Pardos, Z. A., Heffernan, N. T. Accepted (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*. Pittsburg.

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

Zachary A. Pardos[1] 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 initialized values of these four parameters. In this work we simulate data from a model with known parameter values and then brute forced the parameter initialization space of KT to map out which initial values lead to erroneous learned parameters. Through analysis of multi-dimensional visualizations we found that the initial parameter values leading to a degenerate state are not scattered randomly throughput 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 for a more informed selection of initial parameters.

## 1 Introduction

Knowledge Tracing (KT) [1] models are employed by the cognitive tutors [2], used by hundreds of thousands of 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 brute force the parameter initialization space of KT to map out which initial values lead to erroneous learned parameters. Through analysis of multi-dimensional visualizations we found that the initial parameter values leading to a degenerate state are not scattered randomly throughput 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 model parameters and for a more informed selection of those initial states.

[1] 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 for 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 must be specified to give EM starting positions to begin its search. With each iteration the EM algorithm will try to find parameters that better fit the data by maximizing the log likelihood function, a measure of model fit. The two conditions that determine when EM stops and returns learned parameter results are: 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 by 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 packaged used, and a maximum iteration value of 15. The max iteration value used is a 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 a slip rate over 0.50 which asserts that a student is more likely to get an item 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 of 0.30 that 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] in their more recent models.

Alternatives to EM for fitting parameters was explored by Pavlik et al [5], such as using unpublished code by Baker to brute force parameters that maximize log likelihood. Pavlik also evaluated an alternative to KT in the same work [5] and reported an increase in performance compared to the KT results. Gong, Beck & 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, the assumption of the range of the true parameters has to be based on domain knowledge and a more in depth study of parameter learning behavior and accuracy is not possible. 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 not just analyze the learned parameters in terms of plausibility, but also in terms of exact Error of the learned parameters from the ground truth parameter values.

## 2 Methodology

Our methodology involves first synthesizing response data from a known set of parameters and then letting EM try to learn the true parameters from the data using difference initial parameter values. This section describes the details of this proceedure.

### *2.1 Synthesized dataset procedure*

To synthesize a dataset with known parameter values we run a simulation to generate a 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 then set the parameters of the KT model to average values learned across skills in a web based math tutor called ASSISTment [8]. These values which represent the ground truth parameter values are shown in Table 1.

**Table 2.** Ground truth parameters used for student simulation

| $P(L_0)$ | P(T) | P(guess) | P(slip) |
|---|---|---|---|
| Uniform random dist | 0.09 | 0.14 | 0.09 |

Since knowledge is modeled dichotomously, as either learned or unleared, 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 across students. One hundred 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. Su-do code of the simulation procedure is shown bellow.

```
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.** Su-do 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 role a die to determine if a student is in the learned state based on the student's prior and the P(T). 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 are 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 error is shown in the table below.

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

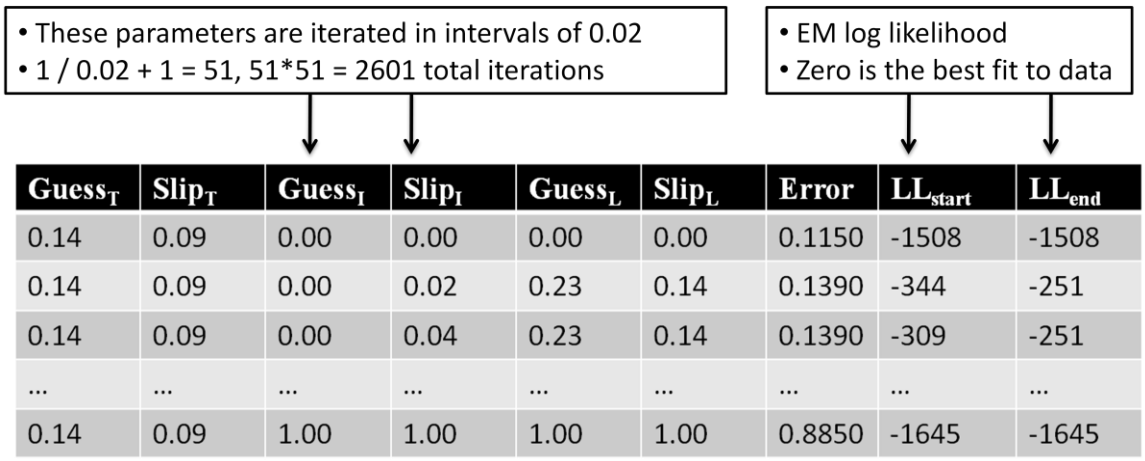
| Parameter | True value | EM initial value | EM learned value |
|---|---|---|---|
| **Guess** | 0.14 | 0.36 | 0.23 |
| **Slip** | 0.09 | 0.40 | 0.11 |
| **Error** = [abs($Guess_{True}$ – $Guess_{Learned}$) + abs($Slip_{True}$ – $Slip_{Learned}$)] / 2 = 0.11 | | | |

The true prior parameter value was set to the mean of the simulated priors (In our simulated dataset of 100 this mean prior was 0.49). Having only two free parameters allows us to represent the parameter space as a two dimensional space 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 move on to the more complex three and four free parameter space.

### 2.2.1 Brute force mapping of the EM 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 brute force exploration of the space was conducted.




| $Guess_T$ | $Slip_T$ | $Guess_I$ | $Slip_I$ | $Guess_L$ | $Slip_L$ | Error | $LL_{start}$ | $LL_{end}$ |
|---|---|---|---|---|---|---|---|---|
| 0.14 | 0.09 | 0.00 | 0.00 | 0.00 | 0.00 | 0.1150 | -1508 | -1508 |
| 0.14 | 0.09 | 0.00 | 0.02 | 0.23 | 0.14 | 0.1390 | -344 | -251 |
| 0.14 | 0.09 | 0.00 | 0.04 | 0.23 | 0.14 | 0.1390 | -309 | -251 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 0.14 | 0.09 | 1.00 | 1.00 | 1.00 | 1.00 | 0.8850 | -1645 | -1645 |

**Figure 3.** Output file of the brute force procedure mapping the EM guess/slip convergence space

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 EM is finished, either because it reached the convergence threshold or the maximum iteration, it returns 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 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 Error as the *Z* coordinate.

## *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 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 bellow which depicts the nature of EM convergence with KT.

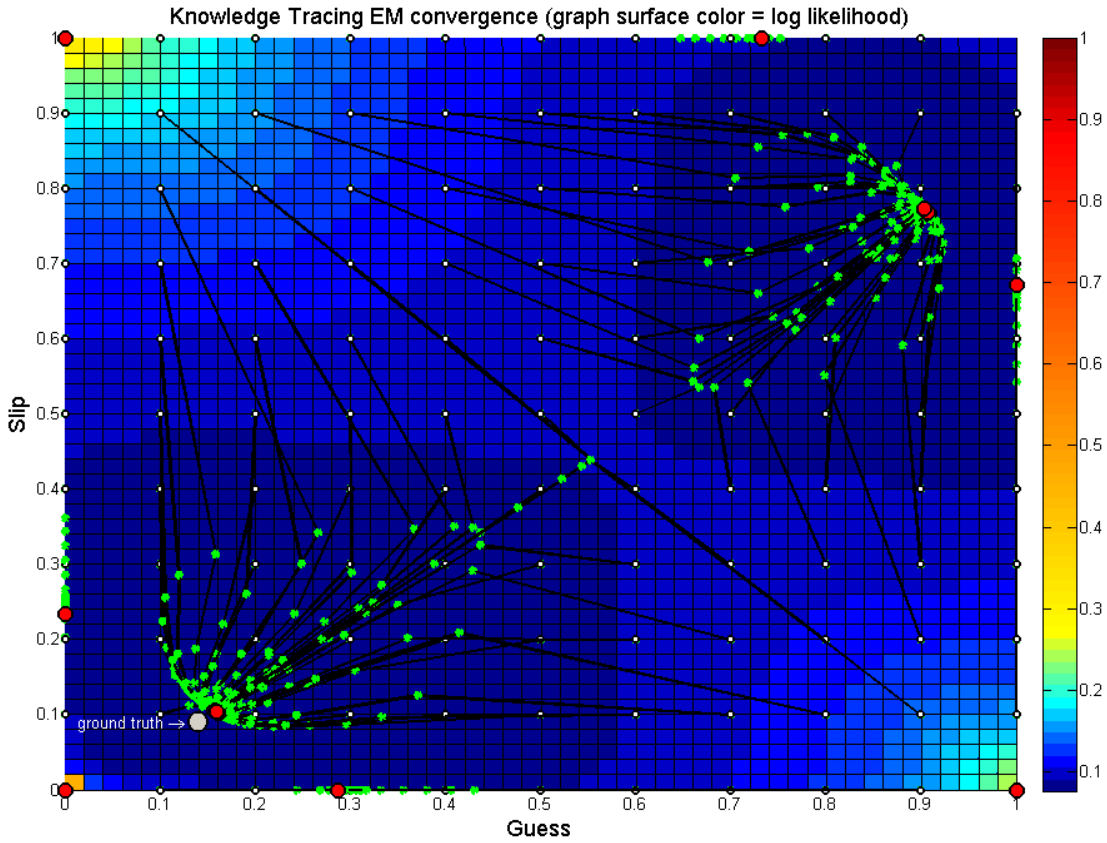



**Figure 4.** Model fit and EM iteration convergence graph of Bayesian Knowledge Tracing

The background surface color of Figure 4 shows the log likelihood ($LL_{start}$) of each guess/slip coordinate. The darker blue regions, with values closer to 0 (see color bar to the right of the figure), represent areas of better fit to the data. This surface visualization clearly depicts the dual global maxima problem of Knowledge Tracing. There are two regions of best fit; 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. The small white dots are starting positions of EM. The green dots are parameter values at each subsequent iteration of EM from the starting value. 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 red dots in the graph are the final learned guess/slip coordinates for each EM run. The graph shows that there are two primary points that EM converges to; 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 <= 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. For the EM iteration plot we tracked the convergence of EM starting positions in 0.10 intervals instead of 0.02 to reduce the clutter created by excessive plot lines.

## *3.2 KT convergence error with 3 free parameters*

In section 2.2 we described how learn rate and prior would be fixed to first get comfortable with how EM behaved with only two free parameters. In this section we now look at how EM converges with only the prior parameter fixed (still at 0.49) and now with the learn rate, guess and slip parameters free. The new error function is now:

**Error** =

$[abs(Learn_{True} - Learn_{Learned}) + abs(Guess_{True} - Guess_{Learned}) + abs(Slip_{True} - Slip_{Learned})] / 3$

The result is largely a replication of the trend observed in the previous section, where guess and slip values that sum to 1 or less tend to converge to the true parameters where as those that sum to greater than 1 converge to a degenerate state. We found that the learn rate is more flexible in its starting position and that higher learn rates (over 0.50) still allow EM to converge to the true triad of parameter values as shown in the figure below. In this 3D volume graphs the surface color represents the resulting error of the learned parameters at the various guess, slip and learn rate initial parameter positions.

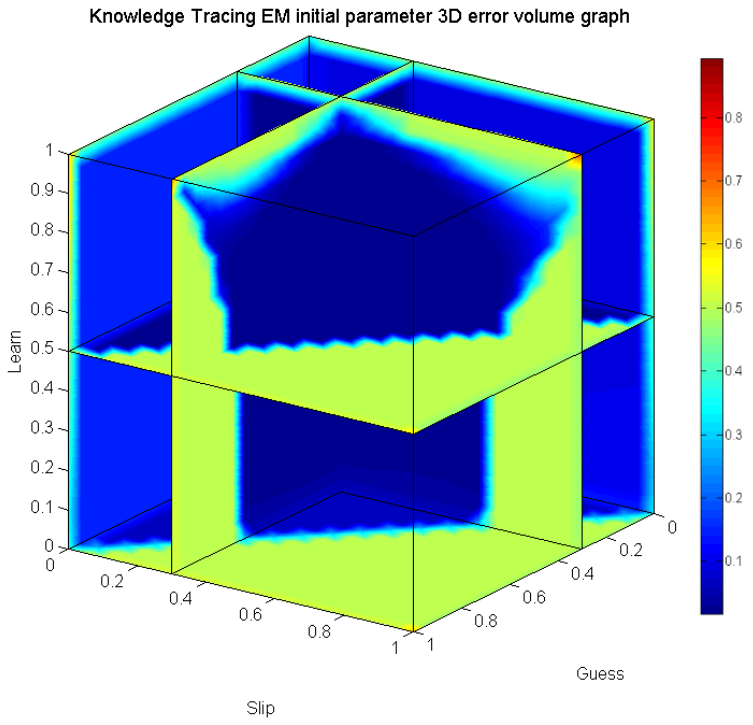



**Figure 5.** KT initial EM position 3D error volume graph with cross section (3 free parameters)

The primary observation from the 3D graph is that even with three free parameters, the error surface is still defined by clear cut boundaries that separate the initial values leading to the ground truth parameters and the initial values leading to the degenerate parameters.

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

We evaluated a recently introduced model [9] that allows for individualization of prior.

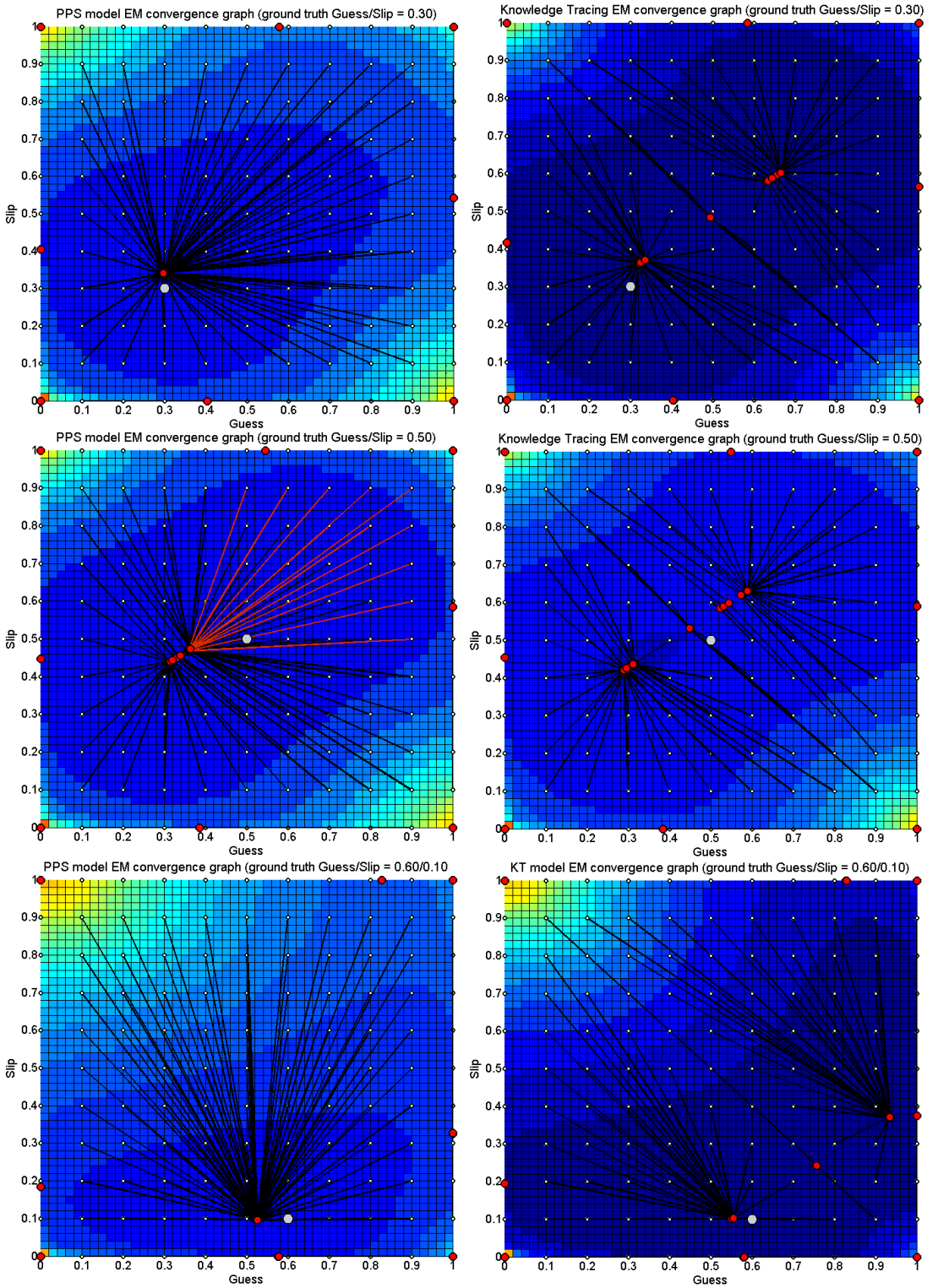



**Figure 6.** EM convergence graphs of the Prior Per Student (PPS) model (left) and KT model (right). Results are shown with ground truth datasets with guess/slip of 0.30/0.30, 0.50/0.50 and 0.60/0.10

The background surface color of Figure 6 represents the model fit space ($LL_{start}$). The small white dots represent initial EM parameter values and the red dots represent ending values. The large white dot represents the ground truth guess/slip value for the synthesized dataset being analyzed. The difference in format between this figure and Figure 4 is that these graphs do not show the iteration points between start and finish. We can observe from Figure 6 that the PPS model has a single point of convergence in all three ground truth parameter datasets while the KT model has three points of convergence. The graphs show that the PPS model will converge to near the ground truth guess/slip values regardless of the initial parameter values (with the exception of setting guess or slip to 0 or 1). In summary, the graphs show that the PPS model does not suffer from the multiple global maxima problem of KT.

The PPS model parameters were learned with fixed true prior values for each of the 100 simulated students, however, we found that the model performed just as well, and in some cases better, when using a cold start heuristic that did not depend on knowing any ground truth prior values. This cold start heuristic essentially specifies two priors, either 0.05 or 0.95. A student associated with one of those two priors depending on their first question response; students who answered incorrectly on question 1 were given the 0.05 prior, students who answered correctly were give the 0.95 prior. This is very encouraging performance since it suggests that single point convergence to the true parameters is possible with the PPS model without the benefit of comprehensive individual student prior knowledge estimates. We also tried synthesized datasets with ground truth learn rate values of 0.20 and 0.30 and guess/slip of 0.14/0.09. We found that these higher learn rates made very little difference in the resulting parameter convergence graphs with either model.

## 4 Discussion and Future Work

When predicting student performance is the goal, it can be argued that if two sets of parameters predict the data equally well the plausibility of the parameter set used does not matter. However, accuracy of the parameters learned becomes crucial when determining student motivation, emotional state or the effectiveness of tutorial feedback based on learned parameters. In these cases scientific conclusions and even pedagogical recommendation are being made based on the values of these parameters. It is therefore imperative for us as a field to understand how our models and fitting procedures behave if we are to confidently advance and expand the adoption of EDM models and techniques. In this work we have explored KT with visualizations to help gain this added understanding of a widely used model and shown how the same methodology can be used to compare and evaluate the behavior of new models.

This research also 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? Under what circumstances is standard Knowledge Tracing a better option? Our comparison of PPS and KT convergence suggested that PPS has a greater chance of converging to the true parameters. This is consistent with reported [9] improved real data prediction of PPS over KT but what does the model fit parameter

convergence space of real world datasets look like? This is an area that is left to be explored.

## 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. Proceedings of the 11th International Conference on User Modeling, 2007, pp. 137-146.

[4] Baker, R.S.J.d., Corbett, A.T., Aleven, V. (2008) More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. 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. Accepted (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.

[8] Pardos, Z. A., Heffernan, N. T., Ruiz, C. & Beck, J. (2008) Effective Skill Assessment Using Expectation Maximization in a Multi Network Temporal Bayesian Network. In Proc. of The Young Researchers Track at the 9th International Conference on Intelligent Tutoring Systems.

[9] Pardos, Z. A., Heffernan, N. T. Accepted (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.). Proceedings of the 2nd International Conference on Educational Data Mining. pp. 151-160. Cordoba, Spain.