

# Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm

Zachary A. Pardos<sup>1</sup> and Neil T. Heffernan  
{zpardos,nth}@wpi.edu  
Worcester Polytechnic Institute

**Abstract.** Bayesian Knowledge Tracing (KT) models are employed by the cognitive tutors in order to determine student knowledge based on four parameters: learn rate, prior, guess and slip. A commonly used algorithm for learning these parameter values from data is the Expectation Maximization (EM) algorithm. Past work, however, has suggested that with four free parameters the standard KT model is prone to converging to erroneous degenerate states depending on the initial values of these four parameters. In this work we simulate data from a model with known parameter values and then run a grid search over the parameter initialization space of KT to map out which initial values lead to erroneous learned parameters. Through analysis of convergence and error surface visualizations we found that the initial parameter values leading to a degenerate state are not scattered randomly throughout the parameter space but instead exist on a surface with predictable boundaries. A recently introduced extension to KT that individualizes the prior parameter is also explored and compared to standard KT with regard to parameter convergence. We found that the individualization model has unique properties which allow it to avoid the local maxima problem.

## 1 Introduction

Knowledge Tracing (KT) models [1] are employed by the cognitive tutors [2], used by over 500,000 students, in order to determine when a student has acquired the knowledge being taught. The KT model is based on two knowledge parameters: learn rate and prior and two performance parameters: guess and slip. A commonly used algorithm for learning these parameter values from data is the Expectation Maximization (EM) algorithm. Past work [3,4,5], however, has suggested that with four free parameters the standard KT model is prone to converging to erroneous degenerate states depending on the initialized values of these four parameters. In this work we simulate data from a model with known parameter values and then run a grid search over the parameter initialization space of KT to map out which initial values lead to erroneous learned parameters. Through analysis of convergence and error surface visualizations we found that the initial parameter values leading to a degenerate state are not scattered randomly throughout the parameter space but instead exist on a surface within predictable boundaries. A recently introduced extension to KT that individualizes the prior parameter is also explored and compared to standard KT with regard to parameter convergence. We found that the individualization model has unique properties which allow for a greater number of initial states to converge to the true parameter values.

---

<sup>1</sup> National Science Foundation funded GK-12 Fellow

## ***1.1 Expectation Maximization algorithm***

The Expectation Maximization (EM) algorithm is a commonly used algorithm used for learning the parameters of a model from data. EM can learn parameters from incomplete data as well as from a model with unobserved nodes such as the KT model. In the cognitive tutors, EM is used to learn the KT prior, learn rate, guess and slip parameters for each skill, or production rule. One requirement of the EM parameter learning procedure is that initial values for the parameters be specified. With each iteration the EM algorithm will try to find parameters that improve fit to the data by maximizing the log likelihood function, a measure of model fit. There are two conditions that determine when EM stops its search and returns learned parameter results: 1) if the specified maximum number of iterations is exceeded or 2) if the difference in log likelihood between iterations is less than a specified threshold. Meeting condition 2, given a low enough threshold, is indicative of algorithm parameter convergence, however, given a low enough threshold, EM will continue to try to maximize log likelihood, learning the parameters to a greater precision. In our work we use a threshold value of  $1e-4$ , which is the default for the software package used, and a maximum iteration count of 15. The max iteration value used is lower than typical, however, we found that in the average case our EM runs did not exceed more than 7 iterations before reaching the convergence threshold. The value of 15 was chosen to limit the maximum computation time since our methodology requires that EM be run thousands of times in order to achieve our goal.

## ***1.2 Past work in the area of KT parameter learning***

Beck & Chang [3] explained that multiple sets of KT parameters could lead to identical predictions of student performance. One set of parameters was described as the plausible set, or the set that was in line with the authors' knowledge of the domain. The other set was described as the degenerate set, or the set with implausible values such as values that specify that a student is more likely to get a question wrong if they know the skill. The author's proposed solution was to use a Dirichlet distribution to constrain the values of the parameters based on knowledge of the domain.

Corbett & Anderson's [1] approach to the problem of implausible learned parameters was to impose a maximum value that the learned parameters could reach, such as a maximum guess limit of 0.30 which was used in Corbett & Anderson's original parameter fitting code. This method of constraining parameters is still being employed by researchers such as Baker et al. [4] and Ritter et al [5] in their more recent models.

Alternatives to EM for fitting parameters were explored by Pavlik et al. [5], such as using unpublished code by Baker to brute force parameters that minimize an error function. Pavlik also introduced an alternative to KT, named PFA [5] and reported an increase in performance compared to the KT results. Gong, Beck and Heffernan [6] however are in the process of challenging PFA by using KT with EM which they report provides improved prediction performance over PFA with their dataset.

While past works have made strides in learning plausible parameters they lack the benefit of knowing the true model parameters of their data. Because of this, none of past work

has been able to report the accuracy of their learned parameters. One of the contributions of our work is to provide a closer look at the behavior and accuracy of EM in fitting KT models by using synthesized data that comes from a known set of parameter values. This enables us to analyze the learned parameters in terms of exact error instead of just plausibility. To our knowledge this is something that has not been previously attempted.

## 2 Methodology

Our methodology involves first synthesizing response data from a model with a known set of parameter values. After creating the synthesized dataset we can then train a KT model with EM using different initial parameter values and then measure how far from the true values the learned values are. This section describes the details of this procedure.

### 2.1 Synthesized dataset procedure

To synthesize a dataset with known parameter values we run a simulation to generate student responses based on those known ground truth parameter values. These values will later be compared to the values that EM learns from the synthesized data. To generate the synthetic student data we defined a KT model using functions from MATLAB's Bayes Net Toolbox (BNT) [7]. We set the known parameters of the KT model based on the mean values learned across skills in a web based math tutor called ASSISTments [9]. These values which represent the ground truth parameters are shown in Table 1.

**Table 1.** Ground truth parameters used for student simulation

Prior	Learn rate	Guess	Slip
Uniform random dist	0.09	0.14	0.09

Since knowledge is modeled dichotomously, as either learned or unlearned, the prior represents the Bayesian network's confidence that a student is in the learned state. The simulation procedure makes the assumption that confidence of prior knowledge is evenly distributed. 100 users and four question opportunities are simulated, representing a problem set of length four. Each doubling of the number of users also doubles the EM computation time. We found that 100 users were sufficient to achieve parameter convergence with the simulated data. Figure 1 shows pseudo code of the simulation.

```

KTmodel.lrate = 0.09
KTmodel.guess = 0.14
KTmodel.slip = 0.09
KTmodel.num_questions = 4
For user 1 to 100
    prior(user) = rand()
    KTmodel.prior = prior(user)
    sim_responses(user) = sample.KTmodel
End For

```

**Figure 1.** Pseudo code for generating synthetic student data from known KT parameter values

Student responses are generated probabilistically based on the parameter values. For instance, the Bayesian network will roll a die to determine if a student is in the learned state based on the student’s prior and the learn rate. The network will then again role a die based on guess and slip and learned state to determine if the student answers a question correct or incorrect at that opportunity. After the simulation procedure is finished, the end result is a datafile consisting of 100 rows, one for each user, and five columns; user id followed by the four incorrect/correct responses for each user.

## 2.2 Analysis procedure

With the dataset now generated, the next step was to start EM at different initial parameter values and observe how far the learned values were from the true values. A feature of BNT is the ability to specify which parameters are fixed and which EM should try to learn. In order to gain some intuition on the behavior of EM we decided to start simple by fixing the prior and learn rate parameters to their true values and focusing on learning the guess and slip parameters only. An example of one EM run and calculation of mean absolute error is shown in the table below.

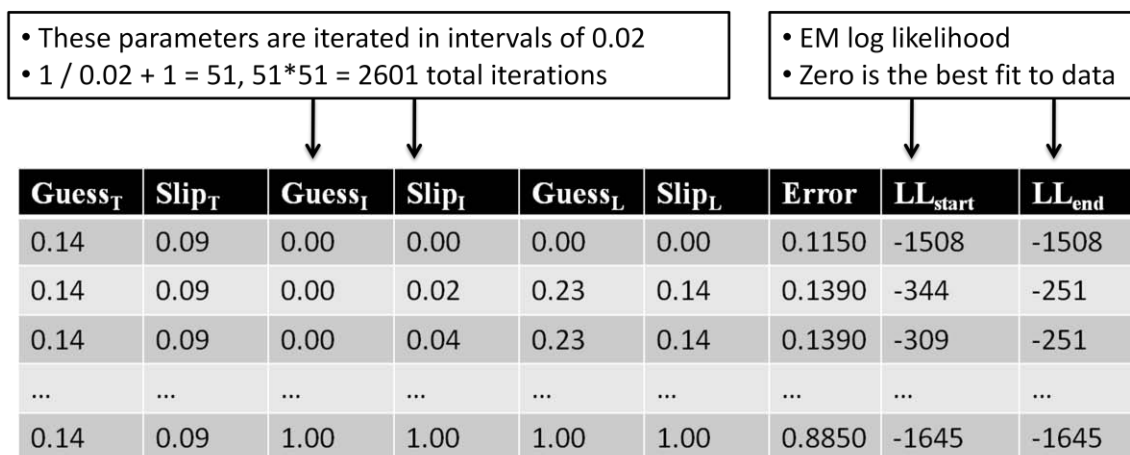
**Table 3.** Example run of EM learning the Guess and Slip parameters of the KT model

Parameter	True value	EM initial value	EM learned value
<b>Guess</b>	0.14	0.36	0.23
<b>Slip</b>	0.09	0.40	0.11
<b>Error</b> = $[\text{abs}(\text{Guess}_{\text{True}} - \text{Guess}_{\text{Learned}}) + \text{abs}(\text{Slip}_{\text{True}} - \text{Slip}_{\text{Learned}})] / 2$ = 0.11			

The true prior parameter value was set to the mean of the simulated priors (In our simulated dataset of 100 the mean prior was 0.49). Having only two free parameters allows us to represent the parameter space in a two dimensional graph with guess representing the X axis value and slip representing the Y axis value. After this exploration of the 2D guess/slip space we will explore the more complex four free parameter space.

### 2.2.1 Grid search mapping of the EM initial parameter convergence space

One of the research questions we wanted to answer was if the initial EM values leading to a degenerate state are scattered randomly throughout the parameter space or if they exist within a defined surface or boundary. If the degenerate initial values are scattered randomly through the space then EM may not be a reliable method for fitting KT models. If the degenerate states are confined to a predictable boundary then true parameter convergence can be achieved by restricting initial parameter values to within a certain boundary. In order to map out the convergence of each initial parameter we iterated over the entire initial guess/slip parameter space with a 0.02 interval. Figure 2 shows how this grid search exploration of the space was conducted.



**Figure 3.** Output of the grid search procedure exploring the initial EM guess/slip parameter space of KT

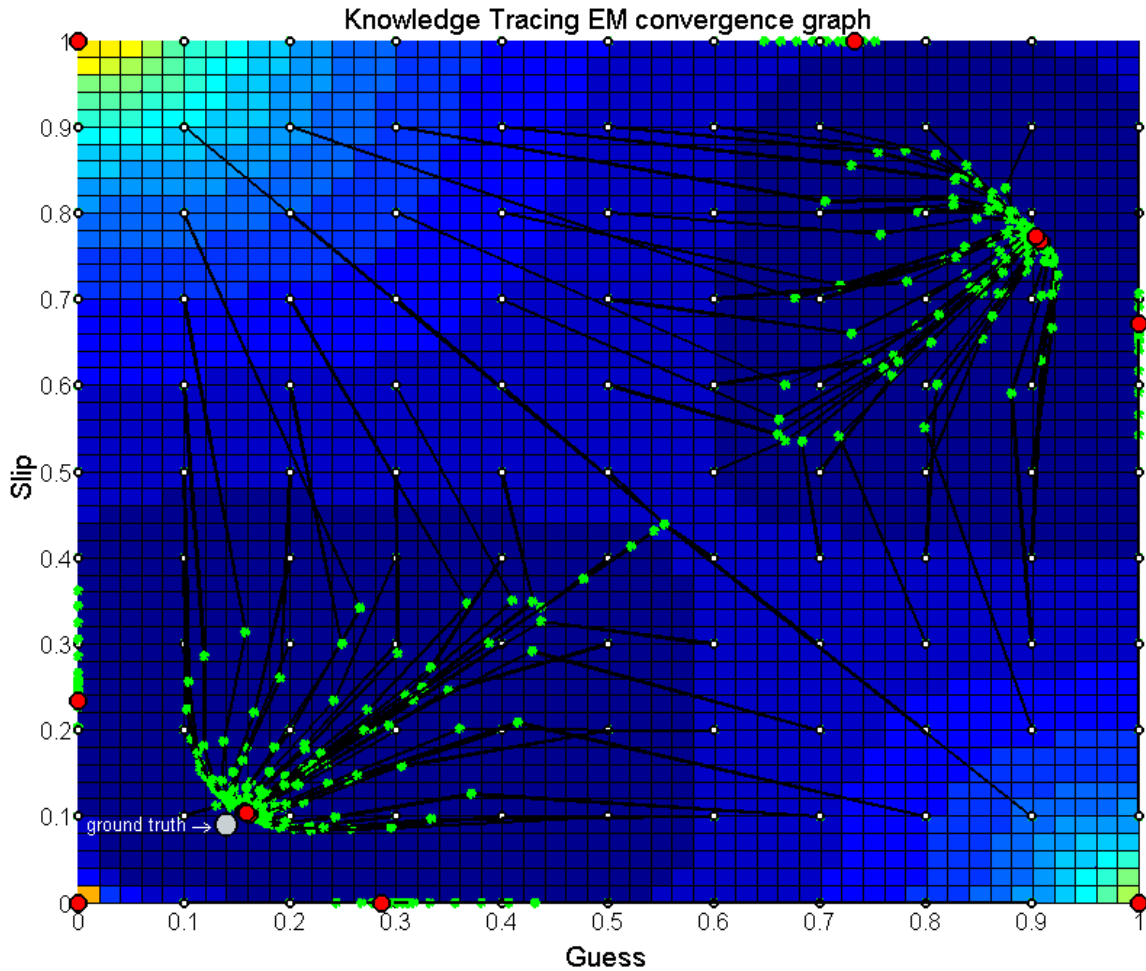
We started with an initial guess and slip of 0 and ran EM to learn the guess and slip values of our synthesized dataset. When an EM run finished, either because it reached the convergence threshold or the maximum iteration, it returned the learned guess and slip values as well as the log likelihood fit to the data of the initial parameters and the learned parameters (represented by  $LL_{start}$  and  $LL_{end}$  in the figure). We calculated the mean absolute error between the learned and true values using the formula in Table 3. We then increased the initial slip value by 0.02 and ran EM again and repeated this procedure for every guess and slip value from 0 to 1 with an interval of 0.02.

### 3 Results

The analysis procedure produced an error and log likelihood value for each guess/slip pair in the parameter space. This allowed for visualization of the parameter space using  $Guess_{initial}$  as the  $X$  coordinate,  $Slip_{initial}$  as the  $Y$  coordinate and either log likelihood or mean absolute error as the error function.

#### 3.1 Tracing EM iterations across the KT log likelihood space

The calculation of error is made possible only by knowing the true parameters that generated the synthesized dataset. EM does not have access to these true parameters but instead must use log likelihood to guide its search. In order to view the model fit surface and how EM traverses across it from a variety of initial positions, we set the  $Z$ -coordinate (background color) to the  $LL_{start}$  value and logged the parameter values learned at each iteration step of EM. We overlaid a plot of these EM iteration step points on the graph of model fit. This combined graph is shown below in figure 4 which depicts the nature of EM's convergence with KT. For the EM iteration plot we tracked the convergence of EM starting positions in 0.10 intervals to reduce clutter instead of 0.02 intervals which were used to create the model fit plot. No EM runs reached their iteration max for this visualization. Starting values of 0 or 1 (on the borders of the graph) do not converge from the borders because of how BNT fixes parameters with 0 or 1 as their initial value.



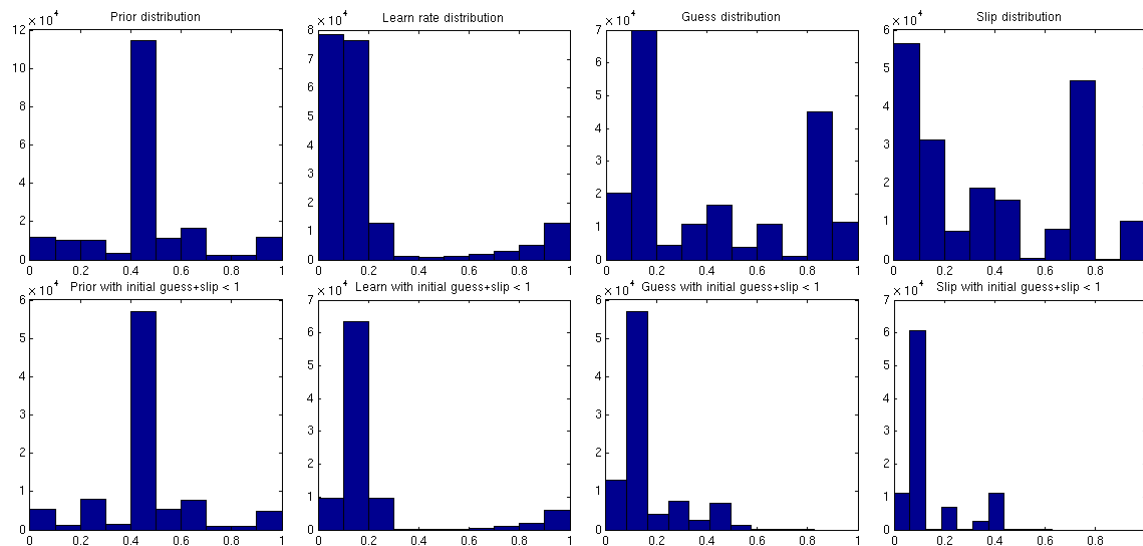
**Figure 4.** Model fit and EM iteration convergence graph of Bayesian Knowledge Tracing. Small white dots represent parameter starting values. Green dots represent the parameter values at each EM iteration. The red dots represent the resulting learned parameter values and the large white dot is ground truth. The background color is the log likelihood ( $LL_{start}$ ) of the parameter space. Dark blue represent better fit.

This visualization depicts the multiple global maxima problem of Knowledge Tracing. There are two distinct regions of best fit (dark blue); one existing in the lower left quadrant which contains the true parameter values (indicated by the white “ground truth” dot), the other existing in the upper right quadrant representing the degenerate learned values. We can observe that all the green dots lie within one of the two global maxima regions, indicating that EM makes a jump to an area of good fit after the first iteration. The graph shows that there are two primary points that EM converges to with this dataset; one centered around guess/slip = 0.15/0.10, the other around 0.89/0.76. We can also observe that initial parameter values that satisfy the equation:  $guess + slip \leq 1$ , such as guess/slip = 0.90/0.10 and 0.50/0.50, successfully converge to the true parameter area while initial values that satisfy:  $guess + slip > 1$ , converge to the degenerate point.

### 3.2 *KT convergence when learning all four parameters*

For the full four parameter case we iterated through initial values of the prior, learn rate, guess and slip parameters from 0 to 1 with a 0.05 interval. This totaled 194,481 EM runs

( $21^4$ ) to traverse the entire parameter space. For each set of initial positions we logged the converged learned parameter values. In order to evaluate this data we looked at the distribution of converged values for each parameter across all EM runs.

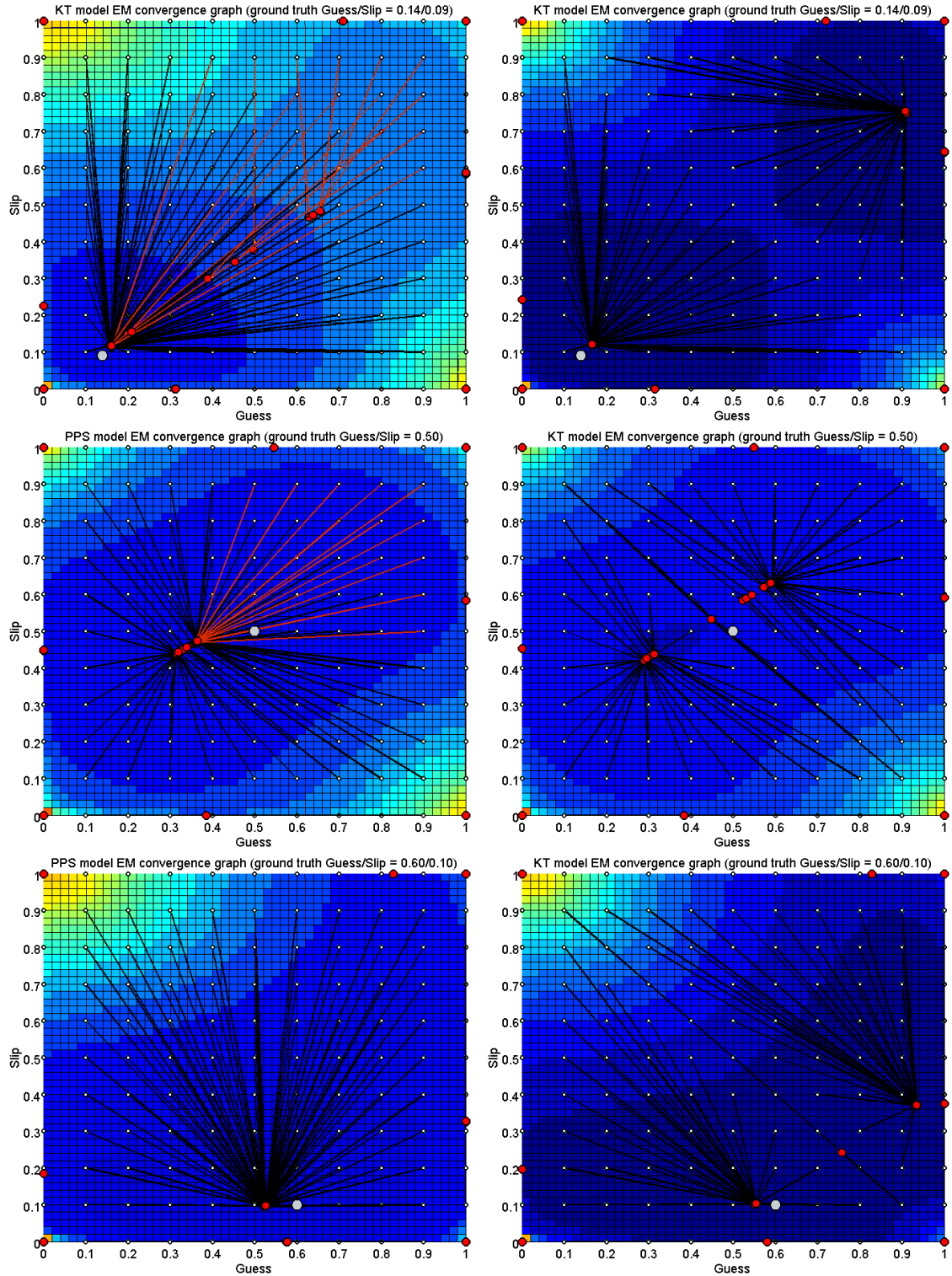


**Figure 5.** Histograms showing the distribution of learned parameter values for each of the four Knowledge Tracing parameters. The first row shows the parameter distributions across all the EM runs. The second row shows the parameter distributions for the EM runs where initial guess and slip summed to less than 1.

The first row of histograms in Figure 5 shows the distribution of learned parameter values across all EM runs. Generally, we can observe that all parameters have multiple points of convergence; however, each histogram shows a clear single or bi-modal distribution. The prior and learn rate appear to be the parameters that are easiest to learn since the majority of EM runs lead to values near their true values. The guess and slip histograms exhibit more of the bi-modal behavior seen in the two parameter case, with points of convergence at opposite ends of the parameter space. In the two parameter case, initial guess and slip values that summed to less than one converged towards the ground truth coordinate. To see if this trend generalized with four free parameters we generated another set of histograms but only included EM runs where the initial guess and slip parameters summed to less than one. These histograms are shown in the second row.

### 3.3 Evaluating an extension to KT called the Prior Per Student model

We evaluated a model [9], recently introduced by the authors, that allows for individualization of the prior parameter. By only modeling a single prior, Knowledge tracing makes the assumption that all students have the same level of knowledge of a particular skill before using the tutor. The Prior Per Student (PPS) model challenges that assumption by allowing each student to have a separate prior while keeping the learn, guess and slip as parameters of the skill. The individualization is modeled completely within a Bayesian model and is accomplished with the addition of just a single node, representing student id, and a single arc, connecting the student node to the first opportunity knowledge node. We evaluated this model using the two-parameter case, where guess and slip are learned and learn rate and prior are fixed to their true values.



**Figure 6.** EM convergence graphs of the Prior Per Student (PPS) model (left) and KT model (right). Results are shown with ground truth datasets with guess/slip of 0.14/0.09, 0.50/0.50 and 0.60/0.10

The KT models, in the right column of figure 6, all show multiple points of convergence with only one of the points near the ground truth coordinate (white dot). Unlike KT, the



PPS models, in the left column, have a single point of convergence regardless of the starting position and that single point is near the ground truth value. The red lines in two of the PPS models indicate that the maximum iteration count was reached. In the 0.14/0.09 model it appears that PPS with starting parameters in the upper right region were converging towards the true values but hit the max iteration count before arriving. The PPS model was shown [9] to provide improved prediction over standard knowledge tracing with real world datasets. The visualizations shown in figure 6 suggest that this improved prediction accuracy is likely due in part to the PPS model's improved parameter learning accuracy from a wider variety of initial parameter locations.

In the case of the PPS models show above there were as many prior parameters as there were students and these parameters were all set to the values that were generated for each simulated student as seen in the line "KTmodel.prior = prior(user)" in figure 1. Accurately inferring many initial prior values would be difficult in practice; however, a heuristic is described in Pardos et al [9] that seeds each individual prior based on the student's first response. Applying this same heuristic to our synthesized dataset with ground truth guess/slip values of 0.14/0.09 we found that all points converged to the true parameter location without EM reaching its maximum iteration count. This performance suggests that single point convergence to the true parameters is possible with the PPS model without the benefit of individual student prior knowledge estimates. A more detailed description and analysis of this technique is in work that is in preparation.

## 4 Discussion and Future Work

An argument can be made that if two sets of parameters fit the data equally well then it makes no difference if the parameters used are the true parameters. This is true when prediction of responses is the only goal. However, when inferences about knowledge and learning are being made, parameter plausibility and accuracy is crucial. It is therefore important to understand how our student models and fitting procedures behave if we are to draw valid conclusions from them. In this work we have depicted how KT exhibits multi-modal convergence properties due to its multi-modal log likelihood parameter space. We demonstrated that with our simulated dataset, choosing initial guess and slip values that summed to less than one allowed for convergence towards the ground truth values in the two parameter case and in the four parameter case, applying this same rule resulted in a convergence distribution with a single mode close to the ground truth value. Lastly, we found that use of the Prior Per Student model eliminated the multiple maxima dilemma in the two parameter case for our synthesized datasets and use of a prior seeding heuristic for PPS resulted in performance comparable to having perfect knowledge of the individual prior confidence probabilities.

This research raises a number of questions such as how KT models behave with a different assumption about the distribution of prior knowledge. What is the effect of increased number of students or increased number of question responses per student on parameter learning accuracy? How does PPS converge with four parameters and what does the model fit parameter convergence space of real world datasets look like? These are questions that are still left to be explored by the EDM community.

## Acknowledgements

We would like to thank all of the people associated with creating ASSISTments listed at [www.ASSISTments.org](http://www.ASSISTments.org). We would also like to acknowledge funding from the US Department of Education, the National Science Foundation, the Office of Naval Research and the Spencer Foundation. All of the opinions expressed in this paper are those of the authors and do not necessarily reflect the views of our funders.

## References

- [1] Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4, 253–278.
- [2] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30–43.
- [3] Beck, J. E. and Chang, K. M. Identifiability: A fundamental problem of student modeling. In *Proceedings of the 11th International Conference on User Modeling*, 2007, pp. 137-146.
- [4] Baker, R.S.J.d., Corbett, A.T., Alevan, V. (2008) 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.
- [5] Pavlik, P.I., Cen, H., Koedinger, K.R. (2009). Performance Factors Analysis - A New Alternative to Knowledge Tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*. Brighton, UK, 531-538
- [6] Gong, Y, Beck, J. E., Heffernan, N. T. In Press (2010) Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting. In *Proc. of The 10th International Conference on Intelligent Tutoring Systems*, Pittsburgh.
- [7] Kevin Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33, 2001.
- [9] Pardos, Z. A., Heffernan, N. T. In Press (2010) Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In *Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, Hawaii.
- [10] Ritter, S., Harris, T., Nixon, T., Dickison, D., Murray, C., Towle, B. (2009) Reducing the knowledge tracing space. In Barnes, Desmarais, Romero, & Ventura (Eds.). In *Proceedings of the 2nd International Conference on Educational Data Mining*. pp. 151-160. Cordoba, Spain.