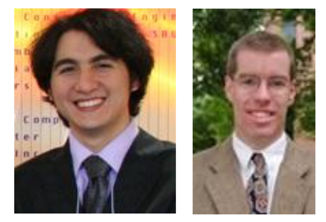
Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm



By Zachary A. Pardos and Neil T. Heffernan

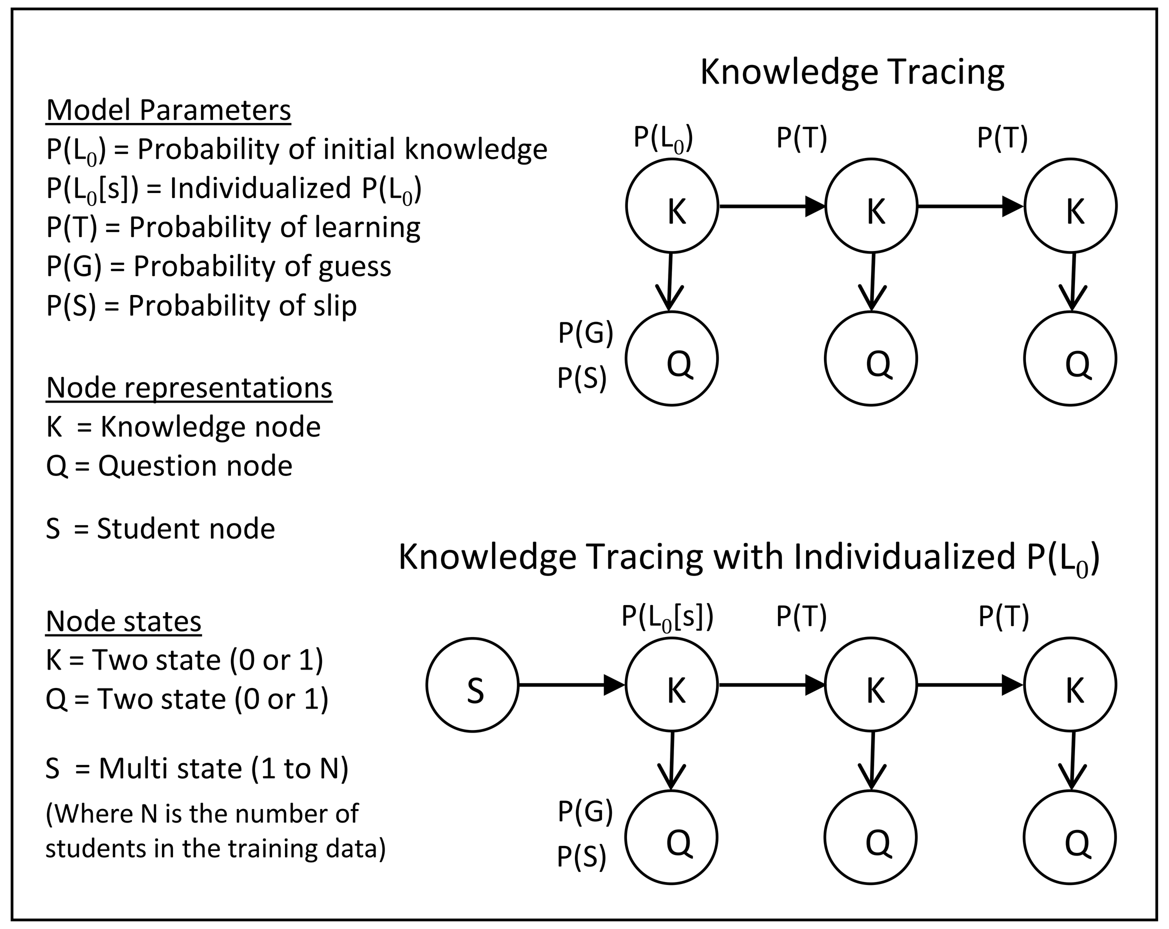
Sponsors

The Spencer Foundation

Goal

To provide a better understanding of the behavior and accuracy of the Expectation Maximization algorithm in fitting Knowledge Tracing models. By using synthesized data that comes from a known set of parameter values we are able to measure the exact error of the learned parameters from the ground truth parameter values and map out the parameter convergence space

Knowledge Tracing models



•Knowledge Tracing (KT) models are employed by the cognitive tutor intelligent tutoring system, used by hundreds of thousands of students. These models are used to infer when a student has acquired the knowledge being taught based on analysis of the student's incorrect and correct responses to the tutor. 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.

•A recently introduced extension to KT adds individualization to the prior parameter, allowing each student to have his/her own prior knowledge value in the model. **This new model is called the Prior Per Student model or PPS.** PPS* is a model that uses the first response of each student to set his/her prior. We sampled from the KT model with known parameters to create a synthesized dataset of 100 users each with 4 responses. The regular PPS model will use the hidden ground truth prior values.

Problem background

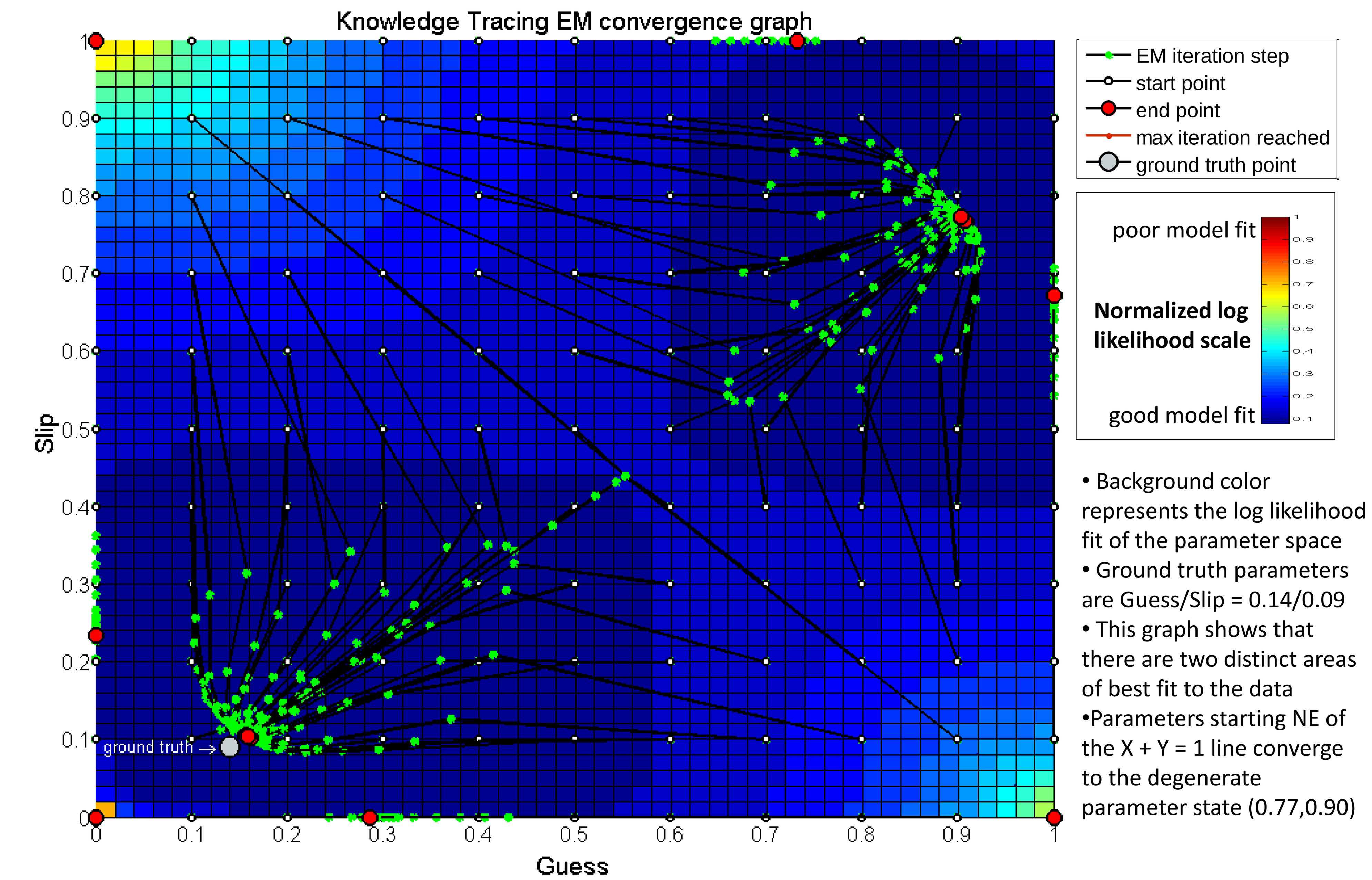
• Past work has suggested that the standard KT model is prone to converging to erroneous degenerate states depending on the initialized values of these four parameters. Beck & Chang explained this problem describing one set of learned parameters as the plausible set, or the set that was in line with the authors' knowledge of the domain and the other set as the degenerate set. A solution using domain knowledge to constrain the parameters was proposed.

•Corbett & Anderson's 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.

•Other works has suggested using brute force methods instead of EM

•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.

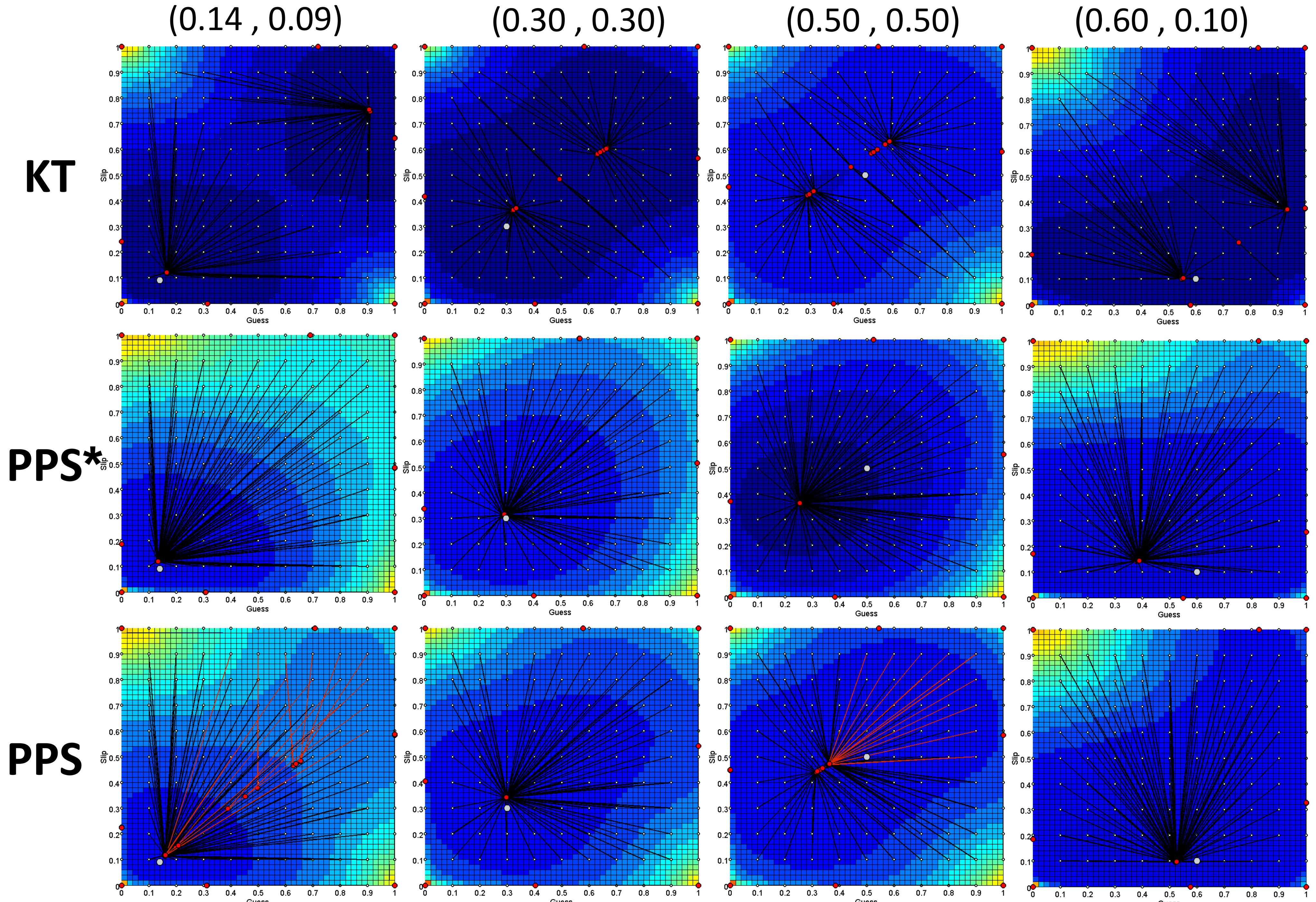
EM Convergence and Model Fit Visualization



• Background color represents the log likelihood fit of the parameter space
• Ground truth parameters are Guess/Slip = 0.14/0.09
• This graph shows that there are two distinct areas of best fit to the data
•Parameters starting NE of the X + Y = 1 line converge to the degenerate parameter state (0.77,0.90)

Comparison of KT and PPS model convergence

Additional synthesized datasets with(Guess, Slip) ground truth parameters



The KT model exhibits one or two points of degenerate convergence for each synthesized dataset while the PPS models have only one point of convergence that is near the ground truth parameter value

The first author is a National Science Foundation GK-12 Fellow

Iterative EM initial parameter analysis

• To start off we fixed the prior and learn parameter at their true values and only learned the guess and slip parameters in order to build intuition about the model behavior with just two free parameters

• We wanted to explore how EM converged based on initial parameter values so we iterated through the entire space of starting values from 0 to 1 in 0.02 steps

Calculation of error based on learned parameter values

Parameter	True value	EM initial value	EM learned value
Guess	0.14	0.36	0.23
Slip	0.09	0.40	0.11

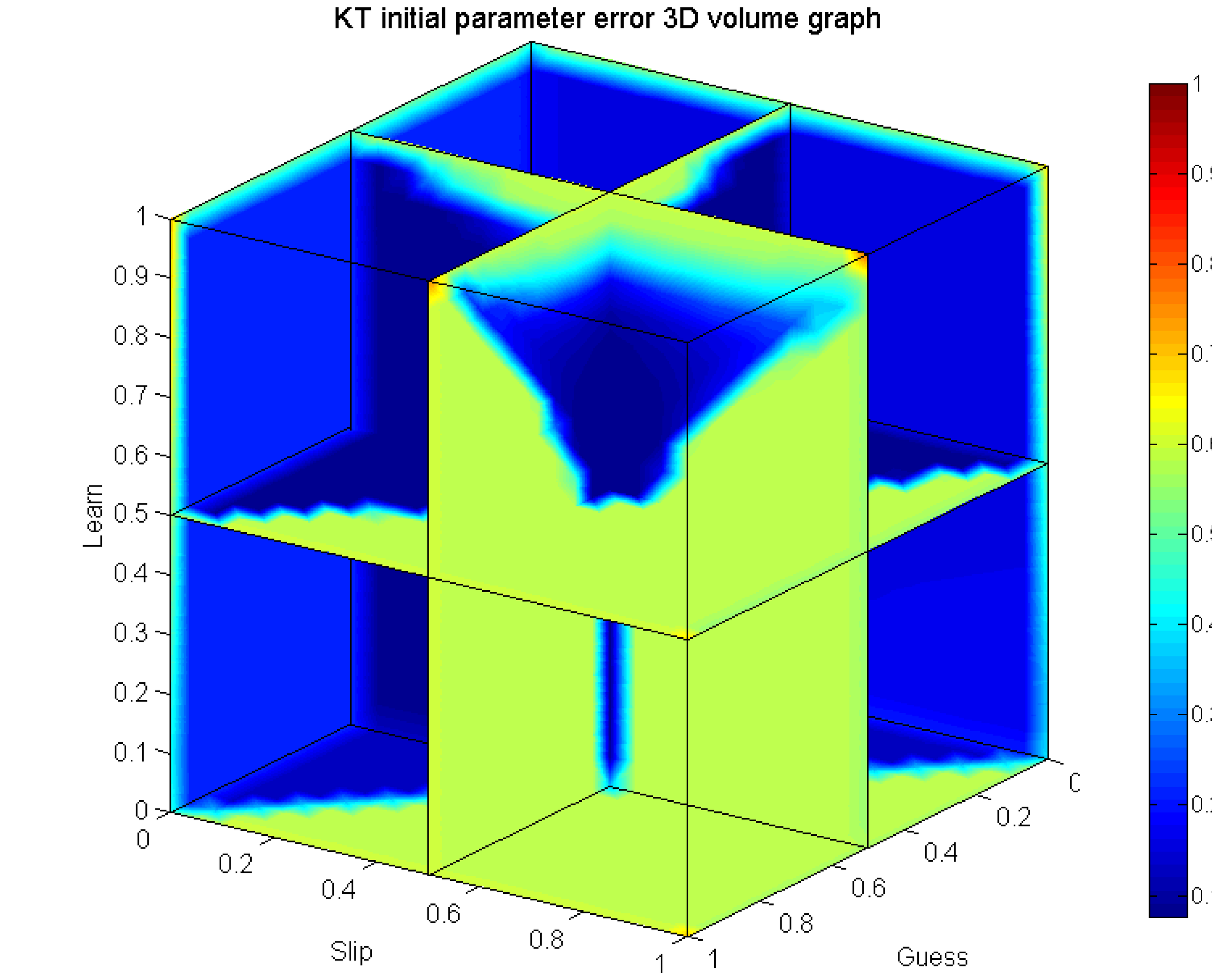
Error = [abs($Guess_{True}$ – $Guess_{Learned}$) + abs($Slip_{True}$ – $Slip_{Learned}$)] / 2 = 0.11

• These parameters are iterated in intervals of 0.02
• 1 / 0.02 + 1 = 51, 51*51 = 2601 total iterations

• EM log likelihood
• Higher = better fit to data

GuessT	SlipT	GuessI	SlipI	GuessL	SlipL	Error	LLstart	LLend
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

Analysis with three free parameters



• The color of the graph at any given point represents the mean average error of the parameters that EM converges to from that point (lower is better)
• We observed a replication of the trend shown in the two parameter analysis, where starting guess and slip values that sum to greater than one end up leading to degenerate states. The starting value of the learn parameter was more flexible than guess and slip. A high learn starting value would still allow for convergence to a low error state

Contributions

• Clearly depicted the dual global maxima nature of the KT model
• Demonstrated how the parameter space of a model can be explored to better understand how it will behave under various circumstances
• Revealed the single maximum property of the PPS model and its ability to learn accurate ground truth parameters from data

Publications based on this work

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.*

Pardos, Z. A., Heffernan, N. T. Under Review (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*.