Comparing Bayesian Knowledge Tracing Model Against Naïve Mastery Model

Vanesa Getseva and Amruth N. Kumar

Ramapo College of New Jersey, Mahwah NJ 07430, USA
{vgetseva, amruth}@ramapo.edu

Abstract. We conducted a study to see if using Bayesian Knowledge Tracing (BKT) models would save time and problems in programming tutors. We used legacy data collected by two programming tutors to compute BKT models for every concept covered by each tutor. The novelty of our model was that slip and guess parameters were computed for every problem presented by each tutor. Next, we used cross-validation to evaluate whether the resulting BKT model would have reduced the number of practice problems solved and time spent by the students represented in the legacy data. We found that in 64.23% of the concepts, students would have saved time with the BKT model. The savings varied among concepts. Overall, students would have saved a mean of 1.28 minutes and 1.23 problems per concept. We also found that BKT models were more effective at saving time and problems on harder concepts.

Keywords: Programming Tutors, Bayesian Knowledge Tracing, Evaluation.

1 Introduction

Student model is essential for facilitating adaptation in intelligent tutoring systems. Bayesian Knowledge Tracing (Corbett et al. 1992) is one of the more popular methods of modeling student's knowledge. The model consists of four parameters per concept. In the past, in order to estimate the four parameters, researchers have used baseline approach (Beck 2007), bounded guess and slip approach, Dirichlet Priors (Beck et al. 2007), contextual estimation (Baker et al. 2008) and empirical probabilities (Hawkins et al. 2014). In this study, we present an empirical approach based on legacy data collected by intelligent tutors. Our approach differs from earlier attempts in that we calculate guess and slip parameters for each problem, not just each concept. We used the calculated BKT model to evaluate its effectiveness in terms of time and effort saved for the students represented in the legacy data.

Currently, our tutors use a naive mastery model to determine whether the student has learned a concept during practice. In this model, a student is said to have mastered a concept if the student solves at least 2 problems on the concept and solves at least 60% of the problems correctly. For the concepts that do not occur as frequently in programming, the mastery criterion was set to at least 1 problem solved and at least 50% of the problems solved correctly. If the Bayesian Knowledge Tracing model could determine that a student has learned a concept with fewer practice problems, using it would reduce the number of unnecessary problems solved and time spent by the student with our tutors.

For the current study we used legacy data collected by tutors on `while` loops and `for` loops from multiple institutions as shown in Table 1. In the table, multi-problem records are the records of students who solved more than one problem on a concept. Each tutor covers one topic, and each topic consists of multiple concepts. The tutors use pretest-practice-posttest protocol during every tutoring session (Kumar 2014).

Table 1. Statistics about the Data Collected by Programming Tutors.

Topic	Number of Concepts	Number of Semesters	Total Records	Multi-problem Records
while Loops	9	9	4,933	2,030
for Loops	10	9	40,124	5,817

The tutors presented only code-tracing problems wherein students were asked to identify the output of a given program. The student grade on each problem was normalized to 0→1.0: 0 when the answer was incorrect, 1.0 when it was correct and a value in between for partially correct answers. The tutors logged the grade and time spent on each problem by each student.

Bayesian Knowledge Tracing (Baker et al. 2008) uses four parameters: L_i, T, G, S. We calculated $P(L_0)$, the probability that a concept was mastered before using the tutor as the percentage of the users who solved the pretest problem on the concept correctly (among Total Records in Table 1). $P(L_0)$ was 0.80 or greater on 32% of the concepts across both tutors. Given the high values of $P(L_0)$, we used 0.98 instead of the traditional 0.95 as the mastery criterion for the BKT model. We computed $P(T)$, the probability of transferring from un-mastered to mastered state for a given concept as the percentage of students who solved the pretest problem on the concept incorrectly, and went on to solve the post-test problem correctly. These were the students who learned the concept by using the tutor.

We computed $P(G)$, the probability a student guesses the correct answer to a practice problem on an un-mastered concept (from Multi-problem Records in Table 1) as the percentage of students who solved the previous problem on the concept incorrectly or partially, but solved the current problem correctly. Similarly, we computed $P(S)$, the probability a student slips, i.e., solves a practice problem on a mastered concept incorrectly or partially as the percentage of students who solved the previous problem on the concept correctly, but solved the current problem incorrectly or partially. For the first practice problem, we approximated this to be 0.01 since the tutors never presented a practice problem unless the pretest problem was solved incorrectly.

Figure 1 illustrates the BKT model for a concept. Several attempts have been made to individualize BKT parameters per student with the aim of improving its fit (Bhatt et al. 2020). *Our approach is different in that we have tried to customize performance parameters G and S to the problems solved by the students because no two problems are alike in terms of the provided context or the expected answer.*

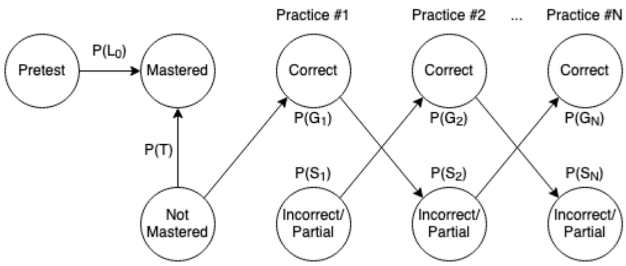

Fig. 1. BKT Model with two parameters (L,T) per concept and two parameters (G,S) per practice problem.

2 Evaluating the BKT Model

We used *k*-fold cross-validation to estimate the performance of our predictive BKT model: We used each of the *k* subgroups to find the number of students who would have saved time, made no difference, or lost time with the BKT model constructed using the other k – 1 groups. We used 25 as the size of each group and rounded up our sample size to the nearest multiple of 25 using stochastic oversampling. After cross-validation runs, we computed the mean of the time and practice problems saved per student across all the cross-validation runs.

The tutor on `while` loops covered 9 concepts. Table 2 lists the results for `while` loop tutor. Note that most students would have saved time with the BKT model on all the concepts. Concepts 8 and 9 are on nested loops and take longer to solve: those are the concepts on which students would have saved the most time with the BKT model.

Table 2. Results of Evaluating BKT Model on `while` Loop Data

Concept	**Mean # of Students who**			**Mean Time Saved (in Minutes)**	**% of Total**	**Mean # of Problems Saved**	**% of Total**	**Across *k* runs**
	Saved Time	**Made no Difference**	**Lost Time**	**(per Student)**				
1	16.60	6.60	1.80	0.96	24	1.13	21.8	10
2	16.64	4.73	3.64	1.27	31.75	1.08	15.25	11
3	19.50	5.50	0.00	0.41	13.67	0.86	25.93	2
4	20.71	3.57	0.71	0.90	30	1.31	31.31	7
5	19.65	4.47	0.88	1.35	33.75	1.91	44.59	17
6	17.64	5.14	2.21	1.92	48	1.58	26.63	14
7	19.33	5.33	0.33	0.65	21.67	1.33	42.45	3
8	14.00	5.50	5.50	2.28	57	1.04	11.82	12
9	14.11	7.89	3.00	2.05	51.25	1.09	11.64	9
Weighted Mean	**17.26**	**5.35**	**2.39**	**1.51**	**38.62**	**1.35**	**25.43**	**Total of 85**

The tutor on `for` loops covered 10 concepts. Table 3 lists the results for `for` loop tutor. Concepts 5 and 10 are minor variations of a regular loop – these were also the concepts on which nearly as many students had no difference as saved time with the BKT model. Concept 2 is on tracing the behavior of two loops, the second loop's iterations dependent on the first. It takes longer to solve. Students saved the most time and problems on this concept.

Table 3. Results of Evaluating BKT Model on `for` Loop Data

Concept	Mean # of Students who			Mean Time Saved (in Minutes)	% of Total	Mean # of Problems Saved	% of Total	Across *k* runs
	Saved Time	Made no Difference	Lost Time	(per Student)				
1	24.08	0.92	0.00	0.90	30	1.52	27.22	13
2	16.03	6.31	2.67	2.12	53	1.78	27.53	36
3	13.00	7.13	4.88	0.26	8.67	0.62	20.9	8
4	12.13	8.10	4.77	1.48	37	1.07	15.93	31
5	10.00	9.59	5.41	0.62	20.67	0.54	8.22	17
6	24.41	0.59	0.00	1.03	34.33	1.81	50.99	37
7	12.88	6.42	5.67	1.68	42	0.93	11.8	43
8	13.80	9.80	1.40	0.35	11.67	0.92	30.67	10
9	14.20	7.97	2.83	0.63	21	0.95	18.51	30
10	12.50	11.14	1.36	0.52	17.33	0.67	16.14	14
Weighted Mean	**15.63**	**6.28**	**3.08**	**1.20**	**33.19**	**1.19**	**23.55**	**Total of 239**

We found that, on average across the two tutors, students would have saved time with the BKT model on 64.23% of the concepts. This is similar to the results of another study that recently found that using BKT models saved time (Bhatt et al 2020), although unlike them, our results were based on the use of legacy data. Students would have saved time/practice problems on some concepts more than others. The pattern that emerged is that students saved more time with BKT model on harder concepts on which it took longer to solve problems. When students neither saved nor lost time with BKT model compared to naïve mastery model, it was on simpler concepts. So, BKT model was found to be more beneficial for harder concepts than easier concepts.

For this study, we did not consider the relationships among the various concepts, i.e., we treated all the concepts as being independent and mutually exclusive. This is a fallible assumption in programming domain. In the future, we plan to use a Bayesian network to account for the relationships among these concepts.

Acknowledgements

Partial support for this work was provided by the National Science Foundation under grant DUE-1432190.

References

Baker, R.S.J., 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, Berlin, Germany. pp. 406-415.

Beck, J. (2007). Difficulties in inferring student knowledge from observations (and why you should care). Educational Data Mining: Supplementary Proceedings of the 13th International Conference of Artificial Intelligence in Education. pp. 21-30.

Beck, J.E., Chang, K-m. (2007). Identifiability: A Fundamental Problem of Student Modeling. Proceedings of the 11th International Conference on User Modeling (UM 2007). pp. 137-146.

Bhatt, S., Zhao, J., Thille, C., Zimmaro, D., Gattani, N. (2020). Evaluating Bayesian Knowledge Tracing for Estimating Learner Proficiency and Guiding Learner Behavior. L@S '20: Proceedings of the Seventh ACM Conference on Learning @ Scale. Association for Computing Machinery, New York, NY, USA, pp. 357-260.

Corbett, A.T., Anderson J.R. (1992). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction 4. pp. 253-278.

Hawkins W.J., Heffernan N.T., Baker R.S.J.D. (2014). Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. In: Trausan-Matu S., Boyer K.E., Crosby M., Panourgia K. (eds) Intelligent Tutoring Systems. ITS 2014. Lecture Notes in Computer Science, vol 8474. Springer. pp. 150-155.

Kumar, A.N. (2014). A Model for Deploying Software Tutors. IEEE 6th International Conference on Technology for Education (T4E), Amritapuri, India. pp. 3-9.