Parametric Constraints for Bayesian Knowledge Tracing from First Principles

Denis Shchepakin, Sreecharan Sankaranarayanan, and Dawn Zimmaro
Amazon.com, Inc., USA
{dshch, sreeis, dzimmaro}@amazon.com

ABSTRACT

Bayesian Knowledge Tracing (BKT) is a probabilistic model of a learner's state of mastery for a knowledge component. The learner's state is a "hidden" binary variable updated based on the correctness of the learner's responses to questions corresponding to that knowledge component. The parameters used for this update are inferred/learned from historical ground truth data. For this, BKT is often represented as a Hidden Markov Model and the Expectation-Maximization algorithm is used to infer the parameters. The algorithm, can however, suffer from issues including settling into local minima, producing degenerate parameter values (such as stating that learners who do not know the skill are more likely to answer correctly than those who do), and a high computational cost during fitting. To address these, we take a "from first principles" approach to derive necessary constraints that can be imposed on the BKT parameter space. Starting from the basic mathematical truths of probability and using conceptual behaviors expected of the BKT parameters in real systems, we derive succinct constraints for the BKT parameter space. As necessary conditions, applying the constraints prior to fitting reduces computational cost and the issues emerging from the EM procedure. We further introduce a novel algorithm for estimating BKT parameters subject to the newly defined constraints. While the issue of degenerate parameters has been reported previously, this paper is the first, to the best of our knowledge, to derive necessary constraints from first principles and also present an algorithm that respects those constraints.

Keywords

Adaptive Learning Systems, Student Modeling, Learner Modeling, Bayesian Knowledge Tracing, Hidden Markov Model, Expectation-Maximization

1. INTRODUCTION

Bayesian Knowledge Tracing (BKT) [6] was introduced as a way to model the changing knowledge states of students who were interacting with an adaptive learning system for skill acquisition. To this day, it remains the most widespread way to model student learning, predominantly owing to its sufficient complexity for many use cases [7, 11, 15, 16]. The model considers the learner's state of mastery as a "hidden" or latent binary variable with two possible states – Mastery and Non-Mastery (Sometimes called Proficient and Not-yet-proficient [4, 14], or Knows and Does-not-know [10]). It then uses four parameters – the initial probability of mastery, the transition probability from non-mastery to mastery over a single learning opportunity, the probability of a correct answer with the learner in the non-mastery state (guess), and the probability of an incorrect answer with the learner in the mastery state (slip), to "predict" (i.e., calculate) whether a given learner is in the mastery state or not. In order to learn the value of these parameters, BKT is most often represented as a Hidden Markov Model [3], and the parameters are determined using an Expectation-Maximization (EM) algorithm [5, 8] on historical data.

As with any other EM algorithm, this may result in multiple sets of (highly dissimilar) parameters that fit the data equally well [3] which, in the case of BKT, affects interpretability. Further, the parameters may be degenerate, i.e., fit the data but violate the conceptual meaning, such as by stating that a learner is more likely to answer correctly if they don't know the skill than if they do [1]. This can lead to incorrect decision-making in real systems. Finally, if the algorithm needs to be re-run after it is post-hoc determined to produce degenerate parameters, then that becomes computationally expensive. Several approaches have been suggested that can partially resolve the issue, including determining the starting values that lead to degenerate parameters [17] (and avoiding them), computing Dirichlet priors for each parameter and using that to bias the search [24], clustering parameters across similar skills [12, 21], and machine-learned models for some of the parameters [1]. Approaches that provably avoid degenerate parameters have also been discussed in literature, but they instead sacrifice the precision provided by the EM algorithm [10]. Thus, in this paper, we first derive parametric constraints for the BKT parameters from first principles, that, if satisfied, necessarily avoid degenerate parameters. Then, we present a novel EM algorithm that respects the derived constraints thus allowing them to be used in practice.

While similar constraints have previously emerged by studying fixed points of the BKT model [23], here we derive them from first principles applied to the conceptual meaning of the modeled process. Moreover, we prove that these less strict

constrains are sufficient compared to the ones derived in [23]. Finally, we also present a novel EM algorithm that respects these constraints.

2. DEFINING THE BKT MODEL

BKT assumes that for each knowledge component (KC), a learner can be in either the Proficient or Not-yet-proficient state at a given point in time. After attempting an assessment, the learner receives feedback, either explicitly or gleans it from the fact that their response was marked correct or incorrect. Thus, this is an opportunity to become proficient in the corresponding knowledge component. If the learner learns successfully, they transition from the not-yet-proficient state to the proficient state for that corresponding KC. Once the learner becomes proficient in a KC, they cannot transition back to the not-yet-proficient state (Note that variations of the BKT model that incorporate "forgetting" allow this transition [13]). A BKT model is then constructed and applied for each KC independently.

Let $L_t^{(d)}$ be an event that learner d is proficient after receiving t rounds of feedback; $C_t^{(d)}$ is an event that learner d answers assessment t correctly; $G_t^{(d)}$ is an event that a learner d guesses a correct answer for an assessment t while not being proficient; $S_t^{(d)}$ is an event that a learner d makes a mistake ("slips") at an assessment t while being proficient; and $R_t^{(d)}$ is an event that a non-proficient learner d transitions to a proficient state after receiving t rounds of feedback. The classic BKT model assumes that probabilities of guess, slip, and transition events are independent of the learner and the assessment and depend only on the learner's proficiency state. Moreover, the initial proficiency probability $P(L_0^{(d)})$ is also assumed to be independent from the learner, and, rather, a proportion of learners in the population that are proficient before attempting their first assessment is used. So, we will omit redundant upper and lower indexes for G, S, R, and L_0 events and their probabilities. See Figure 1 for an outline of the model.

The BKT model defines $P(C_{t+1}^{(d)})$ thus –

$$P(C_{t+1}^{(d)}) = P(L_t^{(d)}) \cdot (1 - P(S)) + (1 - P(L_t^{(d)})) \cdot P(G) \quad (1)$$

Using the Bayes' rule, we get –

$$\begin{aligned} P(L_t^{(d)}|C_{t+1}^{(d)}) &= \frac{P(L_t^{(d)}) \cdot (1-P(S))}{P(L_t^{(d)}) \cdot (1-P(S)) + (1-P(L_t^{(d)})) \cdot P(G)}, \\ P(L_t^{(d)}|\overline{C_{t+1}^{(d)}}) &= \frac{P(L_t^{(d)}) \cdot P(S)}{P(L_t^{(d)}) \cdot P(S) + (1-P(L_t^{(d)})) \cdot (1-P(G))} \end{aligned} \quad (2)$$

where $\overline{C_t^{(d)}}$ is an event complementary to $C_t^{(d)}$, i.e., an event that learner d answers assessment t incorrectly. After an attempt at the assessment and receiving a feedback, the learner has a chance of transitioning if they are not already proficient –

$$P(L_{t+1}^{(d)}) = P(L_t^{(d)}|\cdot) + P(R) \cdot (1 - P(L_t^{(d)}|\cdot)) \quad (3)$$

where $P(L_t^{(d)}|\cdot)$ is either $P(L_t^{(d)}|C_t^{(d)})$ or $P(L_t^{(d)}|\overline{C_t^{(d)}})$ depending on the collected data for the learner.

Knowing the values of all parameters of the BKT model will allow us to predict the probability of learner d being proficient in the KC, $P(L_t^{(d)})$ (which we will refer to simply as "proficiency" of learner d).

3. RESTRICTIONS ON THE BKT PARAMETERS

Prior to estimating the BKT parameters, we need to place some restrictions on them to maintain the conceptual meaning of the modeled process when used in real systems. All results obtained in this section are not learner specific. Thus, for the sake of readability, we will omit all learner-specific indexes in this section, e.g., we will use L_t instead of $L_t^{(d)}$.

First, it does not make sense for $P(S)$ and $P(G)$ to be 0 since that would simply eliminate their use as parameters entirely. $P(G)$ being 1 would mean that a learner in the non-mastery state would necessarily guess and get the question right each time which is unrealistic. Similarity, $P(S)$ being equal to 1 would mean that a learner in the mastery state would necessarily slip and get the question wrong each time which obviates the very definition of mastery. Thus, our first constraint is that $P(S)$ and $P(G)$ both vary between 0 and 1 without ever taking the extreme values exactly. Next, $P(R)$ is also between 0 and 1. If $P(R) = 0$, then learners cannot transition, and the learning experience is a priori useless. if $P(R) = 1$, then the learning experience has a 100% success rate. Both situations cannot be guaranteed. Next, if $P(L_0) = 0$, then from ((2)) - ((3)), all $P(L_t) = 0$, which is an uninteresting scenario to consider. Moreover, if $P(L_t) = 0$, then from ((3)), it follows that $P(L_{t-1}|\cdot) = 0$. And from ((2)), we can see it is possible only if $P(L_{t-1}) = 0$. Therefore, by induction, $P(L_0) = 0$. Similarly, $P(L_t) = 1$ if and only if $P(L_0) = 1$. Thus, we can assume –

1. $0 < P(G) < 1$,
2. $0 < P(S) < 1$,
3. $0 < P(R) < 1$,
4. $0 < P(L_t) < 1$ for all $t = 0, \cdots, T$.

There are some additional restrictions we can add for the BKT parameters. Namely, we want correct answers to increase our estimate of learner's proficiency both before and after the transition. Similarly, incorrect answers should lower the proficiency.

$$P(L_t|C_{t+1}) \geq P(L_t) \geq P(L_t|\overline{C_{t+1}}) \quad (4)$$

and

$$P(L_{t+1}|C_{t+1}) \geq P(L_t) \geq P(L_{t+1}|\overline{C_{t+1}}) \quad (5)$$

The inequalities in ((4)) yield a natural restriction on the parameters –

$$1 - P(S) \geq P(G) \quad (6)$$

that is, the probability of answering an assessment correctly is higher if a learner is proficient. Moreover, this restriction is also sufficient for the left inequality in ((5)) to be true. Let us prove these statements.

Proof. Let us consider the first three inequalities.

$P(L_0)$	Prior Proficiency	the probability the learner is in proficient state for the KC prior to first feedback.
$P(G)$	Guess	the probability of correct guess at an assessment while not being proficient at the corresponding KC.
$P(S)$	Slip	the probability of slip (mistake) at an assessment while being proficient at the corresponding KC.
$P(R)$	Transition	the probability of a learner becoming proficient in KC after making an attempt at an assessment and reading the feedback.
$P(L_t^{(d)})$	Proficiency	the probability of learner d being in a proficient state after receiving t rounds of feedback.
$P(C_t^{(d)})$	Correctness of an Attempt	the probability of learner d answering assessment t correctly.

Figure 1: BKT Parameters and Notation

1. $P(L_t|C_{t+1}) \geq P(L_t)$

$$\frac{P(L_t)\cdot(1-P(S))}{P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t))} \geq P(L_t),$$

$$\frac{1-P(S)}{P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t))} \geq 1,$$

$$1-P(S) \geq P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t)),$$

$$1-P(S)-P(G) \geq P(L_t)\cdot(1-P(S)-P(G)),$$

$$(1-P(S)-P(G))\cdot(1-P(L_t)) \geq 0,$$

which is always true if and only if $1-P(S)-P(G) \geq 0$.

2. $P(L_t) \geq P(L_t|\overline{C_{t+1}})$:

$$P(L_t) \geq \frac{P(L_t)\cdot P(S)}{P(L_t)\cdot P(S)+(1-P(L_t))\cdot(1-P(G))},$$

$$1 \geq \frac{P(S)}{P(L_t)\cdot P(S)+(1-P(L_t))\cdot(1-P(G))},$$

$$P(L_t)\cdot P(S)+(1-P(L_t))\cdot(1-P(G)) \geq P(S),$$

$$(1-P(L_t))\cdot(1-P(G)-P(S)) \geq 0,$$

which is always true if and only if $1-P(S)-P(G) \geq 0$.

3. $P(L_{t+1}|C_{t+1}) \geq P(L_t)$:

$$P(L_{t+1}|C_{t+1}) = P(L_t|C_{t+1}) + P(R)\cdot(1-P(L_t|C_{t+1})) \geq P(L_t),$$

$$P(L_t|C_{t+1})\cdot(1-P(R))+P(R) \geq P(L_t),$$

where left-hand side is

$$\frac{P(L_t)\cdot(1-P(S))\cdot(1-P(R))}{P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t))}+P(R)$$

$$=\frac{P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t))\cdot P(R)}{P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t))}.$$

It follows

$$P(L_t)\cdot(1-P(S))+P(G)\cdot(1-P(L_t))\cdot P(R) \geq P(L_t)^2\cdot(1-P(S))+P(G)\cdot P(L_t)\cdot(1-P(L_t)),$$

and can be further simplified to

$$P(L_t)\cdot(1-P(S))\cdot(1-P(L_t)) + (1-P(L_t))\cdot P(G)\cdot(P(R)-P(L_t)) \geq 0,$$

$$P(L_t)\cdot(1-P(S))+P(G)\cdot(P(R)-P(L_t)) \geq 0,$$

$$P(L_t)\cdot(1-P(S)-P(G))+P(G)\cdot P(R) \geq 0,$$

which is true if $1-P(S)-P(G) \geq 0$.

□

The final inequality in ((5)) yields a non-trivial restriction –

$$P(L_t) \geq \frac{(1-P(G))\cdot P(R)}{1-P(S)-P(G)}. \tag{7}$$

Proof. Similar to the previous proof, from

$$P(L_t) \geq P(L_{t+1}|\overline{C_{t+1}})$$

we have

$$P(L_t) \geq \frac{P(L_t)\cdot P(S) + (1-P(L_t))\cdot(1-P(G))\cdot P(R)}{P(L_t)\cdot P(S) + (1-P(L_t))\cdot(1-P(G))},$$

which leads to

$$P(L_t)^2\cdot P(S) + P(L_t)\cdot(1-P(L_t))\cdot(1-P(G)) \geq P(L_t)\cdot P(S) + (1-P(L_t))\cdot(1-P(G))\cdot P(R),$$

and further simplifies to

$$P(L_t)\cdot P(S)\cdot(1-P(L_t)) + (1-P(L_t))\cdot(1-P(G))\cdot(P(R)-P(L_t)) \leq 0,$$

$$P(L_t)\cdot P(S) + (1-P(G))\cdot(P(R)-P(L_t)) \leq 0,$$

$$P(L_t)\cdot(1-P(G)-P(S)) \geq (1-P(G))\cdot P(R).$$

Note that $1-P(G)-P(S) \neq 0$, otherwise $P(G)=1$ or $P(R)=0$. Therefore,

$$P(L_t) \geq \frac{(1-P(G))\cdot P(R)}{1-P(S)-P(G)}.$$

□

Let us define the value in the right-hand side of ((7)) as P^*. It can be shown that if $P^* < P(L_{t^*}) < 1$, then $P^* < P(L_t) < 1$ for any $t > t^*$ and any sequence of attempts. Namely, the following is true –

THEOREM 1. *In a sequence of all failed attempts $P(L_t)$ will asymptotically approach P^* from the right, and in a sequence of all successful attempts $P(L_t)$ will asymptotically approach 1 from the left.*

PROOF. Let us consider a sequence of only failed attempts $F = (F_1, F_2, \cdots, F_T)$. And let

$$P(F_t) = P^* + \frac{\varepsilon}{(1-P(S)-P(G))}$$

for some $0 < \varepsilon < (1-P(S)-P(G))\cdot(1-P^*)$ and some t. Note that using definition for P^* we get

$$P(F_t) = \frac{(1-P(G))\cdot P(R)+\varepsilon}{(1-P(S)-P(G))}. \tag{8}$$

Next, we write $P(F_{t+1})$ in the following form

$$P(R) + \frac{P(F_t)\cdot P(S)\cdot(1-P(R))}{P(F_t)\cdot P(S) + (1-P(F_t))\cdot(1-P(G))}$$

$$= P(R) + \frac{P(F_t)\cdot P(S)\cdot(1-P(R))}{-P(F_t)\cdot(1-P(S)-P(G)) + (1-P(G))}$$

$$= P(R) + \frac{\frac{(1-P(G))\cdot P(R)+\varepsilon}{1-P(S)-P(G)}\cdot P(S)\cdot(1-P(R))}{-(1-P(G))\cdot P(R) - \varepsilon + (1-P(G))}$$

$$= P(R) + \frac{((1-P(G))\cdot P(R)+\varepsilon)\cdot P(S)\cdot(1-P(R))}{(1-P(S)-P(G))\cdot((1-P(G))\cdot(1-P(R))-\varepsilon)}$$

$$= P(R) + \frac{P(S)\Big[P(R)\cdot((1-P(G))\cdot(1-P(R))-\varepsilon)+\varepsilon\Big]}{(1-P(S)-P(G))\cdot((1-P(G))\cdot(1-P(R))-\varepsilon)}$$

$$= P(R) + \frac{P(R)\cdot P(S)}{1-P(S)-P(G)} + \frac{\varepsilon\cdot P(S)}{(1-P(S)-P(G))\cdot((1-P(G))\cdot(1-P(R))-\varepsilon)}$$

$$= P^* + \frac{\varepsilon}{(1-P(S)-P(G))}\cdot\frac{P(S)}{(1-P(G))\cdot(1-P(R))-\varepsilon},$$

where it can be easily shown from ((8)) and the fact that $P(F_t) < 1$ that

$$0 < \frac{P(S)}{(1-P(G))\cdot(1-P(R))-\varepsilon} < 1.$$

Therefore,

$$P^* < P(F_{t+1}) < P^* + \frac{\varepsilon}{1-P(S)-P(G)} = P(F_t).$$

This proves the statement that if $P(F_0) > P^*$, then the sequence $(P(F_1), P(F_1), \cdots, P(F_T))$ asymptotically approaches P^* from the right.

Let us now consider a sequence of only successful attempts $U = (U_1, U_2, \cdots, U_T)$. We know that the sequence of proficiencies $(P(U_1), P(U_2), \cdots, P(U_T))$ is increasing and cannot reach 1 from previous discussion. Let us show, that it asymptotically approaches 1. Let $P(U_t) = 1-\epsilon$ for some $0 < \epsilon < 1$ and some t. Then we can write $P(U_{t+1})$ in the following form:

$$\frac{P(U_t)\cdot(1-P(S))\cdot(1-P(R))}{P(U_t)\cdot(1-P(S)) + (1-P(U_t))\cdot P(G)} + P(R)$$

$$= \frac{P(U_t)\cdot(1-P(S)) + (1-P(U_t))\cdot P(G)\cdot P(R)}{P(U_t)\cdot(1-P(S)) + (1-P(U_t))\cdot P(G)} =$$

$$= \frac{(1-\epsilon)\cdot(1-P(S)) + \epsilon\cdot P(G)\cdot P(R)}{(1-\epsilon)\cdot(1-P(S)) + \epsilon\cdot P(G))} =$$

$$1 - \frac{\epsilon\cdot P(G)\cdot(1-P(R))}{(1-\epsilon)\cdot(1-P(S)) + \epsilon\cdot P(G))}. \tag{9}$$

Note that –

$$(1-\epsilon)\cdot(1-P(S)-P(G))+P(R)\cdot P(G)>0,$$

$$(1-\epsilon)\cdot(1-P(S))-P(G)+\epsilon\cdot P(G)+P(R)\cdot P(G)>0,$$

$$(1-\epsilon)\cdot(1-P(S))+\epsilon\cdot P(G)>P(G)\cdot(1-P(R)),$$

$$\frac{P(G)\cdot(1-P(R))}{(1-\epsilon)\cdot(1-P(S))+\epsilon\cdot P(G)}<1.$$

Using ((9)), gives us

$$P(U_t)=1-\epsilon<P(U_{t+1})<1.$$

Therefore, if $P(U_0)<1$, then the sequence $(P(U_1),P(U_2),\cdots,P(U_T))$ asymptotically approaches 1 from the left.

Finally, for any sequence of attempts $L=(L_1,L_2,\cdots,L_T)$ if $P(L_0)=P(F_0)=P(U_0)$, then sequence F is the lower bound for L, and sequence U is the upper bound for L. □

Thus, we arrive at the following succinct list of restrictions on the BKT model parameters –

$$0<P(G)<1, \tag{10}$$

$$0<P(S)<1, \tag{11}$$

$$0<P(R)<1, \tag{12}$$

$$1-P(S)-P(G)\geq 0, \tag{13}$$

$$\frac{(1-P(G))\cdot P(R)}{1-P(S)-P(G)}<P(L_0)<1. \tag{14}$$

4. ESTIMATING THE PARAMETERS

Let $X=(X^{(1)},X^{(2)},\cdots,X^{(D)})$ be a hidden process of learners' states of proficiency where $X^{(d)}=(X_1^{(d)},X_2^{(d)},\cdots,X_{T^{(d)}}^{(d)})$ is a sequence corresponding to learner d. $X_t^{(d)}$ takes value 0 if learner d is not proficient after t rounds of feedback, and value 1 if learner d is proficient after t rounds of feedback. Analogously, let $Y=(Y^{(1)},Y^{(2)},\cdots,Y^{(D)})$ be an observable process of a learner's attempts at assessments, with $Y^{(d)}=(Y_1^{(d)},Y_2^{(d)},\cdots,Y_{T^{(d)}}^{(d)})$ is a sequence of attempts corresponding to a learner d. Similarly, $Y_t^{(d)}$ takes value 0 and 1 for incorrect and correct answers by learner d at assessment t, respectively. And let y be a realization of Y, i.e., an available dataset. We cannot directly observe a corresponding realization of the latent process X. For maximum likelihood estimation of parameters we would need to maximize the marginal likelihood function with respect to X, which is intractable. Let us consider approaches to estimate the parameters of the BKT model for a given KC: $P(L_0)$, $P(G)$, $P(S)$, and $P(R)$.

4.1 Expectation-Maximization Algorithm

Expectation–maximization (EM) algorithm [8] is an iterative algorithm to find local maximum likelihood estimates of parameters in a model with unobserved latent variables. EM is not guaranteed to converge to an optimal solution, but rather to a local optimal solution. EM is used when computation of the likelihood function is intractable due to presence of latent variables. Starting with a random initial guess for parameters, the algorithm improves the estimate for model parameters in each iteration by guaranteeing that the new parameter values correspond to higher values of the log-likelihood function without explicitly computing it. Let us denote θ as a vector of all model parameters, and θ^* as the parameter estimates at the current step of EM. During each step, the function $Q(\theta)=Q(\theta|\theta^*)$ is constructed as the expected value of log-likelihood function of θ with respect to the current conditional distribution of X given data y and θ^* –

$$Q(\theta|\theta^*)=\mathbb{E}_{X|y,\theta^*}\left[\log P(y,X|\theta)\right]. \tag{15}$$

EM states that –

$$\forall\theta:\log P(y|\theta)-\log P(y|\theta^*)\geq Q(\theta|\theta^*)-Q(\theta^*|\theta^*), \tag{16}$$

that is, any θ that increases value of Q over $Q(\theta^*|\theta^*)$ also improves the value of the corresponding marginal log-likelihood function. EM defines the next value of θ^* as the –

$$\theta^*_{\text{next}}=\arg\max_{\theta}Q(\theta|\theta^*). \tag{17}$$

For a discrete case, like a BKT model, ((15)) becomes –

$$Q(\theta|\theta^*)=\sum_{x\in\mathcal{X}}\log\left[P(y,x|\theta)\right]\cdot P(x|y,\theta^*), \tag{18}$$

where $\mathcal{X}$ is a set of all possible X. Because $P(y,x|\theta^*)=P(x|y,\theta^*)\cdot P(y|\theta^*)$ and $P(y|\theta^*)$ is a constant with respect to θ, maximization of ((18)) is equivalent to a maximization of –

$$\widehat{Q}(\theta|\theta^*)=\sum_{x\in\mathcal{X}}\log\left[P(y,x|\theta)\right]\cdot P(y,x|\theta^*). \tag{19}$$

Sometimes, maximization of ((19)) is more convenient than ((18)).

BKT can be modeled as a Hidden Markov model and, therefore, a special case of EM algorithm, Baum-Welch algorithm [2], can be used. The Baum-Welch algorithm provides closed forms for θ^*_{next} based on θ^* values. It is fully described in Appendix A.

Note, that the Baum-Welch algorithm does not guarantee that ((13)) - ((14)) are satisfied. To avoid cases where Baum-Welch converges to a unsuitable parameters (i.e., degenerate parameters), we offer to use a different approach.

4.1.1 Novel EM Algorithm using the Interior-Point Method

We want an algorithm that will always yield meaningful parameters for the BKT model by satisfying conditions ((10)) - ((14)). That can be achieved if instead of just maximizing $\widehat{Q}$ in ((17)), we maximize it under ((10)) - ((14)) restrictions. Note that the corresponding log-likelihood function will increase due to property ((16)). Conditions ((10)) - ((12)) and

right-hand side of ((14)) are satisfied automatically due to the form of log functions in $\widehat{Q}$, see ((30)). Finally, ((13)) and left-hand side of ((14)) can be combined into a single inequality, resulting in the following non-linear optimization problem –

$$\begin{aligned} \theta^*_{\text{next}} &= \arg\max_{\theta} \widehat{Q}(\theta|\theta^*), \\ &\text{s.t.} \quad (1 - P(S) - P(G)) \cdot P(L_0) \\ &\qquad -(1 - P(G)) \cdot P(R) \geq 0, \end{aligned} \tag{20}$$

where $\widehat{Q}$ has form ((30)). We will use the interior-point method on ((20)). The goal is to find the maximum of the barrier function –

$$B(\theta, \mu) = \widehat{Q}(\theta|\theta^*) + \mu \cdot \log c(\theta), \tag{21}$$

where $c(\theta)$ is the left-hand side of the constrain from ((20)), and μ is a so-called barrier parameter. We will iterate through a decreasing sequence of values for μ parameter $\mu_1 > \mu_2 > \cdots > \mu_W = 0$, finding a maximum of W in each iteration. As μ approaches 0, the maximum of B converges to the solution of ((20)). Next, a dual variable λ is introduced, defined as $c(\theta) \cdot \lambda = \mu$. To find the extremum point of B we need to find zero of the following vector function –

$$F = \begin{bmatrix} \nabla B \\ \lambda \cdot c(\theta) - \mu \end{bmatrix}. \tag{22}$$

We will use a Newton's method to find zero of F. We start with some initial guesses θ_1 and λ_1 by solving and update them by solving –

$$J_F(\theta_k, \lambda_k) \times \begin{bmatrix} \Delta\theta \\ \Delta\lambda \end{bmatrix} = -F(\theta_k, \lambda_k), \tag{23}$$

where J_F is a Jacobian of F; and updating –

$$\begin{aligned} \theta_{k+1} &= \theta_k + \nu \cdot \Delta\theta, \\ \lambda_{k+1} &= \lambda_k + \nu \cdot \Delta\lambda, \end{aligned} \tag{24}$$

where ν is a value small enough, so updated θ_{k+1} and λ_{k+1} satisfy $c(\theta_{k+1}) \geq 0$ and $\lambda_{k+1} \geq 0$. Next,

$$F = \begin{bmatrix} \frac{\partial \widehat{Q}}{\partial P(L_0)} + \lambda \cdot (1 - P(S) - P(G)) \\ \frac{\partial \widehat{Q}}{\partial P(G)} + \lambda \cdot (P(R) - P(L_0)) \\ \frac{\partial \widehat{Q}}{\partial P(S)} - \lambda \cdot P(L_0) \\ \frac{\partial \widehat{Q}}{\partial P(R)} - \lambda \cdot (1 - P(G)) \\ \lambda \cdot (1 - P(S) - P(G)) \cdot P(L_0) \\ \qquad -\lambda \cdot (1 - P(G)) \cdot P(R) - \mu \end{bmatrix}, \tag{25}$$

where partial derivatives of $\widehat{Q}$ are given by ((31)) - ((35)). And the Jacobian J_F has the following form

$$\begin{bmatrix} \frac{\partial^2 \widehat{Q}}{\partial P(L_0)^2} & -\lambda & -\lambda & 0 & 1 - P(S) - P(G) \\ -\lambda & \frac{\partial^2 \widehat{Q}}{\partial P(G)^2} & 0 & \lambda & P(R) - P(L_0) \\ -\lambda & 0 & \frac{\partial^2 \widehat{Q}}{\partial P(S)^2} & 0 & -P(L_0) \\ 0 & \lambda & 0 & \frac{\partial^2 \widehat{Q}}{\partial P(R)^2} & -1 + P(G) \\ \lambda \cdot \frac{\partial c}{\partial P(L_0)} & \lambda \cdot \frac{\partial c}{\partial P(G)} & \lambda \cdot \frac{\partial c}{\partial P(S)} & \lambda \cdot \frac{\partial c}{\partial P(R)} & c, \end{bmatrix} \tag{26}$$

where –

$$\begin{aligned} \frac{\partial c}{\partial P(L_0)} &= 1 - P(S) - P(G), \\ \frac{\partial c}{\partial P(G)} &= P(R) - P(L_0), \\ \frac{\partial c}{\partial P(S)} &= -P(L_0), \\ \frac{\partial c}{\partial P(R)} &= -1 + P(G). \end{aligned} \tag{27}$$

Note from ((31)) - ((35)) that each first partial derivative of $\widehat{Q}$ has the following form –

$$\frac{\partial \widehat{Q}}{\partial P(\cdot)} = \frac{A}{P(\cdot)} - \frac{B}{1 - P(\cdot)} \tag{28}$$

with some values of A and B independent of $P(\cdot)$. Therefore, all corresponding second partial derivatives have the following form –

$$\frac{\partial^2 \widehat{Q}}{\partial P(\cdot)^2} = -\frac{A}{P(\cdot)^2} - \frac{B}{(1 - P(\cdot))^2}. \tag{29}$$

To summarize, we begin with $\mu = \mu_1$ and find zero of function $F(\mu_1)$ starting with some random initial guesses $(\theta_1(\mu_1), \lambda_1(\mu_1))$ and update them using rule ((24)) and formulae ((23)), ((25)) - ((29)), ((31)) - ((35)) until it converges to values $(\theta_k(\mu_1), \lambda_k(\mu_1))$ for some k. Then we apply the same procedure to find zero of function $F(\mu_2)$ using $(\theta_k(\mu_1), \lambda_k(\mu_1))$ as initial guesses. We continue until we converge to $(\theta_{k'}(\mu_W), \lambda_{k'}(\mu_W))$, the solution of $F(\mu_W) = 0$, which maximizes ((21)) and is solution of ((20)).

5. DEMONSTRATING THE EM-NEWTON ALGORITHM ON SIMULATED DATA

The simulated data used in this section is made available along with the paper [22]. In this section, we compare the performance of the method proposed in this paper, which we call EM-Newton algorithm, with the classical Baum-Welch algorithm. First, let us provide an example where the Baum-Welch algorithm yields degenerate parameters, i.e., the ones not satisfying ((10)) – ((14)). We simulated 100 datasets using the following parameter values: $P(L_0) = 0.45$, $P(R) = 0.3$, $P(S) = 0.1$, and $P(G) = 0.25$ (the same values for all datasets). All simulated datasets contained 300 learners answering 10 questions each. We fit each dataset using both EM-Newton and Baum-Welch algorithms starting from the same random initial parameter guesses. The Baum-Welch algorithm yielded two clusters of fitted parameters: the "good estimates", the ones close to the true parameters (80 cases),

and the "bad estimates", the ones far from the true parameters (20 cases). Moreover, the true parameters and all "good estimates" satisfied the conditions ((10)) – ((14)), while all "bad estimates" violated at least one of the conditions. Interestingly, EM-Newton algorithm was able to rescue the "bad estimates" for the corresponding "bad datasets" and produce valid parameter estimates. While both algorithms yielded very similar values of the parameters for the "good datasets", which were close to the true parameter values and satisfied the conditions, EM-Newton algorithm estimates did not fail even for "bad datasets" by forcing the parameters into a non-degenerate space. See Fig. 2.

Next, we repeated the same experiment but instead of fitting 100 dataset once, we randomly selected one of the "good datasets" and fitted it 100 times with both algorithms using 100 random initial parameter guesses (again, using the same initial guesses between two algorithms). Again, Baum-Welch algorithm produced two clusters of the parameter estimates with the same properties as in the previous experiment (80 "good estimates", 20 "bad estimates"), and the EM-Newton algorithm was able to rescue the "bad" cluster. See Fig. 3. Interestingly, this experiment indicates that the simulated datasets are likely not inherently "bad" or "good", since the same dataset produced both "good" and "bad" estimates at the same rate as a series of different datasets. Although the conditions under which Baum-Welch algorithm produces degenerate results are out of scope of this work, it seems to be dependent on the initial parameter guesses. That makes a lot of sense due to the local nature of the Baum-Welch algorithm. It also explains prior work that attempts to solve this problem by determining the starting values that lead to degenerate parameter values in order to avoid them [17].

After providing some examples of situations when Baum-Welch algorithm produces degenerate results, we looked towards more systematic analysis of the comparison between two algorithms for different combinations of the true parameter values. We randomly sampled 100 different sets of parameters from the space of non-degenerate parameters, defined by conditions ((10)) – ((14)), see Fig. 4. Then, for each set of parameters we repeated the first experiment of the section. We compared the Sum of Squared Errors (SSE) for the estimates yielded by the EM-Newton and Baum-Welch algorithms. As expected, the distribution of EM-Newton SSE was shifted to the left compared to the Baum-Welch algorithm: the outputs of EM-Newton were either similar to the Baum-Welch ones (for "good" cases) or closer to true parameter values (for rescued "bad" cases), see Figure 5, **Left**. This logic was further supported by EM-Newton SSE having lower variability than Baum-Welch SSE: a single-clustered output of EM-Newton algorithm had lower range of SSE for each given dataset than a potentially multiple-clustered output of Baum-Welch algorithm, see Figure (5), **Right**. This experiment demonstrated that EM-Newton algorithm has both higher average accuracy (SSE distribution is shifted to the left) and higher precision (SSE have lower variation within a dataset).

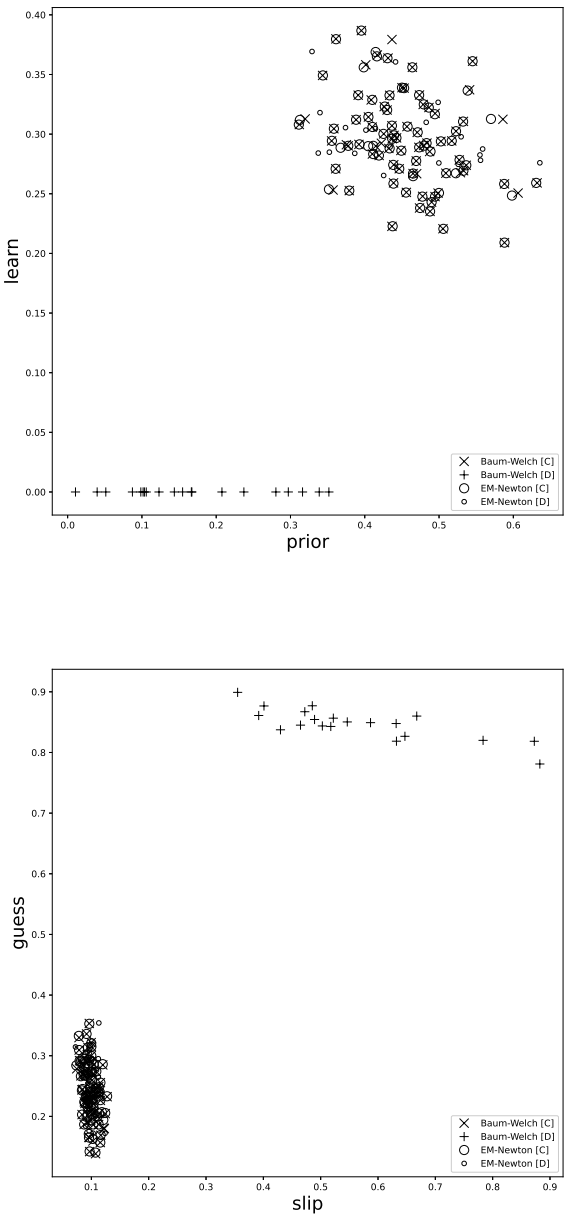
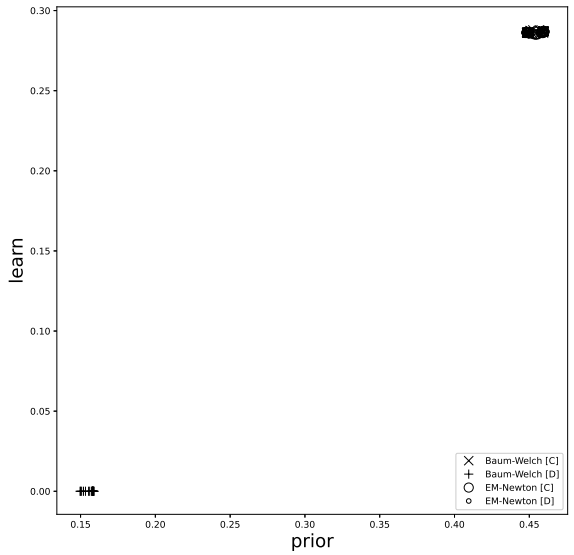


Figure 2: BKT model fit to 100 simulated datasets using classical Baum-Welch algorithm (× and +) and proposed EM-Newton method (○ and ∘). The true parameter values were: $P(L_0) = 0.45$, $P(R) = 0.3$, $P(S) = 0.1$, and $P(G) = 0.25$. Baum-Welch algorithm produced two clusters of estimates: the ones satisfying conditions ((10)) – ((14)) and close to the true parameter values (×; Baum-Welch [C]), and the degenerate ones not satisfying the conditions and far from the true parameter values (+; Baum-Welch [D]). The same datasets were used to produced parameter estimates using EM-Newton algorithm (○; EM-Newton [C] and ∘; EM-Newton [D], respectively). Note that EM-Newton algorithm produced only one cluster of the parameter estimates, that are all close to the true parameter values. Also note that all EM-Newton parameter estimates satisfied conditions ((10)) – ((14)), i.e., not degenerate, by design.

6. DISCUSSION

The paper first derives a list of constraints on the BKT parameter space following from the conceptual meaning of the modeled process. One question that may arise in the reader's



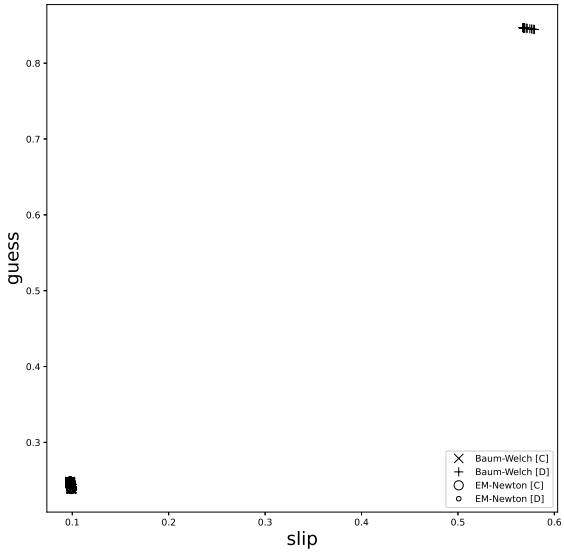


Figure 3: Experiment similar to the one on Fig. 2, but instead of 100 different datasets, the same dataset was fit 100 times using different initial parameter guesses. The outcome of the experiment is virtually identical, with the difference of the estimates grouped much closer. Nevertheless, it is easy to see that the "bad cluster" consists exclusively of a subset of Baum-Welch estimates not satisfying $((10)) - ((14))$ conditions (+; Baum-Welch [D]) (and, which is harder to see, it is the only cluster where they are present).

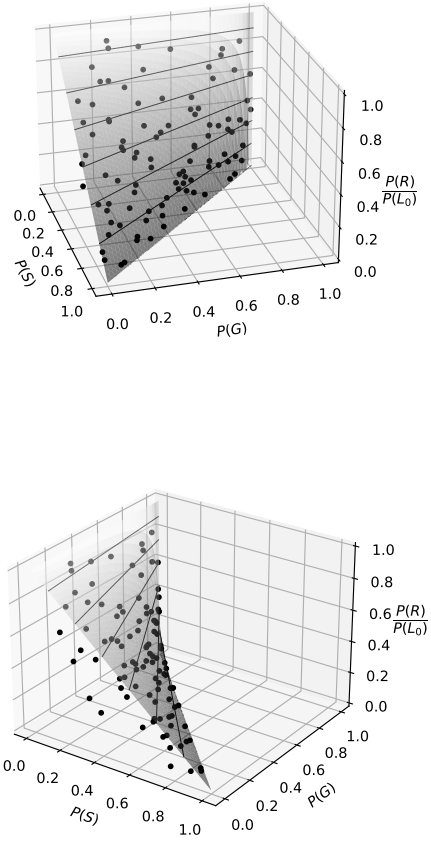


Figure 4: The non-degenerate parameter space defined by $((10)) - ((14))$. Note that there are four parameters, so a 4D plot would be required to actually display the space. Instead, we can rewrite condition $((14))$ with respect to $P(R)/P(L_0)$, and utilize a 3D plot instead. The non-degenerate parameter space is underneath the shown surface depicted from two angles (with additional restriction of all parameters being between 0 and 1). The straight lines on the surface are the contour lines. The sampled non-degenerate parameters are marked as dots.

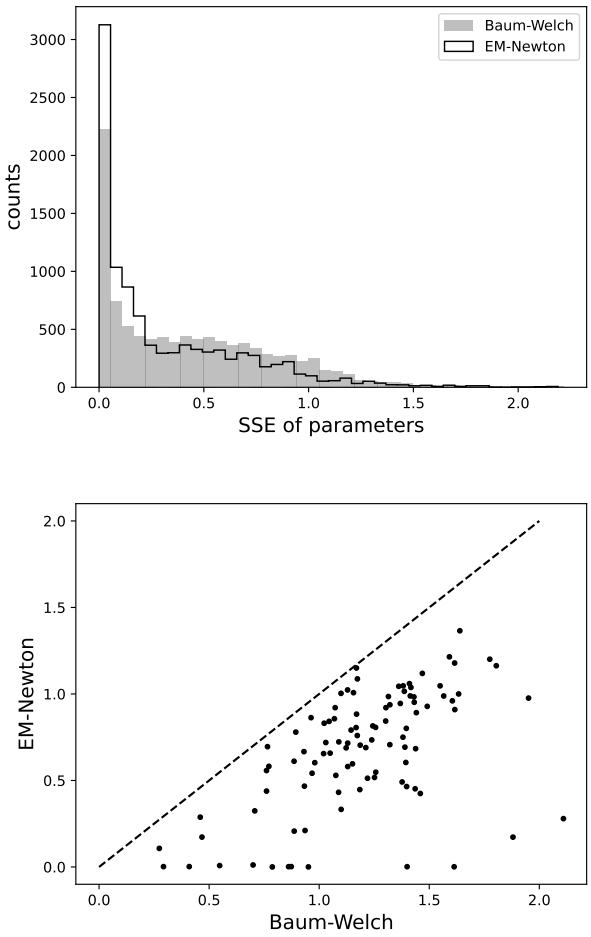


Figure 5: Top. Distribution of SSE for parameter estimates produced by Baul-Welch and EM-Newton algorithms. Bottom. For each dataset and algorithm we found the range of produced SSE (minimum SSE subtracted from maximum SSE). Note that the variation in SSE is always lower for EM-Newton than for Baum-Welch.

mind is around the validity of the constraints imposed on the BKT parameters in practice. While the justifications for the constraints are mentioned in the text, they also assume that the questions are "well-designed". It follows, therefore, that using this process, it is possible to address the complementary issue of identifying poorly performing KCs as those for whom these constraints are violated and flag them to learning designers with appropriately recommended fixes. For example, $P(R) = 1$ being true could mean that the learning experience is not connected to the KC since it is leading to proficiency regardless of mastery. While $1 - P(S) < P(G)$ could tell us the question is worded in such a way that leads to overthinking, i.e., skillful learners are less likely to answer it correctly than unskillful learners guessing the answer by chance.

Ultimately, we derived an algorithm that converges to a set of parameters that are guaranteed to meet the constraints. Additionally, we compared our algorithm to the classic Baum-Welch algorithm used to estimate parameters of Hidden Markov Models, including BKT. We demonstrated that both algorithms converge to similar values of parameters in cases where the values satisfy the derived conditions. We also demonstrated that Baum-Welch algorithm occasionally converges to the values of parameters than are neither close to the true values nor satisfying of the conditions, with our algorithm being able to rescue those cases. Although a single run of the Baum-Welch algorithm is less computational heavy than a single run of our algorithm (ours requires the Newton method to converge on each iteration), the Baum-Welch algorithm is often run multiple times with different initial conditions after post-hoc finding degenerate parameters. Our algorithm can be run once and, therefore, be less computational heavy overall.

Finally, let us also notice that the derivation approach described in the paper can be followed to devise an algorithm subject to a different set of constraints as well, so long as the set of constraints remain tractable. The approach can, therefore, be extended to BKT extensions such as the addition of individual item difficulty [19], individualization [18,25], time between attempts [20], or forgetting [13].

7. CONCLUSION AND FUTURE WORK

This paper derives succinct constraints that can be imposed on the BKT parameter space from first principles. Then, a new Expectation-Maximization algorithm using the Interior-Point Method is introduced that produces parameters subject to those constraints and is, therefore, guaranteed to produce valid, i.e., non-degenerate parameters. While the computational cost savings may not be dramatic for 4-parameter BKT, as presented, they become increasingly more so for extensions to BKT that can still use Expectation-Maximization such as the addition of individual item difficulty [19], individualization [18, 25], time between attempts [20], or forgetting [13] parameters which we will present in future work. Experiments with real-time adaptive learning systems is also currently in progress and will be reported in future work. More complex extensions such as BKT+ [4] incorporate many of these, but, start to commensurately require more complex methods such as Markov Chain Monte Carlo (MCMC) or even deep learning [9] which could make such first principles derivation untenable.

8. REFERENCES

[1] R. S. d. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings 9*, pages 406–415. Springer, 2008.

[2] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[3] J. E. Beck and K.-m. Chang. Identifiability: A fundamental problem of student modeling. In *International Conference on User Modeling*, pages 137–146. Springer, 2007.

[4] S. Bhatt, J. Zhao, C. Thille, D. Zimmaro, and N. Gattani. Evaluating bayesian knowledge tracing for estimating learner proficiency and guiding learner behavior. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 357–360, 2020.

[5] K.-m. Chang, J. Beck, J. Mostow, and A. Corbett. A bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings 8*, pages 104–113. Springer, 2006.

[6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278, 1994.

[7] S. D. Craig, X. Hu, A. C. Graesser, A. E. Bargagliotti, A. Sterbinsky, K. R. Cheney, and T. Okwumabua. The impact of a technology-based mathematics after-school program using aleks on student's knowledge and behaviors. *Computers & Education*, 68:495–504, 2013.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[9] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[10] W. J. Hawkins, N. T. Heffernan, and R. S. Baker. Learning bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. In *Intelligent Tutoring Systems: 12th International Conference, ITS 2014, Honolulu, HI, USA, June 5-9, 2014. Proceedings 12*, pages 150–155. Springer, 2014.

[11] T. Kabudi, I. Pappas, and D. H. Olsen. Ai-enabled adaptive learning systems: A systematic mapping of the literature. *Computers and Education: Artificial Intelligence*, 2:100017, 2021.

[12] T. Käser, S. Klingler, A. G. Schwing, and M. Gross. Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, 10(4):450–462, 2017.

[13] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing?. *International Educational Data Mining Society*, 2016.

[14] Y. Kim, S. Sankaranarayanan, C. Piech, and C. Thille. Variational temporal irt: Fast, accurate, and explainable inference of dynamic learner proficiency. *International Educational Data Mining Society*, 2023.

[15] T. Liu. Knowledge tracing: A bibliometric analysis. *Computers and Education: Artificial Intelligence*, page 100090, 2022.

[16] S. Minn. Ai-assisted knowledge assessment techniques for adaptive learning environments. *Computers and Education: Artificial Intelligence*, 3:100050, 2022.

[17] Z. Pardos and N. Heffernan. Navigating the parameter space of bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. In *Educational Data Mining 2010*, 2010.

[18] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization: 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings 18*, pages 255–266. Springer, 2010.

[19] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization: 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings 19*, pages 243–254. Springer, 2011.

[20] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In *EDM*, pages 139–148, 2011.

[21] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the knowledge tracing space. *International Working Group on Educational Data Mining*, 2009.

[22] D. Shchepakin, S. Sankaranarayanan, and D. Zimmaro. Inferring bayesian knowledge tracing parameters using classical expectation-maximization and a novel newton method-based approach, 2023.

[23] B. van De Sande. Properties of the bayesian knowledge tracing model. *Journal of Educational Data Mining*, 5(2):1–10, 2013.

[24] Y. Wang and J. Beck. Class vs. student in a bayesian network student model. In *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*, pages 151–160. Springer, 2013.

[25] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*, pages 171–180. Springer, 2013.

APPENDIX

A. BAUM-WELCH ALGORITHM

We have –

$$
\begin{aligned}
P(y,x|\theta) &= \prod_{d=1}^{D} P(y^{(d)},x^{(d)}|\theta) \\
&= \prod_{d=1}^{D}\Bigg((1-P(L_0))^{(1-x_1^{(d)})}\cdot P(L_0)^{x_1^{(d)}} \\
&\quad \times \prod_{t=1}^{T^{(d)}}\Bigg\{ \left[P(G)^{y_t^{(d)}}\cdot(1-P(G))^{1-y_t^{(d)}})\right]^{1-x_t^{(d)}} \\
&\quad \times \left[P(S)^{1-y_t^{(d)}}\cdot(1-P(S))^{y_t^{(d)}}\right]^{x_t^{(d)}}\Bigg\} \\
&\quad \times \prod_{t=1}^{T^{(d)}-1}\Bigg\{ \left[(1-P(R))^{1-x_{t+1}^{(d)}}\cdot P(R)^{x_{t+1}^{(d)}}\right]^{1-x_t^{(d)}} \\
&\quad \times \left[x_{t+1}^{(d)}\right]^{x_t^{(d)}}\Bigg\}\Bigg).
\end{aligned}
$$

Thus, ((19)) becomes –

$$
\begin{aligned}
\widehat{Q}\,(\theta|\theta^*) = & \\
\sum_{x\in\mathcal{X}} & \Bigg[\sum_{d=1}^{D}\Bigg((1-x_1^{(d)})\cdot\log(1-P(L_0)) \\
& \quad +x_1^{(d)}\cdot\log P(L_0)\Bigg) \\
& + \sum_{t=1}^{T^{(d)}}\Bigg((1-x_t^{(d)})\cdot y_t^{(d)}\cdot\log P(G) \\
& \quad +(1-x_t^{(d)})\cdot(1-y_t^{(d)})\cdot\log(1-P(G))\Bigg) \\
& + \sum_{t=1}^{T^{(d)}}\Bigg(x_t^{(d)}\cdot(1-y_t^{(d)})\cdot\log P(S) \\
& \quad +x_t^{(d)}\cdot y_t^{(d)}\cdot\log(1-P(S))\Bigg) \\
& + \sum_{t=1}^{T^{(d)}-1}\Bigg((1-x_t^{(d)})\cdot(1-x_{t+1}^{(d)})\cdot\log(1-P(R)) \\
& \quad +(1-x_t^{(d)})\cdot x_{t+1}^{(d)}\cdot\log P(R)\Bigg) \\
& + \sum_{t=1}^{T^{(d)}-1} x^{(d)}\log x_{t+1}^{(d)}\Bigg]\cdot P(y,x|\theta^*)
\end{aligned}
\tag{30}
$$

The maximum of the function can be found by finding extremum of $\widehat{Q}(\theta|\theta^*)$. For $P(L_0)$ we have

$$
\begin{aligned}
&\frac{\partial\widehat{Q}}{\partial P(L_0)} = \\
&\quad \frac{\partial}{\partial P(L_0)}\Bigg[\sum_{x\in\mathcal{X}}\sum_{d=1}^{D}\Bigg((1-x_1^{(d)})\cdot\log(1-P(L_0)) \\
&\qquad + x_1^{(d)}\cdot\log P(L_0)\Bigg)\cdot P(y,x|\theta^*)\Bigg] \qquad (31)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\partial}{\partial P(L_0)}\Bigg[\sum_{d=1}^{D}\log(1-P(L_0))\cdot P(x_1^{(d)}=0,y|\theta^*) \\
&\qquad + \sum_{d=1}^{D}\log P(L_0)\cdot P(x_1^{(d)}=1,y|\theta^*)\Bigg] \\
&= \frac{\sum_{d=1}^{D} P(x_1^{(d)}=1,y|\theta^*)}{P(L_0)} - \frac{\sum_{d=1}^{D} P(x_1^{(d)}=0,y|\theta^*)}{1-P(L_0)}. \qquad (32)
\end{aligned}
$$

For $P(G)$ we have –

$$
\begin{aligned}
&\frac{\partial\widehat{Q}}{\partial P(G)} = \\
&\quad \frac{\partial}{\partial P(G)}\Bigg[\sum_{x\in\mathcal{X}}\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}}(1-x_t^{(d)})\cdot\Bigg(y_t^{(d)}\cdot\log P(G) \\
&\qquad + (1-y_t^{(d)})\cdot\log(1-P(G))\Bigg)\Bigg] \\
&\quad = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}} y_t^{(d)}\cdot P(x_t^{(d)}=0,y|\theta^*)}{P(G)} \\
&\qquad - \frac{\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}}(1-y_t^{(d)})\cdot P(x_t^{(d)}=0,y|\theta^*)}{1-P(G)}. \qquad (33)
\end{aligned}
$$

For $P(S)$ we have –

$$
\begin{aligned}
&\frac{\partial\widehat{Q}}{\partial P(S)} = \\
&\quad \frac{\partial}{\partial P(S)}\Bigg[\sum_{x\in\mathcal{X}}\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}} x_t^{(d)}\cdot\Bigg((1-y_t^{(d)})\cdot\log P(S) \\
&\qquad + y_t^{(d)}\cdot\log(1-P(S))\Bigg)\Bigg] \\
&\quad = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}}(1-y_t^{(d)})\cdot P(x_t^{(d)}=1,y|\theta^*)}{P(S)} \\
&\qquad - \frac{\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}} y_t^{(d)}\cdot P(x_t^{(d)}=1,y|\theta^*)}{1-P(S)}. \qquad (34)
\end{aligned}
$$

And for $P(R)$ we have –

$$
\begin{aligned}
&\frac{\partial\widehat{Q}}{\partial P(R)} = \\
&\frac{\partial}{\partial P(R)}\Bigg[\sum_{x\in\mathcal{X}}\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}-1}(1-x_t^{(d)})\cdot\Bigg((1-x_{t+1}^{(d)})\cdot\log(1-P(R)) \\
&+x_{t+1}^{(d)}\cdot\log P(R)\Bigg)\Bigg] = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T^{(d)}-1} P(x_t^{(d)}=0,x_{t+1}^{(d)}=1,y|\theta^*)}{P(R)}
\end{aligned}
$$

$$- \frac{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}-1} P(x_t^{(d)} = 0, x_{t+1}^{(d)} = 0, y|\theta^*)}{1 - P(R)}. \quad (35)$$

Setting partial derivatives ((31)) - ((35)) to zero, yields a closed form solution for the parameters:

$$\begin{aligned}
P(L_0) &= \frac{\sum_{d=1}^{D} P(x_1^{(d)} = 1, y|\theta^*)}{\sum_{d=1}^{D} P(y|\theta^*)} \\
&= \frac{1}{D} \sum_{d=1}^{D} P(x_1^{(d)} = 1|y^{(d)}, \theta^*), \\
P(G) &= \frac{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} y_t^{(d)} \cdot P(x_t^{(d)} = 0|y^{(d)}, \theta^*)}{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} P(x_t^{(d)} = 0|y^{(d)}, \theta^*)}, \\
P(S) &= \frac{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} (1 - y_t^{(d)}) \cdot P(x_t^{(d)} = 1|y^{(d)}, \theta^*)}{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} P(x_t^{(d)} = 1|y^{(d)}, \theta^*)}, \\
P(R) &= \frac{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}-1} P(x_t^{(d)} = 0, x_{t+1}^{(d)} = 1|y^{(d)}, \theta^*)}{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}-1} P(x_t^{(d)} = 0|y^{(d)}, \theta^*)}.
\end{aligned} \quad (36)$$

Let us now describe and algorithm to find probabilities in ((36)). The hidden proficiency process X is Markov, therefore we can define a transition matrix $A = \{a_{ij}\} = \{P(X_t^{(d)} = j|X_{t-1}^{(d)} = i)\}$ for $i = 0, 1$ and $j = 0, 1$:

$$A = \begin{bmatrix} 1 - P(R) & P(R) \\ 0 & 1 \end{bmatrix},$$

and a so-called emission matrix $B = \{b_j(i)\} = \{P(Y_t^{(d)} = i|X_t^{(d)} = j)\}$ for $i = 0, 1$ and $j = 0, 1$:

$$B = \begin{bmatrix} 1 - P(G) & P(G) \\ P(S) & 1 - P(S) \end{bmatrix}.$$

And a vector of initial states $\pi = \{\pi_i\} = \{P(X_0^{(d)} = i)\}$ for $i = 0, 1$:

$$\pi = \begin{bmatrix} 1 - P(L_0) \\ P(L_0) \end{bmatrix}.$$

Starting with some random initial guess for parameters $P(S)$, $P(G)$, $P(R)$, $P(L_0)$, denoted as vector θ, we compute a so-called Forward Procedure for $d = 1, \cdots, D$ and $m = 0, 1$:

$$\begin{aligned}
\alpha_i^{(d)}(t) = P\big(&Y_1^{(d)} = y_1^{(d)}, \\
&Y_2^{(d)} = y_2^{(d)}, \cdots, Y_t^{(d)} = y_t^{(d)}, X_t^{(d)} = i\big|\theta\big),
\end{aligned} \quad (37)$$

by recursive formulae –

$$\begin{aligned}
\alpha_i^{(d)}(1) &= \pi_i \cdot b_i(y_1^{(d)}), \\
\alpha_i^{(d)}(t+1) &= b_i(y_{t+1}^{(d)}) \cdot \left(\alpha_0^{(d)}(t) \cdot a_{0i} + \alpha_1^{(d)}(t) \cdot a_{1i}\right),
\end{aligned} \quad (38)$$

and a so-called Backward Procedure –

$$\begin{aligned}
\beta_i^{(d)}(t) = P\big(&Y_{t+1}^{(d)} = y_{t+1}^{(d)}, \\
&Y_{t+2}^{(d)} = y_{t+2}^{(d)}, \cdots, Y_T^{(d)} = y_T^{(d)}\big|X_t^{(d)} = i, \theta\big),
\end{aligned} \quad (39)$$

by recursive formulae –

$$\begin{aligned}
\beta_i^{(d)}(T) &= 1, \\
\beta_i^{(d)}(t) &= \beta_0^{(d)}(t+1) \cdot a_{i0} \cdot b_0(y_{t+1}^{(d)}) \\
&\quad + \beta_1^{(d)}(t+1) \cdot a_{i1} \cdot b_1(y_{t+1}^{(d)}).
\end{aligned} \quad (40)$$

Then we can define

$$\begin{aligned}
\gamma_i^{(d)}(t) = P\left(X_t^{(d)} = i\middle| y^{(d)}, \theta\right) &= \frac{P\left(X_t^{(d)} = i, y^{(d)}\middle|\theta\right)}{P\left(y^{(d)}|\theta\right)} \\
&= \frac{\alpha_i^{(d)}(t) \cdot \beta_i^{(d)}(t)}{\sum_{k=0}^{1} \alpha_k^{(d)}(t) \cdot \beta_k^{(d)}(t)},
\end{aligned} \quad (41)$$

and

$$\begin{aligned}
\xi_{ij}^{(d)}(t) &= P\left(X_t^{(d)} = i, X_{t+1}^{(d)} = j\middle| y^{(d)}, \theta\right) \\
&= \frac{P\left(X_t^{(d)} = i, X_{t+1}^{(d)} = j, y^{(d)}\middle|\theta\right)}{P\left(y^{(d)}|\theta\right)} \\
&= \frac{\alpha_i^{(d)}(t) \cdot a_{ij} \cdot b_j(y_{t+1}^{(d)}) \cdot \beta_j^{(d)}(t+1)}{\sum_{k=0}^{1} \sum_{w=0}^{1} \alpha_k^{(d)}(t) \cdot a_{kw} \cdot b_w(y_{t+1}^{(d)}) \cdot \beta_w^{(d)}(t+1)}.
\end{aligned} \quad (42)$$

Therefore, using ((37)) - ((42)), solution ((36)) becomes

$$\begin{aligned}
P(L_0) &= \frac{1}{D} \sum_{d=1}^{D} \gamma_1^{(d)}(1), \\
P(G) &= \frac{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} y_t^{(d)} \cdot \gamma_0^{(d)}(t)}{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} \gamma_0^{(d)}(t)}, \\
P(S) &= \frac{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} (1 - y^{(d)}) \cdot \gamma_1^{(d)}(t)}{\sum_{d=1}^{D} \sum_{t=1}^{T^{(d)}} \gamma_1^{(d)}(t)}, \\
P(R) &= \frac{\sum_{d=1}^{D} \sum_{t=1}^{T(d)-1} \xi_{01}^{(d)}(t)}{\sum_{d=1}^{D} \sum_{t=1}^{T(d)-1} \gamma_0^{(d)}(t)}.
\end{aligned} \quad (43)$$