

Bypassing the intractable problem of student modeling

Joseph E. Beck and Kai-min Chang

joseph.beck@gmail.com
<http://www.andrew.cmu.edu/~jb8n>
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213. USA.

Abstract. In this paper we identify an ambiguity problem in inferring student models: observed student performance corresponds to an infinite family of possible knowledge tracing parameter estimates, all of which make identical predictions about student performance. However, these parameter estimates make different claims about the student's unobserved internal knowledge, some of which are clearly incorrect. We propose methods for evaluating these models to find ones that are more plausible. Specifically, we present an approach using Dirichlet priors to bias model search that results in a statistically reliable improvement in predictive accuracy (AUC of 0.620 ± 0.002 vs. 0.614 ± 0.002). Furthermore, the parameters associated with this model provide more plausible estimates of student learning rate, and better track with known properties of students' background knowledge. The main conclusion is that prior beliefs are necessary to bias the student modeling search, and even large quantities of performance data alone are insufficient to properly estimate the model.

1 Introduction and Motivation

John Self has written about the seemingly intractable difficulties in constructing a student model [1]. His concerns focused on the difficulties of having strong enough models of how students learn and represent problems, and strategies researchers can use to overcome and alleviate those difficulties. In this paper, we focus on a different, seemingly simple but (statistically) intractable, problem: given a set of student performance data and a domain model, infer the student's level of knowledge in each skill. Although knowledge tracing [2] claims to enable such inference, there are statistical difficulties that restrict how can we can interpret its claims. In this paper we identify this difficulty, then propose and validate a method for correcting it. First, we will provide a brief overview of knowledge tracing and identify a crucial shortcoming with the approach.

1.1 Description of Knowledge Tracing

The goal of knowledge tracing is to map student performance (observable) to an estimate of the student's knowledge (unobservable). For example, Figure 1 shows hypothetical student performance (on the left) and learning (on the right) curves. In both graphs, the x-axis represents the number of practice opportunities a student has with a particular skill. For the performance graph, the y-axis is the probability a student with that amount of practice will respond correctly. Since student performance is observable, this value can be directly estimated from the data. For the learning curve, the y-axis is student knowledge, which cannot be directly observed from the data. Instead, we rely on knowledge tracing to provide an estimate of the student's knowledge.

Knowledge tracing uses student performance data to estimate four parameters associated with each skill:

- $K0$: $P(\text{student knows the skill when he starts using the tutor})$
- T : $P(\text{student learns the skill as a result of a practice opportunity})$
- $Slip$: $P(\text{incorrect response} \mid \text{student knows the skill})$
- $Guess$: $P(\text{correct response} \mid \text{student doesn't know the skill})$

The first two parameters, $K0$ and T , are called the learning parameters of the model and represent the student's knowledge of the skill. The final two parameters, $slip$ and $guess$, are called the performance parameters in the model. They are the reason that student performance cannot be directly mapped to knowledge. Perhaps the student generated a correct response because he knew the skill, or perhaps he made a lucky guess? The $slip$ and $guess$ parameters account for several aspects in student performance, including lucky guesses, baseline performance for some testing formats (e.g. multiple choice tests), using partial knowledge to answer a question, using a weaker version of the correct rule to solve a problem, or even the inaccuracy from using a speech recognizer to score the student's performance [3]. All of these factors serve to blur the connection between student performance and actual knowledge.

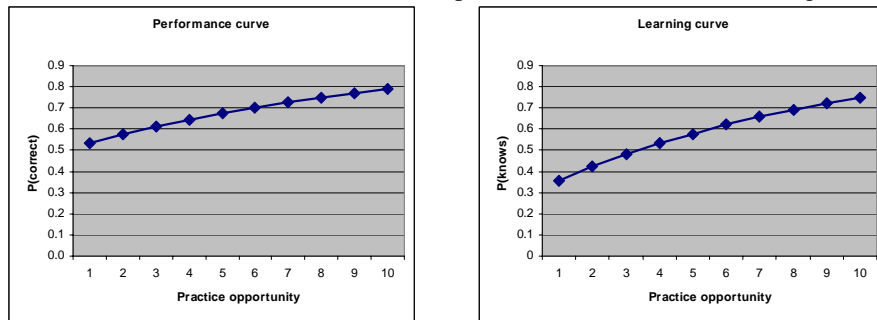


Figure 1. Performance and learning curves

1.2 Problems of Knowledge Tracing

Although the student performance curve can be obtained directly from the data, the learning curve must be inferred statistically. This inference would not be a problem if

there were a unique, best fitting model. Unfortunately, such is not the case. 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 knowledge tracing code¹ used by the Cognitive Tutors [4]. The third model is similar to those used by Project Listen’s Reading Tutor [5] for performing knowledge tracing on speech input [3]. This model’s *guess* parameter is very high because of inaccuracies in the speech recognition signal. As seen in Figure 2, the three models have identical student performance²—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.

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

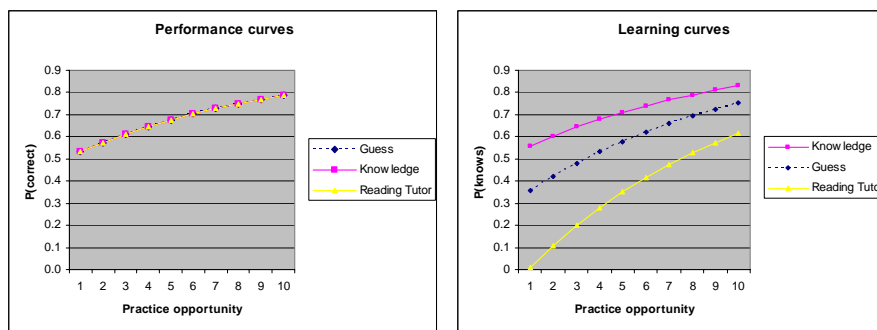


Figure 2. Comparison of three knowledge tracing models illustrating the ambiguity problem

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. In fact, the situation is considerably worse since there is an infinite family of curves that fit the data of which we are presenting three examples. Without loss of generality we will restrict the discussion to the three presented learning curves. One natural question is, given the ambiguity of the performance data in estimating the student’s knowledge, which of three curves is correct? Unfortunately, the situation is more bleak: the question of

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

which model is “correct” is not a meaningful one to ponder. All three of the sets of parameters instantiate a knowledge tracing model that fit the observed data equally well; statistically there is no justification for preferring one model over another.

In general, researchers are not hand generating models and selecting which one fits the data. Instead we use optimization software that finds parameters that best fit our data. The optimization software’s optimization method will drive which set of parameters it returns for a particular skill, and consequently the ensuing estimates of the student’s knowledge—a rather odd dependency. This problem is not one of getting stuck in a local (rather than global) maximum. Rather, the space has several global maxima (for example, the three presented knowledge tracing parameter sets) all of which make different assertions about student knowledge.

1.3 Motivation

One reasonable question is why we should care about the randomness in how knowledge tracing maps observations of student performance to student internal knowledge. After all, all of the models have the same degree of model fit and make identical predictions, so what does it matter which one we select? The difficulty is that we do not train models to fit the data well, we train models to use them in actual adaptive systems. For example, the *knowledge* model predicts students will need 24 practice opportunities to master (have a greater than 95% chance of knowing) a skill, while the *Reading Tutor* model predicts 32 practice opportunities are needed. Which model’s predictions should we believe? Another difficulty is tutorial decisions made on the basis of estimated student knowledge. For example, the Reading Tutor displays instruction if it believes the student’s knowledge is below a certain threshold.

Another rationale is efforts at conducting learning sciences research or educational data mining. Imagine if the desired end product is a graph of the efficacy of a particular treatment when the student’s knowledge is low, medium, or high. If some skills are modeled with parameters similar to *knowledge* while others are modeled with parameters similar to *guess*, the graph will not show the sharp contrast desired.

Generally, we have two desired outcomes for our student models: to accurately describe the student’s knowledge and to make predictions about his behavior. In the past, some researchers have used accuracy in predicting student performance as evidence that the model was an accurate model of the student [3]. Unfortunately, given Figure 2, such claims are unwarranted as very different estimates of knowledge can predict the same performance.

2. Approach

The goal of this work is to address the ambiguity problem by finding a means of biasing the model fitting process to help it find a “good” model. Our approach is first to instantiate knowledge tracing in a graphical modeling framework. The Bayes Net Toolkit for Student Models (BNT-SM) [6] provides a framework for replicating knowledge tracing with a Bayesian network [7]. The reason for using graphical

models is that they may provide a way out of our ambiguity dilemma. We have more knowledge about student learning than the data we use to train our models. As cognitive scientists, we have some notion of what learning “looks like.” For example, if a model suggest that a skill gets worse with practice, it is likely the problem is with the modeling approach, not that the students are actually getting less knowledgeable. The question is how can we encode these prior beliefs about learning?

2.1 Encoding Prior Beliefs as Dirichlets

Graphical models provide a means of encoding prior probabilities of the parameters in the model. A common method for representing such priors is to use the Dirichlet distribution [8]. To determine the Dirichlet priors for each parameter, we first examined histograms from our previous knowledge tracing experiments to see what values each parameter took on. We did not attempt to create Dirichlet distributions that perfectly mimicked those histograms or that minimized the sum-squared error. Our reasoning is that some of the parameter estimates were clearly nonsense. For example, approximately 10% of words had a T parameter that would result in students mastering the word after a single exposure, while 10% of the words would never be mastered. So we used the histograms as a starting point for the prior probabilities, but tempered them with our knowledge of student learning.

Figure 3 shows the prior probabilities we selected for our experiments. The x-axis is each possible value the four knowledge tracing parameters can take and the y-axis is the density of the distribution at that point. For example, the most likely value of the KO parameter is 0.6; it is only about half as likely that KO will take on a value of 0.45. The T parameter peaks at around 0.1, and has a long positive tail; most skills are learned relatively slowly but perhaps some are easier to acquire.

Our Dirichlets take as input two parameters that correspond to the number of positive and number of negative examples seen. The curve represents the likelihood of each possible probability value for $P(\text{positive})$. For example, the KO curve was generated with 9 positive and 6 negative examples, which can be thought of as 9 cases of knowing the skill initially and 6 cases of not knowing the skill initially. The mean of this distribution is $9/(6+9) = 0.6$. The odds that $P(\text{knows the skill initially})$ (i.e. the KO parameter) is 0.3 is quite low, as can be seen from the graph. If instead of 9 positive and 6 negative, we had instead created the distribution with 90 positive and 60 negative examples, the mean would still be 0.6. However, the distribution would have a much sharper peak at 0.6 and consequently a much lower variance. Thus, we control not only the mean of the distribution but also our confidence in how close most skills are to that value. For the T parameter we used 2 and 9 positive and negative examples, for the *guess* parameter 19 and 9, and for the *slip* parameter 1 and 15. The reason for the high guess rate is that we are using a speech recognizer to score student input, and it has a tendency to score incorrect reading as correct [3].

A plausible objection is there is no objective basis for preferring the numbers we used to generate these distributions and therefore this entire step should be omitted. However, skipping the step of creating a distribution is the same as asserting that the distribution is flat across the entire range of $[0,1]$ and all possible values of a parameter are equally likely. Such an assertion seems questionable at best.

2.2 Using Dirichlets to Initialize Conditional Probability Tables

Once we have constructed a set of distributions for KO , T , $slip$, and $guess$, we use their associated parameters to initialize the conditional probability tables (CPT) in the graphical network. The CPTs keep track of counts of different types of events. For example, the Dirichlet distributions for $slip$ and $guess$ would be used to instantiate the CPT shown in Table 2 that maps student knowledge to expected performance.

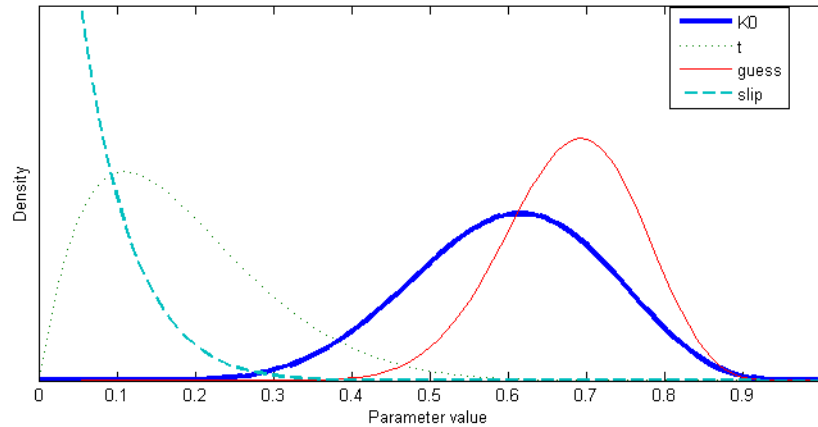


Figure 3. Prior probabilities for knowledge tracing parameters

Table 2. Using Dirichlet priors to initialize conditional probability tables

		Knowledge	
		Doesn't know	Knows
Correct	Incorrect	9	1
	Correct	19	15

The mental model of this process is the Dirichlets are used to seed the CPT, then as actual observations accumulate those are added to the values in the CPT. Thus, if a skill has little training data available it will be biased towards the average distribution. If there are many training data, then that skill's parameters can deviate substantially from the mean. Thus the initialization step does not force parameters to have a particular value, but it simply provides a bias.

3. Experimental Design and Results

The goal of our experiment is to validate whether our approach of using Dirichlet priors results in a better student model. We operationalize “better” as providing more believable estimates of the student's knowledge, without loss in predictive accuracy compared to a baseline model. Our approach is to use the BNT-SM to construct two knowledge tracing models. We constructed the first model using Dirichlet priors with the parameter values described previously. We did not examine the testing data when constructing the Dirichlet priors, and these results are for our first attempt at creating

such priors (so they have not been tuned, and represent a lower bound on improvement). We constructed the second, baseline, model by initializing each parameter to be the mean value that parameter had in a previous experiment (the same data we examined to generate the Dirichlets).

Our data came from 360 children who were mostly between six and eight years old and who used Project LISTEN's Reading Tutor in the 2002-2003 school year. Over the course of the school year, these students read approximately 1.95 million words (as heard by the automatic speech recognizer). On average, students used the tutor for 8.5 hours. For modeling purposes, this paper treats each of the 3532 distinct English words that occurred that year as a separate skill. We had three hypotheses:

1. The approach of using Dirichlet priors would result in a poorer model fit for the training data since the priors provide bias. We were not sure if the Dirichlet approach would better predict the testing data than the baseline.
2. The student learning curves for the models constructed using Dirichlet priors will look more believable.
3. The knowledge tracing parameter values obtained by the Dirichlet priors approach will better fit with our understanding of the domain.

To test our hypotheses, we randomly assigned students to either the train or to the test set with an approximately equal number of data in each set. To test the first hypothesis, we measured each approach's predictive accuracy using the Area Under Curve (AUC) metric for Receiver Operator Characteristic (ROC) curves. AUC is used for binary classification tasks when the true cost of each type of misclassification is unknown [9]. For the training set, both the Dirichlet and baseline had an AUC of 0.653 ± 0.002 . This result contradicts our hypothesis; apparently the Dirichlet priors did not interfere with fitting the training data. For the testing set, the Dirichlet approach had an AUC of 0.620 ± 0.002 while the baseline approach had an AUC of 0.614 ± 0.002 . Thus, Dirichlets resulted in a small but detectable improvement in model accuracy on unseen test data.

To determine which modeling approach resulted in more plausible learning curves, we randomly selected three words that were at the 25th ("bought"), 50th ("Rome"), and 75th ("twist") percentiles for amount of training data. We then plotted the learning curves (see Figure 4) for the models derived using the Dirichlet and baseline approaches. The learning curves associated with Dirichlet priors look reasonable, while the three curves from the baseline approach do not show evidence that students learn. The Dirichlet approach asserts that students would require 24, 16, and 15 exposures to master the words "bought," "Rome" and "twist," respectively. The baseline model believes those numbers should instead be 302, 5306, and 22313 exposures. Reasonable people can disagree over how many exposures are required for the average student to learn a word, but few people would assert that over 5000 exposures of a single word are necessary.

Perhaps these randomly selected three learning curves are not representative? These three words had an average T parameter of 0.14 in the Dirichlet approach vs. only 0.002 in the baseline approach. Across all 3532 words, both the Dirichlet and baseline approaches had an average T parameter of 0.11. However, 918 skills had a T parameter of 0.002 or lower in the baseline, compared to 0 such skills for the Dirichlet approach. Thus, roughly $\frac{1}{4}$ of words with the baseline approach were learned as slowly as those shown in the right graph compared to none with the Dirichlet.

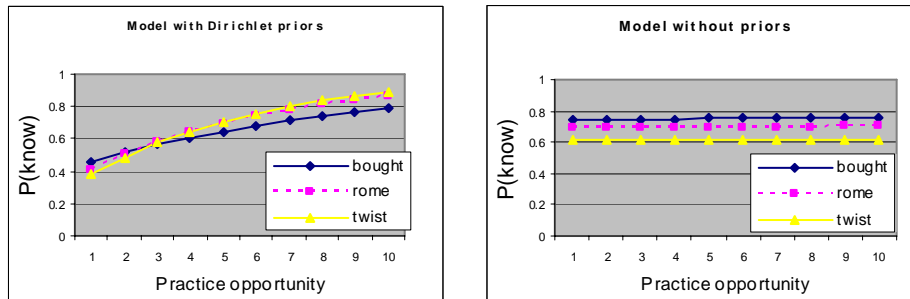


Figure 4. Comparing learning curves for Dirichlet vs. baseline approach

The final evaluation we performed was to examine the individual parameters that make up the knowledge tracing models and determine whether those estimated with Dirichlet priors are more reasonable. One problem is what do reasonable parameter values look like? Except in gross circumstances, such as Figure 4, it is hard to distinguish what makes one set of parameters better than another. However, it is possible to take advantage of the domain we are studying, reading, by using known properties of how students learn words. The KO parameter represents the knowledge students have when they first start using the tutor. In general, students gain competence with a word by being exposed to it. Unfortunately, we do not have a history of all of the words the student encountered before using the Reading Tutor. However, we do have word frequency tables of English text. These tables are not perfect at telling us which words a student has seen, since students will naturally read different material and thus see somewhat different sets of words, but the tables provide a starting point. Specifically, students should be more knowledgeable about words they encounter more frequently. Therefore, there should be a positive correlation between the percent of English text that is made up by a particular word and that word's KO parameter.

The percent of text made up by the most frequent word, “the,” is approximately 7.1%, while the least frequent word (of words occurring in both our training data and our word frequency table) was “shear” which only makes up 0.000085% of text. Since the most frequent word is over 80,000 times more frequent than the least frequent, we performed a log-transform on the percentages before attempting to find a relation. Figure 5 shows the plot of log-percent vs. the KO parameter for both the Dirichlet priors and the baseline approaches. A linear regression for each approach results in an R^2 of 1% for the baseline and 16% for using Dirichlets. Thus, using the Dirichlet approach produces the expected lawful relationship between prior knowledge and exposure to English text while parameters estimated with the baseline approach exhibit almost no such relation. Interestingly, the Dirichlet priors were not tuned based on word frequency, nor did the priors assigned to a word vary based on its frequency. Simply providing the underlying distribution for the various parameters enabled it to correctly assign parameters that track with word frequency.

4. Limitations and Future work

The largest limitation of this work is that it has only been evaluated on one (rather large) data set from Project Listen's Reading Tutor, so it is possible the results may not transfer to other tutoring domains. However, the problem of knowledge tracing sometimes returning parameter values that "just don't look right" is well known. In fact, this work was motivated by the difficulties a researcher (Hao Cen) had with applying knowledge tracing to some Cognitive Geometry Tutor data. Furthermore, the Cognitive Tutor knowledge tracing code's caps on the *slip* and *guess* parameters strongly suggest this problem occurred sufficiently often in the past that it was worth modifying the code. However, validating our approach on another tutor's data set is much needed and high on our priority list.

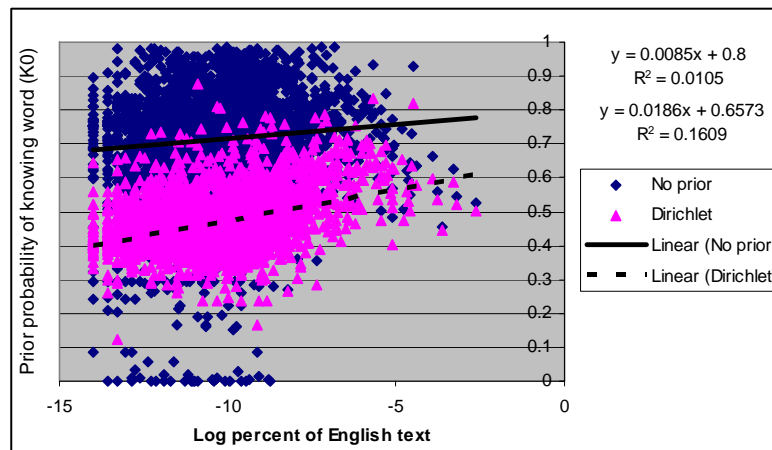


Figure 5. Word frequency vs. estimated prior knowledge

The second big open question is how the Dirichlets should be generated. One possible critique is that our method of finding parameters that looked reasonable isn't replicable across domains. While we acknowledge this argument may have merit, it is not obvious to us whether it is a serious flaw. Encoding the beliefs of those knowledgeable about the domain is something that must be done on a case by case basis. If researchers working on the Geometry Tutor wish to apply our approach of using Dirichlets, they would need to think about what is a reasonable prior distribution for their data. That said, it would be nice to have some guidelines about what reasonable means and variances tend to be, and the best way to use existing data to guide the selection of priors.

5. Contributions and Conclusions

This paper has made several basic contributions to student modeling. First, it identifies the ambiguity problem in knowledge tracing. Given a set of performance data, there is an infinite family of knowledge tracing models that can mimic student

performance. Unfortunately, those models all make rather different claims about the student's knowledge. Although capping the performance parameters alleviates this issue somewhat, it is still a problem. We have not proven it, but we suspect the ambiguity problem occurs not just in knowledge tracing but in any modeling approach that acknowledges student performance is a noisy reflection of his knowledge.

Second, we have shown that predictive accuracy is severely lacking as an evaluation metric for student models. We do not have a strong alternative, but have illustrated that two other techniques, examination of learning curves and inspection of model parameter estimates, can be used to evaluate models.

Third, this paper has proposed and validated a solution to the ambiguity problem. The use of Dirichlet priors is a graceful way of biasing the model search process to result in more sensible models that (at least in this case) are slightly more accurate.

The main conclusion of this paper is that, as system developers and learning science researchers, we must take the assertions of our student model about the student's knowledge with a large grain of salt. Furthermore, acquiring additional training data is not a solution to this problem. Even with an order of magnitude more data, there are still many sets of parameters that will fit the student data equally well. We need to encode prior beliefs in order to do a satisfactory job of modeling student knowledge; performance data are not enough.

References (see www.cs.cmu.edu/~listen for LISTEN publications)

1. Self, J. *Bypassing the intractable problem of student modelling*. in *Intelligent Tutoring Systems*. 1988. p. 18-24.
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*, 2006. **16**: p. 129-143.
4. 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.
5. 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.
6. 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. Jhongli, Taiwan.
7. Reye, J., *Student Modelling based on Belief Networks*. *International Journal of Artificial Intelligence in Education*, 2004. **14**: p. 1-33.
8. Heckerman, D., *A Tutorial on Learning With Bayesian Networks*. 1995, Microsoft Research Technical Report (MSR-TR-95-06).
9. Hand, D., H. Mannila, and P. Smyth, *Principles of Data Mining*. 2001, Cambridge, Massachusetts: MIT Press.