Dynamic knowledge tracing through data driven recency weights

Deepak Agarwal
deepakagarwal39@gmail.com

Ryan S. Baker
University of Pennsylvania,
Philadelphia PA, USA
rybaker@upenn.edu

Anupama Muraleedharan
Educational Initiatives, India
anupama.muraleedharan@ei-india.com

ABSTRACT

There has been considerable interest in techniques for modelling student learning across practice problems to drive real-time adaptive learning, with particular focus on variants of the classic Bayesian Knowledge Tracing (BKT) model proposed by Corbett & Anderson, 1995. Over time researches have proposed many variants of BKT with differentiation based on their treatment of the underlying parameters: (a) general across student and questions; (b) individualized for students; and (c) individualized for questions. Yet at the same time, most of these variants are similar in that they utilize the same Hidden Markov (HMM) architecture to model student learning and share many of the same drawbacks, including less effective balancing between recent and historical student data and assuming that students learn at the same rate across all the attempts irrespective of if they get the question right. At the same time, these variants share the virtue of parameter interpretability, a virtue not seen in recent efforts to re-cast knowledge tracing as a deep learning problem.

This paper proposes a different architecture that replaces learning rate with recency weights which capture student improvement wholly through data rather than assuming constant learning across attempts and manages recent and historical data more appropriately while retaining the interpretability of BKT parameters. The proposed model was tested on multiple public datasets from ASSISTments and Mindspark and performed similarly to classic BKT model on unseen data.

Keywords

Intelligent tutoring system, Bayesian Knowledge Tracing, Student modelling, Hidden Markov Model (HMM)

1. INTRODUCTION

One of the most common forms of adaptivity in intelligent tutoring systems is mastery learning, where a system provides content on a skill until a student demonstrates they know the skill [8]. Most intelligent tutoring systems rely on “Knowledge Tracing” models which predict whether a student has learned a skill or not based on the interactions with the learning resources related to that skill within the tutoring system. Currently, most systems used at scale rely on Corbett and Anderson’s (1995) Bayesian Knowledge Tracing (BKT) model or a close variant of it. Most of these models differ in their treatment of the parameters L_0, G, S and T, but leave the basic structure of the underlying HMM model unchanged, and thus share many of the limitations and drawbacks of the BKT model (e.g. [7, 10, 9, 10]). Recently there have been some attempts to use deep learning-based models in education, termed Deep Knowledge Tracing (DKT) [6, 5]. Though DKT models have performance advantages over BKT, it is extremely difficult to interpret the implicit knowledge model. Khajah and colleagues [6] found that it is possible to make meaningful enhancements to BKT that bring its performance to the same level as DKT models.

In this paper, we propose an algorithm, MS-BKT (Multistate BKT) to address two particular shortcomings of the classic BKT model. First, BKT assumes a constant learning rate after each practice opportunity, irrespective of the student responses. which can lead to bias in estimating student mastery level. Second, BKT represents latent student knowledge as a binary variable with known and unknown states, which is a simplification and assumes that the probability of being in a state at step n depends only on the previous step n-1. We suspect that these assumptions limit the BKT model from considering the entire history of responses for students in a balanced manner by giving unproportionately high weight to the most recent attempt. The MS-BKT addresses these issues through two modifications:

- The MS-BKT model gives more weight to recent responses over older ones during the iterative Bayesian update in order to capture changes in student mastery level from data and excludes learning rate T so there is no assumption of fixed learning after each attempt. Please note that this paper uses ‘Recency’ weights differently than previous papers such as Galyardt & Goldin [3] or Gong et al., [4], where they used a decay function to down-weight the older attempts. In comparison, this paper incrementally increases the weight of the newer attempts.

- MS-BKT expands the knowledge node from the typical 2 states (‘Not learned’, ‘Learned’) to 21 states. Adding multiple states to the knowledge node allows MS-BKT to better capture complex sequences of correct and incorrect responses as multiple states make it possible to fine tune the knowledge level more granularly after each new observation than the 2 state model. Given that real world data can be very noisy, MS-BKT model estimates lead to smoother learning curves than classic BKT models.

2. APPROACH

2.1 Classic BKT Model Architecture

Classic BKT employs a Hidden Markov Model (HMM) with a two-state (‘Not learned’, ‘Learned’) latent node representing student mastery level of the skill and a binary observed node indicating whether the student solved the question correctly or incorrectly as shown in Figure 1. The model assumes that the student can make the transition from not knowing the skill to knowing after every practice opportunity, fit as the learning probability p(T). The model also incorporates the probability that the student may answer a question incorrectly despite knowing the skill (called slip) or may get the answer correct despite not knowing the skill (called guess).

The probability that the student knows the skill gets updated after every practice opportunity through the following equations –

$$p(L_{n-1} \mid C_n{=}1) = \frac{p(L_{n-1}) * (1-p(S))}{p(L_{n-1}) * \big(1-p(S)\big) + \big(1-p(L_{n-1})\big) * p(G)}$$

$$p(L_{n-1} \mid C_n{=}0) = \frac{p(L_{n-1}) * p(S)}{p(L_{n-1}) * p(S) + \big(1-p(L_{n-1})\big) * (1-p(G))}$$

$$p(C_n \mid L_{n-1}) = p(L_{n-1}) * (1\text{-}\ p(S)) + (1\text{-}\ p(L_{n-1})) * p(G)$$

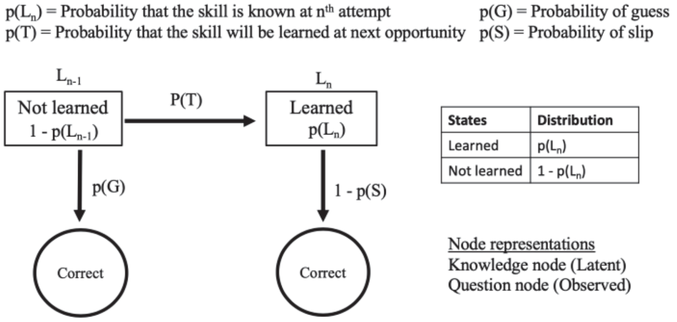
$$p(L_n) = p(L_{n-1} \mid C_n) + (1 - p(L_{n-1} \mid C_n)) * p(T)$$

2.2 Multistate BKT Model Architecture

The architecture for MS-BKT, shown in Figure 2, is similar to that of classic BKT with two changes:

- The “knowledge node” consists of 21 states instead of 2 (Knowledge states are denoted by L^i_n where i is in range 0 to 20 and Σi p(Lin) = 1). 21 discrete states were selected as it was granular enough to give a precise estimate with manageable calculation overhead. The choice of number of states can be explored further in future work, including the possibility of a continuous distribution function.
- A recency weight parameter R is introduced in place of the transition probability p(T). The model assigns a default weight of 1 to the first attempt and thereafter weight increases incrementally by a fixed quantum R for each new attempt. The optimal value of R can be learnt from data. Recent attempts are incrementally weighted more based on the intuition that the recent data will reflect current learning level better but at the same time, older attempts cannot be ignored completely as data can be inherently noisy.

This effectively means that MS-BKT is the same as classic BKT in that new data is integrated with a past estimate aggregating all past data, but differs in that the past estimate is now a distribution and that the weight of the new data increases over time.

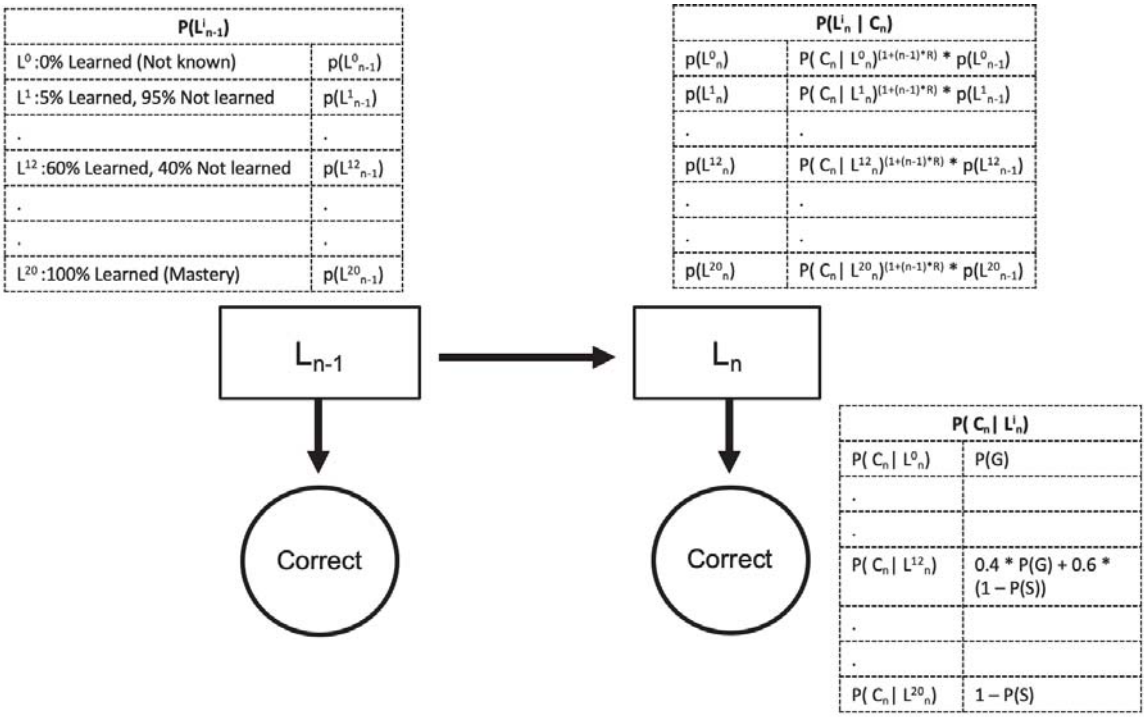


$p(L_n)$ = Probability that the skill is known at nth attempt
$p(G)$ = Probability of guess
$p(T)$ = Probability that the skill will be learned at next opportunity
$p(S)$ = Probability of slip

States	Distribution
Learned	$p(L_n)$
Not learned	$1 - p(L_n)$

Node representations
Knowledge node (Latent)
Question node (Observed)

Figure 1. Classic BKT model



$P(L^i_{n-1})$	
L^0 :0% Learned (Not known)	$p(L^0_{n-1})$
L^1 :5% Learned, 95% Not learned	$p(L^1_{n-1})$
.	.
L^{12} :60% Learned, 40% Not learned	$p(L^{12}_{n-1})$
.	.
.	.
L^{20} :100% Learned (Mastery)	$p(L^{20}_{n-1})$

$P(L^i_n \mid C_n)$	
$p(L^0_n)$	$P(\ C_n \mid L^0_n)^{(1+(n-1)*R)} * p(L^0_{n-1})$
$p(L^1_n)$	$P(\ C_n \mid L^1_n)^{(1+(n-1)*R)} * p(L^1_{n-1})$
.	.
$p(L^{12}_n)$	$P(\ C_n \mid L^{12}_n)^{(1+(n-1)*R)} * p(L^{12}_{n-1})$
.	.
.	.
$p(L^{20}_n)$	$P(\ C_n \mid L^{20}_n)^{(1+(n-1)*R)} * p(L^{20}_{n-1})$

$P(\ C_n \mid L^i_n)$	
$P(\ C_n \mid L^0_n)$	P(G)
.	
.	
$P(\ C_n \mid L^{12}_n)$	0.4 * P(G) + 0.6 * (1 – P(S))
.	
.	
$P(\ C_n \mid L^{20}_n)$	1 – P(S)

Figure 2. MS-BKT model architecture.

2.3 Updating Student Knowledge

Given an observation of the student's response at time opportunity n (correct or incorrect), updated student knowledge (L_n) is calculated using Bayes' rule. Since L_n now consist of 21 states, the probability of each state needs to be updated after every new observation as follows:

$$p(L^i{}_n|C_n) = \frac{p(C_n|L^i{}_{n-1})^{(1+(n-1)*R)} * p(L^i{}_{n-1})}{\alpha(\text{Normalizing factor})} \qquad \text{For i in range 0 to 20}$$

Where:

- $p(L^i{}_n \mid C_n)$ represents the probability of the ith knowledge state given the observation C_n
- $p(C_n \mid L^i{}_{n-1})$ is the likelihood factor. $p(C_n \mid L^i{}_{n-1}) = L^i{}_{n-1} * (1- p(S)) + (1- L^i{}_{n-1})*p(G)$
- $p(L^i{}_{n-1})$ is the prior probability of the ith knowledge state
- $1 + (n-1)*R$ is the weight for the nth response, where n is the number of actions so far and R is a free parameter estimated during model fitting
- α is the normalizing factor which is computed at each iteration to be the value that ensures that probabilities across all the 21 states sum to 1

Once new probabilities are calculated, L_n value is estimated using maximum a posteriori probability (MAP) estimate that equals the mode of the posterior distribution. The advantage of using a MAP estimate over an EAP estimate is that it provides sharper updates even at the initial responses stage. The overall model parameters are learned from data using 'Expectation Maximization'.

3. RECENCY WEIGHTS SUCCESSFULLY CAPTURES REAL TIME LEARNING FROM DATA

In this section we use a hypothetical example to show that the MS-BKT model is capable of capturing learning and forgetting from data itself by the property of recency weights and does not need an external fixed amount of learning to be added after each attempt, unlike classic BKT. This example tracks how the mastery level of three fictitious students changes as they attempt 10 questions on a skill for MS-BKT model. Parameter values used for the below illustration are as follows: L_0: 0.5; G: 0.1; S: 0.1; and T: 0.3.

All three students answer five questions out of 10 correctly, but their patterns are different. Student1 answers questions correctly and incorrectly consecutively. Student2 answers more questions correctly in later attempts, whereas for Student3 the situation is reversed, suggesting that Student2 displays a learning behavior whereas Student3 displays forgetting.

As the following table shows, the mastery level estimate from MS-BKT for Student2 (pattern with learning) is considerably higher than for Student3 (pattern with forgetting), though both students answer 5 out of 10 questions correctly. The mastery estimate of Student1, which was added as a base case, is close to 0.5 as expected.

Table 1. Response patterns used for generating posterior distribution curves

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Mastery Estimate
Student1	0	1	0	1	0	1	0	1	0	1	0.55
Student2	0	0	0	0	1	0	1	1	1	1	0.67
Student3	1	1	1	0	1	0	0	1	0	0	0.37

4. OTHER OBSERVATIONS

In the BKT model, L_n values get updated very aggressively after each observation and result in large fluctuations in the value of L_n (though, with reasonable parameter values, BKT still has lower fluctuation than has been reported for DKT, e.g. Yeung & Yeung, 2018). In comparison to classic BKT model, the MS-BKT model does not fluctuate that widely for the same set of skill parameters. MS-BKT model also takes in account the entire history of the student's responses in a more balanced manner whereas in BKT, a student's response history prior to the third or fourth attempt may become irrelevant due to aggressive updates.

Table 2 and Figure 3 illustrate the above two points using fictitious student data. The underlying BKT and MS-BKT models use the same parameter values for L_0, G, and S; L_0: 0.5, G: 0.1, and S: 0.1. T value for BKT model is 0.1 and R value for MS-BKT is 0.3. The comparison of L_n values for Student4, Student5, and Student6 show that L_n values have significantly higher fluctuations for BKT model in comparison to MS-BKT model. Also, in the cases of Student4 and Student7, L_n estimates are extremely high for the BKT model and does not correspond to the respective response patterns. For Student4, L_n shoots up drastically to 0.75, even though there is a long history of incorrect responses on previous attempts and learning rate is only 0.1. By comparison, the L_n value is around 0.30 for the MS-BKT model. For Student7, L_n value is 0.83 in the case of the BKT model even though 3 out of last 4 responses were incorrect. This is largely due to the fact that the BKT model considers fixed learning rate irrespective of the student responses. The same L_n value for the MS-BKT model is 0.45, as the model is able to derive learning or forgetting directly from the data. Comparison of the response patterns of Student5 and Student6 shows some trade-offs between models. MS-BKT model estimates the L_n value to be 0.55 for Student6 in comparison to 0.90 estimated by the classic BKT model – probably a better fit, since the student has alternated between answering the questions correctly and incorrectly. By contrast, for student5 MS-BKT estimates L_n value to be 0.70 giving the student more credit as for recent responses being correct – perhaps a little too low compared to BKT. Of course, all of these estimates can be adjusted by tuning the parameters during model development.

Table 2. Response patterns used for comparing the two models

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Average	BKT	MS-BKT
Student4	0	0	0	0	0	1	0	1	0.25	0.75	0.30
Student5	0	0	0	0	1	1	1	1	0.50	1.00	0.70
Student6	0	1	0	1	0	1	0	1	0.50	0.90	0.55
Student7	0	1	1	1	0	1	0	0	0.50	0.83	0.45

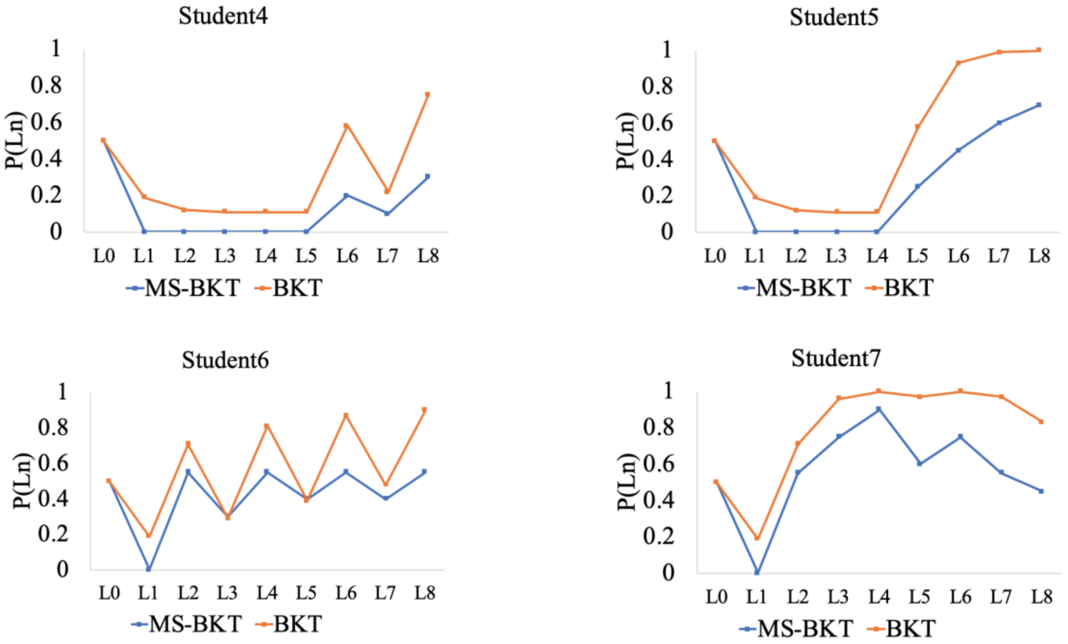


Figure 3. Comparison of L_n estimate for BKT and MS-BKT.

Table 3. L_0**, G, S, T values for BKT and MS-BKT models**

			BKT				MS-BKT			
Dataset	**#Student**	**#Attempts**	L_0	**G**	**S**	**T**	L_0	**G**	**S**	**T**
G6_207	620	6	0.42	0.28	0.15	0.08	0.56	0.27	0.29	0.25
G7_233	540	7	0.73	0.26	0.22	0.01	0.65	0.09	0.25	0.25
G6_217	500	5	0.61	0.30	0.13	0.10	0.60	0.29	0.21	0.25
PER015	855	5	0.50	0.11	0.30	0.15	0.58	0.15	0.29	0.25
WNO021_57	536	6	0.80	0.24	0.18	0.11	0.66	0.27	0.19	0.50
WNO021_48	536	6	0.78	0.30	0.08	0.30	0.74	0.29	0.09	0.25

Table 4. Comparison of BKT and MS-BKT models

			BKT		MS-BKT	
Dataset	**#Students**	**#Attempts**	**AUC ROC**	**RMSE**	**AUC ROC**	**RMSE**
G6_207	156	6	0.707	0.457	0.712	0.460
G7_233	138	7	0.663	0.464	0.640	0.468
G6_217	126	5	0.664	0.442	0.650	0.446
PER015	171	5	0.659	0.480	0.652	0.483
WNO021_57	134	6	0.618	0.421	0.639	0.425
WNO021_48	134	6	0.702	0.337	0.664	0.345

5. PREDICTION QUALITY

We used 6 datasets across 2 different ITS (Assistments - G6_207, G7_233, G6_217; Mindspark - PER015, WNO021_57, WNO021_48) to compare the performance of the MS-BKT model against classic BKT model. Mindspark is an adaptive online tutor for Math and English, developed by Educational Initiatives (EI). Mindspark Math currently has 80,000 users across India, primarily from private schools, in grades 1 to 9. ASSISTments is an online tutor that supports student learning through the use of scaffolding, hints, and immediate feedback. All the datasets consist of student responses in the form of correct or incorrect answers from specific problems tagged by skill. The performance was compared on a hold-out data set consisting of 20% of the data. Table 3 lists out the parameter values for the two models for all the datasets using training data. The parameters for each model were tuned using the simple Brute Force approach. Table 4 compares the performance of both the models on hold-out dataset. Results show that the classic BKT model performs better than MS-BKT model on most of the datasets (except G6_207 and WNO021_57) but the differences are not very large.

6. CONCLUSION

This paper highlights two issues related to the classic BKT model and tries to address them by proposing a new model (MS-BKT). The paper demonstrates that applying a recency adjustment to Bayesian updates can lead to better properties of knowledge estimation, compared to using a static learning rate. The paper also proposes considering latent student knowledge as a multistate variable instead of 2 states, leading to smoother updates in the learning level estimate. In summary, the MS-BKT model displays some useful properties that are worth considering. Ultimately, models should both capture data well and have desirable properties for actual use, whether for use in a running system or discovery with models analysis. There is considerable future work to be done in refining the MS-BKT model further – such as selection of the appropriate number of knowledge states, implementation of recency weights, and effective ways to tune the model parameters.

7. REFERENCES

[1] Baker, R.S., Gowda, S.M., & Salamin, E. 2018. Modeling the learning that takes place between online assessments. *Proceedings of the 26th International Conference on Computers in Education*, 21-28.

[2] Falakmasir, M. H., Yudelson, M., Ritter, S., & Koedinger, K. 2015. Spectral Bayesian knowledge tracing. *Proceedings of the 8th International Conference on Educational Data Mining,* Madrid, Spain, 360-364.

[3] Galyardt, A., & Goldin, I. 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining*. 7, 2, 83–108.

[4] Gong, Y., Beck, J. E., & Heffernan, N. T. 2011. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education*. 21, 1, 27–46.

[5] Jiang, B., Ye, Y., & Zhang, H. 2018. Knowledge tracing within single programming exercise using process data. *Proceedings of the 26th International Conference on Computers in Education*. 89-94.

[6] Khajah, M., Lindsey, R. V., & Mozer, M. C. 2016. How deep is knowledge tracing? *Proceedings of the 9th International Conference on Educational Data Mining*. 94-101.

[7] Pardos, Z. A., & Heffernan, N. T. 2011. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. *User Modeling Adaptation and Personalization Lecture Notes in Computer Science*, 243-254.

[8] Ritter, S., Yudelson, M., Fancsali, S. E., & Berman, S. R. (2016, April). How mastery learning works at scale. *Proceedings of the Third ACM Conference on Learning @ Scale*. 71-79.

[9] Wang, Y., & Beck, J. 2013. Class vs. student in a Bayesian network student model. *Artificial Intelligence in Education. AIED 2013*. Lecture Notes in Computer Science, vol. 7926.

[10] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. 2013. Individualized Bayesian knowledge tracing models. *International Journal of Artificial Intelligence in Education*. 171–180.