Using Similarity to the Previous Problem to Improve Bayesian Knowledge Tracing

William J. Hawkins
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
whawkins90@gmail.com

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
nth@wpi.edu

ABSTRACT

Bayesian Knowledge Tracing (BKT) is a popular student model used extensively in educational research and in intelligent tutoring systems. Typically, a separate BKT model is fit per skill, but the accuracy of such models is dependent upon the skill model, or mapping between problems and skills. It could be the case that the skill model used is too coarse-grained, causing multiple skills to all be considered the same skill. Additionally, even if the skill model is appropriate, having problems that exercise the same skill but look different can have effects on student performance. Therefore, this work introduces a student model based on BKT that takes into account the similarity between the problem the student is currently working on and the one they worked on just prior to it. By doing this, the model can capture the effect of problem similarity on performance, and moderately improve accuracy on skills with many dissimilar problems.

Keywords

Student modeling, Bayesian Knowledge Tracing, Problem Similarity

1. INTRODUCTION

Bayesian Knowledge Tracing (BKT) [3] is a popular student model used both in research and in actual intelligent tutoring systems. As a model that infers student knowledge, BKT has helped researchers answer questions about the effectiveness of help within a tutor [1], the impact of "gaming the system" on learning [5], and the relationship between student knowledge and affect [9], among others. Additionally, it has been used in the Cognitive Tutors [6] to determine which questions should be presented to a student, and when a student no longer needs practice on a given skill.

However, BKT models are dependent upon the underlying skill model of the system, as a separate BKT model is typically fit per skill. If a skill model is too coarse-grained or too fine-grained, it can make it more difficult for a BKT model to accurately infer student knowledge [8].

Additionally, even when a skill model is tagged at the appropriate level, seeing similar problems consecutively as opposed to seeing dissimilar problems may have effects on guessing and slipping, two important components of BKT models. For example, if a student does not understand the skill they are working on, seeing a certain type of question twice or more consecutively may improve their chances of "guessing" the answer using a suboptimal procedure that would not work on other questions from the same skill.

Whether the skill model is not at the appropriate level or seeing consecutive similar questions helps students succeed without fully learning a skill, it may be important to take problem similarity into account in student models. In this work, we introduce the Bayesian Knowledge Tracing – Same Template (BKT-ST) model, a modification of BKT that considers problem similarity. Specifically, using data from the ASSISTments system [4], the model takes into account whether the problem the student is currently working on was generated from the same *template* as the previous problem.

The next section describes the ASSISTments system, its template system and the data used for this paper. Section 3 describes BKT and BKT-ST in more detail, and describes the analyses we performed on these models. The results are reported in Section 4, followed by discussion and possible directions for future work in Section 5.

2. TUTORING SYSTEM AND DATA

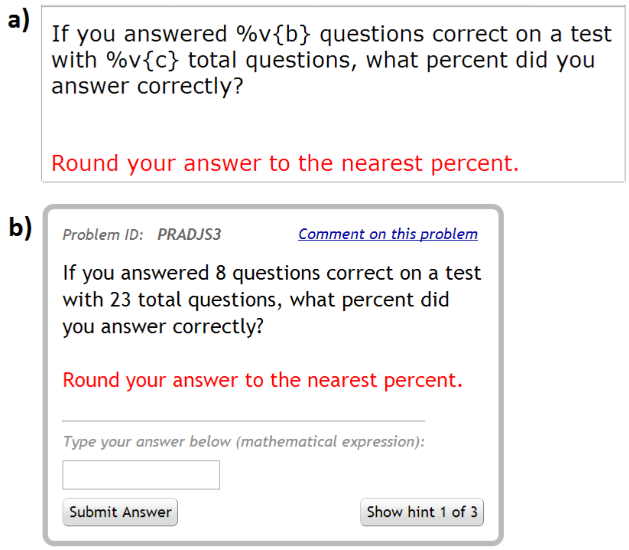
2.1 ASSISTments

ASSISTments [4] is a freely available web-based tutoring system used primarily for middle and high school mathematics. In addition to providing a way for teachers to assess their students, ASSISTments also assists the students in a few different ways: through the use of series of on-demand hint messages that typically end in the answer to the question (the "bottom-out hint"), "buggy" or feedback messages that appear when the student gives a common wrong answer, and "scaffolding" questions that break the original question into smaller questions that are easier to answer.

While teachers are free to author their own content, ASSISTments provides a library of approved content, which includes problem sets called *skill-builders*, which are meant to help students practice a particular skill. While most problem sets contain a fixed number of problems that must all be completed for a student to finish, a skill-builder is a special type of problem set that assigns questions in a random order and that is considered complete once a student answers three consecutive questions correctly on the same day.

While requiring students to answer three consecutive questions correctly on the same day to complete a skill-builder ensures that they have some level of knowledge of the particular skill being exercised, it takes some students many problems to achieve this, meaning they may see the same problem more than once if the skill-builder does not contain enough unique problems.

To ensure this does not happen (or at least make it highly unlikely), ASSISTments has a templating system that facilitates creating large numbers of similar problems quickly. The content creator creates a question as normal, but specifies that it is a *template* and uses variables in the problem statement and answer rather than specific values. Then, they are able to generate 10 unique problems at a time from that template, where each problem is randomly populated with specific values as prescribed by the template. This is especially useful for skill-builders, whose problems should theoretically all exercise the same skill. Figure 1 shows an example of a template (a) and a problem generated from it (b).



a)

If you answered %v{b} questions correct on a test with %v{c} total questions, what percent did you answer correctly?

Round your answer to the nearest percent.

b)

Problem ID: PRADJS3 Comment on this problem

If you answered 8 questions correct on a test with 23 total questions, what percent did you answer correctly?

Round your answer to the nearest percent.

Type your answer below (mathematical expression):

Figure 1. A template (top image) and a problem generated from it (bottom). The variables ‘b’ and ‘c’ in the template are replaced by ‘8’ and ‘23’ in the generated problem.

2.2 Data

In this work, we used ASSISTments skill-builder data from the 2009-2010 school year. This data set consists of 61,522 problem attempts by 1,579 students, spread across 67 different skill-builders. A (student, skill-builder) pair was only included if the student attempted three or more problems on that particular skill-builder, and a skill-builder was included if it was used by at least 10 students and at least one of them completed it.

3. METHODS

In this section, we begin by describing Bayesian Knowledge Tracing, and then move on to our modification of it, called Bayesian Knowledge Tracing – Same Template. Finally, we describe the analyses we performed using these two models.

3.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [3] is a popular student model that uses a dynamic Bayesian network to infer student knowledge using only a student’s history of correct and incorrect responses to questions that exercise a given knowledge component (or “skill”).

Typically, a separate BKT model is fit for each skill. BKT models assume that there are only two states a student can be in for a given skill: the *known state* or the *unknown state*. Using a student’s performance history on a given skill, a BKT model infers the probability that the student is in the *known state* on question *t*, P(K_t).

Fitting a BKT model involves estimating four probabilities:

1. Prior Knowledge – P(L_0): the probability the student knew the skill before answering the first question
2. Learn Rate – P(T): the probability the student will know the skill on the next question, given that they do not know the skill on the current question
3. Guess Rate – P(G): the probability the student will answer the current question correctly despite not knowing the skill
4. Slip Rate – P(S): the probability the student will answer the current question incorrectly despite knowing the skill

Note that forgetting is typically not modeled in BKT: it is assumed that once a student learns a skill, they do not forget it. An example of a BKT model, represented as a static unrolled Bayesian network, is shown in Figure 2.

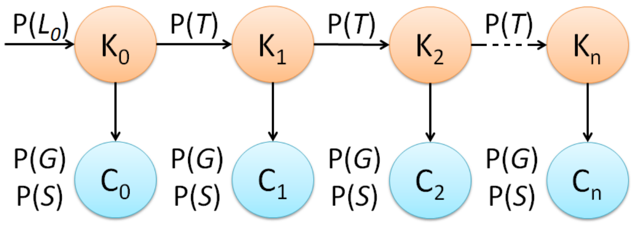


Figure 2. Static unrolled representation of Bayesian Knowledge Tracing. The K_t nodes along the top represent latent knowledge, while the C_t nodes represent performance.

3.2 Bayesian Knowledge Tracing – Same Template

The Bayesian Knowledge Tracing - Same Template (BKT-ST) model differs from the regular BKT model in one way: it takes into account whether the problem it’s about to predict was generated from the same template as the previous problem the student worked on. This is modeled as a binary observed variable that influences performance.

This results in six parameters to be learned per skill: the initial knowledge rate, the learn rate, and two sets of guess and slip rates: one set for when the previous problem and current problem were generated from the same template (P(G|Same) and P(S|Same)), and one for when they aren’t (P(G|Different) and P(S|Different)). The model is shown in Figure 3.

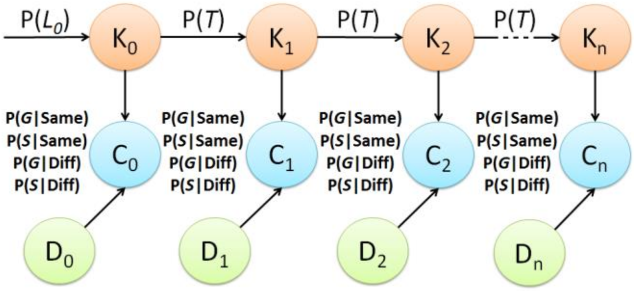


Figure 3. Static unrolled representation of Bayesian Knowledge Tracing – Same Template. The only difference from BKT is the presence of the D_t nodes, which represent whether the previous question was generated by the same template as the current one.

3.3 Analyses

The first analysis in this work simply considers how well the two models fit the data compared to each other overall. This is determined by fitting separate BKT and BKT-ST models for each skill and then predicting unseen student data using five-fold student-level cross-validation. Then, we evaluate each model's ability to predict next question correctness by computing the mean absolute error (MAE), root mean squared error (RMSE) and area under the curve (AUC) for each student and then averaging across students for each type of model. Finally, two-tailed paired t-tests are used to determine the significance of the differences in the metrics.

The second analysis considers what the metrics look like for each model based on how many templates were used for each skill-builder problem set. This is done by splitting the predictions made in the first analysis by how many templates were used in the corresponding skill-builder. We did this to see when it would be worth using BKT-ST over BKT.

Finally we consider the parameter values learned for the BKT-ST model to determine any effects that seeing problems generated by the same template consecutively has on guessing and slipping.

The BKT and BKT-ST models used in these analyses are fit using the Expectation-Maximization (EM) algorithm in the Bayes Net Toolbox for Matlab (BNT) [7]. The initial values given to EM for BKT were 0.5 for $P(L_0)$ and 0.1 for the other three parameters. This was also true for BKT-ST, except the slip rate was set to 0.2 when the current and previous problems were generated from the same template.

4. RESULTS

In this section, we first present the overall comparison of BKT and BKT-ST, then show how they compare to each other based on the number of templates used in each skill-builder. Finally, we examine the learned parameters for the BKT-ST model.

4.1 Overall

The overall results comparing BKT to BKT-ST are shown in Table 1.

Table 1. Overall results of fitting BKT and BKT-ST models.

	MAE	RMSE	AUC
BKT	0.3830	0.4240	0.5909
BKT-ST	0.3751	0.4205	0.6314

According to these results, BKT-ST outperforms BKT in all three metrics. Statistical tests confirmed that these results were reliable (MAE: p < .0001, t(1578) = 9.939; RMSE: p < .0001, t(1578) = 4.825; AUC: p < .0001, t(1314) = -11.095), though according to the values in the table, the only noticeable gain was in AUC.

4.2 By Number of Templates

Next, we considered how well each model did based on the number of templates a skill-builder contained. The results are shown in Figure 4.

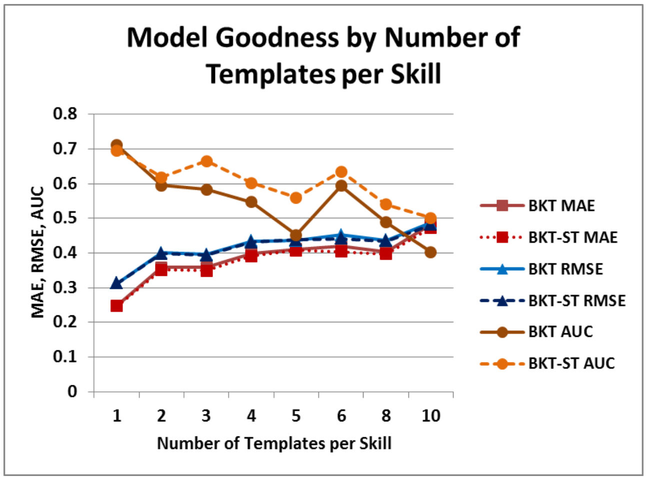


Figure 4. Graph of MAE, RMSE and AUC for the BKT and BKT-ST models, plotted against the number of unique templates per skill.

Interestingly, both BKT and BKT-ST decline rapidly in terms of model goodness as the number of templates per skill-builder increases. This is likely the case because those with more templates are more likely to have more than one skill being tested within them. Interestingly, although both models decline similarly in terms of MAE and RMSE, BKT-ST declines at a slower rate than BKT does in terms of AUC. In fact, BKT-ST outperforms BKT in terms of AUC for every group of skills with more than one template. When grouping the skills by the number of templates they had, BKT-ST achieved an AUC of at least 0.0236 better than BKT for each group that had more than one template, and achieved AUC values that were 0.1086 and 0.0980 better than BKT for skills with five and 10 templates, respectively. Additionally, while BKT performs worse than chance (AUC < 0.5) on skills with eight or more templates, BKT-ST never performs worse than chance.

4.3 Parameter Values

To analyze the parameters learned by BKT-ST, for each skill, we took the average value of each of the six parameters learned across the five folds from the overall analysis.

First, we computed the average value of each parameter across all 67 skills. These are shown in Table 2.

Table 2. Means and standard deviations of BKT-ST parameter values learned across 67 skill-builders

Parameter	Mean	SD
$P(L_0)$	0.6030	0.2617
P(T)	0.2966	0.2500

P(G|Different)	0.1880	0.1655
P(S|Different)	0.2941	0.1737
P(G|Same)	0.3337	0.2495
P(S|Same)	0.1514	0.0848

From the results in Table 2, it appears that on average, seeing consecutive questions generated from the same template both increases the guess rate ($p < .0001$, $t(66) = -4.516$) and decreases the slip rate ($p < .0001$, $t(66) = 7.186$).

Next, we examined how these parameters changed with respect to the number of templates used per skill-builder. The average values of the performance parameters (guess and slip rates for same and different templates) are shown in the graph in Figure 5. The results for skills with one template are omitted since the P(G|Different) and P(S|Different) parameters are meaningless in such cases.

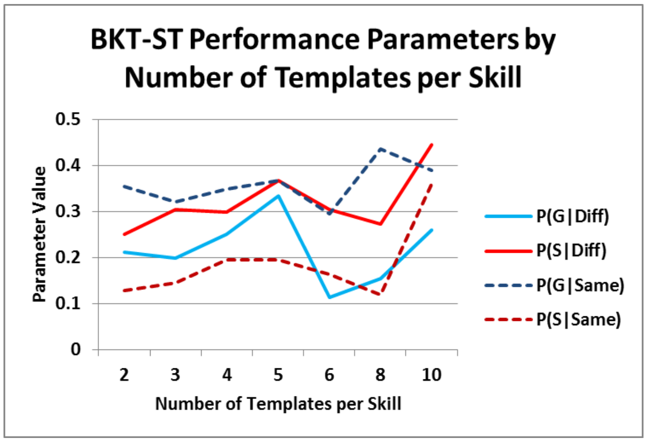


Figure 5. Average value of each performance parameter for the number of templates used per skill-builder.

Although there is no clear pattern for any of the four performance parameters shown in the graph, the average value of P(G|Same) is always higher than that of P(G|Different), and that of P(S|Same) is always lower than that of P(S|Different), with respect to the number of templates used per skill. This appears to reinforce the notion that seeing consecutive problems generated from the same template makes the latter easier to solve, whether this is due to the skill model being too coarse-grained or familiarity with a certain type of problem within a skill inflating performance.

5. DISCUSSION AND FUTURE WORK

From the results in this work, it appears that modifying Bayesian Knowledge Tracing to take similarity between consecutive problems into account moderately improves cross-validated predictive performance, especially in terms of AUC. Additionally, this work showed that seeing consecutive similar problems improves student performance by both increasing the guess rate – the probability of answering a question correctly despite not knowing the skill – and decreasing the slip rate – the probability of answering a question incorrectly despite knowing the skill. Regardless of the underlying reason for this, whether it is because the skill model is too coarse-grained or simply that familiarity with a type of problem within a skill improves performance, it appears important for student models to take the similarity of the problems students encounter into account when trying to model student knowledge.

One direction for future work would be to try going back further in the problem sequence to see how the similarity of problems earlier in a student's history affects their ability to answer the current problem. Additionally, it would be interesting to determine whether the effect changes in certain situations. For example, what is the effect of seeing two similar problems in a row, followed by one that is different from both?

Another area of interest would be to use a model that takes problem similarity into account when trying to predict a longer-term outcome, such as wheel-spinning [2], retention and transfer, as opposed to simply predicting next question correctness.

Finally, applying this model and others like it to other learning environments and skill models of various grain sizes would be helpful for understanding when it is useful. Presumably, if a skill model is at the appropriate grain size, the difference in predictive performance between BKT and BKT-ST would be reduced. The same would be true of systems that fall to one of two extremes: those whose problem sets are highly repetitive, and those whose problem sets have a rich variety of problems.

6. ACKNOWLEDGMENTS

We acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024).

7. REFERENCES

[1] Beck, J.E., Chang, K., Mostow, J., Corbett, A. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. *Intelligent Tutoring Systems*, Springer Berlin Heidelberg, 2008, 383-394.

[2] Beck, J. E., and Gong, Y. Wheel-Spinning: Students Who Fail to Master a Skill. In *Artificial Intelligence in Education*, pp. 431-440. Springer Berlin Heidelberg, 2013.

[3] Corbett, A. and Anderson, J. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253-278.

[4] Feng, M., Heffernan, N.T., Koedinger, K.R. Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *User Modeling and User-Adapted Interaction*, *19*(3), 243-266.

[5] Gong, Y., Beck, J., Heffernan, N., Forbes-Summers, E, The impact of gaming (?) on learning at the fine-grained level. in *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, (Pittsburgh, PA, 2010), Springer, 194-203.

[6] 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*(1), 30-43.

[7] Murphy, K. The bayes net toolbox for matlab. *Computing science and statistics*, *33*(2), 1024-1034.

[8] Pardos, Z. A., Heffernan, N. T., & Anderson, B., Heffernan, C. L. Using Fine-Grained Skill Models to Fit Student Performance with Bayesian Networks. *Proceedings of the Workshop in Educational Data Mining held at the 8th Interna-*

tional Conference on Intelligent Tutoring Systems. (Taiwan, 2006).

[9] San Pedro, M., Baker, R.S.J.d, Gowda, S.M., Heffernan, N.T. Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In Lane, H.C., Yacef, K., Mostow, M., Pavlik, P. (Eds.) AIED 2013. LNCS, vol. 7926/2013, pp.41-50. Springer-Verlag, Berlin Heidelberg.