

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/225182390>

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

Conference Paper · January 1970

DOI: 10.1007/978-3-642-22362-4\_21 · Source: DBLP

CITATIONS

76

READS

387

2 authors:



**Zachary A. Pardos**

University of California, Berkeley

83 PUBLICATIONS 1,243 CITATIONS

[SEE PROFILE](#)



**Neil T. Heffernan**

Worcester Polytechnic Institute

225 PUBLICATIONS 3,842 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ASSISTments Efficacy Trail with SRI and UMAINE and WPI [View project](#)



PLACEments [View project](#)

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

Zachary A. Pardos, Neil T. Heffernan

Department of Computer Science, Worcester Polytechnic Institute,  
100 Institute Road, Worcester, MA 01609 USA  
zpardos@wpi.edu, 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 affectively 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

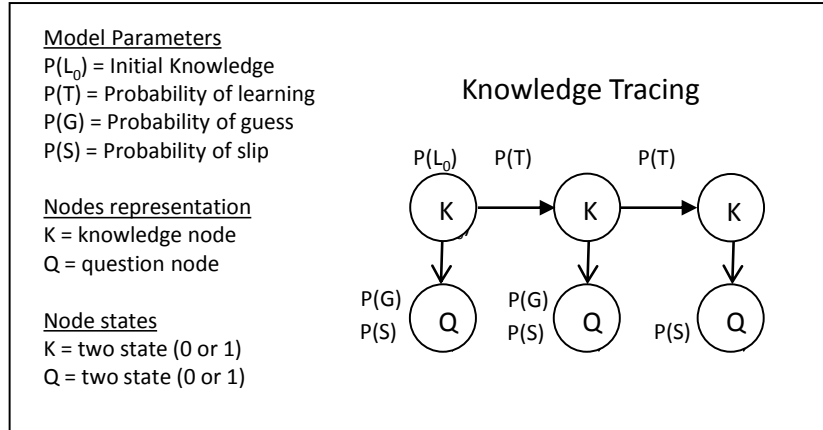
Many models in computer education and assessment take into account difficulty. Item Response Theory (IRT) [1] is one such popular model. IRT is used in Computer Adaptive Testing (CAT) and learns a difficulty parameter per item. This makes IRT models very powerful for predicting student performance; however the model learning processes is expensive and is not a practical way of determining when a student has learned a particular skill. 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 affectively 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 (BKT) model, Fig 1, 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 question 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. 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 or a higher rate of mistakes, or a 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.

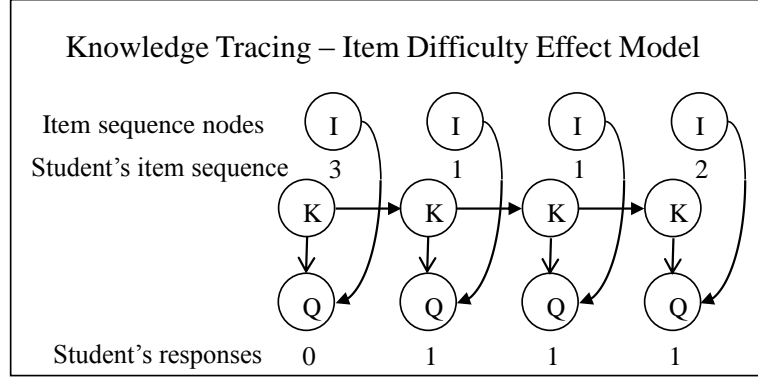


**Figure 1.** The standard Knowledge Tracing model

## 2 Knowledge Tracing: Item Difficulty Effect Model

One of our stated goals was to add difficulty to the classical BKT model without going outside of the Bayesian topology. To do this we use 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 individualized were then conditioned base 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 prior and assigning students to one of those priors based on their first response has proven an affective 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 are 10 distinct items in the skill data, the item node can have values ranging from 1 to 10. The item node is then connected to the question node (Fig 2), 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 and a learn rate and prior, totaling 22 parameters versus BKT'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 models that have equal or even more parameters than data points can still be affective such as was shown in the Netflix challenge and 2010 KDD Cup on Educational Data Mining.



**Figure 2.** The KT-IDEM topology depicting how the question node (and thus the guess/slip) is conditioned on the item node thus adding item difficulty to the 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. By setting a student's item sequence to all 1s, 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 generalizing KT. This model can also be derived by modifying models created by the authors for detecting the learning value of individual items [7].

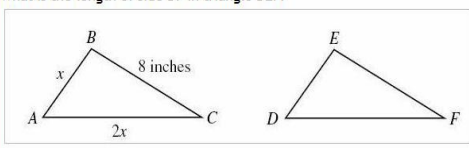
### 3 Datasets

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 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?

*The original question*



[Request Help](#)

Type your answer below (mathematical expression):

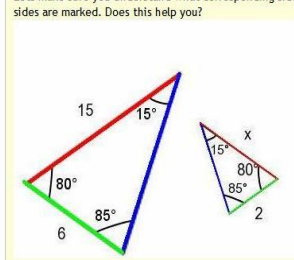
[Submit Answer](#)

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

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

*1st scaffold*

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



[Request Help](#)

Select one:

☒ AB

☐ BC

☐ AC

[Submit Answer](#)

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

*A buggy message*

*A hint*

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

Our first dataset consisted of student responses from ASSISTments [8], a web based math tutoring platform that 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 our web based builder application. For example, a template could 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.

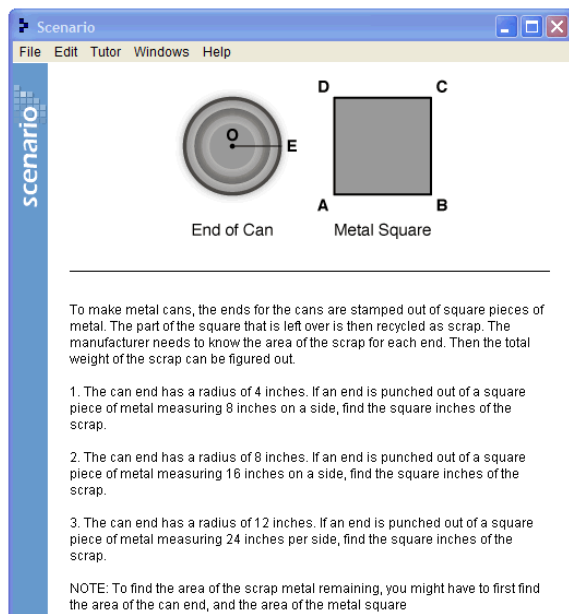
### Skill building datasets

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, 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 12 skill builder datasets with the most data from school year 2009-2001, for this paper. The number of students for each problem set ranged from 637 to 1285. The number of templates ranged from 2-4. This meant that there would be at max 4 distinct sets of guess/slips associated with items in a problem set. Because of the 10 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. Not 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.



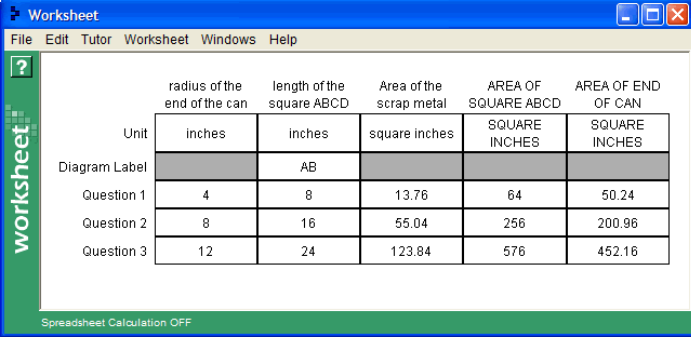
This data was provided as a development dataset in the 2010 KDD Cup competition [10]. The Cognitive Tutor is designed differently than ASSISTments. One very relevant difference to this work is that the Cognitive Tutor presents a problem to a student (Fig 4) that can consist of questions of many skills. 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

**Figure 4.** A Geometry problem within the Cognitive Tutor

determine when a student has mastered a skill. A problem in the tutor can 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 not associated with the mastered 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 (or steps) consisted of multiple skills. Instead of



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

**Figure 5.** Answer entry box for the Geometry problem in Fig 2.

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 10.

## 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 sets of 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 the 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.

## 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. Since no individual student evidence is presented, predictions of the first response are based on the model parameters alone. This means 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. 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 better 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.

**Table 1.** AUC results of KT vs KT-IDEM on the ASSISTments datasets. The AUC of the winning model is marked in bold

Dataset	#students	#templates	AUC	
			KT	KT-IDEM
1	756	3	0.616	<b>0.619</b>
2	879	2	0.652	<b>0.671</b>
3	1019	6	0.652	<b>0.743</b>
4	877	4	0.616	<b>0.719</b>
5	920	2	0.696	<b>0.697</b>
6	826	2	0.750	0.750
7	637	2	0.683	<b>0.689</b>
8	1285	3	0.718	<b>0.721</b>
9	1024	4	0.679	<b>0.701</b>

10	724	4	0.628	<b>0.684</b>
----	-----	---	-------	--------------

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, data points and unique problems in each dataset is 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, on average, how much data there is for each problem, which represents a separate set of guess/slip parameters.

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

Dataset	#students	#datapoints	#probs	#data/#probs	AUC	
					KT	KT-IDEM
1	133	1274	320	3.98	<b>0.722</b>	0.687
2	149	1307	102	12.81	0.688	<b>0.803</b>
3	116	1090	345	3.16	<b>0.612</b>	0.605
4	116	1062	684	1.55	<b>0.694</b>	0.653
5	159	1475	177	8.33	0.677	<b>0.718</b>
6	116	1160	396	2.93	<b>0.794</b>	0.497
7	133	1267	320	3.96	<b>0.612</b>	0.574
8	116	968	743	1.30	<b>0.679</b>	0.597
9	149	1431	172	8.32	0.585	<b>0.720</b>
10	148	1476	177	8.34	0.593	<b>0.626</b>
11	149	1431	172	8.32	0.519	<b>0.687</b>
12	123	708	128	5.53	<b>0.574</b>	0.562

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 reliably difference ( $p = 0.96$ ). As eluded to earlier in the paper, over parameterization is a potential issue when creating a guess/slip per item. In this dataset the problem becomes apparent due to the considerably higher number of problems per dataset than templates per dataset in ASSISTments. Because of this high number of problems, and thus high number of parameters, the number of data points per problem is a significant statistic to observe. With these datasets, the data points per problem (dpr) ratio is highly significant. The five of the twelve datasets with a  $dpr > 6$  were all predicted better by KT-IDEM. 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 ( $p = 0.02$ ).

## 6 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 model. 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 and that the increased accuracy of the model

## Acknowledgements

This research was supported by the National Science foundation via grant “Graduates in K-12 Education” (GK-12) Fellowship, award number DGE0742503 and the Department of Education IES Math center for Mathematics and Cognition grant. We would like to thank the Pittsburg Science of Learning Center for the Cognitive Tutor datasets and Hanyuan Lu for his data preparation assistance.

## References

1. Johns, J., Mahadevan, S. and Woolf, B.: Estimating Student Proficiency using an Item Response Theory Model, in M. Ikeda, K. Ashley and T.-W. Cahn (Eds.): *ITS 2006*, Lecture Notes in Computer Science, 4053, pp 453-462, Springer-Verlag Berlin Heidelberg. (2006)
2. Koedinger, K. R., Corbett, A. T.: Cognitive tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61-78). New York: Cambridge University Press. (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 *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, 406-415. (2008)
5. 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.: Contextual Slip and Prediction of Student Performance

- After Use of an Intelligent Tutor. In *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization*, 52-63. (2010)
6. Pardos, Z. A., Heffernan, N. T.: Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In P. De Bra, A. Kobsa, and D. Chin (Eds.): *UMAP 2010*, LNCS 6075, 225-266. Springer-Verlag: Berlin (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., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar, R., Walonoski, J.A., Macasek, M.A. & Rasmussen, K.P. (2005). The Assistment project: Blending assessment and assisting, In: C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) *Proceedings of the 12th Artificial Intelligence in Education*, Amsterdam: ISO Press. pp. 555-562
  9. Corbett, A. T. (2001). Cognitive computer tutors: solving the two-sigma problem. In: M. Bauer, P. Gmytrasiewicz, & J. Vassileva (Eds.) *User Modeling 2001. LNCS, vol. 2109*, pp. 137--147. Springer Berlin, 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, Special Issue on Knowledge Discovery and Data Mining Cup 2010*. (2011)