Response Tabling– A simple and practical complement to Knowledge Tracing

QING YANG WANG, PAUL KEHRER, ZACHARY A. PARDOS AND NEIL T. HEFFERNAN
Worcester Polytechnic Institute, USA

In this paper we introduce a method of predicting student performance by simply calculating the expected outcome of students with the same sequence or subsequence of responses. This expected outcome, which is simply the percent correct, can be calculated for each response subsequence. The combination of expected outcomes for each subsequence can then be combined for a final prediction of a particular student response. Using skill builder problem sets from the ASSISTments Platform we tested this algorithm against an established model of learning called Knowledge Tracing. Both methods utilized the same data which was only student response data. We found that the Tabling method slightly exceeded knowledge tracing in prediction accuracy. The tabling method training time was minimal, taking only a few seconds to train compared to the 30 minute training time of knowledge tracing. We believe this work offers a valuable alternative to knowledge tracing for use with prediction tasks when information about student learning or knowledge is not required.

Key Words and Phrases: User modeling, sequence mining, ensemble methods, educational data mining

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) aim to improve student learning using reliable assessment, while providing students and teachers immediate feedback on student performance. The fine-grained data produced by these tutoring systems provide an opportunity to model student behavior using various methods. One of the most proven and accepted methods in the ITS field is Knowledge Tracing (Corbett & Anderson 1995) which uses a Dynamic Bayesian Network to track student knowledge. Knowledge tracing provides both the ability to predict future student response values, as well as providing an addition parameter: the probability of student knowledge. For this reason, KT provides insight that makes it useful beyond the scope of simple response prediction. However, KT can be computationally expensive. Model fitting procedures, which are used to train KT, can take hours or days to run on large datasets (Ritter et al. 2009; Bahador & Pardos 2011;Pardos & Heffernan in press). Is this extra computation necessary or can performance be as effectively predicted by calculating simple percent correct features of past response data? This approach was shown to be effective when based on past hint count and response time information [Wang & Heffernan 2011]. We propose an alternative to KT that matches KT's predictive accuracy with minimal computational cost. We suggest that if the task at hand is strictly prediction such as predicting end of year student outcomes or within tutor responses (KDD Cup), our simple tabling model, which ensembles percent correct response predictions, offers a fast and effective solution.

2. DATASET DESCRIPTION

The dataset used in this paper came from the ASSISTments Platform, a web-based tutoring system developed at Worcester Polytechnic Institute and used with 4th to 10th grade math students. The responses are all taken from Skill Building problem sets worked on by students in a suburban middle school in central Massachusetts during the 2009-

Authors' addresses: Department of Computer Science, Worcester Polytechnic Institute, Worcester, Massachusetts 01609. Email: Qin Yang Wang: wangqy@wpi.edu; Paul Kehrer: pkehrer@wpi.edu; Zachary A. Pardos: zpardos@wpi.edu; Neil T. Heffernan: nth@wpi.edu

2010 school year. Skill Building is a special type of problem set, where students are presented with problems from a large pool of problems from a single skill. The student completes problems until he either answers 3 questions correct on the first attempt in a row or completes 10 questions in a given day without getting 3 questions correct in a row, in which case he has "exceeded his daily limit" and must return on a different day. Teachers are also able to change the number of correct problems the student must complete in a row as well as the daily limit. For most of our data the number correct in a row was set to either 3 or 5 but a few teachers set this to as many as 15.

The dataset we used includes student responses from 14 different skill types which are each separated into their own dataset. The skills had an average of 800 student responses per skill. Each data point describes whether the student answered correctly or incorrectly on their first attempt of the question. In each skill, we split the student responses into 5 groups in order to run 5-fold cross validation by student. We trained our table model and a KT model on data from 4 of the folds, and then tested the prediction accuracy of our models on the fifth fold. We did this for all 5 folds. The final results reported are combined from all 5 test folds.

Table 1 shows example rows from our dataset.

Table 1: Example Skill Building Data

Student	Skill	Response Sequence
Student A	Integer Addition	**'0 1 0 1 1 1'**
Student B	Integer Addition	**'1 1 1'**
Student C	Integer Addition	**'0 0 0 1 0 1 1 1'**
Student A	Integer Subtraction	**'0 1 1 1'**
Student D	Integer Subtraction	**'0 0 0 0 1 1 1'**
…		

3. KNOWLEDGE TRACING MODEL DESCRIPTION

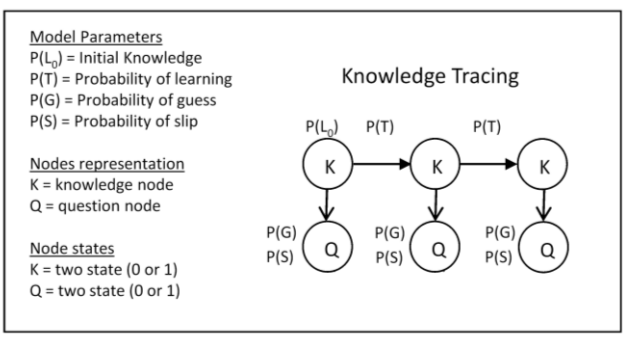

Figure 1: Knowledge Tracing Model

The Knowledge Tracing model has been widely used with ITS to model student knowledge and performance. As shown in Fig 1, Knowledge tracing is a typical 2-variable Hidden Markov Model, with one latent and one observable. There are 4 parameters for each skill. The transmission parameters are comprised of the probability that the student knows the skill before starting, $P(L_0)$, and the probability of learning the

skill from one problem to the next, $P(Knowledge_{t+1}=T \mid Knowledge_t=F)$ or $P(T)$. The emission parameters are the probability of answering correctly while the student doesn't know the skill, "*guessing" P(G)*, and the probability of answering incorrectly when the student does know the skill, "*slip" P(S)*, which can be described as $P(Performance_t|Knowledge_t)$.

In our experiments, we used the Bayes Net Toolbox for Student Modeling (Chang, Beck, Mostow & Corbett 2006) to implement Knowledge Tracing which employs expectation maximization (EM) to fit the model parameters to the training data. We set the initial parameters as follow: initial knowledge: $P(L_0)= 0.50$; guess $P(G) = 0.14$; slip: $P(S) = 0.09$; learning: $P(T) = 0.14$, which are found to be the average parameter values across all skills in previous modeling work conducted using ASSISTments.

4. TABLING MODEL DESCRIPTION

The tabling method evolved out of the simple intuition that common student response sequences may repeat themselves. That is, that overall percent correctness on a problem given a particular past response sequence can predict future performance of other students given the same response sequence. To generalize this idea we take a sequence of correct and incorrect responses, and look at the percentage of correct responses on the next problem. For example, say Student A has answered **'0 1 1'** (0 = incorrect, 1 = correct). We will look at all sequences of student responses that match **'0 1 1'**, observing the response that follows this sequence. So, if 72% of student responses that are preceded by **'0 1 1'** are correct, then we can predict 0.72 for Student A's next response. We describe this prediction as the probability of a correct response given a preceding sequence of **'0 1 1'**, or $P(x=1 \mid$ ***'0 1 1'***$) = .72$. Our model becomes a simple table that maps sequences of preceding responses to the percent correct of the follow-up response. We input a sequence of student responses to the table, and use the percent correct from the row corresponding to that sequence as a prediction for the student's next response.

4.1 TRAINING THE TABLE

For each skill we train one table. The table has a row for each distinct sequence of preceding responses. To best inform our predictions, we used 5 different sequence lengths, observing 0 through 4 previous responses. The rows are labeled: ' ', '**0**', '**1**', '**0 0**', '**0 1**', '**1 0**', '**1 1**', '**000**', etc. There are a total of 31 rows in each table (1 of length 0, 2 of length 1, 4 of length 2, 8 of length 3, and 16 of length 4). For each row in the table, we record how many instances of this response there are, as well as the number of correct and incorrect responses that follow the sequence. We use these numbers to then calculate the percent correct of next responses.

To maximize the number of data points used in training the model, we attempt to use each response of all the students trained against each different sequence length. That is to say for each student's first response, we can only train using an empty preceding sequence. For students' second responses, we can train the empty sequence row as well as the 1-response sequence rows. Table 2 illustrates how the training works, by showing the results of training a table using only one student's response of **'0 1 0 1 1'**.

Table 2: Example for a single student with responses of "01011" for training

Sequence Size	Sequence	Sequence instances	# of next response Correct	# of next response Incorrect	Percent Correct
0	' '	**5**	3	2	0.6
1	'0'	**2** [**0**1011], [01**0**11]	2	0	1.0
	'1'	**2** [**0**1011], [0101**1**]	1	1	0.5
2	'0 0'	**0**	0	0	undefined
	'0 1'	**2** [**01**011], [01**01**1]	1	1	0.5
	'1 0'	**1** [0**10**11]	1	0	1 .0
	'1 1'	**0**	0	0	undefined
	…				

The first row of the table shows that given no prior information, all we can do is count the number of correct responses to calculate the percent correct. In the second row, where the preceding sequence is **'0'**, we see that there are two instances of a **'0'** sequence in the response **'0 1 0 1 1'**. Since the response after the **'0'** is **'1'** in both cases, we record a percent correct of 1.0 for that sequence. We continue this process until we have covered all sequence sizes, and all possible sequence combinations. By counting the number of sequence instances in the table, we see that this single sequence of 5 responses provides our table model with 15 data points.

We counted responses in this manner for every student response in each skill, which leaves us with a table like Table 3, which shows a fully trained table for one of our skills.

Table 3: Fully trained table for sequence lengths 0-3

Sequence Size	Sequence	# of next response Correct	# of next response Incorrect	Percent Correct	Corresponding prediction from KT
0	' '	606	908	0.6674	0.5085
1	'0'	136	255	0.5333	0.2996
	'1'	217	270	0.8037	0.7445
2	'0 0'	42	95	0.4421	0.2349
	'1 0'	27	44	0.6136	0.4451
	'0 1'	91	119	0.7647	0.5717
	'1 1'	110	131	0.8397	0.8864

	‘0 0 0’	12	40	0.3000	0.2205
	‘1 0 0’	8	10	0.8000	0.2762
	‘0 1 0’	14	23	0.6087	0.3278
	‘1 1 0’	13	17	0.7647	0.6398
	‘0 0 1’	22	37	0.5946	0.5050
	‘1 0 1’	14	15	0.9333	0.6978
	‘0 1 1’	64	79	0.8101	0.7870
	‘1 1 1’	31	37	0.8378	0.9535
	…				

4.2 MAKING PREDICTIONS

To test the predictive power of our tabling method, we made predictions for every student response in the test data set. For each response, we used the preceding 4 responses as evidence (or fewer if 4 did not exist such as when predicting the first four responses), and made a prediction. Figure 2 outlines the full prediction process for a single student’s response.

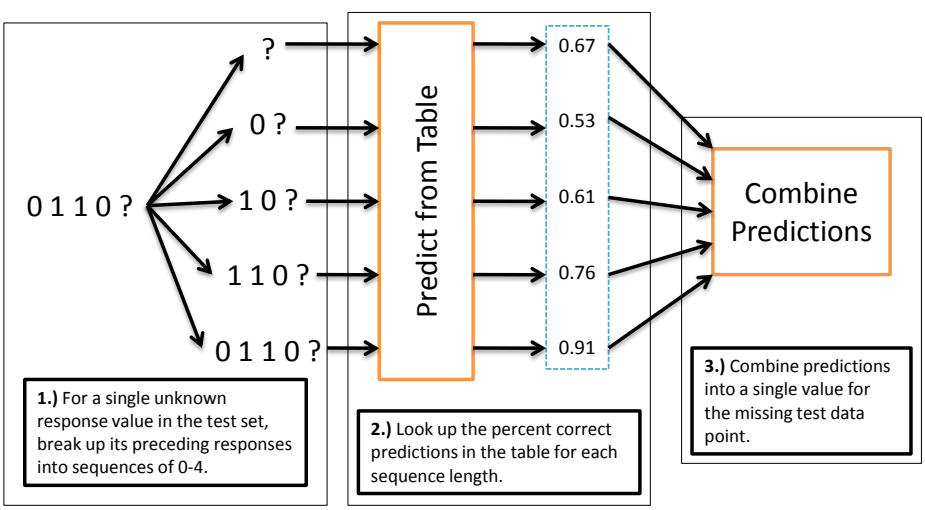

Figure 2: Test Data Prediction Process

First, we would create up to 5 different evidence sequences based on the student’s preceding responses. Then, we would look up each of the 5 sequences in the trained table, yielding 5 different percent correct values. We then took these 5 prediction values, and combined them to come up with a prediction of correctness based on the student’s preceding responses. For the combination step of the process, we did a simple average of our table’s predictions.

4.3 TABLING RESULTS COMPARED WITH KNOWLEDGE TRACING

Table 4 shows RMSE values for our two models predicting student responses on our test data. The table also shows p values from a paired T-Test of each prediction, as well as a T-Test of the squares of the residuals. We found an RMSE of 0.4414 for our tabling model, which is reliably better than the Knowledge Tracing model for this dataset.

Table 4: RMSE of our two models

	RMSE	**Paired TTest Compared to KT**	**Paired TTest on Residual Squared Compared with KT**
KT	0.4534	-	-
Tabling	0.4414	$p << 0.01$	$p << 0.01$

5. ENSEMBLING TABLE PREDCITIONS WITH KNOWLEDGE TRACING

We had the intuition that since our tabling model is so different from the Knowledge Tracing approach, there would be a potential improvement from combining our results from tabling with the predictions from our KT model. The tabling model takes only the previous four responses into account when predicting, and makes a simple mapping of response sequences to percent correct, whereas Knowledge Tracing can use a longer sequence of responses and models the student's probability of knowledge while also making predictions. We hoped that the strengths of the two methods could be ensemble into more accurate prediction.

We combined our results in two ways. We first tried taking the simple average of the tabling model's prediction and KT's prediction for each response. We refer to the second method in the results as "Average-Min-Max", shown in equation 1, motivated by the thought that for each prediction, we would use the model that gave the most definitive response. That is to say, if both predictions were above 0.5, we chose the higher of the two. If both were below 0.5 we chose the lower prediction. If one prediction was above our threshold and the other was below, we chose the average of the two predictions.

$$avgMinMax(P_t, P_k) = \begin{cases} \min(P_t, P_k) \text{ if } (P_t < 0.5) \wedge (P_k < 0.5) \\ \max(P_t, P_k) \text{ if } (P_t > 0.5) \wedge (P_k > 0.5) \\ \text{avg}(P_t, P_k) \text{ if } (P_t < 0.5) \wedge (P_k > 0.5) \\ \text{avg}(P_t, P_k) \text{ if } (P_t > 0.5) \wedge (P_k < 0.5) \end{cases}$$

Equation 1: "Average Min Max" ensembling method

5.1 ENSEMBLING RESULTS

After ensembling our predictions, we found that taking a simple average gave us a significant improvement in RMSE, while using our AverageMinMax technique gave no reliable improvement.

Table 5: RMSE of our ensembled predictions

	RMSE	**Paired TTest Compared to KT**	**Paired TTest on Residual Squared Compared with KT**
KT	0.4534	-	-
Tabling	0.4414	p << 0.01	p << 0.01
Simple Average	0.4391	p << 0.01	p << 0.01
AverageMinMax	0.4438	p << 0.01	p << 0.01

6. RESIDUAL ANALYSIS

We wanted to learn more about exactly how our table model was performing better than KT. We decided to track residuals and RMSE on a per opportunity basis. Figures 3 and 4 show the two graphs for the first 10 student responses. It should be noted that the majority of our student response sequences are 5 or 6 responses long. The behavior of the graphs from 7-10 is based on fewer data points than the rest of the graph.

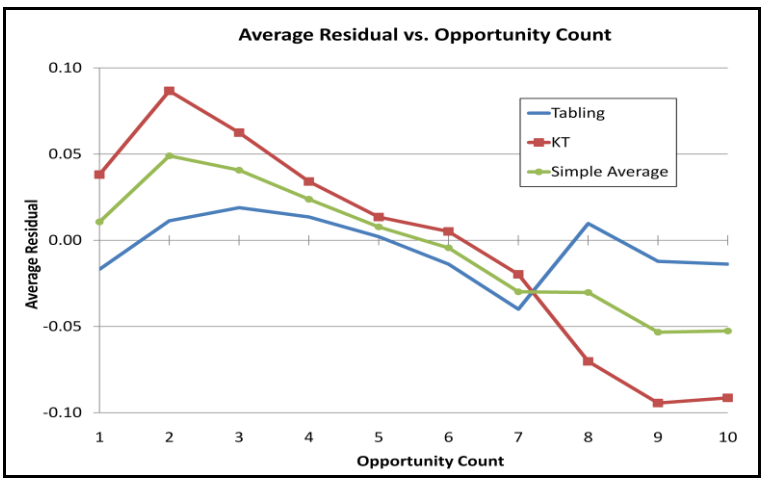

Figure 3: Average Residual Analysis

It is interesting to note from the residual graph in Figure 3 that KT is underpredicting early in the response sequence. Our intuition is that KT is more conservative in its likelyhood of learning increases, and takes a few responses before it can confidently predict correctness. On the other hand, in the later responses of the residual graph, you can see that KT is over predicting, being overly convinced that the student has learned the skill, when he still may be answering questions incorrectly. This fits with our intuition that tabling, which has no concept of a predecided learning curve, would be resistant to drastic over or underpredictions.

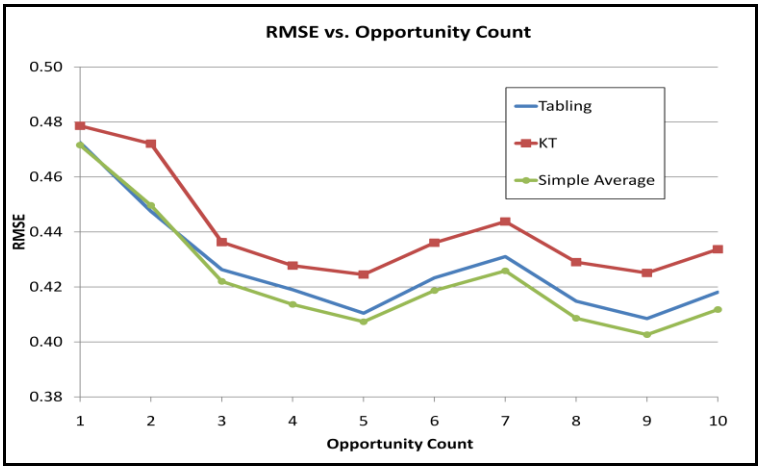

Figure 4: RMSE Analysis

We see in the graph of RMSE in Figure 4, that tabling has a sharp decrease in error on the second response, while KT only has a slight improvement from the first response. This suggests again that KT is underpredicting, and cannot confidently predict correctness with only one response. Tabling, whose average residual at response 2 is close to 0, has a much more accurate prediction for the second response in the sequence.

6.1 BEST AND WORST PREDICTED SEQUENCES

To further examine the performance of tabling versus KT, we found the sequences with the least accurate and the most accurate predictions from our two models. We found that the 3 worst sequences, and the 4 worst sequences were the same for Tabling and KT.

Table 6: Top 3 worst-predicted sequences for Tabling and KT

		RMSE		
#	**Sequence**	**Tabling**	**KT**	**Count**
1	‘**1 1 1 1 0**’	0.8232	0.9924	46
2	‘**1 1 1 0**’	0.7941	0.9835	58
3	‘**1 1 0**’	0.7913	0.9501	168

Table 6 shows the 3 worst sequences for predicting the next response for Tabling and KT. The sequences make intuitive sense, showing responses that have a number of correct responses followed by one incorrect response. It makes sense that it would be difficult to predict the next response, because the previous response could either be an indication that the student doesn’t know the skill, or simply a mistake that would be followed by a correct answer. It is worth noting that for these worst sequences, tabling is more accurate than KT.

Table 7: Top 4 best-predicted sequences for Tabling and KT
(ordered by Tabling's top 4, KT had #3 and #4 swapped)

		RMSE		
#	Sequence	Tabling	KT	Count
1	'**1 1 1 1 1 1**'	0.1478	0.0034	40
2	'**1 1 0 1 1 1 1 1**'	0.1588	0.0056	35
3	'**1 1 0 1 1 1 1**'	0.1641	0.0143	42
4	'**1 1 1 1 1**'	0.1730	0.0091	559

Table 7 shows the best-predicted sequences. As expected, both models are very accurate when the student has gotten a few questions correct in a row before the current question. It makes intuitive sense that the next question is very likely correct. For the best sequences, KT is more accurate than tabling. This is likely because if a student has gotten 6 questions correct in a row, he definitely knows the skill and will very likely answer correctly. KT will have a very high probability of correctness for this sequence, but Tabling will only look at 4 of the responses, and will average the 5 sequence length predictions. The shorter length predictions will generally be much lower than a longer sequence of correct responses.

7. DISCUSSION

What we are suggesting with our Table model is a simple method for doing effective response prediction. Unlike Knowledge Tracing, the Table model offers no interpretability or domain insight. Knowledge tracing can be used for predicting student responses, but it also models a student's probability of knowledge. Interpreting this parameter can be useful in various educational applications. For instance, the Cognitive Tutor uses Knowledge Tracing to determine if a student is finished with the current skill and can move on in the curriculum. The Table model doesn't track any parameters, and doesn't even model students as their own entity. We enter a sequence of evidence responses as input into the table model, and it returns a prediction for the next response. There is nothing else to be learned from the table.

For this reason, we see the table model as having a useful yet limited application. It has the distinct advantage over Knowledge Tracing of being computationally inexpensive. Both training the model and making predictions took seconds, whereas training a dynamic Bayesian Network using Expectation Maximization can take hours using a large enough dataset. If there is the need for interpretability, where predicting the student's response is not enough, our model is not appropriate. However, when the goal is strictly predicting the next response, we believe we have a lightweight compliment to KT that can increase student response prediction accuracy.

8. FUTURE WORK

The model used in the paper relied strictly on previous responses to make predictions. The tabling idea is a simple idea that could easily be expanded to leverage more features into the prediction. We could conceivably have additional rows to our table which show the percentage of correct responses where the student took longer or shorter than a certain amount of time to answer. We could also calculate the percent correct based on the number of previously answered questions on this skill. This way, in our prediction step of testing shown in Figure 2 would include the 5 predictions based on previous responses as well as predictions based on additional features from the dataset. Wang & Heffernan

explored the impact of hints on student performance by finding the percent correct on items based on the number of hints requested (Wang & Heffernan 2011). This is another feature that could be added into our tabling model.

Another area for improvement of the table modeling is the step of combining our predictions from the different table entries. For our results in this paper we simply averaged all of the predictions. Perhaps there is a way to better ensemble these predictions so that the more accurate predictions end up having more weight. The method used to ensemble would have to allow for missing values, because not every response tested will have 4 preceding responses.

ACKNOWLEDGEMENTS

This research was supported by the National Science foundation via grant "Graduates in K-12 Education" (GK-12) Fellowship, award number DGE0742503 and Neil Heffernan's CAREER grant. We would like to thank the organizers of the 2010 KDD Cup at the Pittsburg Science of Learning Center for the Cognitive Tutor datasets and Matthew Dailey for his data preparation assistance.

We acknowledge the contributions of Cristina L. Heffernan, Program Manager and School Liaison for the ASSISTments program. We also acknowledge the many additional funders of ASSISTments Platform found here: http://www.webcitation.org/5ym157Yfr

REFERENCES

BAHADOR, N., PARDOS, Z., HEFFERNAN, N. T. AND BAKER, R. 2011. Less is More: Improving the Speed and Prediction Power of Knowledge Tracing by Using Less Data. *Educational Data Mining 2011 Conference*

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., AND STONE, C. J. 1984. *Classification and Regression Trees*. Wadsworth International Group, Belmont, California

CAURANA, R, NICULESCU-MIZIL, A. CREW, G., KSIKES. 2004. Ensemble selection from libraries of models. *Proceedings of the 21st International Conference on Machine Learning(ICML '04)*

CHANG, K., BECK, J., MOSTOW, J., AND CORBETT, A. 2006. A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. *Intelligent Tutoring Systems, 8th International Conference, ITS 2006,* 104-113

CORBETT, A. T., AND ANDERSON, J. R. 1995. Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4, 253–278.

KOEDINGER, K.R., ANDERSON, J.R., HADLEY, W.H., AND MARK, M.A. 1997. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30–43

PARDOS, Z. A. AND HEFFERNAN, N.T. 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization*

PARDOS, Z. A. AND HEFFERNAN, N. T. 2010. Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. *In Proceedings of the 3rd International Conference on Educational Data Mining*, 161-170.

PARDOS, Z. AND HEFFERNAN, N. In Press. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research*

QUINLAN, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

RITTER, S., HARRIS, T., NIXON, T., DICKISON, D., MURRAY, C., TOWLE, B. 2009. Reducing the knowledge tracing space. *In Proceedings of the 2nd International Conference on Educational Data Mining*, Cordoba, Spain, 151–160

WANG, Y. AND HEFFERNAN, N. T. 2011. The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *The 24th International FLAIRS Conference*