Difficulties in inferring student knowledge from observations (and why you should care)

Joseph E. Beck
joseph.beck@EducationalDataMining.org
Machine Learning Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA USA

Abstract. Student modeling has a long history in the field of intelligent educational software and is the basis for many tutorial decisions. Furthermore, the task of assessing a student's level of knowledge is a basic building block in the educational data mining process. If we cannot estimate what students know, it is difficult to perform fine-grained analyses to see if a system's teaching actions are having a positive effect. In this paper, we demonstrate that there are several unaddressed problems with student model construction that negatively affect the inferences we can make. We present two partial solutions to these problems, using Expectation Maximization to estimate parameters and using Dirichlet priors to bias the model fit procedure. Aside from reliably improving model fit in predictive accuracy, these approaches might result in model parameters that are more plausible. Although parameter plausibility is difficult to quantify, we discuss some guidelines and propose a derived measure of predicted number of trials until mastery as a method for evaluating model parameters.

Keywords: student modeling, mastery learning, parameter estimation, Bayesian networks, Dirichlet priors

1 Introduction

The focus of this paper is on the difficulties in mapping observable student performance to estimate his level of knowledge about underlying skills. This task, better known as student modeling, has been around for at least three decades [1]. Given the long history, it is fair to ask whether there are large, fundamental problems to be overcome. We discuss several problems resulting from the existence of local-maxima as well as multiple global-maxima in the parameter search space for a common student modeling approach. These problems may seem esoteric, but they result in seemingly similar models that make very different claims about the student's knowledge. There are two direct impacts of the results presented in this paper on Educational Data Mining (EDM). First, since estimating student knowledge is relatively well understood, it should be one of the easiest tasks for us to analyze. Given the surprising complexity and difficulty in evaluating models, we should be cautious about our analyses of more exotic phenomena. Second, fine-grained estimates of student knowledge are a useful lever in attacking other data mining tasks. For example, it is difficult to "measure the effect of individual interventions" (from the workshop's call for papers) if we cannot track student knowledge.

The goal of student modeling is to take observations of a student's performance and use those to estimate the student's knowledge, goals, preferences, and other latent characteristics. The key aspect of the problem is that the characteristics of interest are not directly observable. Thus, some means of mapping the observations to characteristics is required. Knowledge tracing [2] is an example of such a technique, and is the one we will focus on in the explorations in this paper. Knowledge tracing, shown in Figure 1, is an approach for taking student observations and using those to estimate the student's level of knowledge. It assumes that students either know a skill or do not; similarly they either respond correctly to an item or not. There are two parameters, *slip* and *guess*, that mediate student knowledge and student performance. These two parameters are called the performance parameters in the model. An assumption of the model is that even if a student knows a skill, there is a chance he might still respond incorrectly to a question that utilizes that skill. This probability is the slip parameter. There are a variety of reasons for an incorrect response, the student could have made a simple typo (e.g. typed '12' instead of '21' for "7 x 3"), he could have been frustrated with the

system and behaving randomly, or he may not have felt like solving the problem and would rather have the tutor tell him how to solve it.

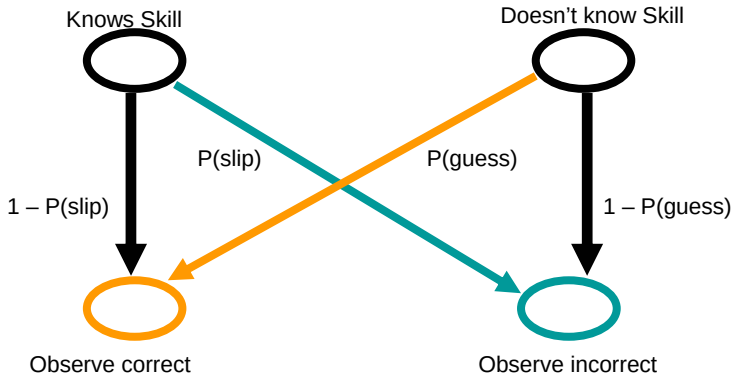

Figure 1. Diagram of knowledge tracing

Conversely, a student who does not know the skill might still be able to generate a correct response. This probability is referred to as the guess parameter. A guess could occur either through blind chance (e.g. in a 4-choice multiple choice test there is a ¼ chance of getting a question right even if one does not understand it), the student being able to utilize a weaker version of the correct rule that only applies in certain circumstances, or through errors in the tutor's scoring mechanism. For example, experiments using knowledge tracing with speech recognition had a very high guess parameter because the speech recognizer had a hard time noticing when students made mistakes [3].

In addition to the two performance parameters, there are two learning parameters. The first is K0, the likelihood the student knows the skill when he first uses the tutor. Initial knowledge can come from a variety of places including material from a prior course or declarative instruction delivered by the tutor before procedural lessons (e.g. [4]). The second learning parameter is t, the probability a student will acquire a skill as a result of an opportunity to practice it. Every skill to be tracked has these four parameters, slip, guess, K0, and t, associated with it.

To train a knowledge tracing model, historical data of student performance is provided to model fitting software. There exist a variety of software packages to perform the optimization such as the curve fitting approach used by some of the cognitive tutors[1] and the Bayes Net Toolkit for Student Modeling (BNT-SM, [5]). To use a knowledge tracing model, when a student begins using an ITS his initial knowledge estimates are set to the K0 estimates for each skill. When a student responds to an item, Bayesian inference is used to update his internal knowledge on the basis of whether he responded correctly or not, and on the skill's slip and guess parameters. The student is then credited with an opportunity to learn the skill via the t parameter.

This work also assumes the tutor's designers already know which skill will be modeled. Although the construction and refinement of domain models from data is an interesting topic (e.g. [6-8]), it is beyond the scope of this paper. This paper uses the knowledge tracing model as a platform for experimentation. We are not asserting knowledge tracing is correct, and in fact would argue the opposite; for one thing, it does not acknowledge that students can forget what they have learned. It is simply an approach that is the closest the AIED community has to a standard. Furthermore, it is the simplest modeling approach that has the promise to be analytically interesting. If we are able to uncover and study basic problems with a model as simple as knowledge tracing, it is probable that more complex modeling schemes will suffer similar drawbacks.

2 Difficulties

Knowledge tracing has two main flaws. The first is that the parameter estimation task is subject to local maxima. That means that given a set of observations, the parameters returned by model fitting software might not the best possible fit to the data. The second problem, less well known, is that there are multiple local maxima. That is, there can be more than one set of parameters that fit the data equally well.

[1] Source code is courtesy of Albert Corbett and Ryan Baker and is available at http://www.cs.cmu.edu/~rsbaker/curvefit.tar.gz

2.1 Addressing local maxima

The first problem, that of local maxima, is an important one to address. The parameters associated with the knowledge tracing model control how it updates its estimates on the basis of student performance. Obviously, we would like estimates that provide the most accurate reflection of the student's knowledge. Since we are unable to observe the student's knowledge directly, we instead settle for preferring parameter estimates that enable the student model to predict student performance.

This study compares two approaches for estimating knowledge tracing parameters. The first approach, curve fitting, uses conjugate gradient descent to find best fitting parameter estimates, and has been used in prior studies with the cognitive tutors. The second approach makes use of the equivalence between knowledge tracing and a simple dynamic Bayesian network (DBN) with two nodes [9]. By casting knowledge tracing as a simple DBN, it is possible to use Expectation Maximization (EM) to perform the parameter estimation. We have constructed a tool, the Bayesian Net Toolkit for Student Modeling (BNT-SM) that takes as input a file of observations and an XML file describing the network structure and outputs the best-fitting parameter estimates [5]. This toolkit is built on top of the Bayes Net Toolkit [10].

Our testbed for this study is data from a 1997 study using the Cognitive Geometry Tutor [11]. These data consist of 24 students with 4101 opportunities to practice the 15 geometry skills that make up the domain. We use these data with each toolkit's implementation of knowledge tracing to obtain best-fitting parameter estimates. It is important to note that in both cases we are using the exact same data and exact same statistical model form. The only aspect that varies is the optimization technique to fit the parameters.

After training each model on the data to obtain the knowledge tracing parameter estimates, we then used them to trace student knowledge on the basis of the performance data. For every student action in the data set, the model compared its estimate of the student's knowledge with his performance. To compare the two modeling approaches, we used the AUC (Area Under Curve) metric for the ROC curve described by the estimated student knowledge and observed performance. The curve fitting approach had an AUC of 0.652±0.01 while the Bayes Net Toolkit had an AUC of 0.696±0.01. Thus, EM appears to provide a somewhat, and statistically reliably, more predictive set of parameters for this study. In a prior study comparing curve fitting with EM using data from Project LISTEN's Reading Tutor, for curve fitting we had an AUC of 0.568, while EM provided a reliably better ($p<0.01$) predictions with an AUC of 0.61. Therefore, this new experimental result provides converging evidence that EM outperforms curve fitting in different domains.

2.2 Addressing multiple global maxima

Multiple global maxima mean there is more than one combination of parameters that minimizes the amount of error. At first consideration, having multiple global maxima does not seem like much of a problem. If there are several best-fitting solutions in the parameter space, that should make the model fitting task easier since there is more than one "right" answer. To understand why multiple global maxima are a problem, consider the three sets of hypothetical knowledge tracing parameters shown in Table 1. The *knowledge* model reflects a set of model parameters where students rarely guess, the *guess* model assumes that 30% of correct responses are due to randomness. This limit of 30% is the maximum allowed in the Cognitive Tutors [12]. The third model is similar to those used by Project Listen's Reading Tutor [13] for performing knowledge tracing on speech input [14]. This model's *guess* parameter is very high because of inaccuracies in the speech recognition signal.

We compute the learning and performance curves for each model by using the knowledge tracing equations and the parameter values from Table 1. Specifically, we initialize P(know) to be K0. After each practice opportunity, we update P(know) = P(know) + (1 – P(know)) * T. For example, at the second practice opportunity the *knowledge* model would have a P(know) of 0.56 + (1 – 0.56) * 0.1 = 0.604. To compute P(correct), we use the knowledge tracing formula to combine the estimated knowledge with the slip and guess parameters: P(correct) = P(know) * (1-slip) + (1 – P(know)) * guess. As seen in Figure 2, the three models have identical student performance[2]—somewhat surprising given that the models appear so different. The much larger surprise is that in spite of having identical performance, their estimates of student knowledge (right graph in Figure 2) are very different.

[2] Technically the three curves are not perfectly identical, however they are equivalent under finite precision arithmetic.

Given the same set of performance data, we have presented three knowledge tracing models that fit the data equally well. Even if the *guess* parameter is capped at 0.3, there are still two competing models that perform equally well. It is not feasible to attack this problem statistically, as all three solutions are best fitting models. If all we have available is the performance data of students attempting to solve a skill, even given arbitrarily large quantities of such data, we are stuck with no way to decide which model we should prefer.

Table 1. Parameters for three hypothetical knowledge tracing models

Parameter	Model		
	Knowledge	Guess	Reading Tutor
K0	0.56	0.36	0.01
T	0.1	0.1	0.1
Guess	0.00	0.30	0.53
Slip	0.05	0.05	0.05

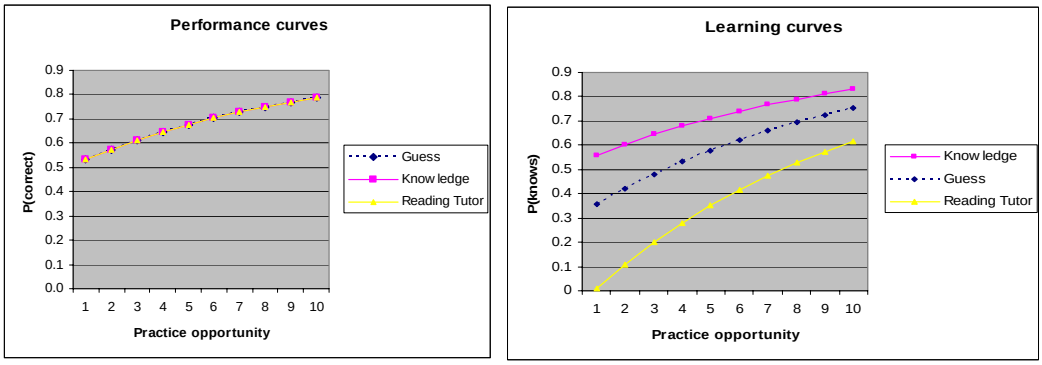

Figure 2. Three hypothetical knowledge tracing models illustrating the identifiability problem

Conceptually, model fitting is finding a set of model parameters, mp, that maximize P(data | mp). That is, the goal is to find a set of model parameters that make the data observed the least surprising. The problem with multiple global maxima is that there are several possible values of mp that maximize this likelihood. One possible workaround is to consider whether some parameter values are more likely than others. For example, few skills are mastered after only one practice opportunity. Therefore, given a choice between two competing models, we would prefer the one that required more than one practice opportunity to master a skill. Mathematically, we instead try to maximize P(data | mp) * P(mp). However, that leaves open the task of specifying the function P(mp).

Bayesian networks provide a natural mechanism, Dirichlet priors [15], for simultaneously maximizing the likelihood of the data and of the parameters. Dirichlet distributions are specified by a pair of numbers (α,β). Figure 3 shows an example (the dashed line) of the Dirichlet distribution for (9,6). This distribution is the one we used for prior experiments in biasing the model fitting process for Project LISTEN's Reading Tutor [16] to help guide the estimation of the K0 parameters. This distribution suggests that few skills have particularly high or low knowledge, and we expect students to have a moderate probability of mastering most skills. Conceptually, one can think of the conditional probability table of the graphical model being seeded with 9 instances of the student knowing the skill initially and 6 instances of him not. If there is substantial training data, the parameter estimation procedure is willing to move away from an estimate of 0.6. If there are few observations, the priors dominate the process. The distribution has a mean of $\alpha/(\alpha+\beta)$. Note that if both α and β increase, as in the solid curve in Figure 3, the mean of the distribution is unchanged but the variance is reduced. If you flip a coin and get 5 heads and 5 tails, you have moderate belief that P(heads) is around 0.5. If you flip the coin 100 times and get 50 heads and 50 tails, your average belief hasn't changed, but you are much less willing to believe that you have observed a biased sample and P(heads) is really 0.7. Similarly, Dirichlets enable researchers to not only specify the most likely value for a parameter but the confidence in the estimate.

We have explained a strategy, optimizing P(D|mp) * P(mp), for determining which model to prefer when several fit the data equally well, and have outlined a computational approach using Dirichlet priors. The problem is now how to specify the Dirichlet distribution. There are several sources of power we can use. One example is expert knowledge of the domain. If someone knows how quickly students tend to master a skill or the likelihood of knowing a skill, that knowledge can be used to set the priors. One complaint is that such an approach is not necessarily replicable as different people will use different experts.

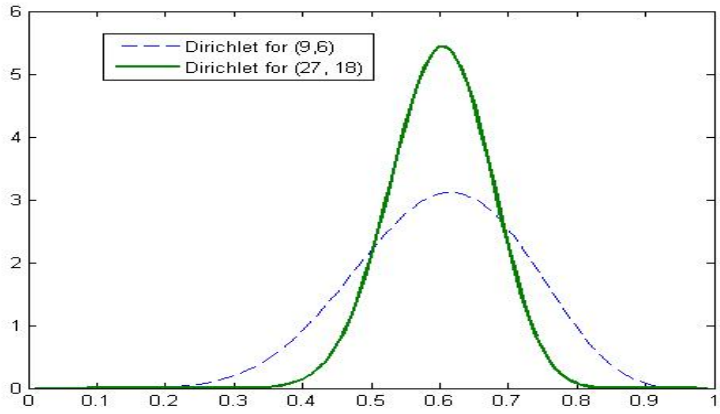

Figure 3. Sample Dirichlet distributions demonstrating decreasing variance

This paper proposes an alternate strategy for determining Dirichlet priors to guide model fitting. If all you have is the performance data for an individual skill, it is not possible to choose which best-fitting set of parameters to use. However, what if you have performance data for multiple skills? Consider a hypothetical example where 14 skills in the domain are estimated to have a guess parameter in the range of 0.1 to 0.2. Imagine if for the 15th skill, there are two equally good models, one has a guess parameter of 0.15 and the other has a guess parameter of 0.5. One heuristic is to prefer models that have parameters that are more similar to the other skills in the domain and therefore select the set of parameter estimates that have a guess value of 0.15.

Based on the idea that it is better to have skills with similar model parameters, we use the following approach to estimate the Dirichlet priors:

1. Use the BNT-SM to estimate the model parameters for the skills in the domain
2. For all four parameters (guess, slip, K0, t)
   - Compute the mean (μ) and variance (σ^2) of the parameter estimates
   - Select α and β to generate a Dirichlet with the same mean and variance as the estimates Specifically, solve for α and β such that:
     - $\mu = \alpha/(\alpha+\beta)$ (mean of the Dirichlet distribution)
     - $\sigma^2 = \alpha\beta / ((\alpha+\beta)^2 (\alpha+\beta+1))$ (variance of the Dirichlet distribution)
3. We now have one Dirichlet distribution described by (α,β) for each of the four parameters
4. Reestimate the value of all four model parameters using their associated Dirichlet priors

This procedure provides a bias for all of the parameters to move towards the mean. If most parameter values are tightly clustered except for a few outliers, the variance will be low, and α and β will be higher. The result of a higher α and β is to provide more weight for moving estimates towards the mean. If the distribution is naturally spread out, α and β will be lower, and there will be little tendency towards correcting extreme parameter estimates. The assumption in step #2 is that a good set of priors for what the parameter estimate should look like is how the parameter is actually distributed as a result of the model fitting process. Consider a skill whose guess parameter, as computed by step #1, is very high. After constructing the Dirichlet distribution for guess, in step #4 there will be a bias against assigning such a high value for guess since it is unlikely. Thus the value will be adjusted downwards towards the mean. The amount of bias towards the mean is proportional to how large α and β are, which is inversely related to the population variance. That is, if the population has a high variance then there is a small bias. Conversely, if a parameter value is tightly clustered, there will be a strong bias towards the mean.

To determine whether our approach of providing Dirichlet priors is effective, we used the same geometry testbed described previously. We split our dataset into training and testing sets, ensuring that we split the data at the level of users (so no user was in both the training and testing set). Even though the no priors and Dirichlet priors models are of equal complexity, it is necessary to split the data into training and testing sets. The process of using Dirichlet priors biases the parameter fitting process away from modeling the training data. After step #1, the parameters are in a local (or perhaps) global maxima. The purpose of step #4 is to move the parameters away from that location to be closer to the average. We should not be surprised if this process causes us to fit the training data less well. Hopefully it should result in improved test set performance.

Using the same evaluation metric as before, AUC, we find that on the training set the no priors approach (step #1) achieves an AUC of 0.707 while the Dirichlet approach achieves 0.705, a slight but not statistically reliable decrease. On the testing data, the Dirichlet approach did slightly but not reliably better with an AUC of 0.692 vs. 0.687 for the no priors approach. Prior experiments using Dirichlet priors resulted in statistically reliable improvements [16]. It is unclear whether we failed to achieve that threshold because of the small data set or because our approach of generating Dirichlet priors from data was not as effective as using human knowledge.

One question is whether it makes sense to iterate the above procedure. That is, once you have estimated the parameters using Dirichlet priors in step #4, loop back to step #2 and take the mean and variance of that distribution, compute α and β, and reestimate. We don't have a theoretic or conceptual model of what would happen, but empirically, model fit statistics dropped on successive iterations. Thus, there seems no point in applying the procedure repeatedly.

3 Evaluating the parameters

In addition to using predictive accuracy of the model, it is also possible to evaluate the parameters of the model directly. However, the question of what constitutes good parameter values is a tricky one. For prior work with the Reading Tutor, we were able to relate the learning parameters to known facts about the domain [16]. For example, the K0 parameter, initial knowledge, should be correlated with a word's frequency in text. Unfortunately, such an approach is difficult to generalize across domains since students are not typically exposed to domain skills in everyday life. For example, how would one count how many times a student would be expected to have an opportunity to learn Newton's Second Law?

However, it is possible to discover trends by visually inspecting the parameter estimates. Table 2 shows the parameter estimates for five random skills in the geometry curriculum for each of the three optimization approaches. Curve fit refers to the classic conjugate gradient descent approach; BNT-SM is the result of fitting a knowledge tracing model with the Bayes Net Toolkit for Student Modeling, and Dirichlets is using the BNT-SM with the four-step procedure described in the previous section. One item is quickly apparent: the curve fitting approach seems to prefer extreme parameter values with it selecting 0.0 (the minimum) or 1.0 (the maximum) for seven parameter estimates. In comparison, the two BNT-SM approaches combined have only one parameter estimate outside the range of [0.05, 0.95]. If we believe that very few skills are known by no one (or everyone), or can be mastered after only 1 exposure, such extreme values should give us pause.

A complementary approach to inspecting the parameter values directly is to see what they imply for the number of trials required to master each skill in the domain. We define mastery the same way as was done for the mastery learning criterion in the LISP tutor [4]: students have mastered a skill if their estimated knowledge is greater than 0.95. We compute the expected number of trials to master a skill by applying the P(know) formula, used to generate the left graph in Figure 2, to each model's K0 and t parameters. We calculated how many practice opportunities would be required until the predicted value of P(know) exceeds 0.95. These values are shown in Table 3. In addition to comparing the three knowledge tracing models, we also compare the parameter estimates generated by LFA (Learning Factors Analysis [7, 17]) for this data set [18]. Focusing first on the knowledge tracing models, it is interesting to note that the BNT estimates tend to believe less practice is needed than the curve fit estimates. Similarly, using Dirichlet priors also reduces the predicted number of trials until mastery.

Which set of numbers to believe is secondary. That the numbers differ so markedly is key. The difference in predictive accuracy between the models is not that great, particularly between the two BNT models. However, the pedagogical implications are great. Some tutors cannot use pure mastery learning and instead advance the entire class at the same rate. Such an approach is needed in many cases to be accepted in actual classrooms where having students spread throughout the curriculum would be unacceptable. When designing the tutor, how much practice should designers assume students will need? Curve fit suggests 160 problems suffice, BNT-SM 136, and BNT-SM with Dirichlet priors says 92. There is nearly a factor of two difference between the high and low estimates. Keep in mind that in many ways this study is a best case scenario for getting similar estimates: the same researcher is conducting all of the analyses, the exact same set of data is being used to train the models, and the exact same statistical model is being trained. The only aspect that varies is the optimization software. Furthermore, relative to problems such as student emotional state, engagement, learning style, etc., the problem of modeling the student's knowledge of a skill is well studied and understood. The moral is that researchers should not be so quick to accept model fit as an arbiter between

competing models. There can still be large differences between models that are not easy to resolve quantitatively.

Table 2. Parameter values for 5 random geometry skills, and summary statistics for all 15 skills

Skill	model	K0	guess	slip	T
circle-area	curve fit	0	0.53	0.06	0.15
	BNT-SM	0.22	0.45	0.09	0.18
	Dirichlets	0.25	0.42	0.10	0.23
Circle-radius	curve fit	1	0.68	0.32	0
	BNT-SM	0.83	0.09	0.27	0.22
	Dirichlets	0.67	0.27	0.25	0.36
parallelogram-area	curve fit	0.56	0.99	0.07	1
	BNT-SM	0.90	0.93	0.06	0.49
	Dirichlets	0.81	0.71	0.06	0.69
pentagon-side	curve fit	0	0.44	0	0.1
	BNT-SM	0.12	0.41	0.03	0.11
	Dirichlets	0.15	0.39	0.06	0.14
trapezoid-height	curve fit	0	0.26	0.03	0.24
	BNT-SM	0.17	0.14	0.22	0.35
	Dirichlets	0.18	0.15	0.21	0.39
Mean (variance) for all 15 skills	curve fit	0.2 (0.11)	0.5 (0.09)	0.12 (0.01)	0.33 (.09)
	BNT-SM	0.41 (0.1)	0.43 (0.08)	0.14 (0.01)	0.31 (0.04)
	Dirichlets	0.4 (0.07)	0.41 (0.04)	0.15 (0.005)	0.45 (0.03)

Table 3. Number of predicted trials to master geometry skills

Skill	Curvefit	BNT-SM	BNT-SM+Dirichlets	LFA
circle-area	18	14	11	21
Circle-circumference	16	7	5	16
Circle-diameter	16	9	3	20
circle-radius	0	6	5	32
compose-by-add	21	17	2	n/a
compose-by-mult	6	5	5	13
parallelogram-area	1	2	2	n/a
parallelogram-side	11	1	2	4
pentagon-area	8	5	5	10
pentagon-side	28	25	20	21
trapezoid-area	9	11	8	13
trapezoid-base	4	8	7	15
trapezoid-height	10	7	6	15
triangle-area	11	13	8	61
triangle-side	1	7	3	8
Total practice to master curriculum	160	136	92	249+?

As an example of relaxing our best case scenario of the same data and statistical model, we also include results for LFA. LFA is a slightly different than knowledge tracing, and does not include performance parameters to account for randomness in student behavior. We used the LFA parameters for this data set generated by [18] and again computed the expected number of trials to master a skill. There were two skills that LFA did not believe students could master due to having negative learning rates[3], so we do not know how much practice it thinks students require for the entire curriculum, only that it is at least 249 practice opportunities. This amount of practice is nearly triple that required by the BNT-SM model that uses Dirichlet

[3] It is possible to constrain the LFA search to require non-negative learning rates to avoid this problem.

priors. To be clear, we are not asserting that this difference implies the Dirichlet model is "three times better" than LFA—for all we know it could be three times worse!

In the absence of a controlled field study, it is difficult to think of a way to resolve which model better models student growth. It may be instructive to look at skills where the various approaches diverge. For example, the skill circle-radius is believed to be known by students at the start in the curvefit model, require 5 or 6 practices in the BNT models, and 32 practices in the LFA model. Perhaps focused testing in areas of maximum divergence among the modeling approaches would be a way to resolve the issue?

One possible approach that is both data-driven and domain independent is to find general guidelines for the number of practice opportunities that are required to master a skill. For reading, a domain expert we collaborate with, Dr. Rollanda O'Connor, said that between 5 and 20 practice opportunities should suffice for learning a word. Although the exact numbers 5 and 20 might not be quite right for other domains, perhaps there are general guidelines that apply to many domains. It is likely that there are few skills that require only 1 or 2 practice opportunities to master, similarly numbers beyond 30 are also implausible.

4 Contributions, Future work, and Conclusions

This paper makes contributions in two areas: methodology for constructing student models and methods for evaluating competing models.

From a methodology standpoint, this paper replicates prior work with using Dirichlet priors [16] to bias student model parameter estimation. It extends our prior results by applying Dirichlet priors to a new domain, geometry. This domain is very different from reading both in the number of skills and in not relying on noisy speech recognition to score student performance. Furthermore, the approach presented here does not rely on beliefs about the true value of the parameters, and instead estimates the parameters to instantiate the Dirichlet distribution directly from the data.

This paper also extends methods for evaluating student models, going beyond the commonly used metric of predictive accuracy (e.g. [19, 20]). We have shown that looking at the model parameters themselves, as well as derived measures such as the expected number of trials to master a skill, are both useful supplements. We do not have a quantitative, domain-independent approach for evaluating these supplementary approaches, but have shown that models with comparable predictive accuracy can differ considerably on those measures. Furthermore, these measures are closer to what we care about, both as system designers and as educational data miners.

Although this paper provides new model construction and evaluation approaches, it raises more issues about how little we know. First, the model parameters estimated by EM appear more sensible than conjugate gradient descent. However, we do not know why. Is there some inductive bias between the two approaches that causes them to prefer different parts of the parameter space? Second, although Dirichlets appear a promising way of biasing the model fitting process, the problem of how to best select the α and β parameters is still unsolved. We have provided a data-driven domain-independent approach that seems plausible, but we have no theoretic reason to believe it is the best we can do. Our approach of using Dirichlets is somewhat restricted by only trying to model the domain. Another way of thinking of the priors is a way of biasing the parameter search. Any source of information that one wishes to use, statistics about the parameter distributions, personal beliefs about the domain, selecting a bias that better corresponds to external test scores, etc. are all permissible. In this paper, for purposes of generality, we restrict ourselves to objective, domain-independent sources of information.

Determining a method for evaluating model parameters directly is also a clear need. Are there recurring standard distributions of parameter estimates across different ITS and domains that we should expect to see? This question is a hard one to address, as typically researchers do not publish their model parameters. Perhaps with additional sharing of parameters, or at least summary statistics, some general trends can be derived.

An examination of the parameters underlying knowledge tracing suggests some room for improvement in how different types of interactions are handled. For example, the four model parameters are defined for each skill regardless of how they are presented. If, for a particular skill, some questions the tutor asks are multiple choice while others are free response, one would expect the slip and guess parameters would be different. However, standard knowledge tracing assumes a fixed slip and guess across question types. Similarly, if certain types of interaction are more conducive to learning than others, the probability of learning the skill (*T*

parameter) should vary. For example, within the domain of reading we have shown that additional practice opportunities for a word on the same day are much less effective than the first practice opportunity [21]. Acknowledging these differences would be a sensible extension to knowledge tracing. As an example of such an extension, we have used the BNT-SM to discover that the receipt of help influences the probability a student acquires a skill [22].

In conclusion, researchers interested in student modeling should consider using the BNT-SM, available at www.educationaldatamining.org under "Resources." It appears to produce parameter estimates with better predictive accuracy than known alternatives, and a manual inspection of the parameter estimates suggests they are more plausible. Furthermore, the BNT-SM is flexible across types of graphical models it can accommodate. Knowledge tracing is a simple example, but the toolkit is capable of handling more complex models (e.g. [22]). Finally, the toolkit is not restricted to just modeling student knowledge estimates; it is capable of representing other latent variables.

The other broad conclusion is that knowledge estimation is a basic building block of both ITS construction and EDM. For ITS, accurate estimates of the student knowledge can be used for mastery learning [4], to determine whether to provide declarative instruction, and to determine on which problem-solving step help should be provided [23]. For EDM, student modeling is useful to measure the effects of interventions at a fine grain size [22] and serve as a feature to predict affective states about the student such as gaming behavior [24]. Furthermore, estimating student knowledge can be thought of as a base case for inferring other latent traits about the student. Although far from solved, understanding how to map student actions to knowledge estimates is better understood than many tasks. So potential difficulties uncovered for student modeling should apply as strongly to other EDM applications.

Acknowledgements

We acknowledge the help of Hao Cen helping us get set up with the geometry tutor data set, and for graciously sharing his model parameter estimates with us. This work was supported by the National Science Foundation, ITR/IERI Grant No. REC-0326153. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Carr, B. and I. Goldstein, *Overlays: a Theory of Modeling for Computer-aided Instruction,*. 1977, MIT: Technical Report, AI Lab Memo 406.
2. Corbett, A. and J. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge.* User modeling and user-adapted interaction, 1995. **4**: p. 253-278.
3. Beck, J.E. and J. Sison, *Using knowledge tracing in a noisy environment to measure student reading proficiencies.* International Journal of Artificial Intelligence in Education (Special Issue "Best of ITS 2004"), 2006. **16**: p. 129-143.
4. Corbett, A.T. *Cognitive computer tutors: Solving the two-sigma problem.* in *International Conference on User Modeling*. 2001. p. 137-147.
5. Chang, K.-m., J. Beck, J. Mostow, and A. Corbett. *A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems*. in *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. 2006. p. 104-113 Jhongli, Taiwan.
6. Barnes, T., J. Stamper, and T. Madhyastha. *Comparative Analysis of Concept Derivation Using the Q-matrix Method and Facets*. in *Workshop at Educational Data Mining at AAAI2006*. 2006. p.
7. Cen, H., K. Koedinger, and B. Junker. *Automating Cognitive Model Improvement by A*Search and Logistic Regression*. in *Educational data mining workshop at National Conference on Artificial Intelligence*. 2005. p.
8. Winters, T., C. Shelton, T. Payne, and G. Mei. *Topic Extraction from Item-Level Grades*. in *Educational Data Mining workshop at AAAI2005*. 2005. p.
9. Reye, J., *Student Modelling based on Belief Networks.* International Journal of Artificial Intelligence in Education, 2004. **14**: p. 1-33.
10. Murphy, K., *Brief Introduction to Graphical Models and Bayesian Networks*. 1998: http://www.cs.ubc.ca/murphyk/Bayes/bnintro.html.

11. Koedinger, K.R. and J.R. Anderson, *Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design*, in *Computers as cognitive tools*, S.P. Lajoie and S.J. Derry, Editors. 1993, Erlbaum: Hillsdale, NJ. p. 15-45.
12. Anderson, J.R., A.T. Corbett, K.R. Koedinger, and R. Pelletier, *Cognitive tutors:Lessons learned.* The Journal of the Learning Sciences, 1995. **4**: p. 167-207.
13. Mostow, J. and G. Aist, *Evaluating tutors that listen: An overview of Project LISTEN*, in *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001, MIT/AAAI Press: Menlo Park, CA. p. 169-234.
14. Beck, J.E. and J. Sison, *Using knowledge tracing in a noisy environment to measure student reading proficiencies.* International Journal of Artificial Intelligence in Education, 2006. **16**: p. 129-143.
15. Heckerman, D., *A Tutorial on Learning With Bayesian Networks*. 1995, Microsoft Research Technical Report (MSR-TR-95-06).
16. Beck, J.E. and K.-m. Chang. *Identifiability: A Fundamental Problem of Student Modeling*. in *International Conference on User Modeling*. 2007. p. Corfu, Greece.
17. Cen, H., K. Koedinger, and B. Junker. *Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement.* in *Intelligent Tutoring Systems*. 2006. p. 164-175 Jhongli, Taiwan: Springer.
18. Cen, H., K. Koedinger, and B. Junker. *Is More Practice Necessary? - Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining.* in *International Conference on Artificial Intelligence in Education*. 2007. p.
19. Beck, J.E., P. Jia, J. Sison, and J. Mostow. *Predicting student help-request behavior in an intelligent tutor for reading.* in *Ninth International Conference on User Modeling*. 2003. p. 303-312 Johnstown, PA.
20. Corbett, A.T. and A. Bhatnagar. *Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model With Declarative Knowledge*. in *Sixth International Conference on User Modeling*. 1997. p.: Springer.
21. Beck, J. *Using learning decomposition to analyze student fluency development*. in *ITS2006 Educational Data Mining Workshop*. 2006. p. Jhongli, Taiwan.
22. Chang, K.-m., J.E. Beck, J. Mostow, and A. Corbett. *Does Help Help? A Bayes Net Approach to Modeling Tutor Interventions*. in *AAAI2006 Workshop on Educational Data Mining*. 2006. p. Boston, MA.
23. Gertner, A. and K. VanLehn. *Andes: A Coached Problem Solving Environment for Physics*. in *International Conference on Intelligent Tutoring Systems*. 2000. p. 133-142.
24. Baker, R.S., A.T. Corbett, K.R. Koedinger, and A.Z. Wagner. *Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System."* in *ACM CHI*. 2004. p. 383-390.