

# KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model

Zachary A. Pardos and Neil T. Heffernan

Department of Computer Science, Worcester Polytechnic Institute,  
100 Institute Road, Worcester, MA 01609 USA  
{Zpardos,nth}@wpi.edu

**Abstract.** Many models in computer education and assessment take into account difficulty. However, despite the positive results of models that take difficulty into account, knowledge tracing is still used in its basic form due to its skill level diagnostic abilities that are very useful to teachers. This leads to the research question we address in this work: Can KT be effectively extended to capture item difficulty and improve prediction accuracy? There have been a variety of extensions to KT in recent years. One such extension was Baker's contextual guess and slip model. While this model has shown positive gains over KT in internal validation testing, it has not performed well relative to KT on unseen in-tutor data or post-test data, however, it has proven a valuable model to use alongside other models. The contextual guess and slip model increases the complexity of KT by adding regression steps and feature generation. The added complexity of feature generation across datasets may have hindered the performance of this model. Therefore, one of the aims of our work here is to make the most minimal of modifications to the KT model in order to add item difficulty and keep the modification limited to changing the topology of the model. We analyze datasets from two intelligent tutoring systems with KT and a model we have called KT-IDEM (Item Difficulty Effect Model) and show that substantial performance gains can be achieved with this minor modification that incorporates item difficulty.

**Keywords:** Knowledge Tracing, Bayesian Networks, Item Difficulty, User Modeling, Data Mining.

## 1 Introduction

Many models in computer education and assessment take into account difficulty. Item Response Theory (IRT) [1] is one such popular model. IRT is used in Computer Adaptive Testing (CAT) and learns a difficulty parameter per item. This makes IRT models very powerful for predicting student performance; however the model learning processes is expensive and is not a practical way of determining when a student has learned a particular skill because it does not model learning. Despite the predictive power of IRT, the Cognitive Tutors [2] employ standard Knowledge Tracing (KT) [3] to model students' knowledge and determine when a skill has been learned. Knowledge Tracing is used because it is a cognitively diagnostic form of

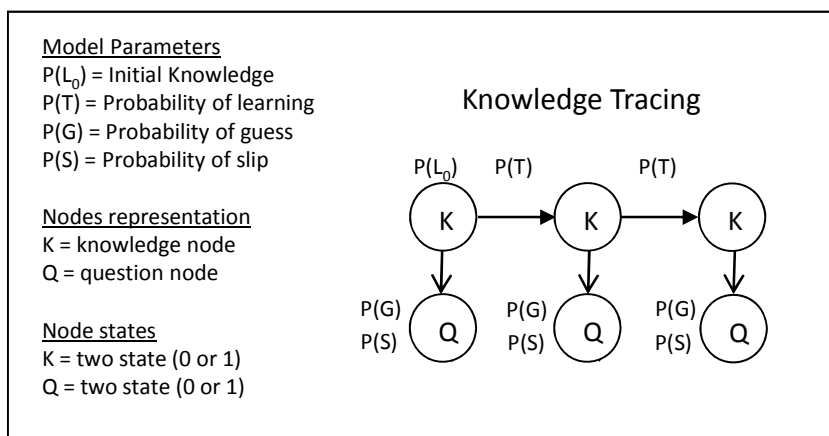
assessment which is beneficial to both student and teacher. The parameters for a KT model need only be learned once, typically at the beginning of the school year (based on the past year's data) and the inference of individual student's knowledge of a skill can be executed with very little computation. Models like IRT that take into account item difficulty are strong at prediction, and model such as KT that infer skills are useful for their cognitively diagnostic results. This leads us to our research question: Can KT be effectively extended to capture item difficulty and improve predictive?

There have been a variety of extensions to KT in recent years. One such extension was Baker's contextual guess and slip model [4]. While this model has shown positive gains over KT in internal validation testing, it has not performed well relative to KT on unseen in-tutor data or post-test data; however, it has proven a valuable model to use alongside other models. Likewise, the contextual slip model [5] also suffered the same inadequacies on in-tutor data prediction. The contextual guess and slip model increased the complexity of KT by adding regression steps and feature generation. The added complexity of feature generation across datasets may have hindered the performance of this model. Therefore, one of the aims of our work in this paper was to make the most minimal of modifications to the KT model in order to add item difficulty and keep the modification limited to slight changes to the topology of the model.

## 1.1 Knowledge Tracing

The standard Bayesian Knowledge Tracing (KT) model has a set of four parameters which are typically learned from data for each skill in the tutor. These parameters dictate the model's inferred probability that a student knows a skill given that student's chronological sequence of incorrect and correct responses to questions of that skill thus far. The two parameters that determine a student's performance on a question given their current inferred knowledge are the guess and slip parameters and these parameters are where we will explore adding question level difficulty. The guess parameter is the probability that a student will answer correctly even if she does not know the skill while the slip parameter is the probability that the student will answer incorrectly when she knows the skill. Skills that have a high guess rate can be thought of, intuitively, as easy (a multiple choice question for example). Likewise, skills that have a low guess and/or a higher rate of mistakes (high slip) can be thought of as hard. Based on this intuition we believe a questions' difficulty can be captured by the guess and slip parameter. Therefore, we aim to give each question its own guess and slip thereby modeling a difficulty per item.

Figure 1 depicts the standard KT model. The three latent nodes representing knowledge are above the three observable nodes representing questions in the tutor. The depiction is showing an unrolled dynamic Bayesian topology for modeling a sequence of three questions but this chain can continue for an arbitrary number of questions a student answers. The guess and slip parameters are represented by  $P(G)$  and  $P(S)$  respectively. The two knowledge parameters, which dictate the state of the knowledge node, are the probability of learning,  $P(T)$ , and probability of initial knowledge,  $P(L_o)$ , also referred to as prior probability of knowledge or just *prior*.  $P(L_o)$  is the probability that a student knows the skill before answering the first question and  $P(T)$  is the probability that a student will transition from not knowing the skill to knowing it.



**Fig. 1.** The standard Knowledge Tracing model

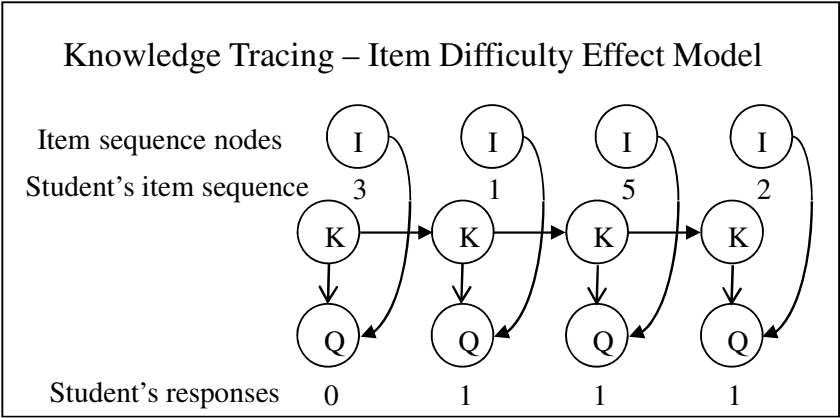
While knowledge is modeled as a binary variable (a student is either in the learned or unlearned state), the inferred probability of knowledge is a continuous value. Once that probability reaches 0.95, the student can be assumed to have learned the skill. The Cognitive Tutors use this threshold to determine when a student should no longer be asked to answer questions of a particular skill.

## 2 Knowledge Tracing: Item Difficulty Effect Model (KT-IDEM)

One of our stated goals was to **add difficulty to the classical KT model without going outside of the Bayesian topology**. To do this we used a similar topology design to that which was demonstrated in Pardos & Heffernan's student individualization paper [6]. In that work **a multinomial node was added to the Bayesian model that represented the student**. The node(s) containing the parameters which the authors wished to individualize were then conditioned based on the student node, thus creating a parameter per student. For example, if one wished to individualize the prior parameter, the student node would be connected to the first knowledge node since this is where the prior parameter's CPT is held. A separate prior could then be set and learned for each student. Practically, **without the aid of a pre-test, learning a prior for every student is a very difficult fitting problem**, however, simplifying the model to represent only two priors and assigning students to one of those priors based on their first response has proven an effective heuristic for improving prediction by individualizing the prior.

In a similar way that Pardos & Heffernan showed how parameters could be individualized by student, **we individualized the guess and slip parameter by item**. This involved creating a multinomial item node, instead of a student node, that represents all the items of the particular skill being fit. This means that if there were 10 distinct items in the skill data, the item node would have values ranging from 1 to 10. These values are simply identifiers for the items which can arbitrarily be assigned. The item node is then connected to the question node (Fig 2) in the topology, thus

conditioning the question's guess/slip upon the value of the item node. In the example of the 10 item dataset, the model would have 10 guess parameters, 10 slip parameters, a learn rate and a prior, totaling 22 parameters versus standard KT's 4 parameters. It is possible that this model will be over parameterized if a sufficient amount of data points per item is not met; however, there has been a trend of evidence that suggests models that have equal or even more parameters than data points can still be effective such as was shown in the Netflix challenge [11] and 2010 KDD Cup on Educational Data Mining [12].



**Fig. 2.** The KT-IDEM topology depicting how the question node (and thus the guess/slip) is conditioned on the item node to add item difficulty to the KT model

Figure 2 illustrates how the KT model has been altered to introduce item difficulty by adding an extra node and an arc for each question. While the standard KT model has a single  $P(G)$  and  $P(S)$ , KT-IDEM has a  $P(G)$  and  $P(S)$  for each item, for example  $P(G|I=1)$ ,  $P(G|I=2)$ ...  $P(G|I=10)$ , stating that there is a different guess parameter value given the value of the item node. In the example in Figure 2, the student sees items with IDs 3, 1, 5 and then 2. This information is fully observable and is used in model training, to fit appropriate parameters to the item  $P(G|I)$  and  $P(S|I)$ , and in model tracing (prediction), to inform which items a particular student has encountered and make the appropriate inference of knowledge based on the answer to the item. By setting a student's item sequence to all 1s during training and tracing, the KT-IDEM model represents the standard KT model, therefore the KT-IDEM model, which we have introduced in this paper, can be thought of as a more general KT model. This model can also be derived by modifying models created by the authors for detecting the learning value of individual items [7].

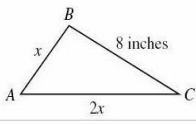
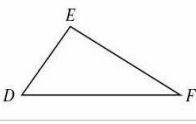
### 3 Datasets

We evaluate the KT and KT-IDEM models with two datasets from two separate real world tutors. The datasets will show how the models perform across a diverse set of different tutoring scenarios. The key factor of KT-IDEM is modeling a separate guess

and slip parameter for every item in the problem set. In these two datasets, the representation of an item differs. In the ASSISTments dataset, a problem template is treated as an item. In the Cognitive Tutor dataset, a problem, which is a collection of steps, is treated as an item. The sections below provide further descriptions of these systems and the data that were used.

### 3.1 The ASSISTments Platform

Triangles ABC and DEF are congruent.  
The perimeter of triangle ABC is 23 inches.  
What is the length of side DF in triangle DEF?

Comment on Problem #4468

Request Help

Type your answer below (mathematical expression):

5

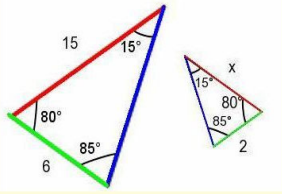
Submit Answer

✗ Sorry, that is incorrect. Let's move on and figure out why!

Comment on Problem #4464

Which side of triangle ABC has the same length as side DF of triangle DEF?

Let's make sure you understand what corresponding sides are. In this picture the corresponding sides are marked. Does this help you?



Comment on Hint #22979

Request Help

Select one:

☒ AB

☐ BC

☐ AC

Submit Answer

Side AB corresponds to side DE of triangle DEF, not DF. Try again, please.

Our first dataset consisted of student responses from ASSISTments [8], a web based math tutoring platform which is best known for its 4th-12th grade math content. Figure 3 shows an example of a math item on the system and tutorial help that is given if the student answers the question wrong or asks for help. The tutorial help assists the student in learning the required knowledge by breaking each problem into sub questions called scaffolding or giving the student hints on how to solve the question. A question is only marked as correct if the student answers it correctly on the first attempt without requesting help.

#### Item templates in ASSISTments

Our skill building dataset consists of responses to multiple questions generated from an item template. A template is a skeleton of a problem created by a content developer in the web based builder application. For example, a template could

**Fig. 3.** An example of an ASSISTments item where the student answers incorrectly and is given tutorial help

specify a Pythagorean Theorem problem, but without the numbers for the problem filled in. In this example the problem template could be: “What is the hypotenuse of a right triangle with sides of length X and Y?” where X and Y are variables that will be filled in with values when questions are generated from the template. The solution is also dynamically determined from a solution template specified by the content developer. In this example the solution template would be, “Solution =  $\sqrt{X^2+Y^2}$ ”. Ranges of values for the variables can be specified and more advance template features are available to the developer such as dynamic graphs, tables and even randomly selected cover stories for word problems. Templates are also used to construct the tutorial help of the template items. Items generated from these templates are used extensively in the skill building problem sets as a pragmatic way to provide a high volume of items for students to practice particular skills on.

### Skill Building Datasets

Skill building is a type of problem set in ASSISTments that consists of hundreds of items generated from a number of different templates, all pertaining to the same skill

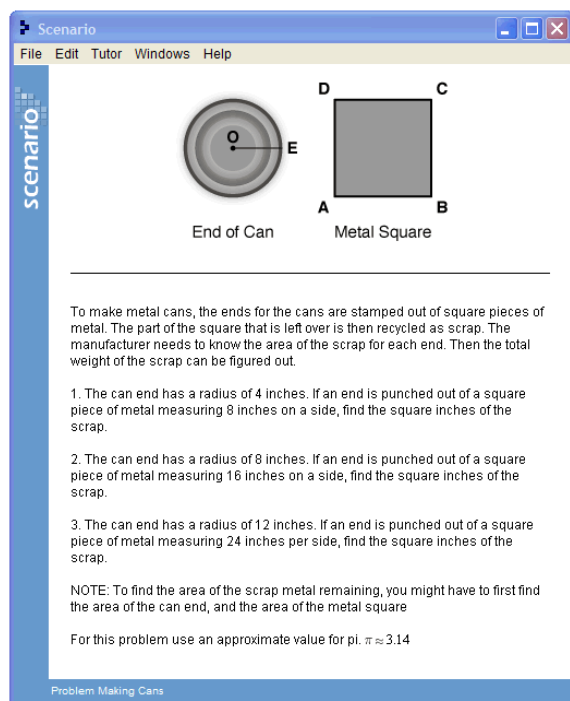


Fig. 4. A Geometry problem within the Cognitive Tutor

or skill grouping. Students are marked as having completed the problem set when they answer three items correctly in a row without asking for help. In these problem sets items are selected in a random order. When a student has answered 10 items in a skill building problem set without getting three correct in a row, the system forces the student to wait until the next calendar day to continue with the problem set. The skill building problem sets are similar in nature to mastery learning [9] in the Cognitive Tutors; however, in the Cognitive Tutors, mastery is achieved when a knowledge-tracing model believes that the student knows the skill with 0.95 or better probability. Much like the other problem sets in ASSISTments, skill builder problem sets are assigned by the teacher at his or her discretion and the problem sets they assign often conform to the particular math curriculum their district is following.

We selected the 10 skill builder datasets with the most data from school year 2009-2010, for this paper. The number of students for each problem set ranged from

637 to 1285. The number of templates ranged from 2-6. This meant that there would be at max 6 distinct sets of guess/slips associated with items in a problem set. Because of the 10 item/day question limit, we only considered a student's first 10 responses per problem set and discarded the remaining responses. Only responses to original questions were considered. No scaffold responses were used.

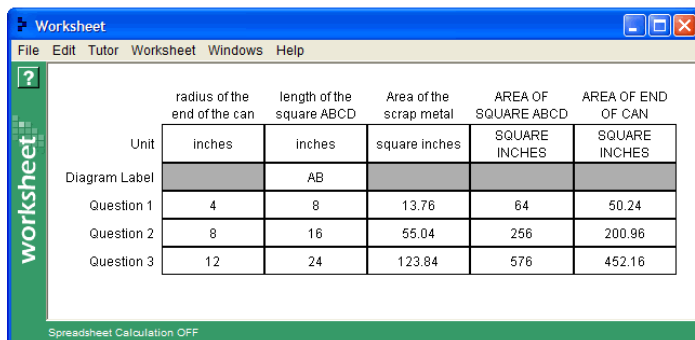
### 3.2 The Cognitive Tutor: Mastery Learning Datasets

Our Cognitive Tutor dataset comes from the 2006-2007 "Bridge to Algebra" system. This data was provided as a development dataset in the 2010 KDD Cup competition [10]. The Cognitive Tutor is designed differently than ASSISTments. One very relevant difference to this work is that the Cognitive Tutor presents a problem to a student (Fig 4) that can consist of questions (also called steps) of many skills. Students may enter their answers to the various questions pertaining to the problem in an answer grid (Fig 5). The Cognitive Tutor uses Knowledge Tracing to determine when a student has mastered a skill. A problem in the tutor can also consist of questions of differing skill. However, once a student has mastered a skill, as determined by KT, the student no longer needs to answer questions of that skill within a problem but must answer the other questions which are associated with the unmastered skill(s).

The number of skills in this dataset was substantially larger than the ASSISTments dataset. Instead of processing all skills, a random sample of 12 skills were selected. Some questions

consisted of multiple skills. Instead of separating out each

skill, a set of skills associated with a question was treated as a separate skill. The Cognitive Tutor separates lessons into pieces called Units. A skill name that appears in one Unit was treated as a separate skill when appearing in a different Unit. Some skills in the Cognitive Tutor consist of trivial tasks such as "close-window" or "press-enter". These types of non-math related skill were ignored. To maintain consistency with the per student data amount used in the ASSISTments dataset, the max number of responses per student per skill was also limited to the first 10.



Unit	radius of the end of the can inches	length of the square ABCD inches	Area of the scrap metal square inches	AREA OF SQUARE ABCD SQUARE INCHES	AREA OF END OF CAN SQUARE INCHES
Diagram Label		AB			
Question 1	4	8	13.76	64	50.24
Question 2	8	16	55.04	256	200.96
Question 3	12	24	123.84	576	452.16

**Fig. 5.** Answer entry box for the Geometry problem in Fig 4

## 4 Methodology

A five-fold cross-validation was used to make predictions on the datasets. This involved randomly splitting each dataset into five bins at the student level. There were



five rounds of training and testing where at each round a different bin served as the test set, and the data from the remaining four bins served as the training set. The cross-validation approach has more reliable statistical properties than simply separating the data in to a single training and testing set and should provide added confidence in the results since it is unlikely that the findings are a result of a “lucky” testing and training split.

#### 4.1 Training the Models

Both KT and KT-IDEM were trained and tested on the same cross-validation data. The training phase involved learning the parameters of each model from the training set data. The parameter learning was accomplished using the Expectation Maximization (EM) algorithm. EM attempts to find the maximum log likelihood fit to the data and stops its search when either the max number of iterations specified has been reached or the log likelihood improvement is smaller than a specified threshold. The max iteration count was set to 200 and threshold was set to the BNT default of 0.001. Initial values for the parameters of the model were set to the following, for both models:  $P(G)$  of 0.14,  $P(S)$  of 0.09,  $P(L_0)$  of 0.50, and  $P(T)$  of 0.14. This set of values were found to be the average parameter values across skills in a previous analysis of ASSISTments data using students from

#### 4.2 Performing Predictions

Each run of the cross-validation provided a separate test set. This test set consisted of students that were not in the training set. Each response of each student was predicted one at a time by both models. Knowledge tracing makes predictions of performance based on the parameters of the model and the response sequence of a given student. When making a prediction on a student’s first response, no evidence was presented to the network except for the item identifier associated with the question. Since no individual student response evidence is presented on the first response, predictions of the first response are based on the models’ *prior* and *guess/slip* parameters alone. This meant that, within a fold, KT will make the same prediction for all students’ first response. KT-IDEM’s first response may differ since not all students’ first question is the same and the *guess/slip* differs based on the question. When predicting the student’s second response, the student’s first response was presented as evidence to the network, and so on, for all of the student’s responses 1 to N.

### 5 Results

Predictions made by each model were tabulated and the accuracy was evaluated in terms of Area Under the Curve (AUC). AUC provides a robust metric for evaluating predictions where the value being predicted is either a 0 or a 1 (incorrect or correct), as is the case in our datasets. An AUC of 0.50 always represents the scored achievable by random chance. A higher AUC score represents higher accuracy.

#### 5.1 ASSISTments Platform

The cross-validated model prediction results for ASSISTments are shown in Table 1. The number of students as well as the number of unique templates in each dataset is



included in addition to the AUC score for each model. A Delta column is also included which shows the KT-IDEM AUC subtracted by the KT AUC score. A positive Delta indicates that there was an improvement in accuracy by using KT-IDEM instead of standard KT. A negative indicates that accuracy declined when compared to KT.

**Table 1.** AUC results of KT vs KT-IDEM on the ASSISTments datasets. The Delta column reports the increase (+) or decrease (–) in accuracy by using KT-IDEM.

Skill	#students	#templates	AUC		
			KT	KT-IDEM	Delta
1	756	3	0.616	0.619	+0.003
2	879	2	0.652	0.671	+0.019
3	1019	6	0.652	0.743	+0.091
4	877	4	0.616	0.719	+0.103
5	920	2	0.696	0.697	+0.001
6	826	2	0.750	0.750	- - - - -
7	637	2	0.683	0.689	+0.006
8	1285	3	0.718	0.721	+0.003
9	1024	4	0.679	0.701	+0.022
10	724	4	0.628	0.684	+0.056

The results from evaluating the models with the ASSISTments datasets are strongly in favor of KT-IDEM (Table 1) with KT-IDEM beating KT in AUC in 9 of the 10 datasets and tying KT on the remaining dataset. The average AUC for KT was 0.669 while the average AUC for KT-IDEM was 0.69. This difference was statistically significantly reliable ( $p = 0.035$ ) using a two tailed paired t-test.

## 5.2 Cognitive Tutor

The cross-validated model prediction results for the Cognitive Tutor are shown in Table 2. The number of students, unique problems and data points in each skill dataset are included in addition to the AUC score for each model. The ratio of data points per problem (the number of data points divided by the number of unique problems) is also provided to show the average amount of data there was per problem.

The overall performance of KT vs. KT-IDEM is mixed in this Cognitive Tutor dataset. The average AUC of KT was 0.6457 while the average AUC for KT-IDEM was 0.6441; however, this difference is not statistically reliable difference ( $p = 0.96$ ). As alluded to earlier in the paper, over parameterization is a potential issue when creating a guess/slip per item. In this dataset this issue becomes apparent due to the considerably high number of problems (avg. 311) compared to the number of templates in ASSISTments (avg. 3). Because of the high number of problems, and thus high number of parameters, the data points per problem ratio (dpr) becomes highly important. The five of the twelve datasets with a dpr > 6 were all predicted more accurately by KT-IDEM, with most showing a substantially higher accuracy over KT (+ 0.10 avg. AUC improvement). Among these five datasets, the average AUC of KT was 0.6124 and the average AUC of KT-IDEM was 0.7108. This

difference was statistically reliably ( $p = 0.02$ ). For the skill datasets with  $dpr < 6$ , the loss in accuracy was relatively low ( $\sim 0.04$ ) with the exception of skill 6 that produced a KT-IDEM AUC of 0.497 a score which was 2 standard deviations lower than the mean KT-IDEM score for Cognitive Tutor. This skill dataset had 396 problems with the most frequent problem accounting for 25% of the data points and the 2<sup>nd</sup> most frequent problem accounting for only 0.3%. This was exceptionally unbalanced relative to the other skill sets and served as an example of the type of dataset that the KT-IDEM model does not perform well on.

**Table 2.** AUC results of KT vs KT-IDEM on the Cognitive Tutor datasets. The AUC of the winning model is marked in bold.

Skill	#students	#prob	#data	#data/#prob	AUC		
					KT	KT-IDEM	Delta
1	133	320	1274	3.98	0.722	0.687	- 0.035
2	149	102	1307	12.81	0.688	0.803	+0.115
3	116	345	1090	3.16	0.612	0.605	- 0.007
4	116	684	1062	1.55	0.694	0.653	- 0.041
5	159	177	1475	8.33	0.677	0.718	+0.041
6	116	396	1160	2.93	0.794	0.497	- 0.297
7	133	320	1267	3.96	0.612	0.574	- 0.038
8	116	743	968	1.30	0.679	0.597	- 0.082
9	149	172	1431	8.32	0.585	0.720	+0.135
10	148	177	1476	8.34	0.593	0.626	+0.033
11	149	172	1431	8.32	0.519	0.687	+0.168
12	123	128	708	5.53	0.574	0.562	- 0.012

## 6 Discussion and Future Work

The training of the models in this paper was accomplished by splitting up a cohort of students into a test and training set through cross-validation. If a previous year's cohort of students were used instead, this may increase the number of training samples due to not requiring a portion of the data to be held out. This will also raise the issue of which guess and slip values to use for an item that has been added after the previous year's data was collected and thus was not in the training set. One approach is to use the average of all the learned guess/slip values or use the standard KT model guess/slip values for that question.

The results for the Cognitive Tutor showed that the average number of data points per problem largely determined if the accuracy of KT-IDEM would be greater than KT. It could be that some problems within a skill dataset have high amounts of data while some problems have low amounts. To improve the accuracy of KT-IDEM, the guess/slip values for the low data problems in the model could be replaced with KT's guess/slip values. This would ensure that when predicting performance on high data items, KT-IDEM parameters would be used and KT parameters would be used on low data items. The model parameter fitting could potentially be improved by using information such as average percent correct and number of hints requested to set the initial guess/slip values for each item instead of using default guess/slip values.

An open area for future work would be to improve assessment speed by choosing items based on their guess/slip values learned with KT-IDEM. The standard computer adaptive testing paradigm is focused on assessment, not learning. To accomplish quick assessment, these tests select the questions that give the optimal amount of information about a student's ability based on their response. In an IRT model, this criterion is called item discrimination. A response to an item with high discrimination results in a larger change in the student's assessed ability level than a response to a lower discrimination item. Likewise, in KT-IDEM, guess and slip can also capture discrimination. When an item has a zero guess and zero slip, the student's response is completely representative of their knowledge; however, when the guess and slip are closer to 0.50, the response has less of an impact on the updated probability of knowledge. In order to optimize the selection of questions for assessment, questions can be selected that maximize the change in probability of knowledge given an incorrect response and the change in probability of knowledge given a correct response to the selected question. Questions eligible for selection should have had sufficient data used to train their guess/slip values, otherwise erroneously high or low guess/slip values are likely to be learned and would not represent the true discrimination of the item. While this method could minimize the number of questions needed to assess a student, the questions which lead to the most learning do not necessarily correspond to the questions which are best for assessment. The Item Effect Model [7] has been used to determine item learning value with a Knowledge Tracing approach and could compliment KT-IDEM for choosing the appropriate questions which blend assistance and assessment.

## 7 Contribution

With the ASSISTments Platform dataset, KT-IDEM was more accurate than KT in 9 out of the 10 datasets. KT scored an AUC of 0.669 on average while KT-IDEM scored an AUC of 0.699 on average. This difference was statistically significant at the  $p < 0.05$  level. With the Cognitive Tutor dataset, overall, KT-IDEM is not statistically reliably different from KT in performance prediction. When dpr is taken into account, KT-IDEM is substantially more accurate (0.10 average gain in AUC over KT). This improvement when taking into account dpr is also statistically reliable at the  $p < 0.05$  level.

We have introduced a novel model for introducing item difficulty to the Knowledge Tracing model that makes very minimal changes to the native topology of the original mode. This new model, called the KT Item Difficult Effect Model (IDEM) provided reliably better in-tutor performance prediction on the ASSISTments Skill Builder dataset. While overall, the new model was not significantly different from KT in the Cognitive Tutor, it was significantly better than KT on datasets that provided enough data points per problem.

We believe these results demonstrate the importance of modeling item difficulty in Knowledge Tracing when sufficient data is available to train the model. The real world implication of improved accuracy in assessment is less student time spent over practicing and improved accuracy of skill reports given to teachers. Accurate guess and slip parameters per item with KT-IDEM also opens up the capability for a tutoring system to select questions with low guess and slip and thus optimizing the number of questions needed for assessment while remaining inside the model tracing paradigm.

**Acknowledgements.** This research was supported by the National Science foundation via grant “Graduates in K-12 Education” (GK-12) Fellowship, award number DGE0742503 and CAREER award. Funding was also provided by the Department of Education IES Center for Mathematics and Cognition grant. We would like to thank Hanyuan Lu, Matt Dailey and the Pittsburg Science of Learning Center for the datasets and dataset preparation.

## References

1. Johns, J., Mahadevan, S., Park Woolf, B.: Estimating Student Proficiency Using an Item Response Theory Model. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 473–480. Springer, Heidelberg (2006)
2. Koedinger, K.R., Corbett, A.T.: Cognitive tutors: Technology bringing learning science to the classroom. In: Sawyer, K. (ed.) *The Cambridge Handbook of the Learning Sciences*, pp. 61–78. Cambridge University Press, New York (2006)
3. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction* 4, 253–278 (1995)
4. Baker, R.S.J.d., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008)
5. Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.A., Kauffman, L.R., Mitchell, A.P., Giguere, S.: 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, pp. 52–63. Springer, Heidelberg (2010)
6. Pardos, Z.A., Heffernan, N.T.: Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010)
7. Pardos, Z., Dailey, M., Heffernan, N.: Learning what works in ITS from non-traditional randomized controlled trial data. *The International Journal of Artificial Intelligence in Education* (in Press, 2011)
8. Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K.R., Junker, B., et al.: The Assistent project: Blending assessment and assisting. In: Looi, C.K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) *Proceedings of the 12th Artificial Intelligence in Education*, pp. 555–562. ISO Press, Amsterdam (2005)
9. Corbett, A.T.: Cognitive computer tutors: Solving the two- sigma problem. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) UM 2001. LNCS (LNAI), vol. 2109, pp. 137–147. Springer, Heidelberg (2001)
10. Pardos, Z.A., Heffernan, N.T.: Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. To appear in *Journal of Machine Learning Research W & CP* (in Press)
11. Bell, R., Koren, Y.: Lessons from the Netflix Prize Challenge. *SIGKDD Explorations* 9, 75–79 (2007)
12. Yu, H.-F., Lo, H.-Y., Hsieh, H.-P., Lou, J.-K., McKenzie, T.G., Chou, J.-W., et al.: Feature Engineering and Classifier Ensemble for KDD Cup 2010. In: *Proceedings of the KDD Cup 2010 Workshop*, pp. 1–16 (2010)