Falakmasir, M. H., Pardos, Z. A., Gordon, G. J., Brusilovsky, P., In press (2013). A Spectral Learning Approach to Knowledge Tracing. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013).* Memphis, TN, U.S.A

A Spectral Learning Approach to Knowledge Tracing

Mohammad H. Falakmasir
Intelligent Systems Program,
University of Pittsburgh
210 South Bouquet Street,
Pittsburgh, PA,
falakmasir@cs.pitt.edu

Zachary A. Pardos
Computer Science AI Lab
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA 02139
zp@csail.mit.edu

Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

Peter Brusilovsky
School of Information Sciences,
University of Pittsburgh,
135 North Bellefield Ave.,
Pittsburgh, PA 15260, USA
peterb@pitt.edu

ABSTRACT

Bayesian Knowledge Tracing (BKT) is a common way of determining student knowledge of skills in adaptive educational systems and cognitive tutors. The basic BKT is a Hidden Markov Model (HMM) that models student knowledge based on five parameters: prior, learn rate, forget, guess, and slip. Expectation Maximization (EM) is often used to learn these parameters from training data. However, EM is a time-consuming process, and is prone to converging to erroneous, implausible local optima depending on the initial values of the BKT parameters. In this paper we address these two problems by using spectral learning to learn a Predictive State Representation (PSR) that represents the BKT HMM. We then use a heuristic to extract the BKT parameters from the learned PSR using basic matrix operations. The spectral learning method is based on an approximate factorization of the estimated covariance of windows from students' sequences of correct and incorrect responses; it is fast, local-optimum-free, and statistically consistent. In the past few years, spectral techniques have been used on real-world problems involving latent variables in dynamical systems, computer vision, and natural language processing. Our results suggest that the parameters learned by the spectral algorithm can replace the parameters learned by EM; the results of our study show that the spectral algorithm can improve knowledge tracing parameter-fitting time significantly while maintaining the same prediction accuracy, or help to improve accuracy while still keeping parameter-fitting time equivalent to EM.

Keywords

Bayesian Knowledge Tracing, Spectral Learning.

1. INTRODUCTION

Hidden Markov Models and extensions have been one of the most popular techniques for modeling complex patterns of behavior, especially patterns that extend over time. In the case of BKT, the model estimates the probability of a student knowing a particular skill (latent variable) based on the student's past history of incorrect and correct attempts at that skill. This probability is the key value used by many cognitive tutors to determine when the student has reached mastery in a skill (also called a Knowledge Component, or KC) [17]. In an adaptive educational system, this probability can be used to recommend personalized learning activities based on the detailed representation of student knowledge in different topics.

In practice, there is a two-step process for inferring student knowledge. In the first step, an HMM is learned for each topic or skill within a tutoring system based on the history of students' interaction with the system. The output of this step is a set of parameters (basic parameters of BKT: prior, learn rate, forget, guess, and slip), which is used in the second step to estimate the mastery level of each student. A popular method for the first step, learning parameters from training data, is Expectation Maximization (EM). However, EM is a time consuming process, and previous studies [2,3,11,14] have shown that it can converge to erroneous learned parameters, depending on their initial values. To address these problems, we propose an alternate method: first we use a spectral learning method [4] to learn a Predictive State Representation [15] of the BKT HMM directly from the observed history of students' interaction. Then we use a heuristic to extract the parameters of BKT directly from the PSR. Our results show that the learned PSR captures the essential features of the training data, allowing a computationally efficient and practically effective prediction of BKT parameters. In particular, we decreased the time spent on learning the parameters of BKT by almost 30 times on average compared to EM, while keeping the mean accuracy and RMSE of predicting students' performance on the next question statistically the same. Furthermore, by initializing EM with our extracted parameters, we can obtain improvements in accuracy and RMSE.

This paper is organized as follows: Section 2 provides a background on BKT parameter learning and spectral learning of the parameters in PSRs. Section 3 describes our methodology and setting. In Section 4 we present the detailed results of our experiments and compare the BKT model with our model from several points of view. We provide analysis and justification of the results in Section 5. Finally, Section 6 is conclusion and future work.

2. BACKGROUND

In BKT we are interested in a sequence of student answers to a series of exercises on different skills (KCs) in a tutoring system [6]. BKT treats each skill separately, and attempts to model each skill-specific sequence using a binary model of the student's latent cognitive state (the skill is learned or unlearned). Treating state as Markovian, we therefore have five parameters to explain student mastery in each skill: probabilities for initial knowledge, knowledge acquisition, forget, guess, and slip. However, in standard BKT [6], it is typical to neglect the possibility of forgetting, leaving four free parameters.

The main benefit of the BKT model is that it monitors changes in student knowledge state during practice. Each time a student answers a question, the model updates its estimate of whether the student knows the skill based on the student's answer (the HMM observation). However, the typical parameter estimation algorithm for BKT, EM, is prone to converging to erroneous local optima depending on initialization. On the other hand, in the past few

years, researchers have introduced a generalization of HMMs called Predictive State Representations (PSRs) [16] that can be extracted from the data using spectral learning methods [8]. The new learning algorithm uses efficient matrix algebra techniques, which avoid the local optima problems of EM (or any other algorithms based on maximizing data likelihood over the HMM parameter space) and run in a fraction of the time of EM. In this section we first review the EM parameter learning of BKT and then provide a brief background on spectral learning of PSRs.

2.1 EM Parameter Learning of BKT

The main problem with BKT parameter learning by EM is the initial values. The EM algorithm is an iterative process. In each iteration, we first estimate the distributions over students' latent knowledge states, and then update the BKT parameters to try to improve the expected log-likelihood of the training data given our latent state distribution estimates. As mentioned before, the iterative nature of EM means that it is prone to getting stuck in local optima. To remedy this problem, researchers often use multiple runs of EM from different starting points; however, the multiple runs can be time-consuming. Calculating the log likelihood of the model in each iteration also involves going through all the training data, which further exacerbates the runtime problem, especially with large data sets.

There are number of studies that try to handle the problems of EM parameter learning by different approaches. In basic BKT [6], the authors tried to solve the problem by imposing a plausible range of values for each parameter—for example setting the maximum value for the guess parameter to be 0.30. Similar approaches have been applied by [2] and [4]. Another study [12] tried to address the local optimum problem by modifying the structure of BKT and using information from multiple skills to estimate each student's prior in particular skills. The same group made an effort [13] to improve BKT by clustering students based on their performance and using different models for students in different clusters.

Beck & Chang [3] discussed another fundamental problem, called identifiability, with learning BKT parameters by maximum likelihood. In their work, they showed that different sets of BKT parameters could lead to identical predictions of student performance. There is still one set that is more plausible based on expert knowledge, but the other set with identical fit tends to predict that the students are more likely to answer a question wrong when they mastered the skill. They recommend the same approach of constraining the values of the parameters into a plausible range based on the domain knowledge. While these studies elucidated the problem of identifiability and gave rules of thumb to follow in order to arrive at plausible parameters, these rules are often specific to a particular domain and do not necessarily generalize. Moreover, constraining EM to move inside a pre-known parameter space is not trivial, and in many cases the optimizer ends up exceeding its iteration threshold walking along the boundaries of the parameter space without converging to the maximum likelihood value.

Pardos & Heffernan [11] suggested running a grid search over the EM parameter initialization space of BKT to try to find which initial values led to good or bad learned parameters. They analyzed the learned parameters and tried to find boundaries for the initial values not based on plausibility but based on the exact error. They showed that choosing initial guess and slip values that summed up to less than one tends to lead EM to converge toward the expert-preferred parameter set.

2.2 Spectral Learning of PSRs

A Predictive State Representation (PSR) [10] is a compact and complete description of a dynamical system. A PSR can be estimated from a matrix of conditional probabilities of future events (*tests* or *characteristic events*) given past events (*histories* or *indicative events*). If the true probability matrix is generated from a PSR or an HMM, then it will have low rank; so, *spectral* methods can approximate a PSR well from empirical estimates of the probabilities [4,5,8,15]. (In practice we estimate a similarity transform of the PSR parameters, known as a Transformed PSR [15].)

We use in particular the spectral algorithm of Boots & Gordon [5] [4]. They applied their method in several applications and compared the results with competing approaches. In particular, they tested the algorithm by learning a model of a high-dimensional vision-based task, and showed that the learned PSR captures the essential features of the environment effectively, allowing accurate prediction with a small number of parameters. Our work uses their published code.[1]

3. METHODOLOGY

We propose replacing the parameter-learning step of BKT with a spectral method. In particular, we use spectral learning to discover a PSR from a small number of sufficient statistics of the observed sequences of student interactions. We then use a heuristic to extract an HMM that approximates the learned PSR and read the BKT parameters off of this extracted HMM. We can finally use these parameters directly to estimate student mastery levels, and evaluate prediction accuracy with our method compared to the standard EM/MLE method of BKT parameter fitting. We call the above method "spectral knowledge tracing" or SKT. We also evaluated using the learned parameters as initial values for EM in order to get closer to the global optimum. Due to the fact that spectral method does not attempt to maximize likelihood, and also some noise in the translation of the PSR to BKT parameters, the returned BKT parameters are close to the global maximum, but further improvement is available with a few EM iterations. The rest of the section presents a short description of the data along with a brief summary of our student model and analysis procedure.

3.1 Data Description

Our data comes from an online self-assessment tool QuizJET for Java programming. This tool is a part of an adaptive educational system JavaGuide [7] that keeps detailed track of students' interaction to provide adaptive navigation support. The system presents and evaluates parameterized questions to students (programming question templates filled in with random parameters); students can try different versions of the same question several times until they acquire the knowledge to answer them correctly or give up. There are a total of 99 question templates, categorized into 21 topics, with a maximum of 6 question templates within a topic.

We consider each topic as a KC and each question template as a Step toward mastery of the KC. Based on the definition of BKT and KC [6,17] we are only considering the first attempt of each student on each question template, assuming that if a student tried a question template several times until success, they will answer the next question within the topic correctly on the first attempt. This mapping is more coarse-grained than the original definition of KC since we are not dealing the data from an intelligent tutoring system. However, the question templates are designed in such a way that answering all of them correctly will result in mastery of the topic.

[1] http://www.cs.cmu.edu/~ggordon/spectral-learning/

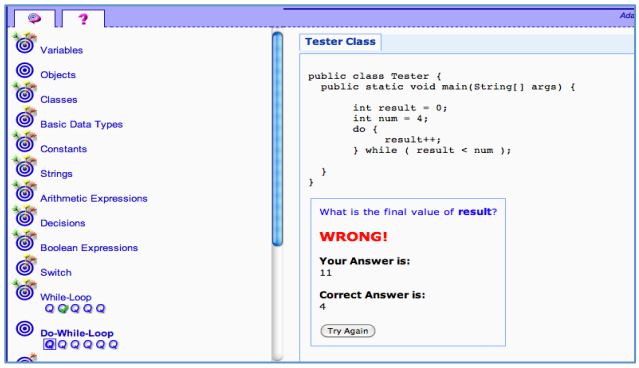

Figure 1: Student view of a question template for the skill "Do-While-Loops".

Figure 1 shows a student view of an example question template. The student can select a topic from the left pane to expand the question templates under each topic. Then s/he can try answering any of the questions under the topic repeatedly whether s/he answers it right or wrong. The system has been in use in the introductory programming classes at the School of Information Sciences, University of Pittsburgh for more than four years. In our study we use data for 9 semesters from Spring 2008 to Fall 2012. Table 1 shows the distribution of records over the semesters.

Table 1: distribution of the records over the semesters.

Semester	#Students	#Topics (Templates) tried	#Records
Spring 2008	15	18 (75)	427
Fall 2008	21	21 (96)	1003
Spring 2009	20	21 (99)	1138
Spring 2010	21	21 (99)	750
Fall 2010	18	19 (91)	657
Spring 2011	31	20 (95)	1585
Fall 2011	14	17 (81)	456
Spring 2012	41	19 (95)	2486
Fall 2012	41	21 (99)	2017
Total	222	21 (99)	10519

The system had no major structural changes since 2008, but the enclosing adaptive system used some engagement techniques in order to motivate more students to use the system. This is the main reason the number of records is higher in the Spring and Fall semesters of 2012.

3.2 Student Model

A time-homogeneous, discrete Hidden Markov Model (HMM) is a probability distribution on random variables $\{(x_t, h_t)\}_{t\in\mathbb{N}}$ such that, conditioned on (x_t, h_t), all variables before t are independent of all those after t. The standard parameterization is the triple (T, O, π) where:

$$T \in \mathbb{R}^{m\times m}, \qquad T_{ij} = \Pr[h_t = i | h_{t-1} = j]$$

$$O \in \mathbb{R}^{n\times m}, \qquad O_{ij} = \Pr[x_t = i | h_t = j]$$

$$\pi \in \mathbb{R}^{m}, \quad \pi_j = \Pr[h_1 = j]$$

O is a mapping from hidden states to output predictions, and T is a mapping between hidden states. Considering our conditional independence properties, T, O, and π fully characterize the probability distribution of any sequence of states and observations [8]. Since the hidden states h_t are not directly observable from the training data, one often uses heuristics like EM to find $\hat{h}_t$, $\hat{T}$, $\hat{O}$ and $\hat{\pi}$ that maximize the likelihood of the samples and the current estimates. In the BKT setting, T is a 2×2 stochastic matrix, so it has two hidden parameters P(learn) and P(forget). O is also a 2×2 stochastic matrix, so it also has two hidden parameters P(guess) and P(slip). And, π is a length-2 probability distribution, so it has one hidden parameter P(init).

Our main contribution is to try extracting these matrices from a learned PSR, giving us the benefit of significantly decreasing training time and avoiding local optima. The details of the spectral algorithm for learning the PSR from the sequence of action-observation pairs are beyond the scope of this paper and can be found in [4]. The algorithm gets a sequence of students' first answers to different question templates within a topic, and builds a PSR using spectral learning. The key parameters of this particular implementation are window sizes used in creating state estimates; we set these to $n_{past} = 10$ and $n_{fut} = 6$. The outputs of the PSR learner are: first, the estimated PSR parameters $\hat{h}_0$, $\hat{A}_1$, and $\hat{A}_2$, and second a set of (noisy) state estimates $\hat{h}_t$, each of which represents a particular time point in the input sequence. We actually added dummy observations before the beginning and after the end of each observation sequence, in order to make the best use of our limited sample size; this means we get four matrices $\hat{A}_i$ from the PSR learner, corresponding to the two original observations plus the two dummy observations. We simply ignore the dummy observations when converting to an HMM.

Nominally, the PSR parameters are related to the HMM parameters by the equations $\hat{\pi} = \hat{h}_0$, $\hat{T} = \hat{A}_1 + \hat{A}_2$, $\hat{O}_i = \hat{A}_i \hat{T}^{-1}$. (Here $\hat{O}_i$ is the diagonal matrix with the ith column of $\hat{O}$ on its diagonal.) However, there is an ambiguity in PSR parameterization: for any invertible matrix S, we can replace each state $\hat{h}_t$ by $S\hat{h}_t$, as long as we replace $\hat{A}_i$ by $S\hat{A}_i S^{-1}$ for $i = 1,2$. When we use the modified parameters to compute likelihoods, each pair $S^{-1}S$ cancels, leaving the predictions of the PSR unchanged. So, we have to choose the right transformation S to be able to find parameters $\hat{T}$ and $\hat{O}$ that satisfy the conditions of BKT (each element should be a probability between 0 and 1, and columns should sum to 1).

To pick the transformation matrix S, we designed a heuristic that looks at the state estimates $\hat{h}_t$: we attempt to guess which points in the learned state space correspond to the unit vectors (1,0) and (0,1) in the desired transformation of the learned state space. (We call these the "transformation points.") Given the transformation points, the matrix S is determined. Our heuristic runs in time linear in the length of the input sequence of correct/incorrect observations. Figure 2 shows an overview of the transformation process and Figure 3 shows the details of the heuristic.

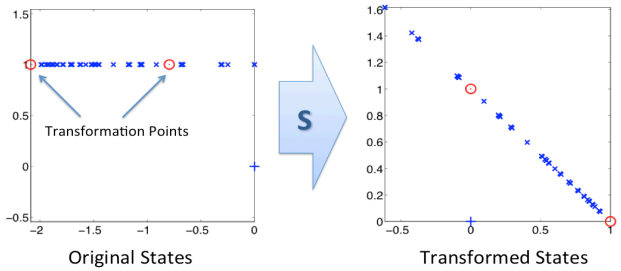

Figure 2: Overview of the transformation scheme.

Algorithm FindTransformationPoints(PSR output States)

Find the minimum and maximum values among the predictive states (mi, ma)

Calculate p = distance between the maximum value among predictive states and the initial state (s_1)

Let n = size(predictive states)

Let step = p / n

For i=1 to n

Fix the first transformation point to $mi - step$

Set second transformation point to $s_1 + i \times step$

Calculate S by linear regression from transformation points to (1,0) and (0,1)

Transform the PSR and calculate $\hat{T}$ and $\hat{O}$

If $\hat{T}$ and $\hat{O}$ have all elements between 0 and 1

Break

End

Figure 3: Our heuristic to find the transformation points

One slightly subtle point is that, due to noise in the parameter estimates, no matter how we choose the transformation S, the matrices $\hat{O}_i = \hat{A}_i \hat{T}^{-1}$ may not be diagonal. In this case, we simply zero out the off-diagonal elements and renormalize.

3.3 Analysis Procedure

To evaluate our new parameter extraction method, we compared the results of our method with EM learning of BKT parameters as a baseline. We compare both runtime and the ability to predict students' correct/incorrect answer to the next question; for the latter, we calculate both Root Mean Squared Error (RMSE) and prediction accuracy (percent correct). We hypothesize that our spectral method has better performance compared to EM in regard to the time spent on extracting the parameters, while keeping the same accuracy and RMSE of predicting the students' answer to the next question. Since the parameters learned from the PSR are an approximation of the actual global best-fit set of BKT parameters, we also hypothesize that if we use the them as the initial parameters of EM, it will result in a better model in both accuracy and RMSE.

4. RESULTS

For the purpose of mimicking how the model may be trained and deployed in a real world scenario, we learn the model from the first semester data and test it on the second semester, learn the model from the first and second semester data and test it on the third semester, and so on. In total, we calculated results for 155 topic-semester pairs. All analysis was conducted in Matlab on a laptop with a 2.4 GHz Intel® Core i5 CPU and 4 GB of RAM.

4.1 EM Results

In our experiments it took around 36 minutes for EM to fit the parameters, which is on average 15 seconds for each topic-semester pair. In 2 out of 155 cases, EM failed to converge within the 200-iteration limit. The average accuracy of predicting a student's answer to the next question using the parameters learned by EM is 0.650 with RMSE of 0.464. Figure 4 shows the boxplot of the parameters learned by EM. The average values for prior, learn, forget, guess and slip are: 0.413, 0.162, 0.019, 0.431, 0.295.

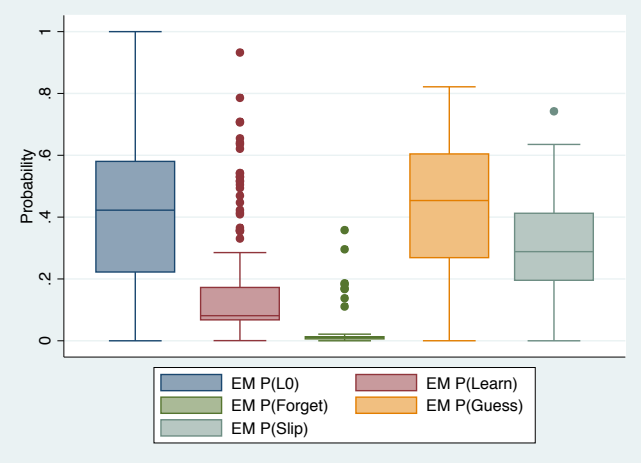

Figure 4: Boxplot of the parameters learned by EM

4.2 SKT Results

It took 1 minute 16 seconds in total for the spectral method to learn the parameters for all semesters and topics; that is almost 30 times faster than EM. The average accuracy of predicting student answer to the next question is 0.664 and RMSE is 0.463. Figure **5** shows the boxplot of the parameters learned by SKT. The average values for prior, learn, forget, guess and slip are: 0.526, 0.268, 0.302, 0.397, 0.271. Note that these values are substantially different from those learned by EM, which means that the calculated student mastery levels will also be different.

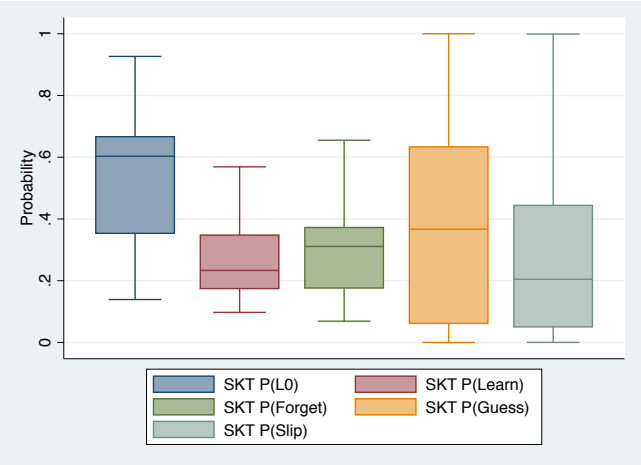

Figure 5: Boxplot of the parameters learned by spectral method

4.3 SEM Results

When we initialized EM with the spectrally learned parameters, the total time was 10 minutes and 40 seconds; that is still substantially faster than plain EM. As expected, the average accuracy of predicting a student's answer to the next question increased to 0.706, and RMSE decreased to 0.422, better than both previous models. Figure 6 shows the boxplot of the refined parameters. The average values for prior, learn, forget, guess and slip are: 0.492, 0.381, 0.360, 0.391, 0.292.

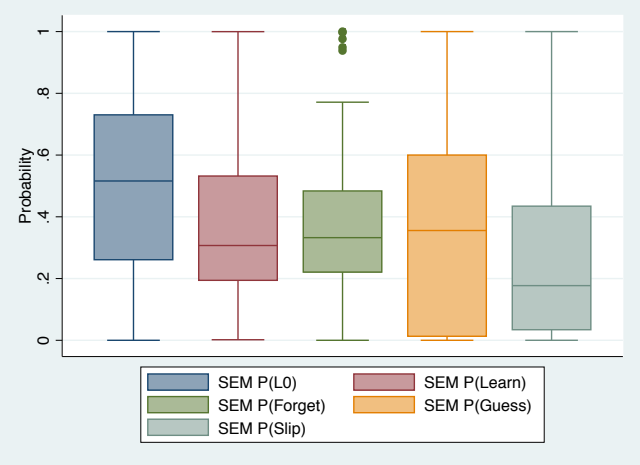

Figure 6: Boxplot of the parameters learned SEM

4.4 Comparison

4.4.1 Time

To get a better understanding of the time complexity of EM and SKT and their relation, we show a semilog plot of the times (Figure 7). We measure the elapsed time of parameter learning using the tic and toc functions of Matlab. Both methods have a similar growth rate as we increase the size of the training data: as we can see in the Figure, the slope of the fitted line for the EM time (green points) is almost the same as the slope of the fitted line for the SKT time (red points). We also tried locally weighted scatter plot smoothing (LOWESS) to compare the runtimes (Figure 8).

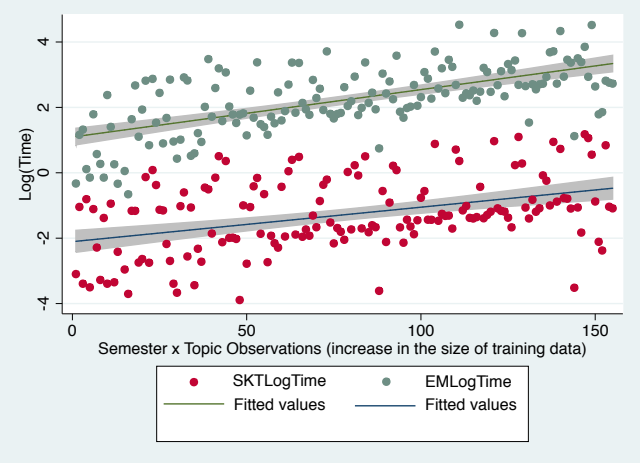

Figure 7: Scatter plot of log(time) with a fitted line

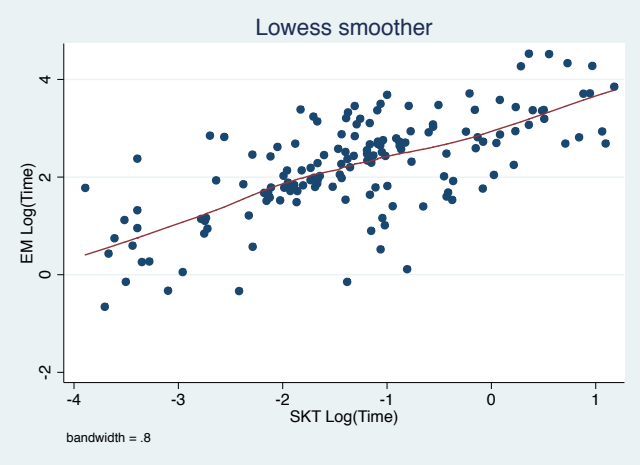

Figure 8: Regression of the Log(time)

The LOWESS plot confirms our intuition that the EM time grows at least linearly compared to the SKT time. To test that hypothesis we tried linear regression on the log-log plot. A 95% confidence interval for the intercept is [2.82, 3.18], which excludes an intercept of 0; a 95% interval for the slope is [.51, .70], which excludes a slope of 1. This can be interpreted as: the time spent learning parameters using EM is on average at least $e^{2.82} \approx 16.77$ times greater than the time spent learning the parameters using SKT, and the scaling behavior of EM is likely to be worse (the ratio gets higher as the data gets larger).

4.4.2 Accuracy and RMSE

Figure 9 and Figure 10 show the histogram of prediction accuracy and RMSE for the 3 models. By looking at the histograms, we can say that the results are approximately normally distributed with about the same variance, but different means.

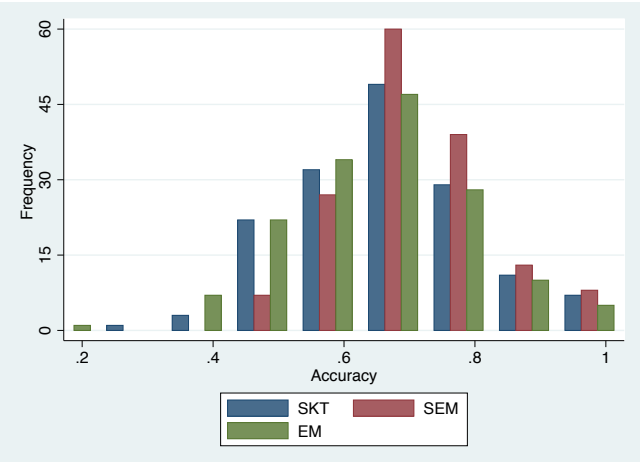

Figure 9: Histogram of Prediction Accuracy

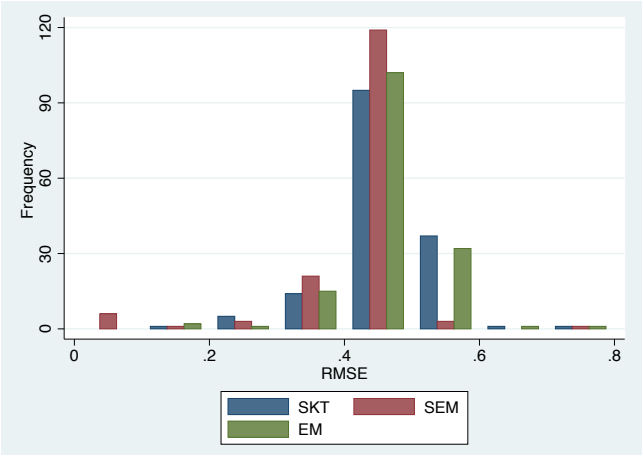

Figure 10: Histogram of the Prediction average RMSE

Regarding prediction accuracy, both of our methods significantly improved the prediction results ($p=0.017$ SKT vs. EM, $p<<0.001$ SEM vs. EM, paired t-test, 153 degrees of freedom). Regarding RMSE, the spectrally learned parameters do not result in a significant improvement compared to BKT, but the combination of SKT with EM leads to a significantly better (lower) RMSE compared to BKT ($p<<0.001$, paired t-test, 153 dof). Table 2 shows the summary of the results. Figure 11 and Figure 12 show the boxplot of the prediction accuracy and RMSE respectively.

Table 2: Summary of the results

Method	Accuracy	RMSE
BKT	0.649 (baseline)	0.465 (baseline)
SKT	0.664 (p=0.017)	0.464 (p=0.348)
SEM	0.706 (p<<0.01)	0.422(p<<0.01)

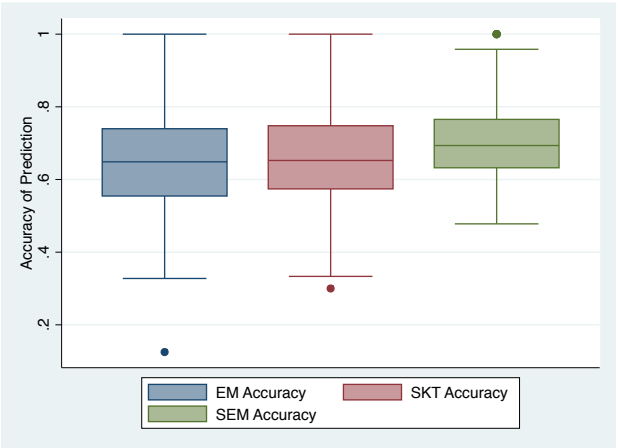

Figure 11: Boxplot of the accuracy

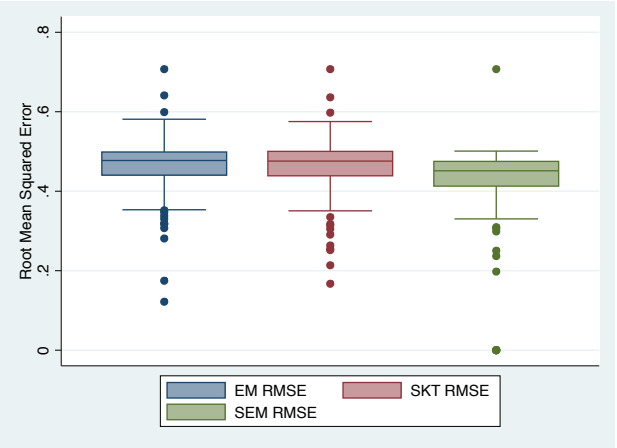

Figure 12: Boxplot of the RMSE

5. DISCUSSION

Based on the results of our study, we found that the spectrally learned parameters can be used directly in the BKT setting, and decrease the time spent on learning parameters by a factor of almost 30 while keeping the same performance in regard to prediction accuracy and RMSE. On the other hand, if we use the spectrally learned parameters to initialize the BKT EM optimization, we can get significantly improved results and still have the advantage of shorter time spent on learning the parameters.

In a setting with a huge number of students and lots of data over several semesters, e.g., an adaptive educational system, the spectrally learned parameters are more helpful in keeping the time spent on building the model for each topic tractable. However, in a more delicate environment, like a cognitive tutor, in which the parameters of BKT are the main basis of the system, we can use the combination method, SEM, and build a more accurate student model in order to predict mastery in different skills.

6. CONCLUSION AND FUTURE WORK

In this paper we presented a novel spectral method for learning the parameters of BKT directly from students' sequences of correct/incorrect responses. One direction for future work would be to compare our method (learn a PSR and extract HMM parameters) to recent algorithms for directly learning an HMM by spectral methods [1], and perhaps combine ideas from these methods with our heuristic.

Another future direction is that, since spectral algorithms have recently been used to learn the parameters of different types of graphical models [9], the results of our study open a new direction for future research on learning complex latent variable models (variations of BKT) directly from student performance data.

From a practical point of view, the results of our study will help us improve our adaptive educational system. Currently, JavaGuide uses a knowledge accumulation approach, based on the total number of correct answers, to estimate students' mastery within each topic for adaptation purposes. The SEM model can be used to improve the system by providing a more accurate (in regard to predicting the student answer to the next question) estimate of student knowledge.

7. ACKNOWLEDGMENTS

This work is partially supported by the Director's Interdisciplinary Graduate Fellowship, and was an extension of a project initiated at the 8th Annual 2012 LearnLab Summer School at CMU.

8. REFERENCES

1. Anandkumar, A., Ge, R., Hsu, D., Kakade, S.M., and Telgarsky, M. Tensor decompositions for learning latent variable models. (2012).
2. Baker, R.S.J., Corbett, A.T., and Aleven, V. More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. *Intelligent Tutoring Systems*, Springer (2008), 406–415.
3. Beck, J.E. and Chang, K. Identifiability: A Fundamental Problem of Student Modeling. *User Modeling 2007 4511*, (2009), 137–146.
4. Boots, B., Siddiqi, S.M., Gordon, G.J., and Byron Boots, S.M.S. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research 30*, 7 (2009), 954–966.
5. Boots, B., Gordon, G.J. Predictive State Temporal Difference Learning. In J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel and A. Culotta, eds., *Advances in Neural Information Processing Systems 23*. 2010, 271–279.
6. Corbett, A.T. and Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and UserAdapted Interaction 4*, 4 (1995), 253–278.
7. Hsiao, I.-H., Sosnovsky, S., and Brusilovsky, P. Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. *Journal of Computer Assisted Learning 26*, 4 (2010), 270–283.
8. Hsu, D., M Kakade, S., Zhang, T., and Daniel Hsu, S.M.K. A Spectral Algorithm for Learning Hidden Markov Models. *Journal of Computer and System Sciences 78*, 1 (2008), 1–22.
9. Ishteva, M., Song, L., Park, H., Parikh, A., and Xing, E. Hierarchical Tensor Decomposition of Latent Tree Graphical Models. *The 30th International Conference on Machine Learning (ICML 2013)*, (2013).
10. Michael L. Littman, R.S.S., Singh, S.P., Littman, M.R.S.S., S Sutton, R., and P Singh, S. Predictive Representations of State. *Neural Information Processing Systems 14*, 14 (2001), 1555–1561.
11. Pardos, Z.A. and Heffernan, N.T. Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. *EDM*, www.educationaldatamining.org (2010), 161–170.
12. Pardos, Z.A. and Heffernan, N.T. *Modeling individualization in a bayesian networks implementation of knowledge tracing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
13. Pardos, Z.A., Trivedi, S., Heffernan, N., Sárközy, G.N., and Zachary A. Pardos, S.T. Clustered Knowledge Tracing. *Intelligent Tutoring Systems ITS 12*, Springer Berlin Heidelberg (2012), 405–410.
14. Pavlik, P.I., Cen, H., and Koedinger, K.R. Performance Factors Analysis --A New Alternative to Knowledge Tracing. *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, (2009), 531–538.
15. Rosencrantz, M., Gordon, G., and Thrun, S. Learning low dimensional predictive representations. *Twenty-first international conference on Machine learning - ICML '04*, ACM Press (2004), 88.
16. Singh, S., James, M.R., and Rudary, M.R. Predictive state representations: a new theory for modeling dynamical systems. *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, AUAI Press (2004), 512–519.
17. Vanlehn, K. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education 16*, 3 (2006), 227–265.