The Impact on Individualizing Student Models on Necessary Practice Opportunities

Jung In Lee
Language Technologies Institute
Carnegie Mellon University
junginl@cs.cmu.edu

Emma Brunskill
Computer Science Department
Carnegie Mellon University
ebrun@cs.cmu.edu

ABSTRACT

When modeling student learning, tutors that use the Knowledge Tracing framework often assume that all students have the same set of model parameters. We find that when fitting parameters to individual students, there is significant variation among the individual's parameters. We examine if this variation is important in terms of instructional decisions by computing the difference in the expected number of practice opportunities required if mastery is assessed using an individual student's own estimated model parameters, compared to the population model. In the dataset considered, we find that a significant portion of students are expected to perform twice as many practice opportunities if the student is modeled using a population-based model, compared to the number needed if the student's own model parameters were used. We also find an additional significant portion of students will be likely to receive less practice opportunities than needed, implying that such students will be advanced too early. Though further work on additional datasets is needed to explore this issue in more depth, our results suggest that considering individual variation in student parameters may have important implications for the instructional decisions made in intelligent tutoring systems that use a Knowledge Tracing model.

1. INTRODUCTION

Both intelligent tutoring systems and live classroom instruction often assume that student learning can be adequately represented using a single model and associated set of model parameters. For example, in this paper we will focus on Knowledge Tracing [8], a popular method for estimating student mastery of skills that has been used in effective cognitive tutor systems [9]. Knowledge tracing is parameterized by 4 variables that are typically assumed to be the same for all students. Note that these population-level models still allow us to represent variation in our estimates of student performance: if two students respond differently to a set of practice opportunities, the model will have different estimates of future student performance for the two cases.

There have been some prior work on KT student models that represent differences in the student's initial knowledge [10]. In addition, several logistic regression-based student models, including Additive Factor Models [5] and Instructional Factors Analysis [7], include a single constant that is individually fit per student. Including this student parameter has been shown to lead to models that better fit the data, and have improved prediction accuracy. However, in all these cases, the parameters related to the progress of student learning and student observations, are fitted to the entire population. Therefore the underlying dynamical process of student learning, and the way in which that is translated to student performance, is assumed to be identical across students.

There's evidence to suggest this assumption is too strong. Standard high schools commonly offer multiple versions of the same class, such as a remedial version, normal version, and honors version. This approach is taken, at least in part, because it is believed both that students may have different learning speeds or prior backgrounds for a subject, and that those differences mean that the students will be best taught in different ways. In other words, instruction will vary not just according to our current estimate of student performance, but also how we anticipate that performance changes over time.

Here we examine the variation among individual student's parameters, and quantify the impact of this variation on pedagogical strategies. To start we consider this in the context of mastery learning, using the Knowledge Tracing framework to estimate and monitor student skill mastery. We already know from Cen et al. [6] that tuning the KT parameters can lead to a significant impact on reducing the amount of necessary practice opportunities; however, this work still uses a single set of KT parameters for all students. Corbett and Andersen [8] did try fitting individual parameters, and found this improved the predictive power of the model, as well as some evidence that this might improve student performance; however, the authors used curve-fitting to find the parameter values[1] and the authors did not examine the difference in practice opportunities needed if a population model was used instead of an individual model.

In this paper we fit Knowledge Tracing model parameters to each individual, on a dataset from the ASSISTment system [10]. We examine the distribution of the resulting parameters, and compare them to computing a single set of KT model parameters for all students. In our second contribution, we compute the difference in expected number of practice opportunities required if mastery is assessed using an individual student's own estimated model parameters, compared to the population model. We find that about 40% of

[1]Evidence [2] suggests that EM, which we use in this paper, finds a better parameter fit than curve fitting.

our student sample falls into one of two cases: students who will be forced to do many more practice opportunities than necessary, and students who are considered to have mastered the material when in reality they likely need additional practice opportunities. This implies that observed variation in student modeling parameters has an important and significant effect on instructional decisions, and that considering individual variation in the model parameters could lead to more effective teaching. We will outline several future ideas for how this could be accomplished in the conclusion of this paper.

2. BACKGROUND

A popular approach is to model a student's knowledge as a latent variable, which changes in response to practice opportunities. The system gets information about this underlying state through the student's responses to practice opportunities. This model is a Hidden Markov model, and estimating the student's current hidden state can be done by performing Bayesian filtering. A special case of this approach is known as Knowledge Tracing [8], which assumes the student has 1 binary hidden state per skill (the student has either mastered or not mastered the skill) and binary observations, corresponding to whether the student gets a question about a skill correct or incorrect. There are 4 model parameters per skill in the Knowledge Tracing framework: $p(L_0)$ is the initial probability the student has already mastered the skill; $p(T)$ is the probability of the student transitioning from not having mastered the skill to having mastered it after a practice opportunity; $p(S)$ is the probability the student gives a wrong answer even though she has mastered the skill; and $p(G)$ is the probability the student gives the correct answer even though she has not mastered the skill. After the student is given a practice opportunity, and gets the problem correct or incorrect, the model updates the probability of the student's underlying mastery state. Though in Knowledge Tracing the standard approach is to first update the estimate of the student's mastery given their observed response, and then update their mastery as to whether they have learned, in this paper we instead adopt the alternate convention (often used in other Bayesian models) of first updating the probability the student has mastered the material given they received a new practice opportunity, and then updating that estimate given the observed student response. This yields the following equations for computing $p(L_{t+1})$, the probability the student has mastered the skill at time step $t+1$, as a function of the probability the student has mastered the skill at the prior time step $p(L_t)$, the observed student response, and the 4 parameters:

$$p(L_{t+1}|c) = \frac{(1-p(S))(p(L_t)+p(T)(1-p(L_t)))}{(1-p(S))(p(L_t)+p(T)(1-p(L_t)))+p(G)(1-p(T))(1-p(L_t))} \quad (1)$$

$$p(L_{t+1}|w) = \frac{p(S)(p(L_t)+p(T)(1-p(L_t)))}{p(S)(p(L_t)+p(T)(1-p(L_t)))+(1-p(G))(1-p(T))(1-p(L_t))} \quad (2)$$

The basic model assumes that the student never forgets a skill, so once it is mastered, it stays mastered.

Often Knowledge Tracing is paired with mastery learning. Here the goal is for the student to master the desired skill (or set of skills). As the student's mastery level is hidden, a typical approach is to continue to provide practice opportunities to the student until, based on the student's responses, the probability that the student has mastered the skill $p(L_t)$ exceeds some prespecified threshold, such as 95%.

3. METHODS

Our interest is in characterizing the distribution of model parameters associated with individual students, and examining how this impacts the amount of practice opportunities needed for a student to reach mastery, versus using parameters fit to the population.

3.1 Parameter Fitting

We fit a KT model for each individual student, where the model consists of the 4 parameters specified in the prior section, $\langle p(L_0), p(T), p(S), p(G)\rangle$. The input data for each student consists of a trajectory of practice opportunities for a particular skill, where the j-th entry contains whether the student got this opportunity correct or incorrect. We would like to compute the model parameters that maximize the likelihood of the observed data. If at each j we knew if the student had mastered or not mastered the skill, then computing the best parameters would simply involve counting. For example, to estimate the probability of slipping $p(S)$ we would simply count up the number of instances where the student had mastered the skill but got the problem wrong, divided by the number of times the student had mastered the skill. However, we don't know if the student has mastered the skill or not. Therefore we use Expectation Maximization to find parameters that locally maximize the likelihood of the observed data.

Expectation maximization (EM) is an iterative algorithm where each iteration consists of two stages. In the first stage we fix the current estimates of the parameters and use these parameters to estimate the probability the student has mastered or not mastered ($\hat{p}(L_t)$ and $1-\hat{p}(L_t)$) the skill at each of the time steps in the trajectory. These estimates can be efficiently computed using the forward-backward algorithm, whose computational complexity is linear in the trajectory length and quadratic in the number of hidden states: here there are only 2 possible states, mastered or unmastered.

In the second stage, new parameter estimates are computed given these estimated probabilities of mastery. The new $p(S)$ parameter is computed by taking all instances where the student got the problem wrong, and summing the probability that the student had really mastered the skill in all those instances, divided by the probability of the student having mastered the skill on all time steps:

$$\hat{p}(S) = \frac{\sum_j \delta_j(w)\hat{p}(L_j)}{\sum_j \hat{p}(L_j)},$$

where $\delta_j(wrong)$ is 1 if the j-th student response was wrong, and 0 otherwise. A new estimate of $p(G)$ is computed as

$$\hat{p}(G) = \frac{\sum_j \delta_j(c)(1-\hat{p}(L_j))}{\sum_j (1-\hat{p}(L_j))},$$

where in the numerator we sum over all instances where the student got the answer correct. Updating the parameter estimate of $p(T)$ involves the probability of the student transitioning from not having the skill mastered to having mastered the skill. $p(L_0)$ is estimated from $\hat{p}(L_0)$.

Algorithm 1 ExpOppNeed

Input: $p^{tsg} = \langle p(T), p(S), p(G)\rangle$, $p(L_j)$, d, p_{path}, ϵ
Output: EO
if $p_{path} < \epsilon$ **then**
  return $EO = 0$;
else
  if $p(L_j) \geq d$ **then**
    return $EO = 0$ {reached mastery}
  else
    $p(c|p(L_j))$ (Eqn. 3) {prob get correct}
    $p(L_{j+1}|c)$ (Eqn. 1) {prob mastery if get correct}
    $p_{path,c} = p_{path} * p(c|p(L_j))$
    {Compute further opp. need if get problem correct}
    EO_c=ExpOppNeed(p^{tsg}, $p(L_{j+1}|c)$, d, $p_{path,c}$, ϵ)
    $p(w|p(L_j))$ (Eqn. 4) {prob get wrong}
    $p(L_{j+1}|w)$ (Eqn. 2) {prob mastery if get wrong}
    $p_{path,w} = p_{path} * p(w|p(L_j))$
    {Compute further opp. need if get problem wrong}
    EO_w=ExpOppNeed(p^{tsg}, $p(L_{j+1}|w)$, d, $p_{path,w}$, ϵ)
    return $EO = 1+p(c|p(L_j))*EO_c+p(w|p(L_j))*EO_w$
  end if
end if

Then the whole process repeats, using the updated parameter estimates to compute new estimates of the underlying probability of mastery. These iterations continue until the process converges to a fixed point, which is guaranteed to occur.

There are several limitations to using EM that are relevant for our purposes. First, EM is only guaranteed to converge to a local optima of the likelihood function. Second, the parameters found may not be semantically plausible. This can occur for the $p(S)$ and $p(G)$ parameters. We expect $p(S)$ to be < 0.5 since if it is larger, it means that it is more likely for a student to get a problem wrong than right when she has mastered the given skill. Similarly, we expect $p(G)$ to be < 0.5, since a student is more likely to get a problem wrong than right if she has not mastered the associated skill.

Given these concerns, we performed a discretized search over the $p(G)$ and $p(S)$ parameters, as well as a discretized search over the initial probability of mastery $p(L_0)$. We included this parameter as we initially had some results where the $p(L_0)$ was fit by EM to be 0 or 1 and we suspect it is probable that students should lie inside these extremes. We ran EM to compute the best $p(T)$ for each tuple of $\langle p(L_0), p(G), p(S)\rangle$ parameters, and selected the model parameter tuple with the highest likelihood.

In addition, we also fit model parameters by aggregating all student data together, and fitting a single population model. In the rest of the paper we will use p_i to denote the parameters fit for the i-th student, and p_{pop} to denote the parameters fit for the whole population of students.

3.2 Expected Time to Mastery

Given the estimated student learning model parameters, we next compute the expected number of practice opportunities it will take for student i to reach mastery. We take the standard approach of using a threshold d to define mastery: when $p(L_t) \geq d$, the student is defined to have reached mastery. Consider a given student i, with her associated parameters $\langle p_i(L_0), p_i(T), p_i(S), p_i(G)\rangle$. First, let $p_i(L_t|obs_{1:t})$ be the probability of student i having mastered the skill after having t practice opportunities, and having made the responses $\langle obs_1, obs_2, \ldots obs_t\rangle$ to each respective practice opportunity (where $obs_j = correct, wrong$). Note this expression can be calculated by sequentially applying either Equation 1 or Equation 2, depending on whether the student got that practice opportunity correct or incorrect.

Algorithm 2 ExpOppNeedPop

Input: $p_i^{tsg} = \langle p_i(T), p_i(S), p_i(G)\rangle$, $p_i(L_j)$,d,p_{path}, ϵ, $p_{pop}^{tsg} = \langle p_{pop}(T), p_{pop}(S), p_{pop}(G)\rangle$, $p_{pop}(L_j)$,
Output: EO
if $p_{path} < \epsilon$ **then**
  return $EO = 0$;
else
  if $p_{pop}(L_j) \geq d$ **then**
    return $EO = 0$ {reached mastery under pop model}
  else
    $p(c|p_i(L_j))$ (Eqn. 3) {prob student i gets correct}
    {Prob. mastery if get correct}
    $p_{pop}(L_{j+1}|c)$ (Eqn. 1) {under population model}
    $p_i(L_{j+1}|c)$ (Eqn. 1) {under student i's model}
    $p_{path,c} = p_{path} * p(c|p_i(L_j))$
    {Compute further opp. need if get problem correct}
    EO_c=ExpOppNeedPop(p_i^{tsg}, $p_i(L_{j+1}|c)$, d, $p_{path,c}$, ϵ, p_{pop}^{tsg},$p_{pop}(L_{j+1}|c)$)
    $p(w|p(L_j))$ (Eqn. 4) {prob student i gets wrong}
    {Prob. mastery if get wrong}
    $p_{pop}(L_{j+1}|w)$ (Eqn. 2) {under population model}
    $p_i(L_{j+1}|w)$ (Eqn. 2) {under student i's model}
    $p_{path,w} = p_{path} * p(w|p(L_j))$
    {Compute further opp. need if get problem wrong}
    EO_w=ExpOppNeedPop(p_i^{tsg}, $p_i(L_{j+1}|c)$, d, $p_{path,c}$, ϵ, p_{pop}^{tsg}, $p_{pop}(L_{j+1}|c)$)
    return $EO = 1+p(c|p_i(L_j))*EO_c+p(w|p_i(L_j))*EO_w$
  end if
end if

A second important pair of quantities is the probability of observing a particular student response (correct or incorrect) at time t to an opportunity, given the current probability of mastery $p_i(L_t)$. These are:

$$p(c|p_i(L_t)) = p(G)(1 - p_i(L_t)) + (1 - p(S))p_i(L_t) \quad (3)$$
$$p(w|p_i(L_t)) = (1 - p(G))(1 - p_i(L_t)) + p(S)p_i(L_t). \quad (4)$$

Using Equations 1,2,3 and 4, we can compute the expected number of practice opportunities for a student to reach mastery using a recursive algorithm. Intuitively, the expected number of additional practice opportunities needed depends on the current probability that the student has mastered the material, $p_i(L_t)$. If $p_i(L_t) \geq d$ then no more practice opportunities are needed. Otherwise at least one more practice opportunity is needed. Depending on whether the student gets that opportunity correct or incorrect, then $p_i(L_{t+1})$ will get updated accordingly, and then we can compute the expected number of additional opportunities needed from the resulting probability of mastery. But we don't know in advance whether the student will get the next question correct or not, so we take the expectation over these two possibilities. More precisely, let $EO_i(p_i(L_t))$ be the expected number of further practice opportunities needed given the

current probability of mastery. Then if $p_i(L_t) \geq d$,

$$EO_i(p_i(L_t)) = 0, \quad (5)$$

otherwise,

$$\begin{aligned} EO_i(p_i(L_t)) &= 1 + p(c|p_i(L_t))EO_i(p_i(L_{t+1}|c)) \\ &+ p(w|p_i(L_t))EO_i(p_i(L_{t+1}|w)). \end{aligned} \quad (6)$$

The full algorithm, ExpOppNeed, is displayed in Algorithm 1. The algorithm is called using the student's initial probability of mastery, $p_i(L_0)$. Calculating the expected number of practice opportunities needed for individual i is done using Equations 5 and 6. The algorithm also maintains a variable p_{path} that represents the probability of the path of observations, given the student's initial probability of mastery $p_i(L_0)$. When p_{path} falls below a threshold ϵ, we also set the expected number of future practice opportunities needed to 0. This is an approximation, but is necessary since in some models the student will never reach a sufficient probability of mastery if she gets all practice opportunities incorrect. Since in the real world such a student would not do an infinite number of problems, and to prevent infinite recursion, we terminate when $p_{path} < \epsilon$.

3.3 Population vs Individual

As mentioned previously, often intelligent tutoring systems operate with a fixed set of parameters for all students, which we denote the population model p_{pop}. We are interested in what impact this assumption means for individual students whose learning parameters may not match the aggregate population model parameters. More specifically, we are interested in the expected needed number of practice opportunities if the probability used to assess mastery is calculated using the population parameters instead of the student's own parameters. Note that we cannot simply apply Algorithm 1 using the population parameters. Doing so would calculate the expected number of practice opportunities needed if we assess mastery using the probability given by the population model, and if correct and wrong observations are generated according to a student who has the same parameters as the population model. In contrast, we are interested in the case when a real student (with different parameters than the population model) is responding correctly or incorrectly to practice opportunities, but that the system is monitoring the student's progress using a different set of model parameters. This is likely to occur in tutors that use population models.

To estimate the expected needed number of practice opportunities in this situation, we need to estimate the probability of a correct or wrong answer being generated according to student i's model parameters p_i, but assess if the student has reached a sufficient probability of mastery using the population parameters p_{pop}. This means that during our calculations we need to maintain two separate estimates of the probability of mastery for the student, $p_{pop}(L_t|obs_{1:t})$ and $p_i(L_t|obs_{1:t})$, which respectively represent the probability under the population model parameters, given the observed sequence of correct and incorrect answers, and the probability under student i's own parameters.

Let $EO_i^{pop}(p_{pop}(L_t), p_i(L_t))$ represent the expected number of additional practice opportunities need to reach mastery, given the current estimate of the probability of mastery under the population model is $p_{pop}(L_t)$, and under the individual model is $p_i(L_t)$. EO_i^{pop} is computed in Algorithm 2, which is a modification of Algorithm 1.

4. EXPERIMENTS

We used a dataset of student responses to 42 problem sets in the ASSISTment system [10]. Each set corresponded to 1 skill, and the number of problems given per skill ranged from 4-13. We first compute individual student parameters. Since the number of data samples was fairly limited per skill, we computed a set of parameters for all skills an individual did problems on. Though this is an approximation, in existing tutoring systems with good learning outcomes, multiple skills are often modeled as having the same parameters. We will consider this and other assumptions made further in the discussion section. We selected the subset of 265 students who did problems on 10 or more skills. For this subset, the mean number of skills per student was 12.79 (range=[10,22]), yielding an average of 69.57 (range=[45,132]) total problem tries per student. We fit learning parameters $\langle p(L_0), p(T), p(S), P(G)\rangle$ to each individual, and to the aggregated dataset across all individuals.

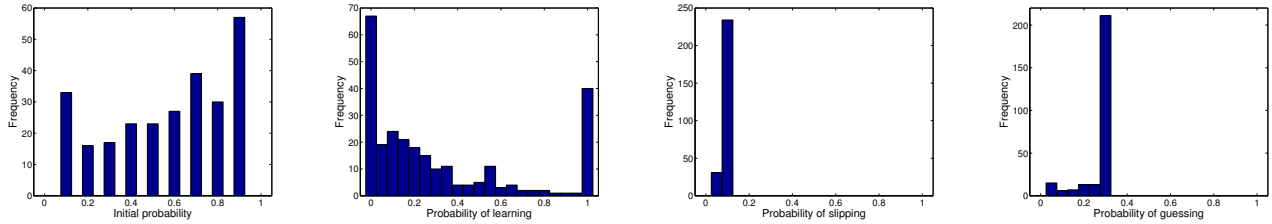
When fitting the model parameters, we restricted $p(S)$ to lie in $[0.05, 0.1]$ and $p(G)$ to lie in $[0.05, 0.3]$, trying values incremented by 0.05: these bounds have been used in prior work [1]. $p(L_0)$ was restricted to lie in $[0.1, 0.9]$, and values were tried at increments of 0.1. $p(T)$ was fit using EM, as described previously. We ran EM for each individual (or the aggregated dataset) with 10 different initializations, and chose the one with the highest log likelihood. The computational time needed for EM to converge (for a single individual's data) was approximately 0.6s.

Following the standard procedure in Knowledge Tracing [8], we set the mastery threshold d to 0.95. We set the expected number of future practice opportunities from $p(L_t|obs_{1:t})$ to 0 if the sequence of observations $obs_{1:t}$ had a probability of less than $\epsilon = 10^{-7}$. We later discuss the effect of ϵ.

5. RESULTS

We first report the resulting distributions of the estimated student learning model parameters, computed for each student separately. Histograms of these distributions are displayed for the four learning parameters in Figures 1(a), 1(b), 1(c), and 1(d). For the slip $p(S)$ and guess $p(G)$ parameters, the distribution is fairly peaked. The majority of individuals have $p(S)$ and $p(G)$ parameters that are close to the parameters for the full population. However, for both the initial probability of mastery, $p(L_0)$, and the probability of mastering a skill after not having understood it, $p(T)$, there is a large spread of values. In each figure we have also included the parameter estimated if the data from all 265 individuals is aggregated, and a single set of model parameters is estimated. It is clear that for both $p(L_0)$ and $p(T)$, there will be many individuals whose best fit parameters are quite far from the population parameters. This suggests that the expected number of practice opportunities needed for some students may differ if student mastery modeling is done using the individual student's own estimated parameters, compared to if we maintain a probability of the student's mastery level using the population parameters.

Indeed, this is what we observe. We start by computing the expected number of practice opportunities needed for



(a) $p(L_0)$. $p_{pop}(L_0) = 0.6$ (b) $P(T)$. $p_{pop}(T) = 0.086$ (c) $P(S)$. $p_{pop}(S) = 0.1$ (d) $P(G)$. $p_{pop}(G) = 0.3$

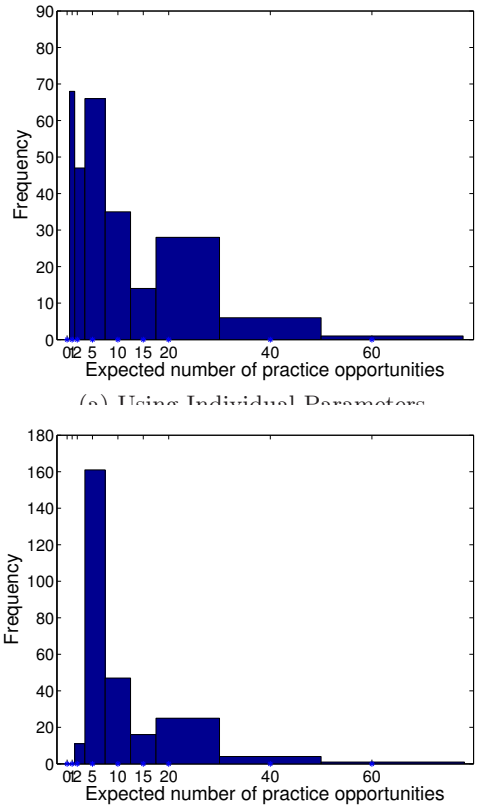
Figure 1: Histogram of parameters fit for each individual.

an individual, given the model parameters p_i estimated for that individual, $EO_i(p_i(L_0))$. The probability of the student reaching mastery is assessed using the student's parameters p_i. A histogram of the expected number of practice opportunities for an individual to reach mastery as evaluated using their own parameters is displayed in Figure 2(a). We have binned the data to illustrate the distribution of the individuals. Over half the individuals are expected to need less than 5 practice opportunities to reach mastery. However, there are also a significant number of individuals that are expected to need a much larger number of practice opportunities to reach mastery: over 40 students require at least 15 practice opportunities, and over 30 of those require 20 or more. This graph suggests that there is a significant spread in the required amount of practice necessary for different students.

We next examine the expected number of practice opportunities needed for a student i, if we evaluate mastery using the population parameters, $EO_i^{pop}(p_{pop}(L_0), p_i(L_0))$. Note that this is the situation that occurs in existing tutors that use a single set of model parameters for all students. Figure 2(b) displays a histogram of the expected number of practice opportunities needed when evaluating mastery using the population parameters. At a rough glance the histogram looks similar to Figure 2(a), but upon closer examination, the individuals histogram has its peak at about 2 whereas the population model has its peak at about 5 expected number of time steps.

We examine this discrepancy more systematically in Figures 3(a) and 3(b). There are three possible situations that could result from estimating the number of practice opportunities if we evaluate mastery learning using a set of population parameters, on an individual with their own set of parameters. In the first case a student might have to do more practice problems than they would if we evaluated mastery using the student's own parameters. For example, this might occur if the probability of mastering an unmastered skill is higher for individual i than for the group, $p_i(T) > p_{pop}(T)$. In the second case an individual will need a very similar number of practice opportunities, whether we evaluate their skill mastery using the estimated population parameters, or the individual's own parameters. In the third case are students who are expected to need more practice opportunities than that predicted if using the population model.

The first and third situations are the ones of concern. In the first case, it would mean that some students are having to do more problems than really needed. Since there is only a finite amount of time in the school year, this means that these students would be likely to cover less material than they are



(a) Using Individual Parameters

(b) Using Population Parameters

Figure 2: The expected number of practice opportunities to reach mastery, where the probability of mastery is evaluated either based on the individual's estimated model parameters, or the estimated population model parameters.

capable of, due to doing extra unnecessary problems. In addition, doing more problems than required might contribute to student boredom and disengagement. In the third case, a student may be recorded as having mastered a skill before he has really reached the required threshold probability of mastery. This means that the student could be advanced to later skills, some of which might assume knowledge of this earlier skill, without the student having actually understood the earlier skill to a satisfactory level.

Therefore our next objective was to ascertain if there was evidence that either case 1 or case 3 occurred in this dataset. For each individual i we computed the difference in expected number of practice opportunities needed using the popula-

tion model to assess mastery, versus the individual model:

$$\Delta(i, pop) = EO_i^{pop}(p_{pop}(L_0), p_i(L_0)) - EO_i(p_i(L_0))). \quad (7)$$

Figure 3(a) displays a histogram of $\Delta(i, pop)$. Note that higher values mean that the expected number of practice opportunities is more under the population parameters, than if one used the individual's parameters, aka case 1. Negative numbers are when the individual is expected to need more opportunities given their own parameters, compared to the population parameters, aka case 3. We can see that there are a number of individuals that fall into either case. One way for defining students in case 1 is students that are expected to be given one or more additional practice opportunities if the population model is used to do assessment of reaching the threshold probability of mastery, versus the expected number using the student's own parameter. Using this definition 146 students (or 55%) fall into case 1, where the mean $\Delta(i, pop)$ of this group is 2.18. Though important, this may not sound like a large number. But the importance of this difference becomes more clear when we consider the ratio of expected practice opportunities,

$$r(i, pop) = \frac{EO_i^{pop}(p_{pop}(L_0), p_i(L_0))}{EO_i(p_i(L_0))}. \quad (8)$$

A histogram of $r(i, pop)$ is displayed in Figure 3(b). This figure shows that over 50 students in our dataset would be expected to do *twice* as many problems if the population parameters are used to assess mastery learning, compared to using the individual's parameters. For example, one student's parameters were $p_i(L_0) = 0.3, p_i(T) = 0.86, p_i(S) = 0.1, p_i(G) = 0.3$. Here $\Delta_i(i, pop) = 2.74$ and $r(i, pop) = 3.32$. For such students, they could potentially be covering twice or more as much material in the same time if their own model parameters were known and used to evaluate mastery. This could have very significant effects on such students' learning, and, equally importantly, at least in this dataset, this effects a significant proportion of the students, over 20% of the student population in our dataset.

Returning to estimates of $\Delta(i, pop)$, Figure 3(a) also reveals that there are a number of individuals whose $\Delta(i, pop) < -1$, indicating that we expect such individuals to need at least one more practice opportunity if we use their own parameters to assess mastery, compared to using the population model parameters. This is case 3, and there are 21 individuals who fall into this category. The average $\Delta(i, pop)$ for this subset is -1.47. An additional interesting quantity in this case is the expected probability of mastery of individual i, if mastery is assessed using the population model parameters. This quantity is displayed in Figure 3(c) for all students whose $\Delta(i, pop) < 0$, aka students that are expected to need more practice opportunities than they will receive if the population parameters are used. Though a subset of these students have an expected probability of mastery that is fairly close to the desired threshold (0.95), 44 students ($\sim$17%) have an expected probability of mastery of 0.6 or less. For example, one student's estimated parameters were $p_i(L_0) = 0.1, p_i(T) = 0.028, p_i(S) = 0.1, p_i(G) = 0.3$, and the expected actual average mastery for this student when his mastery is assessed using the population parameters, was only 0.47, far below the 0.95 threshold of mastery desired. Such individuals are unlikely to have sufficiently understood the current skill, yet would be considered to have reached mastery, and moved on to the next skill.

Therefore, in this dataset, there appear to be a substantial number of students who fall into case 1 or case 3, who would be likely to have to do more problems than is actually necessary, or who would not have reached the desired probability of mastery, respectively.

Some readers might be concerned that the reported results are computed using an algorithm that only approximates the expected number of needed practice opportunities. This approximation arises from setting $EO_i(p(L_t|obs_{1:t})$ to 0 if the probability of the path of observations is less than set threshold, $p(obs_{1:t}) \leq \epsilon = 10^{-7}$, which was done from computational reasons. As one exploration, we repeated our analysis using $\epsilon = 10^{-5}$ and observed very similar results. More generally, our approximation will typically underestimate the difference $\Delta(i, pop)$ in the two cases we are interested in. Intuitively this is because whichever parameter set, either p_i or p_{pop}, has the higher learning rate and/or higher initial probability of mastery, that model will be less effected by terminating low probability paths, because its expected number of time steps will be shorter. We tested this intuition by taking the population parameters (where $p(T) = 0.086$), and creating two alternative models, one "fast" learner where $p(T) = 0.2$, and one "slow" learner where $p(T) = 0.02$. We used different ϵs and evaluated Δ for these different models:

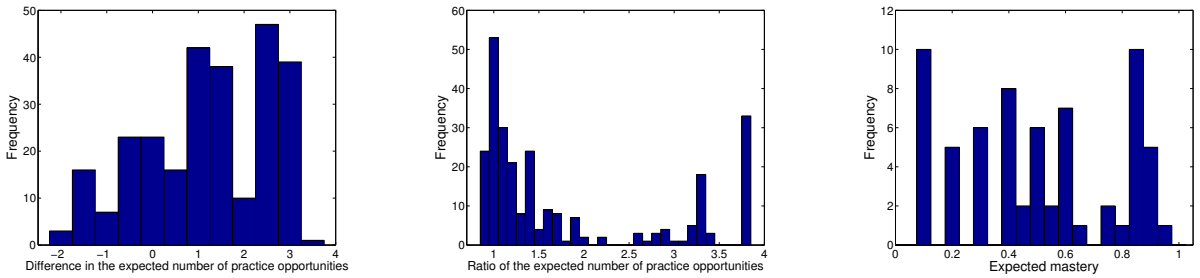
ϵ	10^{-5}	10^{-6}	10^{-7}	$0.5 * 10^{-7}$
$\Delta(fast, pop)$	2.511	2.844	3.103	3.175
$\Delta(slow, pop)$	-1.700	-2.374	-3.100	-3.311

Note that the magnitude of the difference is increasing as $\epsilon \to 0$, indicating that by using a $\epsilon > 0$, we are likely to be underestimating the true difference in expected time steps between an individual model and when using a population model. This suggest that we expect our reported results to be an underestimate of the true significance of the impact of using population parameters instead of individual parameters when assessing mastery.

6. RELATED WORK

Student modeling is naturally of key interest to the intelligent tutoring systems and educational data mining community. Knowledge Tracing [8] has been explored extensively in the research community and is also used in effective intelligent tutoring systems.

Over the last 5 years there has been significant interest in methods for fitting the parameters in the Knowledge Tracing model. Beck and Chang [3] pointed out that in some cases, more than 1 set of KT parameters predict exactly the same student performance (the probability the student will get the next answer correct as the number of practice opportunities increases). This means that the model suffers from an identifiability problem, where there are more than 1 set of parameters that equally well fit the observed data. In addition, Beck and Chang discuss the issue that when model parameters are fit by EM, the resulting model parameters may be implausible from a seminar perspective, for example if the probability of guessing the right answer $p(G)$ is higher than 0.5, then students that have a greater chance of getting a problem right than wrong when they haven't mastered the skill, which seems unlikely. Beck and Chang addressed the issue of identifiability by using hand set Dirichlet priors to introduce domain. Baker, Corbett and Aleven [1] presented a machine learning method to es-



(a) Population-Individual (b) Ratio $\frac{Population}{Individual}$ (c) Expected mastery

Figure 3: The expected number of practice opportunities needed when using the population parameters to assess if the student has reached a threshold of mastery(a), versus the individual's own parameters(b). (c) shows the expected level of mastery of students if using the population parameters to assess mastery, for students i that needed more practice, $\Delta(i, pop) > 0$.

timate the probability a student has slipped or guessed as a way to address both identifiability and plausibility. Rai and colleagues [11; 12] have since investigated learning Dirichlet priors from data, or multiple Dirichlets, to improve parameter plausibility. Ritter et al. [13] clustered similar skills together to reduce the number of model parameters required in order to find better parameters given a fixed set of data.

There has also been a limited amount of work on trying to individualize the Knowledge Tracing parameters. In their key paper, Corbett and Andersen [8] investigated individualizing the KT parameters by first learning a set of parameters for all students, and then computed weights to adjust these parameters to individual students. While the resulting weighted parameters lead to predictions that correlated better on student test performance than a nonweighted model, the weighted model did not generally have a better predictive accuracy on the student's test score. In contrast to weighting, Pardos and Heffernan [10] simply enabled the initial probability of mastery, $p(L_0)$, to vary among individuals during the initial fitting process. The resulting model predicted student responses better than a Knowledge Tracing model where all parameters are identical for all students. Our work is related to Pardos and Heffernan, except we fit all 4 KT parameters individually to each student.

There are a number of alternate mathematical models of student learning and performance, including Learning Factors Analysis [5]. A number of these models use a logistic regression approach to directly modeling the probability a student will get the next practice opportunity correct, without an additional representation of the student's latent skill mastery state. However to our knowledge, unlike the Knowledge Tracing approach, these models have not been used to help make decisions about how much additional practice a student needs. One simple extension to the idea of thresholding when the student's probability of mastery has reached a sufficient level would be to continue providing the student with practice opportunities until the logistic regression model exceeds a prefixed threshold of the student getting the next problem correct.

7. DISCUSSION AND FUTURE WORK

Our results suggest there is a large amount of variation in student learning parameters in this dataset, and this variation has important implications for the amount of practice opportunities that should be given to individual students.

There are a number of ways that the work presented in this paper could be further improved in the future. Right now we are combining a brute force discretized grid search over a subset of the Knowledge Tracing parameters, with performing Expectation Maximization over the remaining parameters. It would be interesting to explore other methods for fitting KT parameters to data, such as those previously proposed [3; 1; 11; 12; 13].

We currently assume all the skill parameters for an individual student are the same. This was done in order to have a larger number of data points to use in order to fit the student model parameters, and because there is prior precedent in successful tutors of using the same skill parameters for multiple skills. However, we suspect that at least for some skills, the parameters differ. Therefore we have two axes of potential variation: the individual skills, and the individual students. One natural concern is that increasing the number of model parameters (by modeling individual skills or individual students) leads to a danger of overfitting, since there will be less data for each subset of parameters we wish to estimate. Indeed, prior work by Ritter et al. [13] found that by clustering skills into groups, and fitting KT parameters to each cluster (instead of each skill), the resulting clusters generalized better to new students than fitting data to individual skills. We plan to perform a related analysis on seeing if there appear to be different clusters of students, all of which share similar skill parameters. More generally, we are interested in developing hierarchical models of skill parameters, which could fit individual student and individual skill parameters, but do so in a way that encourages clustering of both skills and students. Latent Dirichlet Allocation [4], a method for modeling collections that share features, such as documents sharing subsets of topics, may be a relevant approach.

Though our results suggest that individual variation in student model parameters can exist, and lead to overpractice or underpractice if these differences are ignored by using a population model, we have only analyzed a single dataset. Further experiments should be performed on other datasets to see if similar results are obtained.

Assuming these results hold in other cases, to us the most exciting issue is how to modify automated mastery teaching to enhance student learning, assuming that individual students' parameters vary. In the current paper we fit individ-

ual model parameters to the student data after the students have already completed all practice opportunities. In real tutoring settings, the student parameters will be unknown. Therefore, in order to test whether using individual student parameters benefits students using real classroom studies, we will need to infer students' parameters while simultaneously monitoring their performance and deciding whether the student has reached a sufficient threshold probability of mastery. Corbett and Andersen [8] used a fixed initial set of exercises as a diagnostic period to learn student parameter weights. After this mastery learning proceeds using parameters that incorporate the student's individual weights. Of course, this begs the question of how long a diagnostic period is required to fit a good estimate of the individual's parameters, while simultaneously being short enough that the individual may benefit from having individualized parameters: if we can only fit individual parameters after the student has reached mastery, this method is unlikely to be effective. The success of this approach also depends on other modeling assumptions. For example, if we assume that an individual has the same set of parameters for all skills, then one approach is to use the population skill parameters when the student is learning the first skill, and then take the student data from that first skill and fit a set of KT parameters to that data for the individual. This set could then be used as the student learns other skills. One could also imagine taking a Bayesian approach and explicitly modeling the uncertainty over a student's learning parameters. This uncertainty could be updated as the student responds to practice opportunities. One challenge in this setting is that in a Knowledge Tracing model the student's true state of mastery is never observed. This poses some interesting technical challenges when updating a distribution over model parameters.

Finally, in this paper we focused on a Knowledge Tracing model of student learning. KT models have been often used in mastery tutoring systems. In contrast, we are not aware of prior work that use logistic regression student learning models (e.g. [5]) as part of an adaptive instructional strategy that decides when to stop giving the student practice opportunities. Logistic regression student models often incorporate both student-specific and shared population parameters. Therefore in the future we are interested in incorporating logistic regression models into a tutoring strategy, and comparing this with our individualized KT model mastery-learning approach, in terms of their effect on the expected amount of practice needed.

8. CONCLUSIONS

In this paper we fit all 4 Knowledge Tracing parameters to each individual student from a set of tutoring data, and examined the resulting parameter distributions. The resulting observed parameter distribution was found to have interesting implications for instruction versus using a single model for all students. About 20% of students would have to do approximately double the number of practice problems in order to reach the threshold of mastery defined using the population parameters, compared to using the student's own parameters to assess probability of mastery. Another $\sim$17% of students would be expected to have a probability of mastery of only 60% or less when the population model would expect the student is at a probability of mastery of 95% or higher. This suggests that using a single set of population parameters for all students in a tutoring system may result in a significant portion of students covering less content than they are capable of, due to being required to complete redundant practice problems, and equally importantly, may advance some students to later skills before they are ready. In the future we will investigate how to learn and incorporate individualized parameters during the tutoring process.

9. ACKNOWLEDGEMENTS

The authors thank the Pittsburgh Science of Learning Center and Google for support.

10. REFERENCES

[1] S. J. d. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian estimation. In *ITS*, 2008.

[2] J. Beck. Difficulties in inferring student knowledge from observations (and why you should care). In *AIED Workshop on EDM*, 2007.

[3] J. Beck and K. Chang. Identifiability: A fundamental problem of student modeling. In *UM*, 2007.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[5] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis - a general method for cognitive model evaluation and improvement. In *ITS*, 2006.

[6] H. Cen, K. Koedinger, and B. Junker. Is Over Practice Necessary? Improving Learning Efficiency with the Cognitiive Tutor using Educational Data Mining. In *AIED*, 2007.

[7] M. Chi, K. Koedinger, G. Gordon, P. Jordan, and K. VanLehn. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *EDM*, 2011.

[8] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[9] K. R. Koedinger, J. Anderson, W. Hadley, and M. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8:30–43, 1997.

[10] Z. Pardos and N. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *UMAP*, 2010.

[11] D. Rai, Y. Gong, and J. Beck. Using Dirichlet priors to improve model parameter plausibility. In *EDM*, 2009.

[12] D. Rai, Y. Gong, and N. Heffernan. Using multiple Dirichlet distributions to improve model parameter plausibility. In *EDM*, 2010.

[13] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the knowledge tracing space. In *EDM*, 2009.