Tutor Modeling Versus Student Modeling

Zachary A. Pardos, Neil T. Heffernan

Department of Computer Science

Worcester Polytechnic Institute

zpardos@wpi.edu, nth@wpi.edu

Abstract

The current paradigm in student modeling has continued to show the power of its simplifying assumption of knowledge as a binary and monotonically increasing construct, the value of which directly causes the outcome of student answers to questions. Recent efforts have focused on optimizing the prediction accuracy of responses to questions using student models. Incorporating individual student parameter interactions has been an interpretable and principled approach which has improved the performance of this task, as demonstrated by its application in the 2010 KDD Cup challenge on Educational Data. Performance prediction, however, can have limited practical utility. The greatest utility of such student models can be their ability to model the tutor and the attributes of the tutor which are causing learning. Harnessing the same simplifying assumption of learning used in student modeling, we can turn this model on its head to effectively tease out the tutor attributes causing learning and begin to optimize the tutor model to benefit the student model.

Introduction

The beginning of the current paradigm in student modeling, known as Knowledge Tracing (Corbett & Anderson 1995) started with Atkinson's approach to modeling instruction (Atkinson & Paulson 1972). An adaptation of the Bayesian computations from Atkinson and a simplification of the more complex ACT-R cognitive architecture (Anderson 1993), Knowledge Tracing has firm roots in learning theory. However, it is its use in practice that has drawn the majority of attention to the model. The Cognitive Tutors™, used by over 500,000 students, annually, employ Knowledge Tracing to determine when a student has learned a particular skill and when to subsequently end practice of that skill. The real world adoption of the model has made it a popular yard stick for gauging the relative performance of new models, of which there have been many (Desmarais & Baker 2011).

There has been a focus in the literature on within-tutor predictive performance as the primary benchmark of comparison between models (Pardos et al. 2012). This was also the benchmark used to rank solutions to the recent Knowledge Discovery and Data Mining (KDD) Cup on Educational Data, a high profile annual data mining competition organized by the Association for Computing Machinery (ACM). An extension to Knowledge Tracing which individualized model parameters per student was part of a solution that placed 4th in the competition (Pardos & Heffernan, in press). While the primary application of Knowledge Tracing has been to infer student knowledge, the model can be extended to make inferences about the effect of various components of the tutor on learning.

In this paper we overview the techniques in which Knowledge Tracing's Bayesian framework has been extended to incorporate attributes of the student to improve prediction. We also look at how model extensions have expanded to various attributes of the tutor and allowed for the learning effect of those attributes to be observed.

The Bayesian Knowledge Tracing Model

An average student can be modeled as a statistical processes with probability *P(L0)* of knowing the skill being practiced before instruction beings. If the student begins with not knowing the skill then she will likely answer the first problem incorrectly but can guess the correct answer with probability *P(G)*. If the student begins with knowing the skill then she will likely answer the first problem correctly but can make a mistake, or slip, with probably *P(S)*. A student who begins with not knowing the skill will learn the skill with probability *P(T)* between the first and second opportunities and between all subsequent opportunities until the skill is learned. These probabilities; *P(L0)*, *P(G)*, *P(S)* and *P(T)* comprise the set of parameters of Knowledge Tracing with which student knowledge and performance is modeled. This process is equivalent to that of a Hidden Markov Model (HMM). In an HMM, *P(G)* and *P(S)* are referred to as the emission parameters, while *P(T)* is the transition parameter. In the context of Intelligent Tutoring Systems, *P(G)* and *P(S)* are referred to as the performance parameters, with *P(L0)* and *P(T)* being the knowledge parameters. In Knowledge Tracing, the

probability of forgetting is fixed at zero. The parameters *P(L0)* and *P(T)* affect the projected probability of knowledge over time in a similar fashion to learning curve analysis (Martin et al. 2005). Note that the projected probability of knowledge at the next opportunity to answer a question of the same skill, $P(L_{n+1})$, does not involve the performance parameters and is calculated with the following formula:

$$P(L_n|Response_n) + \big(1 - P(L_n|Response_n)\big)P(T)$$

If no response at opportunity n exists then the prior probability of L_n is used.

Reasoning about the value of the latent given observations of correct or incorrect responses is a separate task involving the guess and slip parameters. The closer to zero the guess and slip parameters, the less uncertainty exists about the latent of knowledge, given an observation. Given a high guess value, a longer sequence of correct responses would be necessary to have 0.95 or greater certainty in the skill being known (the threshold at which the Cognitive Tutors reach the conclusion of mastery). The posterior probability of knowledge, which is the updated probability of knowledge after observing some evidence, $P(L_n|Response_n)$, is calculated by the following formula, given an observation of a correct answer to a question:

$$\frac{P(L_n)\big(1 - P(S)\big)}{P(L_n)\big(1 - P(S)\big) + \big(1 - P(L_n)\big)P(G)}$$

Given an observation of an incorrect answer to a question, the following formula is used:

$$\frac{P(L_n)P(S)}{P(L_n)P(S) + \big(1 - P(L_n)\big)\big(1 - P(G)\big)}$$

The initial introduction of Knowledge Tracing by Corbett & Anderson used Bayesian update rules to calculate the inference of knowledge, however; it wasn't until 2004 that Reye demonstrate that these update rules could be completely modeled within the framework of a Dynamic Bayesian Network (Reye 2004). The work referred to in this paper uses static, unrolled Dynamic Bayesian Networks, which are the equivalent of a DBN for a fixed number of time steps.

Parameter fitting

Either grid-search or Expectation Maximization (EM) can be used to fit the parameters of the model to the data. Details of both methods and their predictive performance have been an active topic of discussion in the student modeling literature (Pardos et al. 2012). With the standard knowledge tracing parameters, grid-search runs faster but its runtime increases exponentially with the addition of parameters to the model. The runtime of EM, however, follows a power function with increasing numbers of parameters and is a widely used algorithm for fitting parameters of HMMs, making it a preferred choice when fitting the more complex, individualized models which will be presented in later sections.

Identifiability

The standard objective in training parameters of a model is to achieve goodness of fit to the data. The objective in training parameters for a model being used for cognitively diagnostic purposes is two-fold. With such a model, parameter plausibility is also an objective. With four parameters it is possible that the same goodness of fit to the data can be achieved with two entirely different sets of parameter solutions (Beck & Chang 2007). While this is not an issue for data prediction, it is problematic for meaningful inference of the latent of knowledge, which is the primary use of Knowledge Tracing in the Cognitive Tutors. Various mends to the problem have been employed such as bounding parameter values when using grid-search , setting the initial parameter position to plausible values instead of random values when using EM, and individualizing the prior parameter to achieve an improved baseline of traction for plausible parameter convergence (Pardos et al. 2012).

Modeling Student Individualization

Standard Knowledge Tracing makes the simplifying assumption that all students learn a skill at the same rate and begin practicing a skill with the same prior knowledge. Individualization of these parameters can break this simplifying assumption and has shown improvement over standard Knowledge Tracing in performance prediction in the Cognitive Tutor for Algebra (Pardos & Heffernan, in press) and for Genetics as well as the ASSISTments tutor's non-skill building problem sets (Pardos & Heffernan 2010), although; using prior knowledge individualization did not improve prediction in the ASSISTments skill-building problem sets (Pardos et al. 2012).

Corbett & Anderson took a regression approach to individualization that trained the general set of four parameters learned per skill and then used a regression to add in a student weight for each parameter that spanned skills. While incorporation of individual weights resulted in higher correlation of predictions to a post-test, the weights did not improve the accuracy of the predictions of within-tutor student responses. We will discuss an individualization approach proposed by Pardos & Heffernan (2010) that takes a similar angle to Corbett & Anderson but adheres to a strictly Bayesian formulation. New criticism of the model will also be presented as well as novel suggestions for improvement.

Student Individualization (multistep)

The individualization model used in the KDD Cup competition used a multistep training process of individualizing the student parameters whereby a separate model was first trained for each student and then combined with a model trained for each skill (Pardos & Heffernan, in press). This resulted in $U + S$ models being trained where

U was the number of students and S was the number of skills.

The first step was to learn parameters for each student. In standard Knowledge Tracing, skill parameters are learned by training from a dataset where the rows are different students who have provided responses to the skill and the columns are the students' answers to the skill at different opportunities. To train student parameters, the dataset was transformed to have the rows be different skills a particular student has provided responses to and the columns be the student's responses to those skills at different opportunities. Figure 1 shows the difference between a dataset organized for skill parameter training vs. one organized for student parameter training.

Skill Dataset (Pythagorean Theorem)					
	Op.1	**Op.2**	**Op.3**	**Op.4**	**Op.4**
John	0	1	1	1	
Christopher	0	1	0	1	1
Sarah	1	1	1		

Student Dataset (Christopher)					
	Responses				
	Op.1	Op.2	Op.3	Op.4	Op.4
Addition	1	1	1		
Pythagorean	0	1	0	1	1
Subtraction	0	1	0	1	1

Figure 1. Example datasets prepared for training skill parameters (above) and student parameters (below)

The result of the first step was a *P(L0)*, *P(G)*, *P(S)* and *P(T)* parameter fit for each student. The next step was to train per skill models that incorporated all of the student parameters. For simplicity of presentation here we will demonstrate incorporating only the individual student learning rate, *P(T)*, although the technique generalizes to the other parameters as well.

Figure 2 shows a Bayesian network approach to incorporating the individual student learn rates, represented in the *H* node, into the skill model. In this step, *P(L0), P(G), P(S)* and *P(T|H)* parameters are learned per skill. The student parameters, *P(H|Student)*, are fixed to the values learned in step 1 and are constant for each skill model. They are stored in a Conditional Probability Table (CPT) belonging to the *H* node, which is a binary node that stands for *High Learner*. A student ID is included in each row of the skill response dataset in order to reference the appropriate individual student learn rate associated with the evidence. The individual learn parameters dictate the probability that the *H* node is true or not. Since the learning rate per skill is conditioned on the value of the binary *H* node, two learning rates per skill are trained; one for *high learners*, H, and one for *non high learners*, $\overline{H}$. The formula for calculating the probability of knowledge at the next opportunity, $P(L_{n+1})$, in this model is:

$$P(L_n|Response_n) + \big(1 - P(L_n|Response_n)\big)P(T|H)P(H|Student) + \big(1 - P(L_n|Response_n)\big)P(T|\overline{H})(1 - P(H|Student)$$

The formulas for calculating the posterior probabilities and probabilities of correct answers do not differ from standard Knowledge Tracing.

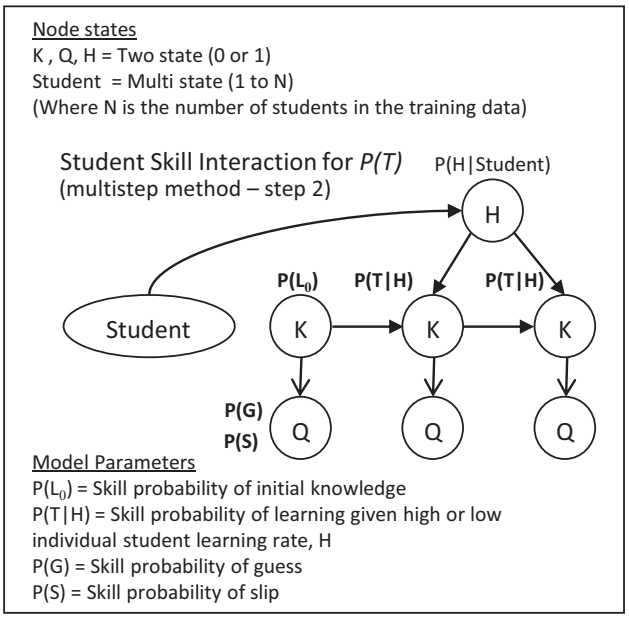

Figure 2. Bayesian network of the multistep model which incorporates the individualized student learning rates

The strength of this model is that it incorporates individual student learn rates into the model in a way that is massively parallelizable at each step. The student parameter models can be learned completely independently of one another, as can the skill models, after the student models have completed. This is of significant benefit to computation time if cluster resources are available and a large dataset is being processed, such as the 21010 KDD Cup datasets, one of which had 6,000 users and 900 skills.

There are several weaknesses to this parallelizable two-step approach, however. One is that the students must have answered a similar distribution of skills (by difficulty) in order for the individual student learning rates to be comparable to one another. For example, if an average learning rate student answers only skills which are easy to learn, she will likely receive a high individual learn rate. However, if a high learning rate student answers only skills which are difficult, she will have a learning rate lower than the other student but only because the two students completed skills of disparate difficulty. The second weakness is lack of normalization of the individual parameters when incorporated in the skill model. The

effect of this is that the difference in a skill's *high learner* learning rate and *not high learner* learning rate can only be as large as the difference between the smallest and the largest individual student learning rate. The individual parameters must be normalized to allow for greater play in the skill learn rates. Normalizing probabilities is a concern, however, in the case where the trained model is applied to a new student with an individual learning rate that is higher or lower than the minimum or maximum pre-normalized student learning rate.

Student Individualization (single step)

The two issues of 1) an equal skill distribution requirement and 2) lack of normalization in the *high learner* node, which exist in the multistep model, can be addressed with a single step individualized model. This model trains skill and student parameters simultaneously. This allows for individual student parameters to be fit in the context of all skill models, thus no longer requiring equal skill distribution among students. It also allows for the individual student parameters, such as the learn rates in the *high learner* node, to be of any magnitude between 0 and 1 that best fit the global model, instead of being limited to the minimum and maximum student *P(T)* values. This serves to no longer confine the disparity between *high learner* and *non high learner* conditioned skill learn rates.

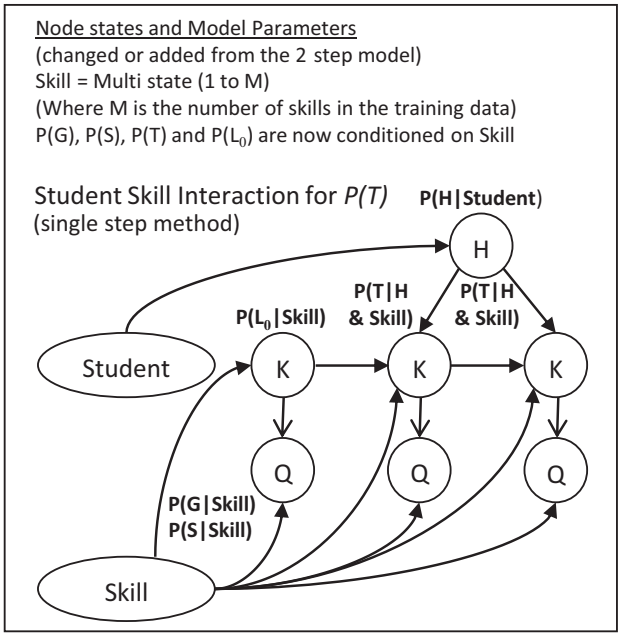

Figure 3. Bayesian network of the single step model which simultaneously fits skill and student parameters

This single step model, shown in Figure 3, trains skill and student parameters simultaneously by adding a *Skill* node to the model, which is a multinomial node with values ranging from 1 to M where M is the number of skills in the training data. The skill parameters are made conditionally dependent on the Skill node, allowing for *P(G), P(S), P(T|H)* and *P(L0)* parameters to be trained per skill, for all skills at once. A student ID as well as a Skill ID is included in the rows for the skill dataset to properly associate the evidence with both skill and student. The individualized student learn parameters in the *high learner* node must be initialized to some values before training. This might appear to be an initialization and convergence problem for large numbers of students but this is no more a problem than was present in the multistep method. In both methods, the initial values of the student parameters can be set to the same value or initialized randomly within some plausible bound. The additional data present in this single step model should help constrain the parameter values and result in better overall model performance compared to the multistep method.

The drawback to this approach is that the model is fit not just in a single step but in a single training of EM. This means high single threaded compute time for EM convergence as well as high memory load, since the entire dataset is being fit to at once instead of a single user's data or a single skill's data at once as was the maximum load seen in the multistep method. One way in which to reduce the data size while still fitting parameters for all students and skills is to cluster students and or skills at some K and only include the response sequences, or a sampling of response sequences, representative of the clusters during training. At K equal to M or N, the result would be equivalent to using all data. As K decreased, so should the model fit but a happy medium value of K should exist such that the data size is tractable and performance is still above that of the multistep model.

Modeling the Effects of the Tutor

Individualization at the student level tells us something interesting about the student; how fast they learn, how much they have retained from past instruction, but learning something about the tutor and how it affects learning can be more actionable as it sheds light on ways in which to improve instruction to better assist and assess the student.

Individualization of Educational Content in the Tutor

Before the effects of the tutor on learning can be measured, the difficulty of individual questions, or piece of educational content in the tutor, must be controlled for. In order to accomplish this, a separate guess and slip parameter can be fit for each question in a skill or problem set. Fitting separate guess and slip parameters per question modulates the difficulty and also the information gain among the questions. As described in the introduction section, guess and slip values closer to zero allow for lower uncertainty in the inference of the latent of knowledge. Different guess and slip values for each question allows for the appropriate amount of information, about whether or

not a correct answer should translate to knowledge of the skill, to be gained from a response. A correct response and inference of knowledge should, by virtue of the HMM design, transfer to the next opportunity to answer the next question of the same skill. Therefore, the amount of information gain for each question, set through the guess and slip parameters, expresses the relative relation between performance and knowledge among the questions. The utility of individualization of question guess and slip is maximized when the order in which questions are presented to students is randomized for each student.

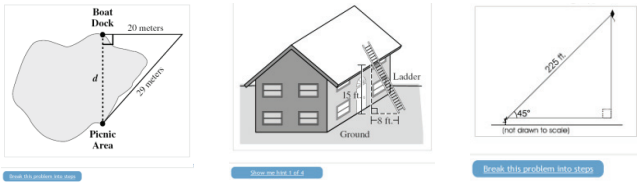

Figure 4. Pythagorean theorem questions (A), (B) and (C)

Consider the three Pythagorean theorem questions (A,B,C) in Figure 4. All three questions ask the student to find the hypotenuse length; (A) does so with a lake cover story, (B) uses a house cover story and (C) uses no cover story at all. They all have a button bellow the picture that provides the student with assistance if pressed. The first two questions provide help in the form of hints while the third question provides help in the form of step by step tutored problem solving, otherwise known as scaffolding. A dataset representing student answers to these questions might look like the following, in Figure 5, where the identifying letter IDs of the questions serve as the attribute values.

Skill Dataset (Pythagorean Theorem)						
	Responses			Attribute		
	Op.1	Op.2	Op.3	Op.1	Op.2	Op.3
John	0	1	1	C	A	B
Christopher	0	1	0	B	A	C
Sarah	1	1	1	A	B	C

Figure 5. Example dataset of student responses and question IDs serving as the attribute at each opportunity

It could be imagined, given more data, that these questions vary in difficulty among one another, with question C being answered correctly 33% of the time, B being answered correctly 66% of the time, and question A being answered correctly 100% of the time. The model in Figure 6 shows how question level individualization of difficulty, via the guess and slip parameters, has been accomplished in a Bayesian network (Pardos & Heffernan 2011b).

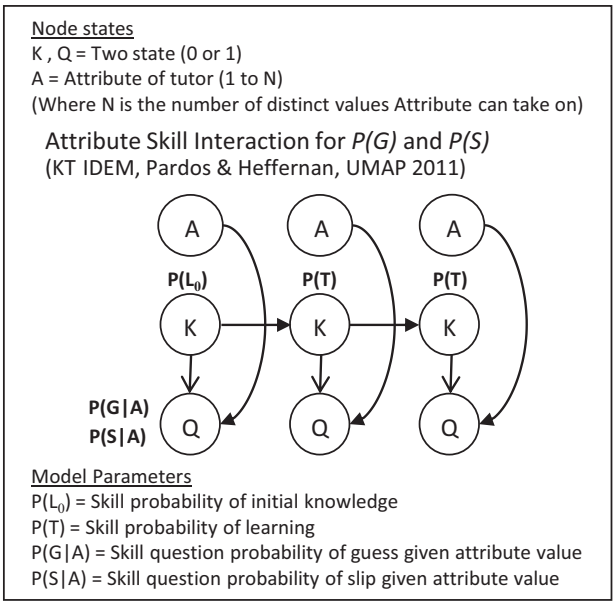

Figure 6. Bayesian network of the Knowledge Tracing Item Difficulty Effect Model (KT-IDEM), showing the conditional dependence of *P(G)* and *P(S)* on Attribute.

In this model, the question node is conditionally dependent on the attribute value which changes at each opportunity and is representing the different Pythagorean theorem questions from our dataset example. Applying this model has shown to significantly benefit skill-builder problem sets (randomized) in the ASSISTments Platform as well as linear sequence Cognitive Tutor for Algebra except for skills in which very small amounts of data per problem exist to train the individual guess and slip parameters (Pardos & Heffernan 2011b). When greater than 6 data points existed per problem on average, the KT-IDEM model outperformed regular KT.

While this example describes individualizing question guess and slip based on question ID, any other attribute, such as answer field type (multiple-choice or fill in the blank, for example), could take its place as an attribute.

Now that the difficulty (or information gain) of each question is controlled for, the endeavor of measuring the learning effect of each question can be taken on. The *P(T)* parameter in Knowledge Tracing is the probability of learning between each opportunity. Imagine if instead of a constant *P(T)* at every opportunity, the probability of learning between opportunities was dependent upon which Pythagorean theorem question was just viewed. Since the questions also provide different tutoring, a difference in learning could be expected between them. The application of this intuition is shown in the model in Figure 7 (Pardos & Heffernan 2011).

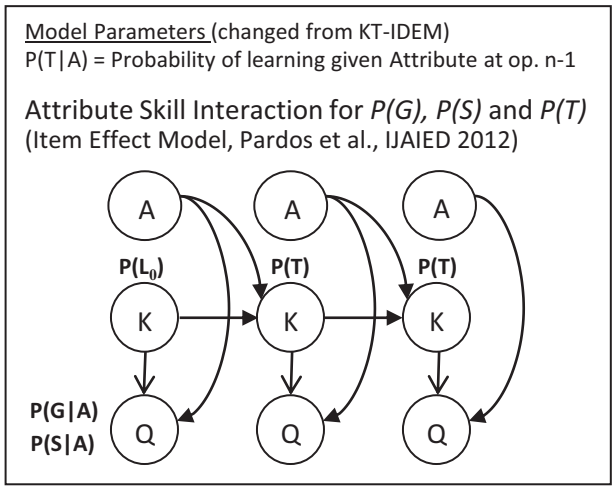

Figure 7. Bayesian network of the Item Effect Model showing the condition dependence of *P(T)* on 'A' at n-1.

Figure 7 shows the slight modification of making the *P(T)*, at opportunity *n,* conditionally dependent upon the attribute value at opportunity *n 1*. Using the example of the three questions as attribute values, this model captures the learning rate attributed to each question (and its tutoring). Relative question learning rate information can bring content with low learning value to the attention of content creators to either revise or replace. It also allows researchers to evaluate what aspects of the tutor are promoting student learning so that these aspects, such as effective pedagogy and content ordering, can be replicated.

Like the KT-IDEM model, this model is not limited to using question ID as the attribute values. In the question example, the tutorial help types of scaffold and hint could be the attribute values as was done in Pardos, Dailey & Heffernan (2012) where this model was used to evaluate the effectiveness of different tutorial strategies across different skill-builder problem sets. A learning gain analysis was also run on the data and the Bayesian model's tutorial strategy learning rates correlated with the learning gains in 10 of the 11 problem- sets. Further research using in vivo experiment data to validate against is ongoing.

Conclusion

In this paper we have overviewed techniques for individual student and tutor parameter incorporation into the Bayesian Knowledge Tracing Model and summarized work of ours that has demonstrated some of the potential in this approach. The Bayesian formulation of student and tutor modeling appears to be an elegant one for representing different hypothesis of how learning may or may not be taking place in the tutor.

Acknowledgments

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 NSF CAREER grant, award number REC0448319. We also acknowledge the many additional funders of the ASSISTments Platform found here: http://www.webcitation.org/5ym157Yfr

References

Anderson, J. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum Associates

Atkinson, R. C., Paulson, J. A. An approach to the psychology of instruction.*Psychological Bulletin*, 1972, 78, 49 61.

Beck, J.E., Chang, K.M. (2007) Identifiability: A Fundamental Problem of Student Modeling. In *User Modeling 2007*. LNCS, vol. 4511/2009, pp. 137 146. Springer Berlin.

Corbett, A. T., & Anderson, J. R. (1995), Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User Adapted Interaction 4(4)*, 253 278.

Desmarais, M.C. and Baker, R. (2011). A Review of Recent Advances in Learner and Skill Modeling in Intelligent Learning Environments. *User Modeling and User Adaptive Personalization*, *21*, (to appear)

Martin, B., Koedinger, K., Mitrovic, T., & Mathan, S. (2005). On Using Learning Curves to Evaluate ITS. *Proceedings of the Twelfth International Conference on Artificial Intelligence in Education*, pp. 419 426,Amsterdam

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

Pardos, Z. & Heffernan, N. (2010b) Navigating the parameter space of Bayesian Knowledge Tracing models. In Baker, R.S.J.d. et al. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*. pp. 161 170.

Pardos, Z.A., Dailey, M. & Heffernan, N. (2011), Learning what works in ITS from non traditional randomized controlled trial data. *The International Journal of Artificial Intelligence in Education*.

Pardos, Z. & Heffernan, N. (2011b) KT IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Konstant et al. (Eds.) *19th International Conference on User Modeling, Adaptation and Personalization* (UMAP 2011). Pages 243 254.

Pardos, Z.A., Gowda, S. M., Baker, R. S.J.D., Heffernan, N. T. (2012), The Sum is Greater than the Parts: Ensembling Models of Student Knowledge in Educational Software. To appear in *ACM's Knowledge Discovery and Data Mining Explorations, 13(2)*

Pardos, Z.A., Heffernan, N. T. (In press), Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. To appear in the Journal of Machine Learning Research W & CP..

Reye, J. (2004). Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education: Vol. 14*, 63 96