Degree of Error in Bayesian Knowledge Tracing Estimates From Differences in Sample Sizes

Stefan Slater
University of Pennsylvania
3700 Walnut St., Philadelphia PA 19104
slater.research@gmail.com

Ryan S. Baker
University of Pennsylvania
3700 Walnut St., Philadelphia PA 19104
ryanshaunbaker@gmail.com

The authors claim no conflicts of interest for this work.

Abstract (100-150 words)
Bayesian Knowledge Tracing (BKT) is a knowledge inference model that underlies many modern adaptive learning systems. The primary goal of BKT is to predict the point at which a student has reached mastery of a particular skill. In this paper we examine the degree to which changes in sample size influence the values of the parameters within BKT models, and the effect that these errors have on predictions of student mastery. We generate simulated datasets of student responses based on underlying BKT parameters and the degree of variance they involve, and then fit new models to these datasets and compared the error between the predicted parameters and the seed parameters. We discuss the implications of sample size in considering the trustworthiness of BKT parameters derived in learning settings and make recommendations for the number of data points that should be used in creating BKT models.

Keywords
Bayesian Knowledge Tracing
Knowledge Inference
Student Model
Simulated Data

Introduction

Modeling student knowledge is a critical component of modern adaptive learning systems (Desmarais & Baker, 2012). These models make inferences about what students know, which are used for several purposes. The most common use of models of student knowledge is to assess whether a student has reached mastery of a given skill, driving mastery learning (Koedinger & Corbett, 2006). They are also extensively used to understand the properties of different skills in adaptive learning systems (Ritter et al., 2009), and as components in other analyses (Hershkovitz et al., 2013). The problem of student knowledge modeling in adaptive learning systems differs from traditional contexts of psychometric application in that students are learning as they are being assessed – in other words, the student knowledge state is changing during the process of assessment itself. While psychometric methods have more recently been applied to this problem (e.g. Chen, Lee, & Chen, 2005; Pelánek, 2016; Wilson et al., 2016), the literatures of student modeling, intelligent tutoring systems, and educational data mining have artificial intelligence-based methods for modeling and assessing student knowledge in these contexts that goes back to the early 1990s. Perhaps the most popular framework for doing so is referred to as Bayesian Knowledge Tracing (BKT, Corbett & Anderson, 1995), used within a range of widely-used adaptive learning systems, including perhaps most notably the Cognitive Tutor family of curricula (Koedinger & Corbett, 2006) used by hundreds of thousands of students a year.

The BKT framework is a simple Hidden Markov Model (and also a simple Bayesian Network – Reye, 2004) that tries to estimate a single binary latent variable – has a student mastered the current knowledge component or not? The probability of a specific student's knowledge at a specific time is calculated based on data using four parameters, which are estimated separately for each distinct skill or knowledge component within the dataset:

> $p(L_0)$, the probability that a student has mastered the skill before it is attempted for the first time
>
> p(G), the probability that a student correctly answers despite being in the no-mastery state (a guess)
>
> p(S), the probability that a student incorrectly answers despite being in the mastery state (a slip)
>
> p(T), the probability that, despite an incorrect answer, a student has learned from the attempt and reached the mastery state before they reach their next opportunity to practice the skill

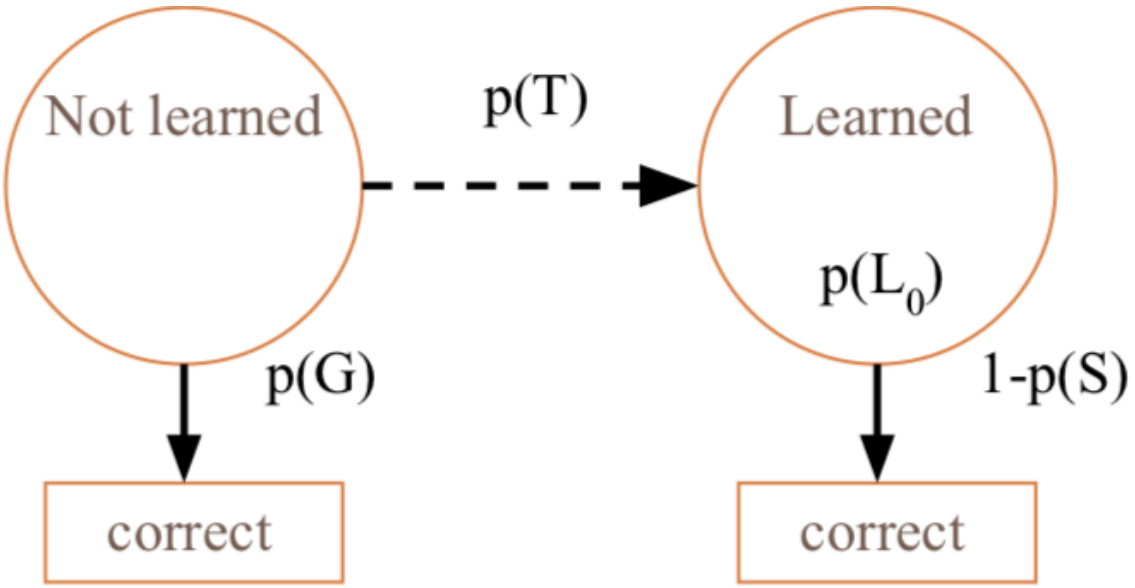

Figure 1. Bayesian Knowledge Tracing (diagram reproduced from Baker, 2015)

BKT has been successful at predicting future student performance within adaptive learning (e.g. Corbett & Anderson, 1995) and performance on later tests as well (Corbett & Anderson, 1995; Baker et al., 2010; Pardos et al., 2011). Considerable scholarly work has focused on expanding the original BKT framework in order to improve its predictive accuracy. To give just a few examples, Baker, Corbett, & Aleven (2008) generated contextual estimates of the probability that each student attempts represented a guess or a slip; this approach led to better predictive performance on some data sets but unstable, poorer performance on others. Pardos & Heffernan (2011) generated item-level parameters in order to estimate difficulty of specific problems by adding problem-specific guess and slip parameters. They found that the addition of these parameters improved the performance of the knowledge tracing model in some cases (but not in all), and suggest that the inconsistent benefit could be due to over-parametrization of the item-level model. Wang & Heffernan (2013) proposed methods for allowing partial-credit, rather than binary correct and incorrect responses. In this work, partial correctness was determined by the number of hints that a student used before attempting the problem, with a higher number of hints decreasing the percent correctness of the problem attempt. They found that the partial credit model was significantly better at predicting student performance than BKT's original formulation.

In addition to comparing different variants of BKT to one another, many papers have attempted to compare the effectiveness of BKT to other knowledge inference models. BKT typically performs comparably to models of similar complexity (e.g. Gong, Beck, & Heffernan, 2010; Baker et al., 2011). More complex extensions to BKT have performed comparably to more complicated models such as recurrent neural networks (Khajah, Lindsey, & Mozer, 2016); however, both of these approaches lose a great deal of the interpretability that BKT provides, and it has not been established whether these approaches predict long-term student performance as well as BKT. Overall, it seems likely that there is a limit to how well BKT can perform overall at predicting future

student performance. Beck and Xiong (2013) developed a series of 'cheating experiments' to identify the ideal performances of various permutations of BKT models. They found that knowledge tracing likely has a limit to its ability to predict future student performance within adaptive learning, with AUC ROC maxing out between 0.70 and 0.75.

Other work has focused on underlying assumptions of knowledge tracing models. For instance, Fancsali, Nixon, & Ritter (2013) tested the performance of BKT when applied to datasets constructed by drawing simulated student response values generated from random BKT parameters, rather than fitting students to a stable underlying set of parameters. Because classical BKT calculates parameters for individual skills, across students, it assumes that any subgroups of students in the dataset have similar initial knowledge and learning rates, and that they guess and make careless errors at similar rates. Their work found that heterogeneous datasets do impact the performance of BKT, increasing the proportions of students who are either prematurely identified as being in the mastery state, or incorrectly identified as needing additional practice. When the data sets used to generate BKT parameters are heterogeneous, which is often the case, effective estimation by BKT is harder.

Figuring out how much data is needed to estimate valid BKT parameters is a key practical question for adaptive learning system developers using BKT. In mature systems like Cognitive Tutor (Koedinger & Corbett, 2006), BKT parameters can be estimated on very large datasets – Cognitive Tutor is used by hundreds of thousands of students each year. However, BKT is used in many smaller systems as well – and in some cases, researchers have even tried fitting separate BKT models for individual students (e.g. Lee & Brunskill, 2012).

In this paper, we examine the degree to which sample size influences the estimation of BKT parameters, and in turn the predictions of student knowledge that BKT makes. Prior published work has ranged from datasets consisting of thousands of students per model (Ritter et al., 2009; Beck & Xiong, 2013) to a single student per model (Lee & Brunskill, 2012). In most situations, larger sample sizes produce better estimates and increase statistical power (Cohen, 1992), but decisions about sample sizes in student modeling are typically made using heuristics and "rules of thumb". In this paper, we examine the degree of error associated with BKT parameter estimation, across different numbers of students and lengths of problem sets.

To accomplish this, we generated multiple simulated sample datasets using various initial BKT parameters, with different numbers of students and practice opportunities. We then fit new BKT models to this data, comparing the error between the fit parameters and the initial parameters, as well as the error between fit parameters across multiple datasets generated using the same parameters. We aggregate and discuss these results, and then evaluate the degree to which differences in parameter estimates influence subsequent predictions about student knowledge. We conclude with recommendations for practice by practitioners developing models of BKT going forward.

The methods and results are presented in two sections. In the first section, we discuss our methods and findings for evaluating error in parameter estimates, and in the second we discuss our methods and findings for evaluating error in knowledge estimates.

Methods – Evaluation of Error in Parameter Estimates

BKT uses four parameters to generate its continually-updating inferences of student knowledge: $p(L_0)$, p(G), p(S), and p(T). The student starts out with a probability of $p(L_0)$ of knowing the skill (being in the mastery state). Then, for each problem the student encounters, the model checks whether the student answered the problem correctly or not. In most formulations of BKT, only the first attempt at a problem is considered evidence of whether the student knows the problem or not (Corbett & Anderson, 1995) – an exception to this is seen in Wang & Heffernan, 2013. BKT's estimate of whether the student had mastered the skill *before* attempting the problem is updated based on the student's performance on the problem, using Bayes Theorem:

$$p(L_{n-1}|ans) = \ correct * \frac{p(L_{n-1}) * \big(1 - p(S)\big)}{p(L_{n-1}) * \big(1 - p(S)\big) + \big(1 - p(L_{n-1})\big) * p(G)} + incorrect * \frac{p(L_{n-1}) * p(S)}{p(L_{n-1}) * p(S) + \big(1 - p(L_{n-1})\big) * (1 - p(G))}$$

Equation 1. Calculations for updating students' probability of mastery before attempt.

The system then calculates the probability the student had mastered the skill *after* attempting the problem. Most formulations of BKT assume that a student does not forget a skill he or she has mastered (but see Khajah et al., 2016) – however, a student who does not know the skill can learn it, with probability p(T). To get the student's probability of having mastered the skill *after* attempting the problem, we take the probability that the student was in the mastery condition at the start of the previous problem, and we add to that the probability that, if the student was not in the mastery condition, they now are after completing the problem:

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

Equation 2. Calculations for updating students' probability of mastery after attempt.

Given these four parameters, we can also calculate the probability that a student will answer a question correctly given their current probability of mastery. This probability is equal to the probabilities that the student is not in the mastery condition and guessed, plus the probability that the student is in the mastery condition and did not slip:

$$p(corr) = \big(1 - p(L_n)\big) * \ p(G) + \ p(L_n) * (1 - p(S))$$

Equation 3. Calculations for the probability of a correct answer on the next attempt.

Generating a Simulated Data Set

In this work, we chose to use simulated data rather than secondary data sources for several reasons. First of all, we wanted to know what the true parameters were that generated the data, to see how closely BKT could re-capture these parameters. With real data, it is impossible to know what student's true probability of learning or slipping is. Second, we wanted to experimentally compare and fit multiple different potential sets of model parameters, and locating a large number of different data sets that mapped to a broad and systematic range of parameters would have been a very time-consuming process. Third, we wanted to compare across multiple different sample sizes. While we could have sampled from existing datasets, we also wanted to ensure homogeneity of subjects (a concern raised for real-world data by Fancsali et al., 2013) to produce a consistent environment for calculating parameter error across underlying parameters. Simulating data also ensured that we had a perfect mapping between items and knowledge components, within the knowledge model, as an inaccurate mapping would lower the quality of our model fit for reasons other than the primary factors we are studying here.

Our first step was to generate simulated logs of student interactions with intelligent tutoring systems – lists of 1s and 0s that correspond to correctness and incorrectness across multiple problems on a given problem set. We wanted to make sure that these response patterns were derived from an underlying BKT model, so that we could compare experimental model fit to the 'ground truth' that produced the data.

We first took seed parameters that we wanted to construct the model on. Starting with these parameters, we calculated the probability that a student would answer a problem correctly based on them, using Equation 3. We then selected a random number between 0 and 1 and compared the probability of correctness to this random value. If p(corr) was lower, we said that the student got the problem wrong, and if it was equal to or greater, the student got the problem right. This allowed us to convert continuous probabilities of correctness from p(correct) to the binary 0/1 response necessary to fit a BKT model. We then adjusted student knowledge using the equations listed above and continued the process until we had generated a data set with the requisite number of responses and students.

Using this data set, we fit a new BKT model to our simulated data. We used the L-BFGS-B function minimization process in Python 2.7's SciPy library to obtain our estimates, using 30 random starts for each estimation of the parameters. We then compared the parameter estimates of the fitted model to the seed parameters used to construct the simulated response values. For each unique set of seed parameters, we repeated this process 30 times, each with a unique set of generated responses. We then calculated averages, deviations, errors, and ranges of the fitted parameter estimates for each set of seed parameters.

We generated a total of 51,030 unique datasets, and fit a BKT model to each. The parameters and conditions that we varied are in Table 1.

Parameter	Values
Sample Size	5, 10, 25, 50, 100, 500
Problem Set Length	3, 6, 10
$p(L_0)$	0.10, 0.25, 0.60
p(G)	0.02, 0.10, 0.25
p(S)	0.02, 0.10, 0.25
p(T)	0.05, 0.15, 0.40

Table 1. Parameters and sample sizes used for simulation trials.

We chose our parameter values for simulated data generation to represent a range of values, subject to reasonable limits. For $p(L_0)$ we chose a minimum value very close to 0. This would represent a sample of students for whom the skill being learned is relatively novel – there is a very small probability that any given student has mastered the skill being taught before seeing the first problem. For an upper value, we chose $p(L_0) = 0.60$ as a relatively high level of prior knowledge. A $p(L_0)$ of 1 would be meaningless to model within BKT, as it would represent a test of understanding for students who have already reached mastery – only the slip parameter would be relevant.

For p(G) and p(S), it is necessary to limit values to avoid problems of model degeneracy, where getting a problem wrong is more evidence for mastery than getting it right (Baker, G, & Aleven, 2008). Past research efforts have proposed varying upper limits for p(G) and p(S), such as p(G) + p(S) <= 1 (Beck, 2014), p(G) and p(S) each <= 0.50 (Baker, Corbett, & Aleven, 2008), and p(S) = 0.10, p(G) = 0.30 (Corbett & Anderson, 1995). In this paper, during model generation, we chose to limit P(G) and P(S) to being no higher than 0.25. This both avoids model degeneracy, and avoids the "high-guess/high-slip" parameters seen mostly in systems with poorly-fit skill models, items of highly variant difficulty, or very careless students (see discussion in Baker et al., 2010). For the modeling process, we used the more liberal thresholds of 0.50 for both p(G) and p(S), in order to determine if data sets with relatively limited p(G) and p(S) can nonetheless yield models with high p(G) or p(S) values. Model degeneracy was prevented with the threshold chosen, as Pardos & Heffernan (2010) find that degenerate values can often fit data equally as well as appropriate values, simply by flipping the link between knowledge and performance.

For p(T), we picked relatively lower parameters than for $p(L_0)$ for data generation because low rates of learning are common in intelligent tutoring systems, and in many learning contexts in general. Despite some arguments that students are not making progress if they need 10 or more attempts to learn a skill (e.g. Beck & Gong, 2013), some skills legitimately need a considerable amount of practice to master. Additionally, BKT models with high p(T) rapidly approach mastery, and leave little variance to be modeled. A maximum p(T) of 0.40 allows for a high rate of learning while still producing an amount of data that can be analyzed for trends. The $p(L_0)$ and p(T) parameters were allowed to take on any values between 0 and 1 during model fitting.

Results – Evaluation of Error in Parameter Estimates

Confirming Degree of Model Convergence to Seed Parameters

Our first step in studying our simulations is to confirm that our fitted models converged on the seed parameters used to construct the data. If fitted BKT models don't generally produce the same parameters as the original seed parameters, this would indicate a problem in either the simulation or the fitting procedure – or a serious problem in BKT itself. Table 2 shows the average deviation of the parameter estimates from the seed parameters, averaged across all 30 trials and across all parameters (standard deviations will be given in the following section). In other words, we took the average of the averages of each parameter estimate, per sample size and problem set length. n_s is the number of simulated students in the sample, while n_p is the number of simulated problems.

	$n_p = 3$	$n_p = 6$	$n_p = 10$
$n_s = 5$	0.116	0.078	0.070
$n_s = 10$	0.040	0.027	0.013
$n_s = 25$	0.008	0.008	~ 0
$n_s = 50$	0.004	-0.001	0.002
$n_s = 100$	0.007	0.001	-0.001
$n_s = 250$	~ 0	0.002	~ 0
$n_s = 500$	0.003	0.001	~ 0

Table 2. Average deviation from seed parameters for each block of trials.

As Table 2 shows, $n_s = 5$ and $n_s = 10$ do not appear to converge to the seed parameters, while n (s) = 25 and up did. This finding suggests that our BKT-based simulations are indeed producing data which can be modeled using BKT for sample sizes of at least 10. For $n_s = 5$ and $n_s = 10$, however, this approach does not appear to work – the average of parameter estimates across 30 trials is not the same as the seed parameters used to generate the data. Therefore, we do not report findings for $n_s = 5$ and $n_s = 10$ in the rest of the paper.

After confirming that our simulation process had worked, we next examined the differences in parameter estimates across different numbers of students and problem set lengths. Table 3 shows the standard deviations of our estimates of $p(L_0)$, p(G), p(S), and p(T) for each block of 30 simulations that we conducted. For these tests, we wanted to determine the amount of variance for each parameter within the 30 trials that we ran for each condition.

	$p(L_0) = 0.10$	$p(L_0) = 0.25$	$p(L_0) = 0.60$
$n_s = 25$	0.105	0.136	0.159
$n_s = 50$	0.091	0.105	0.117
$n_s = 100$	0.068	0.079	0.088
$n_s = 250$	0.044	0.053	0.059
$n_s = 500$	0.033	0.038	0.043

	p(G) = 0.02	p(G) = 0.10	p(G) = 0.25

$n_s = 25$	0.046	0.078	0.121
$n_s = 50$	0.032	0.058	0.100
$n_s = 100$	0.021	0.044	0.076
$n_s = 250$	0.014	0.030	0.051
$n_s = 500$	0.011	0.022	0.038

	p(S) = 0.02	p(S) = 0.10	p(S) = 0.25
$n_s = 25$	0.052	0.076	0.111
$n_s = 50$	0.040	0.062	0.094
$n_s = 100$	0.028	0.046	0.076
$n_s = 250$	0.017	0.031	0.055
$n_s = 500$	0.015	0.024	0.041

	p(T) = 0.05	p(T) = 0.15	p(T) = 0.40
$n_s = 25$	0.080	0.107	0.146
$n_s = 50$	0.050	0.075	0.108
$n_s = 100$	0.034	0.049	0.077
$n_s = 250$	0.021	0.033	0.049
$n_s = 500$	0.015	0.022	0.035

Table 3. Standard deviations for parameter estimates of $p(L_0)$, p(G), p(S), and p(T) across different student sample sizes.

Table 3 shows several effects. First, the standard deviation of parameter estimates decreases as student sample size increases. Larger numbers of students to fit a BKT model to leads to less error in parameter estimates. Second, increasing parameter values increases prediction error. The difference between the lowest and highest $p(L_0)$ that we tested was relatively small, 0.01 at $n_s = 500$, while error from the lowest to highest values of p(G), p(S), and p(T) doubled. Finally, we found that estimates for $p(L_0)$ seem to be the most error-prone, with generally higher standard deviations than the other parameters we tested.

Evaluating Differences in Parameter Estimates Based on Problem Set Length

In addition to varying the number of students simulated, we also varied the number of opportunities to practice a skill that a given student has. Longer problem sets provide more information that the BKT model can use to infer student learning and derive appropriate parameters. Table 4 shows the standard deviations for each of the four BKT parameters across each condition of problem set length that we tested. As with Table 3, each cell consists of a sample of 30 simulated datasets.

	$p(L_0) = 0.10$	$p(L_0) = 0.25$	$p(L_0) = 0.60$
$n_p = 3$	0.126	0.148	0.168
$n_p = 6$	0.105	0.128	0.153
$n_p = 10$	0.095	0.122	0.149

	p(G) = 0.02	p(G) = 0.10	p(G) = 0.25

$n_p = 3$	0.060	0.086	0.127
$n_p = 6$	0.058	0.074	0.107
$n_p = 10$	0.051	0.072	0.097
	p(S) = 0.02	p(S) = 0.10	p(S) = 0.25
$n_p = 3$	0.094	0.113	0.144
$n_p = 6$	0.052	0.068	0.090
$n_p = 10$	0.033	0.043	0.061
	p(T) = 0.05	p(T) = 0.15	p(T) = 0.40
$n_p = 3$	0.131	0.164	0.187
$n_p = 6$	0.093	0.115	0.148
$n_p = 10$	0.075	0.094	0.140

Table 4. Standard deviations for parameter estimates of $p(L_0)$, p(G), p(S), and p(T) across different problem set lengths.

There are several differences between how problem set lengths and student sample size each impact the stability of parameter estimates. While longer problem sets were associated with more stable parameter estimates, the effect was not as large as for increases in student sample size. Additionally, the standard deviation of estimates for p(T) remained particularly high compared to changes in student sample size. This suggests that BKT estimation is most effective if students complete more problems, and indeed that BKT may perform even better if students complete more than 10 problems. However, many researchers argue that learning systems should be designed to avoid having students work for more than 10 problems on the same skill (e.g. Beck & Gong, 2013).

Identifying Instances of Extreme Parametrization

Finally, we wanted to examine unexpected outcomes that we noticed in the earlier analyses. Even for relatively high student sample sizes and problem set lengths, some simulated datasets produced parameter estimates that were unexpectedly large – reaching the maximum bound adopted during estimation. This phenomenon is often seen in real-world use of BKT, but was previously often assumed to represent a flaw in the skill being studied rather than a property of BKT.

We aggregated the number of times a simulation produced an extreme parameter, and these counts are shown in Table 5. We define an extreme parameter here as one which reaches the bounds of the model ($p(L_0)$ or p(T) = 0 or 1, p(G) or p(S) = 0 or 0.50), in the direction *away* from the seed parameter. That is, an extreme parameter for $p(L_0) = 0.10$ would be 1 (but not 0.01), while an extreme parameter for $p(L_0) = 0.60$ would be 0.01 (but not 1). For middle parameters, such as p(S) = 0.10, we counted both the lower extreme and upper extreme as extreme parameter estimates. Table 5 shows the percent of all simulated trials of a given student sample size and problem set length (n = 2430) which produced either an extremely low or extremely high parameter estimate.

	$n_p = 3$		$n_p = 6$		$n_p = 10$	
$p(L_0)$	Low	High	Low	High	Low	High
$n_s = 25$	12.67%	< 1%	12.51%	None	10.86%	None
$n_s = 50$	4.69%	< 1%	5.19%	None	5.76%	None
$n_s = 100$	3.13%	< 1%	2.30%	None	2.14%	None
$n_s = 250$	< 1%	None	< 1%	None	< 1%	None
$n_s = 500$	< 1%	None	< 1%	None	< 1%	None
p(G)						
$n_s = 25$	41.56%	< 1%	34.90%	< 1%	31.03%	< 1%
$n_s = 50$	34.57%	< 1%	27.98%	< 1%	23.54%	< 1%
$n_s = 100$	26.67%	< 1%	17.04%	None	15.23%	< 1%
$n_s = 250$	20.49%	None	11.03%	None	7.90%	None
$n_s = 500$	13.83%	< 1%	5.76%	None	5.84%	None
p(S)						
$n_s = 25$	38.68%	3.00%	30.04%	< 1%	24.86%	None
$n_s = 50$	36.30%	1.85%	25.02%	< 1%	19.84%	None
$n_s = 100$	30.37%	1.11%	19.51%	None	13.17%	None
$n_s = 250$	25.51%	< 1%	11.60%	None	6.01%	None
$n_s = 500$	21.15%	< 1%	6.67%	None	2.06%	None
p(T)						
$n_s = 25$	23.12%	< 1%	13.50%	< 1%	4.94%	< 1%
$n_s = 50$	16.63%	< 1%	6.01%	< 1%	1.23%	None
$n_s = 100$	11.93%	< 1%	3.46%	None	< 1%	None
$n_s = 250$	6.34%	None	< 1%	None	None	None
$n_s = 500$	3.87%	None	None	None	None	None

Table 5. Number of cases with extreme parameter values (out of 2430) by parameter, student sample size, and problem set length.

Findings from Table 5 suggest that longer problem sets are less susceptible to extreme parametrization estimates than shorter problem sets. A similar effect was observed for greater student sample size, however, even with large student sample sizes, extreme parameters remained common at $n_p = 3$ and $n_p = 6$. Increasing problem set length appears to result in a greater reduction in extreme parametrizations than increasing student sample size. Additionally, extreme parametrizations of p(G) and p(S) were somewhat more common than for $p(L_0)$ and p(T). This could either be due to more variance being inherent in predictions of p(G) and p(S), or it could be a consequence of the narrower range of values that these parameters can take in most BKT models, to avoid model degeneracy.

Methods – Evaluation of Error in Knowledge Estimation

The preceding analyses showed that model parameters can deviate considerably from the original "ground truth" in the data simulation, for small numbers of students, and especially for small amounts of data per student. However, these deviations in values do not entirely communicate the practical significance of these model errors. The larger question is how effective the models are at predicting when a student reaches mastery. To put it another way, how much does a larger standard deviation for a specific parameter translate into differences in the point at which a student is predicted to master a skill? How sensitive are BKT mastery predictions to variance in parameter estimates?

Once we had completed the process of data generation and model fitting, we computed the variance in estimations of the point of student mastery. In other words, we wanted to know the degree to which the models varied in their inferences about whether a student had reached mastery based on changes in the underlying parameters.

To accomplish this, we conducted a second simulation, taking a similar approach to Fancsali, Nixon & Ritter (2013). This approach starts by using the $p(L_0)$ parameter according to its original semantic meaning in (Corbett & Anderson, 1995); as an indicator of what proportion of students have mastered the skill before starting to work in the system. We randomly assign simulated students to the mastery or no mastery state, according to this probability. For instance, $p(L_0) = 0.30$ means that 30% of a sample of simulated students begin in the mastery state. Then, for each opportunity to practice the skill, an additional proportion of students who have not yet mastered transition to the mastery state *after* completing the problem. By the definition of BKT, this proportion is equal to p(T). Once we have determined when each student reached mastery, correct and incorrect answers can be generated – a student in the mastery state generates correct answers with probability (1-p(S)), while a student in the no mastery state generates correct answers with probability p(G). Following the generation of this simulated dataset we fitted a new BKT model to the student correctness data, and compared when the ground truth model indicated that a student reached mastery to when the fitted model inferred that the student had reached mastery.

As Fancsali and colleagues (2013) note, there will inherently be some disagreement between when the student reaches mastery and when the model infers that they have reached mastery. Generally, BKT is unable to predict that a student is in the mastery condition from the first attempt, except for abnormally high values of $p(L_0)$. Fancsali and colleagues term this the 'acceptable lag' – the period of time where BKT is unable to make a prediction of mastery because insufficient data has been obtained about a specific student to make a valid inference about that student. In Fancsali et al. (2013), this is operationalized as the theoretical minimum value that $p(L_n)$ can reach, given a string of infinite incorrect answers. For this paper, we use a slightly different operationalization: we define the theoretical minimum $p(L_n)$ for a student as p(T). We choose this value for the following reason: After a given problem, a student with $p(L_n) = 0$ has a probability p(T) of transitioning to the mastery state. We chose to use this particular approach because we wanted to rely on knowing the state of the hidden node as much as possible, rather than inferring it from data, and $p(L_n) = 0$ reflects a student who is not in the mastery state.

From there, we calculate the number of consecutive correct answers necessary to go from $p(L_n) = p(T)$ to $p(L_n) = 0.95$. The number of attempts this process takes is what we define as the acceptable lag, and we subtract this acceptable lag from the difference between ground truth mastery and the model's predictions of mastery. We use $p(L_n) = 0.95$ as it is a commonly-used (almost always-used in systems that use BKT) threshold for determining whether a student has reached mastery.

Conducting such an analysis achieves two goals; first, it assists in establishing a principled recommendation for the number of problems and students that should be used to conduct BKT analyses. Second, this analysis establishes whether extreme parameter values observed are a serious problem: how much of an impact do these extreme parameter values have on predictions of mastery? Using the procedure described above, we generated a single simulated data set using a set of parameters that might reflect a 'normal' knowledge component: $(L_0) = 0.25$, $p(G) = 0.10$, $p(S) = 0.10$, $p(T) = 0.15$. For this set of analyses, using a single dataset allowed us to compare differences in knowledge estimation between models without variance between multiple different simulated datasets that used the same underlying parameters.

We then performed a series of trials to fit new models onto this simulated dataset. We varied the degree of 'error' present in the models by choosing a range of parameters centered around the ground truth used to generate the original dataset. For $p(L_0)$ we used the parameters {0.01, 0.10, 0.20, 0.22, 0.25, 0.28, 0.30, 0.50, 0.75}, for p(G) and p(S) we used the parameters{0.01, 0.05, 0.07, 0.09, 0.10, 0.11, 0.13, 0.15, 0.25, 0.45}, and for p(T) we used the parameters{0.01, 0.05, 0.10, 0.12, 0.15, 0.18, 0.20, 0.25, 0.35}. For each of these models, we took the difference between a student's true point of mastery in the simulated data, and that student's estimated point of mastery in the fit model, accounting for the acceptable lag in each model. This produced a measure for each student of the error between the ground truth and the model equal to the model prediction, minus the ground truth observation, minus the acceptable lag. We then calculated the standard deviation of these values. We performed these trials for each parameter, while holding all other parameters constant at their seed parameter value – for each value of $p(L_0)$ that we tested, p(S) and p(G) were fixed at 0.10, and p(T) was fixed at 0.15.

Results – Evaluation of Error in Knowledge Estimation

To avoid parameter instability due to insufficient data, we simulated 5000 students; to give the simulation enough data for most of the students to reach mastery, we simulated 20 problems per student. Nonetheless, it is possible for students in the simulated model to never reach mastery, and for the fitted model to fail to predict mastery. The simulated model, using ground truth parameters, contained 673 students who did not reach mastery, 13.5% of the sample; because of different ground truth parameters, the number of students predicted to not reach mastery in the fitted model changed for each trial of the simulation. Data of students who reached mastery, and for whom the models did predict mastery, are presented in Table 6.

$p(L_0)$	Students who reached mastery	SD of prediction error	p(G)	Students who reached mastery	SD of prediction error
0.01	4965	1.367	0.01	4999	2.880
0.10	4965	1.395	0.05	4969	1.387
0.20	4965	1.395	0.07	4967	1.393
0.22	4965	1.395	0.09	4966	1.393
0.25	**4965**	**1.395**	**0.10**	**4965**	**1.395**
0.28	4965	1.395	0.11	4965	1.395
0.30	4965	1.395	0.13	4945	1.064
0.50	4965	1.400	0.15	4943	1.089
0.75	4965	1.397	0.25	4926	1.375
			0.45	4790	1.930

p(S)	Students who reached mastery	SD of prediction error	p(T)	Students who reached mastery	SD of prediction error
0.01	4965	1.431	0.01	4881	1.489
0.05	4965	1.395	0.05	4933	1.223
0.07	4965	1.395	0.10	4945	1.064
0.09	4965	1.395	0.12	4950	1.035
0.10	**4965**	**1.395**	**0.15**	**4965**	**1.395**
0.11	4965	1.395	0.18	4966	1.393
0.13	4966	1.393	0.20	4966	1.393
0.15	4966	1.390	0.25	4969	1.379
0.25	4969	1.386	0.35	4974	1.575
0.45	4969	1.288			

Table 6. Standard deviations of differences between time when simulated student reached mastery and when model estimates mastery reached. Seed parameters are given in boldface.

Table 6 suggests that there are differences in error of mastery predictions based on differences in parameter estimates. The 'base rate' for prediction error in BKT seems to be about 1.4 opportunities to practice – even at the seed parameters, there was an error of 1.395 problems between the ground truth mastery point and the model estimation of mastery. For $p(L_0)$, this error rate stayed quite stable, and there was very little difference in prediction error whether $p(L_0)$ was very small or very large. Very low and very high values of p(G) led to higher error rates, while p(G) of 0.13 and 0.15 produced less error than the seed parameter value. For p(S), very low values led to higher error rates, while very high values of p(S) led to less error than seed parameter values. For p(T), very low and high values both led to higher error rates, while the lowest error rates were at p(T) = 0.10 to 0.12.

Discussion

Variance in Model Parameters Based on Sample Sizes

Our analyses suggest a high degree of variance in the parametrization of BKT models based on both the number of students included in a dataset as well as the number of problems included in a problem set. We found that there is about four times as much error in parameter estimations for $n_s = 25$ compared to $n_s = 500$, and about twice as much error in parameter estimations for $n_p = 3$ compared to $n_p = 10$.

Additionally, we found differences in parameter estimation based on the value of the underlying parameters being used. As a general trend, larger parameter values appear to have more error associated with subsequent predictions; an effect that seems to occur largely independent from changes in n_s and n_p. This suggests that learning environments or skills where students are very likely to understand the material before starting or where students receive strong scaffolding between problems may not be as appropriate for applying BKT. Because these settings are likely to produce high estimates of $p(L_0)$ and p(T), respectively, they are more prone to high error rates in predictions of parameters and estimates of student knowledge. On the other hand, errors in these contexts – where students need minimal practice -- are more likely to produce over-practice than under-practice, and the correct performance these already-mastered students achieve suggests that they will soon reach mastery anyways.

Sample Sizes Needed to Avoid Extreme Parameters

One potential issue that we noticed with fitting these models is that BKT has a tendency in some cases to produce extreme parameter values, such as a p(T) = 1 for an initial parameter of p(T) = 0.15, in some cases. These extreme parameter values are problematic because of the interpretations of learning that they lead to – for instance, a skill with p(T) = 1 is a skill where every student, without fail, attains mastery of the skill they are learning after completing a single problem. High values of p(G) and p(S) present similar problems. A model with p(G) = 0.50 reflects a model where students in the no mastery state still answer correctly half the time; a model with p(S) = 0.50 reflects a model where students in the mastery state still answer incorrectly half the time; a model with p(G) = 0.50 *and* p(S) = 0.50 reflects a chance model, where nothing about mastery can be inferred; and when p(S) or p(G) go above 0.5, students who have not mastered a skill are expected to perform better than those who have. Extreme parameters are common when applying BKT to small numbers of students and small numbers of problems completed per student, in specific cases occurring up to 42% of the time. However, they are much rarer when BKT is applied to larger numbers of students who completed more problems. The number of students in the data set appears to play a larger role in whether extreme parameter values are seen than how many problems students receive. For p(G) and p(S) however, even at a student sample size of 500 and a problem set length of 10, there was still a considerable number of trials that resulted in extreme parameters. Even larger sample sizes and problem set lengths may be necessary to ensure that parameter values are reliable for a specific dataset; but more than 10 problems completed is not desired for many learning systems, making this goal hard to achieve.

Sample Size Recommendations Based on Mastery Predictions

Perhaps the best indicator of whether a set of parameters is problematic is how much impact it has on the number of problems the student receives before being assessed to have mastery. As such, this paper also examines the degree to which variance in parameter estimates led to variance in predictions of mastery for students. We found very little difference in prediction errors for $p(L_0)$, regardless of its estimation. Since $p(L_0)$ is replaced by estimates of $p(L_n)$ after the first opportunity to practice a skill, based on actual student performance, it is perhaps unsurprising that $p(L_0)$ does not influence mastery predictions much – as a problem set gets longer the influence of $p(L_0)$ rapidly decreases. We can see a similar range of values for p(G), with p(G) = 0.05 to p(G) = 0.25 all producing similar error rates in estimates of student mastery. Once the model starts moving into higher parameter ranges, like p(G) = 0.45, the error in mastery predictions starts to increase, reaching a standard deviation of nearly two opportunities to practice. For p(S), however, rates of error actually *improve* as p(S) approaches 0.45. p(T) has the narrowest window of optimal parameters out of all four, from about p(T) = 0.10 to p(T) = 0.12.

Summary

Our analyses suggest a set of criteria that BKT analyses should meet. If the primary goal of an analysis is to predict future correctness, it appears to be feasible to estimate BKT parameters on as few as 25 students, provided that parameter values are relatively low (Table 2). If the goal of the BKT model is to predict student mastery, at least 25 students and at least 3 opportunities to practice a given skill seems to be sufficient. At these sample sizes, even large error rates in parameter estimates will not change mastery estimates by more than two or three problems. If a lower degree of error is desired due to having relatively lengthy problems for students to complete, error in mastery estimates can be brought under two problems for low values of P(T) by fitting models to data from 250 students, and for high values of P(T) by fitting to data from 500 students.

However, if the goal of the model is to make inferences about properties of the underlying skills, significantly larger sample sizes appear to be necessary. Parameter error continues to decrease through even up to samples of 500 students and 10 problem completed per skill for the student, and even larger sample sizes and problem set lengths may produce better estimates. Additionally, to avoid extreme parameter estimates, sample sizes of at least 250 students and 6 problems per skill per student appear warranted.

Future Work

Although the work presented here establishes some findings about the robustness of BKT, and the sample sizes needed, it may be valuable for future work to also investigate the effects of even larger numbers of students on parameter estimates. We chose a maximum of $n_s = 500$ in this paper, due to the goal of establishing the consequences of the small samples often used to develop BKT models for new adaptive learning systems. However, some larger adaptive learning systems have data available for thousands or even tens of thousands of students. It's currently unclear how much more improvement can be obtained to BKT's properties under these conditions, an area of potential relevance to more widely-used adaptive learning systems.

It is also worth considering how these findings apply when considering extensions to the BKT framework, to examine whether other variants of BKT offer less error in predictions of parameters or of mastery. For predictions of student mastery, even when ground truth parameters and fitted parameters were matched, there were still errors in estimations of mastery of about a problem and a half. It's unclear whether this error reflects the ceiling of BKT's accuracy (e.g. Beck & Xiong, 2013) or whether other variants can offer better accuracy in their predictions. Though BKT remains the most widely-used student knowledge modeling algorithm in adaptive learning, other algorithms such as Performance Factors Analysis (PFA), ELO, and Deep Knowledge Tracing (DKT) may have different impacts from changes in sample size.

By investigating more thoroughly how sample size interacts with the reliability of student knowledge model parameters and predictions, we can better understand the practical implications of different sample sizes. This in turn can influence practice in when BKT is applied in a learning system's life-cycle, and ensure that its use is principled and valid – producing better pedagogical decisions and hopefully better student outcomes as well.

References

Baker, R.S. (2015) Big Data and Education. 2nd Edition. New York, NY: Teachers College, Columbia University.

Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., MacLaren, B.M., Kauffman, L.R., Mitchell, A.P., Giguere, S. (2010) Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization, 52-63.

Baker, R. S., Corbett, A. T., & Aleven, V. (2008). More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems*, 406-415.

Baker, R.S.J.d., Pardos, Z., Gowda, S., Nooraei, B., Heffernan, N. (2011) Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems. *Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization*, 13-24.

Beck, J. (2014). The Field of EDM: Where we Came From, and Where We're Going. Presentation at the 7th International Conference on Educational Data Mining.

Beck, J. E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. In *International Conference on Artificial Intelligence in Education*, 431-440.

Beck, J., & Xiong, X. (2013). Limits to accuracy: how well can we do at student modeling?. In *Educational Data Mining 2013*.

Chen, C. M., Lee, H. M., & Chen, Y. H. (2005). Personalized e-learning system using item response theory. *Computers & Education*, *44*(3), 237-255.

Cohen, J. (1992). Statistical power analysis. *Current directions in psychological science*, *1*(3), 98-101.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*(4), 253-278.

Desmarais, M.C., Baker, R.S.J.d. (2012) A Review of Recent Advances in Learner and Skill Modeling in Intelligent Learning Environments. *User Modeling and User-Adapted Interaction*, 22 (1-2), 9-38.

Fancsali, S., Nixon, T., & Ritter, S. (2013). Optimal and worst-case performance of mastery learning assessment with bayesian knowledge tracing. In *Educational Data Mining 2013*.

Gong, Y., Beck, J. E., & Heffernan, N. T. (2010). Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *International conference on intelligent tutoring systems*, 35-44. Springer, Berlin, Heidelberg.

Hershkovitz, A., Baker, R. S. J.d., Gobert, J., Wixon, M., & Pedro, M. S. (2013). Discovery with models: A case study on carelessness in computer-based science inquiry. *American Behavioral Scientist*, *57*(10), 1480-1499.

Khajah, M., Lindsey, R. V., & Mozer, M. C. 2016. How Deep is Knowledge Tracing? In *Proceedings of the 2016 Educational Data Mining Conference*.

Koedinger, K. R., & Corbett, A. (2006). *Cognitive tutors: Technology bringing learning sciences to the classroom*. *Cambridge Handbook of the Learning Sciences*, 61-71.

Lee, J. I., & Brunskill, E. (2012). The Impact on Individualizing Student Models on Necessary Practice Opportunities. *International Educational Data Mining Society*.

Pardos, Z. A., Gowda, S. M., Baker, R.S.J.d., Heffernan, N. T. (2011) Ensembling Predictions of Student Post-Test Scores for an Intelligent Tutoring System. In *Proceedings of the 4th International Conference on Educational Data Mining*, 189-198.

Pardos, Z. A., & Heffernan, N. T. (2011). KT-IDEM: introducing item difficulty to the knowledge tracing model. In *International Conference on User Modeling, Adaptation, and Personalization*, 243-254.

Pelánek, R. (2016). Applications of the Elo rating system in adaptive educational systems. *Computers & Education*, *98*, 169-179.

Reye, J. (2004). Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education*, *14*(1), 63-96.

Ritter, S., Harris, T. K., Nixon, T., Dickison, D., Murray, R. C., & Towle, B. (2009) Reducing the Knowledge Tracing Space. In *Proceedings of the Educational Data Mining Conference 2009*, 151-159.

Wang, Y., & Heffernan, N. (2013). Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *International Conference on Artificial Intelligence in Education*, 181-188.

Wilson, K. H., Karklin, Y., Han, B., & Ekanadham, C. (2016) Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. *Proceedings of the 2016 Educational Data Mining Conference*, 539-544.