into two separate categories: one for all retries representing the first attempt to solve the problem, and one for more than the first attempt to solve the problem.[2] The number of parameters for this model was $2 + 2 \cdot 2$ per skill. We also considered a model that assigned separate parameters for both different number of questions and question retires (denoted KT-IDEM-COUNT), which was also discretized to two values. The number parameters for this model was $2 + 2 \cdot 2 \cdot \#(\text{num. questions})$ per skill.

Lastly, students from different ages and grades may differ in their performance for the same questions, so it makes sense to condition the model on the student's grade level. Therefore we used a new model that assigns separate guess and slip parameters for different school grades (e.g., 4th or 5th grade). The number parameters for this model was $2 + 2 \cdot 2 \cdot \#(\text{num. grades}) \cdot \#(\text{num. questions})$ per skill.

For each of these models, we implemented two methods for dealing with partial credit. The first method (called "Weight"), was the approach using Equation 11 to compute the posterior probability of learning a skill given the evidence, and fitted parameters using a modification of Empirical Probabilities approach [11] that included Equations 9 and 10.

The second method for dealing with partial credit scores (called "Threshold"), reduced the student's score to a binary variable by assigning a correct response if the score was above a given threshold.

$$\text{evidence} = \begin{cases} \text{correct} & \text{if score} \geq \text{threshold}, \\ \text{incorrect} & \text{otherwise} \end{cases} \quad (12)$$

This method used the original EP approach for parameter fitting and for computing the posterior probabilities over the skill $P(L_n)$, as described by Hawkins et al. [11].

5. EMPIRICAL METHODOLOGY

We evaluated each of the partial credit approaches in the models described above on two domains. The first domain was the K9 dataset described in Section 3, while the second domain was the ASSISTment data set used by Ostrow et al. [14].

Table 1 shows the distribution of the questions over four years of use of the K9 system, including the number of different questions, the number of students and the number of responses. Note that only partial information was collected for 2015, hence the smaller number of response instances compared to the other years.[3]

We set the prior parameter values for all models in a way that is similar to those set by other works in the BKT literature. Specifically, the prior probability over skill was set to

[2]The number of categories was determined using a held-out set of data that was not used for testing purposes.

[3]The records in this dataset were anonymized and the study was approved by the institutional review board of Ben-Gurion university.

Year	#(responses)	#(students)	#(questions)
2009	1,865,737	11,601	8,613
2011	1,438,702	14,349	8,367
2014	1,061,161	5,384	5,714
2015	448,222	5,384	5,714
Total	4,813,822	49,294	32,099

Table 1: K9 Database statistics

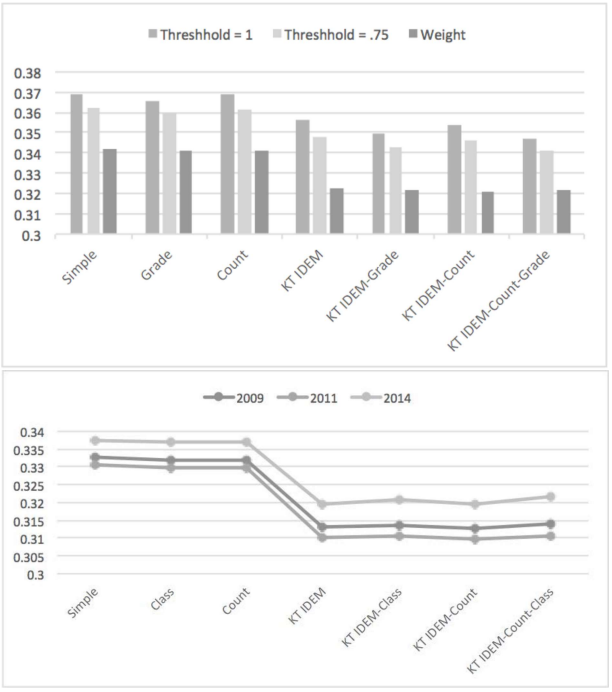

Figure 3: RMSE measures (y-axis) for the different BKT models (top) and breakdown by year (bottom) on K9 dataset

$P(L_0) = 0.5$; the probability of acquiring a skill given feedback was set to $P(T) = 0.3$; the probability of guess and slip was set to $P(G) = 0.2$ and $P(S) = 0.15$ respectively. We used the same prior parameter values for all skills (topics).

We used the root mean square error (RMSE) for evaluating the different models. RMSE is calculated for each item by comparing the predicted correctness ($P(\text{correct}_i)$) of a response to the actual response, within the range $[0, 1]$ for partial credit scores. RMSE has been shown to be the strongest performance indicator for BKT with significantly higher correlation than LL and AUC [18].

We begin with presenting the results from the K9 dataset. We employed a standard five-fold cross-validation technique which randomly split the data set into five folds, each including data from the four years of use of the system. Figure 3 (top) shows the average RMSE value (y-axis) over five rounds in which different folds of the data served for training and testing. For each of the models, we report results using the weight approach for modeling partial scores, as well as the approach using thresholds which were set to 1 and 0.75. All reported results were statistically significant in the $p < 0.05$ range using t-test measures.

As shown in the figure, for all BKT models, reasoning about partial credit scores with the Weighted method achieved

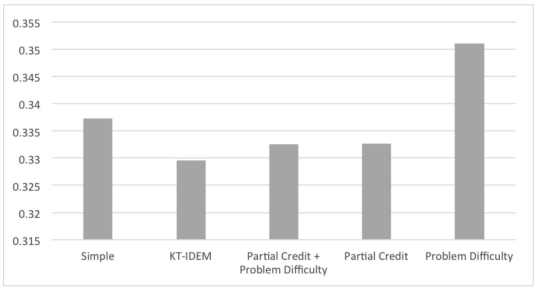

Figure 4: RMSE measures (y-axis) for the different BKT models on ASSISTMENT dataset

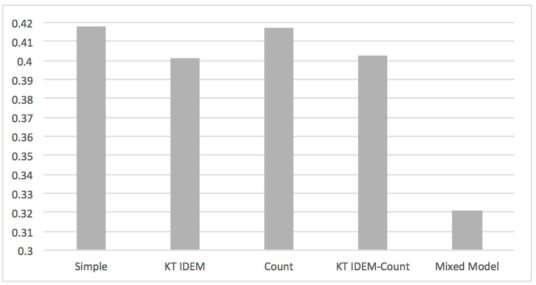

Figure 5: Comparison of MixedModel to other approaches on subset of K9 dataset from 2015

better performance (lower RMSE scores) than both of the threshold approaches. Additionally, the best performance was obtained by the KT-IDEM-Count model. Figure 3 (down) breaks down the performance of the weighted partial credit approach for the different models for each year. As shown in the figure, all methods exhibited consistent performance when testing on different years.

The second domain was the ASSISTment dataset used by Ostrow et al. [14], which was compiled from problem logs from the ASSISTments platform during the 2012-2013 school year.[4] Here, we used two of the BKT models (BASIC and KT-IDEM) using the partial credit score approach described above.[5] We compared the performance of these techniques to the ones suggested by Ostrow et al., which addressed the partial credit score issue by using predefined probability tables that depended on problem difficulty, partial credit scores, or both. All of the BKT models we used employed the Weighted partial credit score approach that was described in the previous section.

Figure 4 compares the RMSE scores (y-axis) for the Simple and KT-IDEM approaches (left side of the figure) to the Partial Credit, problem difficulty, and combined approaches of Ostrow et al. (right side of the figure). As shown in the figure, the KT-IDEM model obtained lower RMSE score than did all other approaches.

6. MODEL SELECTION

In this section we address the task of choosing which of the BKT models to use when interacting with new students in the classroom, who were not used to train the model. A natural candidate is to use the model with the best average performance, as reported in Section 3. However, a single model cannot be optimal for predicting student's performance on all questions (as suggested by Pardos and Heffernen [17]). For example, we observed that for specific instances, the simple BKT model was able to outperform more complex models like the KT-IDEM model. Following this insight, we employed a model selection approach, that chose which BKT model to apply for a given response based on the performance of a set of candidate models on the training set of past questions.

The model selection process receives the following as input: a set of possible BKT models, a new student's response to a test-question, and number of retries for this student and question. It selects from the training set the set of all questions matching the test question and number of retries, and returns the model with the lowest average RMSE score on the training set, which is used for prediction on the test-question. The performance on the training set is used as a proxy for the performance of the chosen BKT model on the actual test question.

We evaluated this MixedModel approach on both K9 and ASSISTment dataset. Figure 5 compares the performance of the MixedModel on the K9 dataset in terms of RMSE (y-axis). The set of candidate models included the subset of models equals to the top performing RMSE models of Section 3: the SIMPLE model, the COUNT model, the KT-IDEM model, and the KT-IDEM-Count model. To choose the best model to use in the test-set environment, we computed the RMSE values for each of the candidate models on the training set. In K9, we used the subset of the dataset from years 2009,2011 and 2014 as training, and tested on data from 2015, to simulate situations in which new students will be using the system, with little to no prior history of their interactions. As can be seen by the figure, the MixedModel approach achieved significantly lower RMSE scores than did the other models. The higher RMSE values of the other models as compared to other years can be explained by the lower number of records compared to other years.

We also compared the model selection approach in the ASSISTment data set (not shown in the figure) with the SIMPLE and KT-IDEM BKT models. We used a standard five-cross validation technique to train and test the MixedModel approach. The results were similar, in that the MixedModel approach was able to achieve the best performance on new data and significantly outperform the KT-IDEM method. Consequently we decided to adopt the MixedModel approach when deploying our sequencing algorithm in schools.

7. THE BKT-SEQUENCE ALGORITHM

In this section, we describe an algorithm that utilizes BKT for the purpose of selecting questions in realtime in the classroom, called BKT-Sequence. The algorithm receives as input a BKT model and a set of questions. Each question is assumed to have an associated skill. The output of the algorithm is the next question to present to the student.

The BKT-Sequence algorithm is shown in Figure 6. The algorithm first determines the minimal and maximal predicted scores for each question in Q (lines 2-3). Line 5 computes the student's "intended" score for each question. The

[4]This dataset is publicly available at http://tiny.cc/LaS2015Submission.

[5]We did not use the grade model because the student's grade was not available in the data, and we did not use the count model because it was superceded by the KT-IDEM model.

1: **function** ChooseNextQuestion(Q)
2: MinScore = $\min\{P(\text{score}_{n'}) \mid n' \in Q\}$
3: MaxScore = $\max\{P(\text{score}_{n'}) \mid n' \in Q\}$
4: For all questions $n \in Q$ compute
5: $\text{WantedScore}_n = \text{MinScore} + (\text{MaxScore} - \text{MinScore}) \cdot (1 - P(L_n) + \epsilon_n)$
6: $\text{Diff}_n = |P(\text{score}_n) - \text{WantedScore}_n| \cdot \text{Penalty}_n$
7: $Q' =$ all questions of skill $\text{argmin}_{m' \in Q} P(L_{m'})$
8: Return question j such that $j = \text{argmin}_{j \in Q'} \text{Diff}_j$

Figure 6: BKT-Sequence Algorithm

interval maxScore − minScore determines the range of possible scores for the student for question n. It then computes an intended score that the student should get for each question in order to advance its learning, with a value that is between these minimal and maximal scores. This score depends on the current mastery level of the student in that skill. Low levels of knowledge of the skill (as determined by $P(L_n)$) should lead to higher intended scores, The intended score is the point within the range between MaxScore and MinScore that best matches the student's mastery level of the skill for the question. This point is determined by the lack of knowledge that the student has over the skill, which is $(1 - P(L_n) + \epsilon_n)$, where ϵ_n is a correction term for n that is explained below. As the knowledge of a particular skill grows, the intended score for a question of this skill will be lower (closer to MinScore than to MaxScore).

Lastly, the the algorithm returns the question in Q with a predicted score that is closest to that of the intended score, i.e., closest to the score of the question that is the best match for the student. Specifically, Line 6 computes the absolute difference between these two scores and in line 8 the algorithm returns the question that minimizes this difference. As the knowledge of a particular skill grows, more difficult questions will be selected because they are associated with a lower predictive score than easier questions.

We note the following implementation details. First, our discussions with pedagogical experts who were concerned that transitioning rapidly to more difficult questions may reduce students' motivation. We capped the increase or decrease in value to the posterior over the skill level at 0.1, which was based on showing examples to the pedagogical experts.

$$P(L_n) = \begin{cases} \min & \{P(L_n \mid \text{evidence}_{n-1}), P(L_n) + 0.1\} \\ & \text{if } P(L_n \mid \text{evidence}_{n-1}) > P(L_n) \\ \max & \{P(L_n \mid \text{evidence}_{n-1}), P(L_n) - 0.1\} \\ & \text{otherwise} \end{cases} \quad (13)$$

Second, we employed a correction term ϵ_n for identifying questions with degenerate parameter values in which answering incorrectly does not change the posterior ($P(L_n \mid \text{incorrect}_n) = P(L_{n-1})$).

Third, we introduced a penalty score for each question n that is equal to the number of retries for the question. In this way, we avoid selecting questions that were attempted many times in the past.

To illustrate the use of this algorithm, consider a set of questions selected by a teacher or student and belonging to the skill of matching triangles. Suppose that the $P(L_n)$ for this student is 0.6952. The minimum predicted grade for the student (MinScore) is 0.2317 and the maximum predicted grade (MaxScore) is 0.8592. Table 2 shows a few of the candidate questions in the set of questions that provide the lowest intended scores for the student, showing for each question: the difficulty level, the predicted score, the number of retries (attempts) by the student, the corrective term ϵ_n, the parameter values $P(G)$ and $P(S)$, and the difference between the predicted score and the intended score. In this case, all of the questions belong to the same topic, so the algorithm will choose the question with the lowest difference between the predicted and intended score (itemID 9805), which is associated with a difficulty level of 3. Note that there are other questions in the set with the same difficulty level, which were not chosen. For example, the last question (ItemID 11625) in the list was not chosen because of a high value for the $P(S)$ (slip) parameter, which resulted in a lower predicted score, thus increasing the gap between the intended score and the predicted score for this question. The other questions (ItemIDs 11575 and 11625) were not chosen because of their high number of retries, which penalized their score.

Item ID	Difficulty Level	Predicted Score	Num Attempts	ϵ_n	$P(G)$	$P(S)$	Diff
9805	3	0.696	1	0.417	0.462	0.189	0.003
11575	3	0.659	3	0.347	0.287	0.158	0.010
965	1	0.708	4	0.411	0.478	0.178	0.051
11625	3	0.474	1	0.575	0.232	0.406	0.323

Table 2: Example of candidate questions for BKT-Sequence algorithm

8. DEPLOYMENT IN CLASSROOMS

In cooperation with educational researchers and the developers of the K9 system, we were able to deploy the sequence algorithm in two different schools in Israel. The default use of the system in all schools is that students select a topic and a level of difficulty, and the system selects random questions within these categories. The number of consecutive retries allowed for each question was limited to three (the same question could appear in another session). The students could also choose not to answer a question. Students received feedback from the system about their score for each question. An analysis of the data of students' interactions shows that students choose to solve (mostly) easy problems and that they exhibit overall high grades.

We hypothesized that using the BKT-Sequence algorithm to generate educational content would get students to attempt to solve more difficult problems, without harming their performance on these problems or their satisfaction from using the e-learning system, when compared to an alternative sequencing approach.

Our experiment was conducted in two separate schools that did not use the K9 system prior to the experiment. The system was introduced in both schools at the same time, and the experiment was conducted during the last two months of the school year, between April 28th, 2015 and June 29th, 2015. Students interacted with K9 in sessions of 15 questions. We compared between two approaches to sequence questions to students.

The first approach used the BKT sequence algorithm of Figure 6. The second approach was determined by pedagogical experts, and included a set of questions randomly

School	Group size (BKT, ASC)	Num. questions (BKT, ASC)	Time in system (BKT, ASC)	Num. of logins (BKT, ASC)
A	(12, 9)	(921, 1054)	(60, 31)	(8.75, 10)
B	(26, 26)	(4253, 3534)	(74, 42)	(22, 14)

Table 3: Statistics for both BKT and ASC sequencing approaches

sampled from different level of difficulties as follows: 20% questions of level 1 difficulty; 30% questions of level 2 difficulty; 40% questions of level 3 difficulty; 10% questions of level 4 difficulty. (The pedagogical experts did not want to include questions of the most difficult level.) The questions were sequenced to students in ascending order of difficulty.

In each school, we chose two classrooms, one of which was randomly assigned to use the BKT-Sequence algorithm (denoted BKT), while the other classroom was assigned to the ascending (ASC) algorithm. To gauge the level of students in each classrooms we administered a single session of 15 questions in mathematics that were sampled from different topics and difficulty levels in mathematics. The questions for this preliminary test were chosen by a domain expert. There was no statistically significant difference between the two groups in each school in the average score on this preliminary test. Hence we asserted that the students in each group exhibited similar knowledge baselines of the material.

Each classroom in K9 used its respective condition. We restricted ourselves to sessions from students who completed the preliminary session. We did not track the individual students' progress during the experiment nor control any of the conditions in the classroom beyond the use of the sequence algorithm. Students could use the system with no supervision and practice as many questions as they want. Table 3 shows the distribution of the number of students in each group, the total number of questions solved, the average time spent on each question (in seconds), and the number of logins to the system.

Figure 7 shows the performance of students using the BKT-Sequence algorithm and the ASC algorithm in both schools. The x-axis shows the difficulty levels of the different questions from easy (1) to hard (5), while the y-axis shows the average grade obtained by students. As shown in the figure, for schools A and B, for low levels of difficulty (levels 1 and 2) students using the ASC sequencing outperformed students using the BKT-Sequence algorithm. For medium level of difficulty (level 3), there was no difference between students' performance using both algorithms. For higher level of difficulty (level 4), students using the BKT-Sequence algorithm achieved higher performance than those using the ASC algorithm.

Table 4 shows the number of questions posed by each algorithm for each level of difficulty by each school. As shown in the table, there were significantly more difficult questions posed by the BKT sequencing algorithm than the ASC algorithm.

A natural question that arises is whether students were less motivated to work with the BKT-Sequence algorithm because it gave them harder questions. As shown by Table 3, the average time spent on solving questions in the system was significantly higher (for all schools) for students using the BKT-Sequence algorithm than the ASC algorithm. In addition, we were able to conduct a survey in one of the schools (School B) that participated in the experiment. We

Level	ASC A	BKT A	ASC B	BKT B
1	358	131	1104	667
2	324	191	1176	1089
3	291	297	936	1216
4	81	302	318	1281
5	0	218	0	525

Table 4: Number of questions by difficulty level in the BKT-Sequence and Ascending Algorithm Conditions in school A and school B.

asked the students in each sequencing condition the following question: "How much did you feel the question helps you to understand the relevant topic?" Students could choose to answer that (1) the question did not help at all, (2) helped a little, or (3) helped a lot. The average score for the BKT algorithm was 2.2, while the score for the ASC algorithm was 1.53, which was also verified to be significantly lower ($p < 0.05$). We can thus conclude that reasoning about student's skill knowledge in the BKT-Sequence algorithm resulted in students receiving questions that are more suitable for them, as also determined by their subjective opinions.

9. DISCUSSION AND CONCLUSION

In this work, we used a computational knowledge tracing approach to augment existing e-learning systems to sequence educational content to students in the classroom. Our approach followed three main steps. First, we extended the Bayesian Knowledge Tracing approaches to account for the challenges of modeling classroom data. We showed that an augmented BKT model that reasoned about partial credit scores, item difficulty and multiple retries was able to outperform existing models from the literature in predicting student's scores.

Second, we provided a model selection approach that selects the best model to use out of a given set of candidate BKT models and a student's response. We showed the efficacy of this approach when predicting new student's scores.

Third, we designed an algorithm that used the model selection approach to choose questions to students by reasoning about their inferred skill knowledge. The algorithm follows the mastery learning paradigm, by identifying questions that are predicted to advance student's knowledge within skills needing improvement. It provides the student with the question that is predicted to advance her knowledge while keeping her in the zone of possible learning [21] by suggesting questions that are within the range of the student's inferred capabilities.

Our algorithm was subsequently deployed in an e-learning system for mathematics and evaluated in two different schools with new students. It was compared to a baseline sequencing approach which sampled questions of varying difficulty levels according to a domain expert. It was shown to lead students to solve more difficult questions, spend more time in the system, and report higher levels of satisfaction from the systems than students who used the alternative method. Our results demonstrate the benefits of a general technique for augmenting existing e-learning systems in a way that improves students' performance.

We address several issues arising from our study. First, we chose to evaluate our algorithms in real classrooms over the course of several weeks. Students varied widely in their

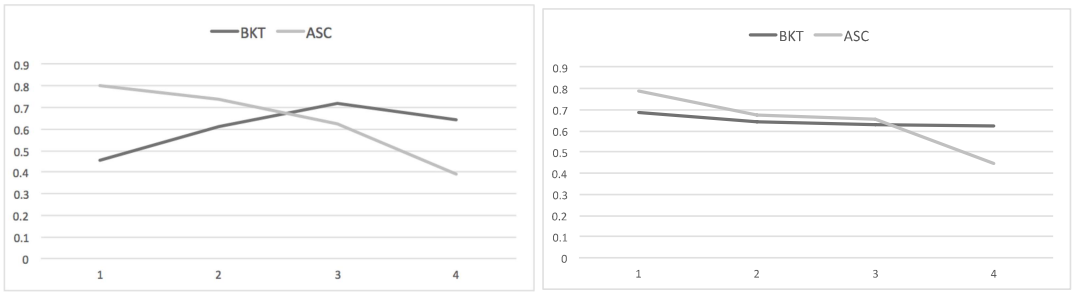

Figure 7: Performance comparison between BKT-Sequence and ASC algorithm for each skill level in school A (left) and school B (right). The x-axis shows the difficulty levels, while the y-axis shows the average grade.

use of the system in the different classrooms, including time of use (whether at class or at home), and whether they were incentivized to use the system by their teachers. Because we did not use controlled laboratory conditions, we chose to forego the use of pre and post tests to measure the effects of our algorithms on students' performance and learning gains. However, the fact that the students in both algorithm conditions exhibited similar performance on a standardized set of questions before commencing the study confirms that it was conducted on equal grounds.

Second, we note that in all schools, students using the ascending algorithm performed better on easier questions (levels 1 and 2) while students using the BKT-Sequencing algorithm performed better on harder questions (levels 3 and 4). This can be explained by the fact that there were significantly more easier questions proposed by the ascending algorithm than the BKT-Sequence algorithm. However, we were able to show that it is possible to get students to solve more difficult questions - a primary goal of e-learning systems in school. Furthermore students expressed more satisfaction from using this system, despite having to solve more difficult questions.

We are currently extending the work in several ways. First, we will compare the algorithm to other ASC manual strategies and to a more elaborate baseline approach that considers students' performance when deciding on the next question to ask. Second, we are designing new BKT models that consider gamification elements which are becoming more prevalent in student's work.

10. ACKNOWLEDGMENTS

The authors wish to thank the Lnet software company (https://lnet.org.il/) for providing the infrastructure for data collection and sequencing. Special thanks is due to Iris Tabak for her very helpful advice on experiment design and previous drafts. Thanks also to Guy Shani on helpful comments on previous drafts of this paper.

11. REFERENCES

[1] F. B. Baker and S.-H. Kim. *Item response theory: Parameter estimation techniques*. CRC Press, 2004.

[2] M. Bieliková, M. Šimko, M. Barla, J. Tvarožek, M. Labaj, R. Móro, I. Srba, and J. Ševcech. Alef: from application to platform for adaptive collaborative learning. In *Recommender Systems for Technology Enhanced Learning*, pages 195–225. Springer, 2014.

[3] J. H. Block, P. W. Airasian, B. S. Bloom, and J. B. Carroll. *Mastery learning: Theory and practice*. Holt, Rinehart and Winston New York, 1971.

[4] P. Brusilovsky, L. N. Cassel, L. M. Delcambre, E. A. Fox, R. Furuta, D. D. Garcia, F. M. Shipman, and M. Yudelson. Social navigation for educational digital libraries. *Procedia Computer Science*, 1(2):2889–2897, 2010.

[5] B. Clement, P.-Y. Oudeyer, D. Roy, and M. Lopes. Online optimization of teaching sequences with multi-armed bandits. In *International Conference on Educational Data Mining (EDM)*, 2014.

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

[7] A. T. Corbett and A. Bhatnagar. Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model With Declarative Knowledge. In *User Modeling*, pages 243–254. Springer Vienna, Vienna, 1997.

[8] R. S. d Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415, 2008.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[10] M. C. Desmarais and R. S. J. de Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[11] W. J. Hawkins, N. T. Heffernan, and R. S. J. de Baker. Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. *Intelligent Tutoring Systems*, 8474(Chapter 18):150–155, 2014.

[12] T.-C. Hsieh, M.-C. Lee, C.-Y. Su, et al. Designing and implementing a personalized remedial learning system for enhancing the programming learning. *Educational Technology & Society*, 16(4):32–46, 2013.

[13] Y.-M. Huang, T.-C. Huang, K.-T. Wang, and W.-Y. Hwang. A markov-based recommendation model for exploring the transfer of learning on the web. *Educational Technology & Society*, 12(2):144, 2009.

[14] K. Ostrow, C. Donnelly, S. Adjei, and N. Heffernan. Improving student modeling through partial credit and problem difficulty. In *Proceedings of the 2nd ACM Conf on L@ S*, pages 11–20, 2015.

[15] Z. Pardos and N. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*, volume 6075, pages 255–266. 2010.

[16] Z. A. Pardos, Y. Bergner, D. T. Seaton, and D. E. Pritchard. Adapting Bayesian Knowledge Tracing to a Massive Open Online Course in edX. *EDM*, pages 137–144, 2013.

[17] Z. A. Pardos and N. T. Heffernan. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In *UMAP*, 2011.

[18] R. Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 2015.

[19] L.-p. Shen and R.-m. Shen. Learning content recommendation service based-on simple sequencing specification. In *Advances in Web-Based Learning–ICWL 2004*, pages 363–370. Springer, 2004.

[20] Y. Wang and N. Heffernan. Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *Artificial Intelligence in Education*, pages 181–188. Springer, 2013.

[21] J. V. Wertsch. *Vygotsky and the social formation of mind*. Harvard University Press, 1988.