How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis

Yue Gong, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA
ygong@wpi.edu

Joseph E. Beck, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA
josephbeck@wpi.edu

Neil T. Heffernan, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA
nth@wpi.edu

Abstract. Student modeling is a fundamental concept applicable to a variety of intelligent tutoring systems (ITS). However, there is not a lot of practical guidance on how to construct and train such models. This paper compares two approaches for student modeling, Knowledge Tracing (KT) and Performance Factors Analysis (PFA), by evaluating their predictive accuracy on individual student practice opportunities. We explore the space of design decisions for each approach and find a set of “best practices” for each. In a head to head comparison, we find that PFA has considerably higher predictive accuracy than KT. In addition to being more accurate, we found that PFA’s parameter estimates were more plausible. Our best-performing model was a variant of PFA that ignored the tutor’s transfer model; that is, it assumed all skills influenced performance on all problems. One possible implication is that this result is a general one suggesting there is benefit from considering models that incorporate information from more than the typical handful of skills associated with a problem in the transfer model. Alternately, an explanation for this result is the transfer model that our tutor uses is particularly weak. We also found that both KT and PFA have relatively low predictive accuracy for cases where students generate incorrect responses, and 2/3 of the model’s errors are false positives, indicating a better means of determining when students will make mistakes is needed.

Keywords. Student modeling, knowledge tracing, performance factors analysis, expectation maximization, machine learning, model fitting approaches, data aging

INTRODUCTION

Student modeling is one of the major research issues for Intelligent Tutoring System (ITS) as it has been widely used for making inferences about the student’s latent attributes. Student modeling’s general approach is to take observations of a student’s performance (e.g., the correctness of the student’s response in a practice opportunity) or a student’s actions (e.g., the time he spent working on a question), and then use those to estimate the student’s underlying hidden attributes, such as knowledge, goals, preferences, and motivational state, etc. Those attributes cannot be observed directly, which is why student modeling techniques have always attracted a great deal of attention.
In ITS, student modeling has three common usages.

1. To track student knowledge. For example, the Cognitive Tutor, one of the most successful ITS (Koedinger, Anderson, Hadley & Mark, 1997), uses a student modeling technique, knowledge tracing (Corbett & Anderson, 1995), to track student knowledge in order to determine student mastery during problem practicing.

2. Predict student behaviours within the tutor, such as student performance on the next practice opportunity, or on external measures. For example, Feng, et al. employed item response theory (van der Linden et al., 1997) and ITS data to estimate student performance in an end-of-year high stakes state test (Feng, Heffernan & Koedinger, 2009). Beck, et al. constructed a student model capable of predicting whether the learner will request help from the tutor (Beck, J. E., Jia, P., Sison, J. & Mostow, J.).

3. Obtain plausible and explainable parameter estimates, in order to answer questions scientific about human learning. Plausibility concerns how believable the parameters are, which is often tested by comparing them to an external gold standard. Being explainable indicates the parameter estimates produced by the model have practical meaning, which enable the researcher to produce meaningful scientific output for the broader community. Beck, et al. investigated the utility of help provided in an ITS by interpreting parameters estimated from learning decomposition (Beck & Mostow, 2008) and an augmented knowledge tracing model (Beck, Chang, Mostow & Corbett, 2008). Other work (Arroyo & Woolf, 2005; Gong, Rai, Beck & Heffernan, 2009) inspected student motivational effects on student learning, by interpreting their model parameter estimates. Especially for student misuse of intelligent tutoring systems (Baker, Corbett, & Koedinger, 2004), the effects of ‘gaming the system’ and off-task behaviour have been examined by using many different student models and interpreting the model parameters (Baker & de Carvalho, 2008; Cocea, Hershkovitz, & Baker, 2009; Gong, Beck, Heffernan & Forbes-Summers, 2010).

Student models are usually evaluated by how well they predict students’ behaviors, as well as by parameter plausibility (Beck, 2007; Beck & Chang, 2007).

Given the importance of student modeling, considerable work has been done in order to improve student modeling techniques, pursuing higher predictive accuracy and (or) higher parameter plausibility. Baker, et al. introduced the concept of contextual guess and slip into knowledge tracing to improve the model accuracy (Baker, Corbett & Aleven, 2008). Pardos, et al. integrated individualization into knowledge tracing and showed a reliable improvement in prediction of real world data (Pardos & Heffernan, 2010). Pavlik et al. presented a new alternative student model, Performance Factor Analysis (PFA) and found that it is somewhat superior to knowledge tracing (Pavlik, Cen & Koedinger, 2009). Moreover, there is also a great deal of work focusing their efforts on examining and improving student model parameter plausibility (Beck & Chang, 2007; Gong, Beck & Heffernan, 2010; Pardos & Heffernan, 2010; Rai, Gong & Beck, 2009) as well.

As the field is making progress, however, we face a crucial problem: across various models and approaches, fair and reasonable evaluations are hard to achieve for a number of reasons. Different studies use different data sets and report different measurement metrics, which makes it difficult to compare the models. Even if they did avoid the above differences, it is still likely that they fit their models using slightly different procedures, even though the models they used are the same. Unfortunately, for some models, such as knowledge tracing, the method of fitting has a non-trivial impact on performance. In addition, many models have their own weaknesses or even problems. In this case, the ways of handling the problems could also have impact on model performance, yet researchers don’t necessarily deal with them in the same way. Therefore, it is of critical importance to perform comparative analyses of various approaches while taking those issues into account. In this study, we focus our efforts on the following two aspects. We
compare two competitive student modeling techniques: knowledge tracing (KT) vs. performance factor analysis (PFA). We perform ‘within comparisons’, where for KT, the comparisons concern the knowledge tracing models obtained from different model fitting approaches and with different methods of handling the model’s problem (which will be illustrated later). For PFA, we propose and compare some additional variants and determine which settings of PFA result in optimal performance. Finally, we also perform ‘between comparisons’ in which we give comprehensive comparisons between KT and PFA.

STUDENT MODELING TECHNIQUES

Knowledge tracing model

There are a variety of student modeling techniques. The knowledge tracing model (Corbett, Anderson, 1995), shown in Fig. 1, has been broadly used. It is based on a 2-state dynamic Bayesian network, where student performance is the observed variable and student knowledge is the latent. The model takes student performances and uses them to estimate the student’s level of knowledge. As part of training the model, two performance parameters are estimated: slip and guess, which mediate student knowledge and student performance. The guess parameter represents the fact that the student may sometimes generate a correct response in spite of not knowing the correct skill. The slip parameter acknowledges that even students who understand a skill can make an occasional careless mistake. There are also two learning parameters. The first is initial knowledge (K0), the likelihood the student knows the skill when he first uses the tutor. The second is the learning rate, the probability a student will acquire a skill as a result of an opportunity to practice it.

When knowledge tracing is used for prediction, the model uses knowledge tracing parameters, usually estimated from the training data (a large number of students), as well as the student’s actual performances. The model, using K0 at the very beginning, iteratively updates student knowledge according to the student performance on the previous practice opportunity. The pseudo code used to update the student knowledge in his i-th practice opportunity is derived from Bayes’s rule and shown in the following.

Based on the student’s estimated knowledge, the predicted performance at that particular practice opportunity can be calculated by first applying the knowledge tracing equation, shown in Equation 1 to compute
To obtain the probability of a correct response on the $i^{th}$ practice opportunity, $P_{\text{correct}_i}$, one uses Equation 1. In this way, the model is capable of predicting student performance step by step for every problem solved by the student.

$$P_{\text{correct}_i} = K_i(1 - \text{slip}) + (1 - K_i)\text{guess} \quad (1)$$

**Performance factor analysis**

A new alternative student modeling approach was presented by Pavlik Cen, and Koedinger (2009), Performance Factor Analysis (PFA). PFA is based on reconfiguring Learning Factor Analysis (LFA) (Cen, Koedinger & Junker, 2007). LFA's standard form is shown in Equation 2, where $m$ is a logit value representing the accumulated learning for student $i$ (ability captured by $\alpha$ parameter) using one or more knowledge components (KCs) $j$. The easiness of these KCs is captured by the $\beta$ parameters for each KC, and the benefit from prior practice for each KC is a function of, $n$, the number of prior practices for student $i$ with KC $j$, and of $\gamma$, the amount of learning on that KC for each practice. Equation 4 is the logistic function used to convert $m$ strength values to predictions of observed probability (Pavlik, Cen & Koedinger, 2009).

$$m(i, j \in \text{KCs}, n) = \alpha_i + \sum_{j \in \text{KCs}} (\beta_j + \gamma_jn_{i,j}) \quad (2)$$

$$p(m) = \frac{1}{1 + e^{-m}} \quad (3)$$

The formula for PFA is very similar as LFA, and also uses a logistic model for making predictions and has student performance as the dependent variable. As shown in Equation 4, PFA reconfigures LFA on its independent variables, by dropping the student variable ($\alpha$) and replacing the knowledge component variable with the question identity (i.e., one parameter per question). The model estimates a parameter for each item which represents the item’s difficulty. Thus $\beta$ parameters no longer capture the difficulty of the KCs (also called skills in this paper), but that of the items. The model also estimates two parameters ($\gamma$ and $\rho$) for each skill reflecting the effects of the prior successes and prior failures achieved on that skill. The conversion from the logit value to the prediction of student performance is done by following Equation 3.

$$m(i, j \in \text{KCs}, q \in \text{questions}, s, f) = \beta_q + \sum_{j \in \text{KCs}} (\gamma_jn_{i,j} + \rho_jf_{i,j}) \quad (4)$$

The PFA model can also be viewed as a learning decomposition model (Beck & Mostow, 2008) in that it estimates the different effects of getting a practice opportunity correct or incorrect.

**ISSUES WITH KNOWLEDGE TRACING**

**Model fitting approaches**

As pointed out in (Beck & Chang, 2007; Pardos & Heffernan, 2010; Rai, Gong & Beck, 2009), knowledge tracing suffers from two major problems with trying to estimate parameters: local maxima and multiple global maxima. The first problem one is common to many error surfaces and has known solutions such as multiple restarts. The second difficulty is known as identifiability and means that for the same model structure, given the same data, there are multiple (differing) sets of parameter values that fit the data equally.
well. Based on statistical methods, there is no way to differentiate which set of parameters is preferable to the others. Different model-fitting approaches have various criteria for fitting the data, and this produces different parameter estimates and further lead to different predictive abilities. Therefore, we explored the impact of model-fitting approaches on model accuracy and plausibility.

The Expectation Maximization (EM) algorithm (Moon, 1996) is a model-fitting approach for KT. EM finds a set of parameters that maximize the data likelihood (i.e., the probability of observing the student performance data). EM processes student performance as a piece of evidence with a time order, and uses this evidence for the expectation step where the expected likelihood is calculated. The model then computes the parameters which maximize that expected likelihood. The algorithm iteratively runs these two steps until it finds the final best fitting parameters. With EM, there is no guarantee of finding a global, rather than a local, maxima.

Recently, the brute force approach has been proposed for estimating parameters for KT (Baker, Corbett & Aleven, 2008). Contrary to EM, which maximizes the data likelihood, brute force attempts to minimize the sum of squared error (SSE). Originally, KT’s parameters are continuous, so that there is no way to compose a finite search space, which, however, is a must for an exhaustive search. We used the source code provided by Ryan Baker, which resolves the issue by only considering two decimal places of precision. In this way, the parameter space is reduced from infinity to $99^4$ possible parameter sets for each skill (i.e., there are four parameters for each skill and each of them has 99 possible values ranging from 0.01 to 0.99). Initially, every parameter starts from the value of 0.01 and is incremented by 0.01 on every iteration. Ultimately, for each skill, it finds the set of parameters resulting in the lowest SSE.

The major drawback is that the method suffers from high computational cost due to the large search space, so often the search space is reduced by setting search boundaries. In this study, we applied the brute force approach on the knowledge tracing model with two different settings. With the first one, no boundaries are designated and the search is free to explore the entire parameter space and try every combination of the four parameters. The second setting is to set search boundaries. Prior work by Pavlik, Cen, and Koedinger (2009) compared PFA and KT fitted with brute force. In our study, a comparison process with minimal differences is preferred. Therefore, in order to make a careful comparison, we followed the same protocol as the original work followed. Specifically, we used the same set of bounded ceiling values for the four parameters, so that the maximum probabilities of initial knowledge, guess, slip and learning are 0.85, 0.3, 0.1 and 0.3, respectively.

Conjugate Gradient Descent, an optimization method used to solve systems of equations, is used to estimate parameters in the CMU cognitive tutors. Chang et al. (Chang, Beck, Mostow & Corbett, 2006) found that EM produces models with higher predictive accuracy than Conjugate Gradient Descent.

Unlike the KT model, the family of logistic learning decomposition models is based on the form of standard logistic regression, so that the model-fitting procedure is ensured to reach global maximia; thus resulting in unique best fitting parameters. Consequently, for PFA, the model-fitting approach is not an issue.

Problem with multiple skill questions

The KT framework has a major weakness: when there is more than one skill involved in a question (called a multi-skill question), the model lacks the ability to make a prediction considering all of the skills simultaneously. As seen in the pseudo code for KT, it looks at observations on a skill. However, in some tutors, a question is usually designed to require multiple skills to achieve a correct answer. If a student model cannot accommodate this common phenomenon well, its ability of making plausible parameter estimates and accurate prediction is likely to be weakened.

A common solution is to associate the performance on the multi-skill question with all requested skills, by listing the student performance on this item multiple times, once for each required skill (e.g., Pardos,
Beck, Ruiz & Heffernan, 2008). Thus, when we train a KT model, a multiple skill question is split into multiple single skill questions. This strategy enables parameter estimation to proceed, but increases the probability of overfitting and also results in an accompanying problem: multiple predictions. Each duplicated performance is associated with a particular skill that has its own set of guess and slip parameters, which are used to calculate the student’s predicted performance (Equation 2). It is highly likely that those predicted values are not equivalent, which means for the same student, on the same practice opportunity, our models make different claims about how likely he is to produce a correct response. Given the conflicting predictions, some means of making a distinct prediction is needed.

In this study, we attempted two approaches to address the problem. The first is similar to an approach taken by Pardos, Beck, Ruiz, and Heffernan (2008) and is inspired by the joint probability in Probability Theory. The probability a student generates a correct answer in a multi-skill question is dependent on his ability to achieve correctness in all required skills. Therefore, we multiplied each skill’s predicted performance together and assign the product as the new predicted performance for all corresponding observations. Yet, the reasonableness of this method relies on an assumption: how likely a student can answer correctly for one skill must be independent of the probability that he responds correctly in another skill.

The second approach, takes the minimum probability of the predicted performances as the final value. The intuition behind this “weakest link” strategy is the likelihood of a student gets a correct answer is dominated by his knowledge of his weakest skill.

The above strategies, or some variant of them, are necessary options for KT due to its lack of ability to handle multi-skill performances. However, it is not the case for PFA, as it has the ability to handle multi-skill items. To perform a fair comparison and to account for possible effects on model accuracy by duplicating questions, we tested PFA by duplicating items identically to how we did for KT. In addition, we also examined PFA when it works in its natural spirit where we fit the model with the data that still keep the original multi-skill performances intact.

ISSUES WITH PFA

The PFA model estimates the parameters of item difficulty, and the effects of prior successes and prior failures for each skill. When the model counts the numbers of prior successful and failed practices, it doesn’t consider the order in which these practices occurred, which might be a potential problem. Consider the following example, a student is about to solve a question involving the Pythagorean Theorem. He has answered four prior questions involving the same skill, two of them correctly and the other two incorrectly. PFA is not concerned with the order in which the practices occurred. But consider the case of a student getting the first two correct and the last two incorrect, as compared to a student whose performance was reversed. We would expect the second student to perform better on the fifth item. Therefore, we attempt to address this concern with PFA by giving more weight to recent performances.

Our approach is to employ data aging to take into account the performance order. Based on the assumption that the further back the past practice was, the less it impacts the current question, we introduced a decay factor, \( \delta (0 < \delta < 1) \), that updates the counts by decreasing the importance of prior performances. The formulas are as follows:

\[
\text{success\_count}_t = \sum_{1 \leq k \leq t-1} P_k \ast \delta^{t-1-k} \tag{5}
\]

\[
\text{failure\_count}_t = \sum_{1 \leq k \leq t-1} |P_k - 1| \ast \delta^{t-1-k} \tag{6}
\]
These two formulas replace $s_{i,j}$ and $f_{i,j}$ in the original PFA formula, where in formulation, $s_{i,j}$ and $f_{i,j}$ are represented as:

$$s_{i,j} = \sum_{1 \leq k \leq t-1} P_k$$  

(7)

$$f_{i,j} = \sum_{1 \leq k \leq t-1} |P_k - 1|$$  

(8)

In other words, the original PFA model is identical to the one with data aging with $\delta = 1$. The same as in the original PFA, the new count function only considers questions involving the same skills. In Equations 5 and 6, $t$ indicates that the current question is the $t$th question that the student is about to solve. $P_k$ is the correctness in the $k$th practice opportunity. $\delta$ is the decay factor. As an example, suppose the student has completed four questions involving the Pythagorean Theorem and that the sequence of performances was: correct, correct, incorrect and incorrect (1,1,0,0). Further suppose the decay factor is 0.9. According to our formulas, the number of prior successes would be the sum of $1 \times 0.9^3 + 1 \times 0.9^2 + 0 \times 0.9^1 + 0 \times 0.9^0$, which equals to 1.5, while the number of prior failures would be the sum of $0 \times 0.9^3 + 0 \times 0.9^2 + 1 \times 0.9^1 + 1 \times 0.9^0$, which equals to 1.9. Contrariwise, a student who got the first two items right and the last two items wrong would have a correct count of 1.9 and an incorrect count of 1.5. In this way, the model is able to differentiate the performance orders by the two students.

By giving more weight to the more recent practices, those practices’ effects are considered more important than ones further in the past. This new approach has the benefit of enabling the exploration of various assumptions in terms of the decay impact of previous practices. Smaller $\delta$ implies that older practices rapidly become less important. On the contrary, when $\delta = 1$, we have the classic PFA. In this study, we chose $\delta = 0.9$ as we don’t want to eliminate the effects of further practices too quickly.

We use the formulation in Equations 5 and 6 for expositional clarity. However, it is possible to compute the aging values without maintaining the student history on the skill, and instead at each time step simply multiply the success and failure counts by $\delta$.

**METHODS**

For this study, we used data from ASSISTment (Razzaq et al., 2005), a web-based math tutoring system. The data are from 343 twelve- through fourteen-year old 8th grade students in an urban region in the Northeast United States. These data consisted of 193,259 problems completed in ASSISTment during Nov. 2008 to Feb. 2009. Performance records of each student were logged across time slices for 104 skills (e.g., area of polygons, Venn diagram, division, etc.).

Previous work compared KT and PFA models, and found PFA to be superior (Pavlik, Cen & Koedinger, 2009). In this study, we ran a replication study, but also focused on examining design decisions of each model. For KT, we investigated the impacts of using different model fitting approaches. We also attempted and tested different approaches to handling multiple-skill problems, described previously (multiplication and minimum). For PFA, we examined several PFA variants in order to optimize the model performance. We present our comparison results based on both predictive accuracy and parameter plausibility.

We used Bayesian Network Toolkit for Student Modeling (BNT-SM) (Chang, Beck, Mostow & Corbett, 2006) which uses the EM algorithm for the knowledge tracing model to estimate the model parameters. We also used Ryan Baker’s unpublished java code to accomplish the brute force model fitting procedure for KT. We also replicated the PFA model using the same model settings as in Pavlik, Cen, and Koedinger (2009), except where noted below.
We fit the three models with the pre-processed data in terms of converting multi-skill questions into multiple single skill questions. For PFA, we also examined it by fitting the original data which keeps multi-skill questions as a single unit. We refer to the PFA model handling multi-skill performances as PFA-Multi and the other dealing with multiple single questions as PFA-Single. We did 4-fold crossvalidation at the level of students, and tested our models on unseen students, which is different from what Pavlik et al. did (Pavlik, Cen & Koedinger, 2009). They conducted 7-fold crossvalidation and tested their models on seen students’ unseen performances. We prefer to hold out at the student level since that results in a more independent test set.

In the next section, we report the mean values of measurements across the four test folds. We also used Cohen’s $d$ to indicate the effect sizes and report them in parentheses in the tables, except where noted. To evaluate the models, we also perform statistical tests. All statistical tests we used, except where noted, are paired two-tailed $t$-tests using the results from the crossvalidation with degrees of freedom of $N-1$, where $N$ is the number of folds.

**RESULTS: PREDICTIVE ACCURACY**

Predictive accuracy is the measure of how well the instantiated model fits the test data. We used two metrics to examine the model predictive performance on the unseen data set: Efron’s $R^2$ and AUC (Area Under Curve) of ROC curve (Receiver Operating Characteristic curve). Efron’s $R^2$ is a measure of how well a model performs in terms of accurately predicting values for each test data point, where the baseline it compares to is predicting the data points by just guessing the mean of the sample. AUC of ROC curve evaluates the model’s performance on classifying the target variable which has two categories. In our case, it measures the model’s ability to differentiate students’ positive and negative responses.

One crucial point to keep in mind in examining these results is the $R^2$ values are for predicting individual student trials. That is, for each student response our models make a prediction and are scored on their accuracy. We compute error $= (predicted \ value - actual \ value)^2$, and compute Efron’s $R^2$ as:

$$R^2 = 1 - \frac{\sum (predicted_value - actual_value)^2}{\sum (mean - actual_value)^2}$$

Note that for predicting individual trials, $R^2$ values are typically fairly low (Heathcote, Brown & Mewhort, 2002; e.g., Beck & Mostow, 2008). If instead we predict aggregate performance and plot learning curves as are frequently seen in talks and papers, we have an $R^2$ of 0.88; so our skill model is reasonably accurate and our data register student learning. Our position is that the lower $R^2$ is not a sign that there is a problem with our models; rather it is what results when one closely examines the factors that influence student performance at the micro level: predicting individual student performances is much more difficult than predicting average performance for a set of performances.

**Knowledge tracing: Using different model fitting procedures**

In the first part of this section, we compare the predictive accuracy of the knowledge tracing models fitted by the brute force algorithms with two different settings. The first model, where the brute force was restricted by the boundaries, is labelled as BF-Restricted. The other model, which explored the entire search space, is labelled as BF-Full. We also reported the size of the search spaces.

The first two columns of Table 1 show the results of the comparisons for the two approaches on our metrics. The values for each metric are calculated by averaging the corresponding numbers obtained in the
Table 1
Crossvalidated predictive accuracy comparison between two KT models with brute force of two settings

<table>
<thead>
<tr>
<th></th>
<th>R²</th>
<th>AUC</th>
<th>Size of search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF-Restricted</td>
<td>0.036</td>
<td>0.656</td>
<td>$7.65 \times 10^5$</td>
</tr>
<tr>
<td>BF-Full</td>
<td>0.072 (2.49)</td>
<td>0.662 (0.45)</td>
<td>$9.6 \times 10^7$</td>
</tr>
</tbody>
</table>

4-fold crossvalidation. The numbers in the parentheses indicate the effect sizes of the comparisons using Cohen’s $d$.

$$Cohen's\ d = \frac{\text{mean}_1 - \text{mean}_2}{\sqrt{\frac{\text{var}_1}{n} + \frac{\text{var}_2}{n}}}$$ (8)

Since $R^2$ measures a model’s predictive ability by comparing to a naïve model, which guesses the mean as the prediction value for each individual data point, 0 indicates that the model has no prediction power once knowing the mean value of the target to be predicted. Note that with Efron’s $R^2$ it is possible to obtain a negative value, as that indicates the model’s predictions are worse than simply guessing the mean. For AUC, the baseline is 0.5, which would indicate random predictions and no relationship between the predicted value and the true value. Our results suggest that the size of the parameter space considered by the algorithm does matter. Based on a paired two-tailed $t$-test using the results from the crossvalidation, we found that BF-Full is able to outperform its counterpart and also reliably so in both metrics ($p < 0.05$). The effect size of $R^2$ is large, while even in AUC, the effect size is medium. However, one thing worth pointing out is that the improvement is less exciting if we take into account its cost. For BF-Full, the search space is 130+ times as big as used by the restricted brute force. Considering the baseline is already quite large (765,000 data points), BF-Full needs much more resources and computational time. For our dataset, over five days were required for each fold of the cross-validation for BF-Full.

Other model fitting procedures than brute force can be used to estimate parameters. Table 2 compares the models’ predictive accuracy when applying brute force and expectation maximization (see the rows in bold). EM outperforms BF-Restricted in both metrics ($p < 0.01$ in $R^2$; $p = 0.02$ in AUC), and has essentially identical accuracy as BF-Full. The effect size of EM against BF-Restricted (for a clear representation, we didn’t list it in the table) is relatively smaller than that of BF-Full: 2.44 for $R^2$ and 0.38 for AUC. With respect to computational time, although EM needs to run several iterations until convergence occurs, the time consumed is approximately the same as BF-Restricted.

**Knowledge tracing: Handling multi-skill questions**

Given the problem of multi-skill questions in KT, we compared the two proposed approaches for predicting performance (multiplication and min()) with the default models, which have no designed techniques to handle multi-skill questions. Those default models, after splitting multiple skill questions into multiple single skill questions, would make multiple, different predictions, on student performance on the question, and one for each skill.

We found the approach of calculating the product results in worse predictive accuracy for all model-fitting approaches. In a sense, this result means the predicted performances on each skill are not truly independent of each other. In contrast, taking the minimum value of the predicted performances provides more accurate models. As shown in Table 2, comparisons between mean values suggest that the min models
Table 2
Crossvalidated comparisons of the default models and the models solving multi-skill questions

<table>
<thead>
<tr>
<th></th>
<th>R²</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.036</td>
<td>0.656</td>
</tr>
<tr>
<td>BF-Restricted Multiplication</td>
<td>−0.144</td>
<td>0.656</td>
</tr>
<tr>
<td>Min</td>
<td>0.046 (0.72)</td>
<td>0.670 (1.2)</td>
</tr>
<tr>
<td>Default</td>
<td>0.072</td>
<td>0.662</td>
</tr>
<tr>
<td>BF-Full Multiplication</td>
<td>−0.191</td>
<td>0.634</td>
</tr>
<tr>
<td>Min</td>
<td>0.073 (0.15)</td>
<td>0.677 (1.36)</td>
</tr>
<tr>
<td>Default</td>
<td>0.072</td>
<td>0.661</td>
</tr>
<tr>
<td>EM Multiplication</td>
<td>−0.175</td>
<td>0.633</td>
</tr>
<tr>
<td>Min</td>
<td>0.073 (0.14)</td>
<td>0.676 (1.28)</td>
</tr>
</tbody>
</table>

are generally better than the default models. The AUC values are reliably different (p < 0.05) in every pair of the comparisons, whereas in R² we failed to find any reliable differences. Another consistent finding lies in the effect sizes (shown in parentheses). The model of taking the minimum value achieved a large effect size in AUC, but not so much in R². This indicates that this approach helps better discriminate correct and incorrect student responses, but does not markedly change the prediction errors, especially for BF-Full and EM, which are the two models that have explored the search space completely.

PFA: Bounding learning rates

One problem of PFA models is they could produce negative “learning” (really the change from answering a question successfully) rates due to over-fitting or due to sampling bias in the data. Therefore, in the original work on PFA (Pavlik, Cen & Koedinger, 2009), the researchers set 0 as the lower bound for the impact of getting a question correct. In order to understand whether and how much the potential for negative learning rates would impact the model’s performance, we examined the models where the negative learning rates were maintained, as well as the models with manual intervention where the learning rates are bounded as non-negative in the training procedures.

We tested the impact of allowing negative learning rates for all of our variants of the PFA algorithm (except the All Skill model, described below). We found that in all versions of PFA, the models produced a small number of skills with negative learning rates, which varied from 0 to 5 among the 104 skills. Moreover, the impact of the negative learning rates is also very small. For most of the 12 sets of parameters generated (three versions of PFA and four folds for each version), the R² values and AUC values are effectively unchanged by restricting learning rates to be non-negative. Only in two comparisons, R² values vary since the fourth decimal and one pair of AUC values don’t agree with each other from the third decimal. Take the comparison between the PFA model and the PFA model with bounded learning rate as example, the averaged R² values of the two models are 0.167512 and 0.167646; the AUC values of the models are 0.74475 regardless of whether learning rates are restricted to be non-negative. Most comparisons exhibit this behavior, thus for we do not report the comparative results in a table.

In our prior work (Gong, Beck & Heffernan, 2010), we used a computationally expedient approach, which may have been flawed and diverged slightly from the approach used in (Pavlik, Cen & Koedinger, 2009), of preventing learning rates from being negative. For those learning skills whose learning rates were
less than 0, we replaced the learning rate with a 0. This approach differs from that taken by Pavlik, Cen, and Koedinger (2009), as they forced the parameter to be non-negative, which potentially results in a slightly different model-fitting solution. Therefore in this study another goal was to investigate how much the results would vary when using the correct approach. The results from this study verified that the approach we used in our prior work (Gong, Beck & Heffernan, 2010) is an acceptable alternative (at least for our dataset where the automatically machine learnt models yielded few skills with negative learning rates.) as we found the differences in the $R^2$ values of the two approaches to be tiny, with differences starting in the fourth decimal.

**PFA: Handling multi-skill questions**

In contrast to the knowledge tracing model, the PFA model has the inherent ability to handle problems that require multiple skills. In Pavlik et al.’s work, when comparing KT and PFA, the researchers gave PFA the dataset organized in the same way as for KT by duplicating multi-skill questions (Pavlik, Cen & Koedinger, 2009). Therefore, the PFA model also treats a multiple-skill question as multiple single-skill questions. In our study, first of all, we wanted to replicate the previous work (Pavlik, Cen & Koedinger, 2009) as closely as possible, so we also trained and tested PFA (PFA-Single) by giving it the identical datasets as KT used. However, we were also curious to know how PFA performs in its natural way by handling multi-skill questions directly. To make this comparison, we also fitted and tested the PFA model (PFA-Multi) using the same datasets as we used for KT but without duplicating the multi-skill questions, i.e. we fitted all KT and PFA models to the same data, which however were organized in different ways.

As seen in Table 3, PFA-Multi results in stronger performances in both metrics, and the differences are statistically reliable with $p$ values < 0.01. The effect size, Cohen’s $d$, is large. This result implies that the ability of PFA to handle multiple skill questions directly indeed benefits its accuracy, and PFA should be used in its original spirit as it would result in better predictive performance.

**PFA: Addressing the issue of performance order**

One potential problem of the PFA model is that it ignores the order of practice opportunities, i.e. the model treats all observations equally important regardless of when they happened. We are interested in understanding whether there is a benefit from considering the order of the observations. We introduced a decay factor, $\delta$, to address this issue. A sequence of prior practices with its own correctness order is mapped to a unique count of prior successes and a unique count of prior failures (see Equations 5 and 6). With this new feature, the PFA model is able to predict the student’s next performance not only based on the number of previous responses, but also the order of the responses occurred.

Table 4 shows the comparison between the classic PFA and the decay version of PFA. In both metrics, the PFA with decay factor, $\delta=0.9$, produces reliably higher ($p < 0.001$) predictive accuracy than the classic

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Crossvalidated comparisons between the PFA models handling multi-skill questions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R^2$</td>
</tr>
<tr>
<td>PFA-Single</td>
<td>0.151</td>
</tr>
<tr>
<td>PFA-Multi</td>
<td>0.168 (2.27)</td>
</tr>
</tbody>
</table>
Table 4
Crossvalidated comparisons between the classic PFA and a variant PFA that handles the problem of performance order

<table>
<thead>
<tr>
<th></th>
<th>R²</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFA-Classic</td>
<td>0.168</td>
<td>0.745</td>
</tr>
<tr>
<td>PFA-Decay (δ = 0.9)</td>
<td>0.172(0.51)</td>
<td>0.748(0.45)</td>
</tr>
</tbody>
</table>

PFA. We noticed that although our proposed approach results in reliably better performance, the effect size is not large, suggesting that more work is needed here.

**PFA: Ignoring the transfer model**

The classic PFA model uses item difficulty and student success or failure on prior practices on skills required for this item to predict student performance. This approach ignores the potential impact of other skills and only considers as relevant those skills which have been tagged by our subject-matter expert being related to the item. This assumption is reasonable and easily understandable, as only the student’s performances on those relevant skills would contribute the question solving. In this study, however, we also presented a new variant of PFA which takes an opposing view to that assumption: rather than just considering the skills believed to be involved in the question, the model is trained and tested by considering all the skills in the dataset. The intuition is that there might be relationships that are not well captured by the transfer model. For example, perhaps overall student competency is an important predictor? Or perhaps items involve a broader range of skills than our subject matter expert believed. This model assumes that the probability a student successfully solves a problem might also depend on his proficiencies on other skills. However, there are no easy ways to identify which other skills are important to a given skill, therefore in this study, we used all skills.

We compared the proposed PFA variant with the classic PFA. As seen in Table 5, the PFA-All skills model seems to beat the classic PFA in both metrics. However, we failed to find any reliable differences between these two models, even though the mean values appear a trend suggesting PFA-All skills is probably better than the classic PFA and the Cohen’d values suggest that the effect size is large.

One reason for the lack of reliable difference is the relatively low statistical power of the t-tests since we only have four independent observations (one for each fold of the cross validation). Therefore, we designed a special t-test for this comparison. We conducted a paired two tailed t-test between the R² values of the data points of each student. In each fold of the cross validation, we calculated R² treating students as units. Specifically, a student’s performances were grouped together and treated as a subset of the test data. For each subset, we computed the R² value. As a result, we had N R² values, where N is the number of students that were in the test dataset. These N R² values were the independent observations in the t-test. We did

Table 5
Crossvalidated comparisons between the classic PFA and all-skill PFA

<table>
<thead>
<tr>
<th></th>
<th>R²</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFA-Classic</td>
<td>0.168</td>
<td>0.745</td>
</tr>
<tr>
<td>PFA-All skills</td>
<td>0.181 (0.89)</td>
<td>0.756 (1.04)</td>
</tr>
</tbody>
</table>
four $t$-tests in total, and one for each fold. In all four comparisons, the mean $R^2$ values of PFA-All skills are superior to PFA-Classic, and the $p$ values from the four $t$-tests vary from 0.005 to $3.81 \times 10^{-32}$. This statistical test indicates that PFA-All skills model has better performance in terms of predictive accuracy. Interpreting this result is complicated, and more work, using transfer models developed by subject matter experts and for other domains, will help considerably in better understanding it. At present we view the PFA-All skills approach as an intriguing option.

Overall comparison: Predictive accuracy across KT and PFA

In this section, we compare four main models. The knowledge tracing model fitted by brute force with restricted search area and the PFA model taking multiple single questions have been compared by other researchers (Pavlik, Cen & Koedinger, 2009). Although the above two models are not the best, we still include them, so as to replicate the comparisons using the same model settings as used by Pavlik, Cen, and Koedinger (2009) and to understand the models’ performance across different studies.

Other than these two models, we also examined the knowledge tracing model fitted by the expectation maximization algorithm, as we want to conduct a direct comparison between the PFA model and the KT model with the alternative model fitting procedure. In addition, we compared the above three models with the best variant of PFA where it considers all skills. In our results, we also reported the number of parameters produced by each model. In this section, we didn’t list effect sizes in the table as most of them have been presented before.

Divergent with the finding of Pavlik et al. (Pavlik, Cen & Koedinger, 2009), that the PFA model is somewhat superior to the KT model, the PFA model, at least on our dataset, outperforms the knowledge tracing model by a great amount regardless of model fitting procedures used for KT. Since we fit both the PFA and KT models to the same dataset where multiple skill questions were split into multiple single skill questions, a fair comparison is possible as such dataset favours neither model. We have already shown that PFA-Single is the weakest among all PFA variants, so the comparison in Table 6 provide a lower bound for the PFA vs. KT difference. In the last row of Table 6, we see that compared to KT+EM, the $R^2$ value is improved 150% by the PFA-All skills model. The effect sizes in both metrics are also large: 6.4 for $R^2$ and 7.07 for AUC.

We noticed that PFA produced more parameters than KT, which seems inconsistent with the results reported in (Pavlik, Cen & Koedinger, 2009). But, given that the number of parameters in PFA = 2*# of skills + # of items, while the number of parameters in KT is 4*# of skills. Consequently having fewer items favors PFA, while fewer skills benefits KT with respect to yielding fewer parameters.

Although we have reported AUC values for each model, which are used to evaluate the models’ classification performances, it is important for researchers to understand which models are better and under which conditions. We used confusion matrices to visually understand the models’ misclassifications. We

Table 6

<table>
<thead>
<tr>
<th></th>
<th>$R^2$</th>
<th>AUC</th>
<th># of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT + EM</td>
<td>0.072</td>
<td>0.661</td>
<td>416</td>
</tr>
<tr>
<td>KT + BF-Restricted</td>
<td>0.036</td>
<td>0.656</td>
<td>416</td>
</tr>
<tr>
<td>PFA-Single</td>
<td>0.151</td>
<td>0.732</td>
<td>1013</td>
</tr>
<tr>
<td>PFA-All skills</td>
<td>0.181</td>
<td>0.756</td>
<td>1013</td>
</tr>
</tbody>
</table>
Table 7 provides a deeper examination about the performances of the best-performing models for KT and PFA. For KT, we used both EM and the BF-Full. For PFA, we used the decay model to account for performance ordering and the PFA variant that ignored the transfer model, since those versions performed the best.

In Table 7, the four cells in each model’s sub-table represent the percentages of instances that fall into each category. For example, 26% of the time, the KT+EM model predicted a correct response, but the student actually responded incorrectly. In the previous sections, we showed that the family of the PFA models outperforms the KT models by a substantial margin. The confusion matrices expose where the power comes from. When we compare the two KT models and the two PFA models, we see the big difference lies in the rate of false positives, which corresponds to the top-right corner in each sub-table. This cell indicates that the model claims the students would produce a correct response, which is an incorrect prediction as the students got those items wrong. We found the PFA models perform much better in this scenario, although with substantial room for improvement. Of the errors, about 2/3 of PFA’s errors are false positives, with only 1/3 of the errors being false negatives. Most of the models followed this general trend of performing better when the students’ responses are correct. Among all correctly answered questions, 86-87% instances are correctly predicted by the four models, while for incorrect responses the numbers are 30% for KT and about 45% for PFA. This finding suggests that even given the best model we have so far, the classification accuracy for incorrect student responses is less than 50% which seems disappointing, but the upside is that the numbers guide our research efforts towards reducing false positives in the predicting student performance.

RESULTS: PARAMETER PLAUSIBILITY

Predictive accuracy is a desirable property, but ITS and educational data mining researchers are also interested in interpreting models to make scientific claims. Therefore, we prefer models with more plausible parameters when we want to use those for scientific study. However, deciding that one set of model parameters is more plausible is non-trivial due to the lack of gold standards to quantify parameter plausibility. In our study, we used the two metrics we explored in Rai, Gong, and Beck (2009).

**Broken learning rates**

For the first metric, we inspected the number of practice opportunities required to master each skill in the domain. We assume that skills in the curriculum are designed to neither be so easy to be (on average)...
mastered in three or fewer opportunities nor too hard as to take more than 50 opportunities. We computed the number of opportunities needed to master a skill by treating student performance as unobserved, and used the learning rate to update the probability of knowledge. We define mastery as the same way as was done for the mastery learning criterion in the LISP tutor (Corbett, 2001): students have mastered a skill if the estimated probability of knowing it is greater than 0.95. Based on students’ prior knowledge and learning parameters, we calculated the number of practice opportunities required until the predicted knowledge exceeds 0.95. Then, we compared the number of skills with implausible values (fewer than 3 or more than 50).

Table 8 shows the average numbers of skills calculated across 4-fold crossvalidation and the corresponding standard deviations in the parentheses. The results are consistent in the two conditions. BF-Restricted results in 0 skills with an overly fast mastery rate and the fewest skills mastered too slowly. It is reasonable, as one of the reasons of assigning boundaries to this model fitting approach for KT is to prevent the model from producing unbelievable parameters. We also see the support from the opposite side where when brute force is released from the bounded search area, the model, BF-Full, generated the most extreme cases in both conditions. EM is in the middle with respect to the number of implausible parameters. In contrast to BF-Restricted, EM is allowed to explore the entire search surface and no manually assigned boundaries are needed.

Student parameter accuracy

The second metric is using external measurement to evaluate parameter plausibility.

The students in our study had taken a 33-item algebra pre-test before using ASSISTments. Taking the pre-test as external measure of incoming knowledge, we calculated the correlation between the students’ initial knowledge estimated by the models and their pretest scores.

In other to acquire student’s initial knowledge parameter, we used KT to model the students instead of skills (see Rai, Gong & Beck, 2009 for details). Since PFA has no student parameter by default, we tweaked

Table 9

<table>
<thead>
<tr>
<th></th>
<th>KT+BF-Restricted</th>
<th>KT+EM</th>
<th>PFA-Multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation</td>
<td>0.865</td>
<td>0.827</td>
<td>0.895</td>
</tr>
</tbody>
</table>
it to include student as an additional independent variable. In Table 9, we see that the PFA model using multi-skill performances produces the strongest correlation (i.e., the most plausible estimate of student initial). KT+BF-Restricted shows a higher ability to estimate plausible parameters than KT+EM. KT+EM is reliably different ($P<0.05$) than PFA-Multi; none of the other differences is reliable.

**CONTRIBUTIONS, FUTURE WORK AND CONCLUSIONS**

This paper extends the literature on evaluating student modeling techniques and comparing techniques against each other. As data sets become larger and we are capable of evaluating models more thoroughly, it is important to extend our understanding of the techniques we are using in order to find best practices for the field.

Specifically, we found that PFA substantially outperforms KT on a variety of metrics. This result is not in agreement with those obtained by Pavlik et al., who found that PFA is somewhat superior to the KT model fitted by brute force (Pavlik, Cen & Koedinger, 2009). We are uncertain for the reason for this divergence in results yet, but suspect a difference in the datasets, a difference in the cross-validation procedure (this work holds out all of the data for a student, the prior work by Pavlik et al. holds out some data from each student), or that we modelled difficulty as being associated with items rather than with skills. We also showed that the model parameters estimated by PFA were more plausible than those estimated by KT.

Within KT, we explored several aspects of the design space. We examined a brute force search as an approach for estimating model parameters, and reported results for a complete search of the space, as well as for a restricted portion of the space as has been done in previous work (Gong, Beck & Heffernan, 2010). We found that restricting the parameter space BF considers results in a moderate decrease in predictive accuracy, but completes much more quickly and results in more plausible parameters. Thus, restricting BF to only consider parameter values considered to be likely is a tradeoff. When compared to EM, we showed that BF is roughly equivalent to somewhat worse. BF-Full was about as accurate as EM, but at a cost of lower parameter plausibility and a much longer time to estimate the parameters. BF-Restricted was less accurate than EM, but had comparable parameter estimation time and more plausible parameter estimates.

KT also has an issue of how to model problems that contain multiple skills. For predicting student performance, we found that a simple heuristic of using the skill with the minimum student proficiency outperformed both multiplying the skills together (i.e., an independence assumption), and making simultaneous predictions with each of the skills.

Similarly, we explored the space of design decisions to come up with some “best practices” for using PFA. We first explored the possible deficiency of PFA with regard to negative learning rates. We found, at least for our data, this problem is relatively uncommon, and correcting it by setting negative learning rates to 0 or by using a procedure which forces the value to be non-negative, results in a negligible difference. In introducing PFA (Pavlik, Cen & Koedinger, 2009), the approach used for multi-skill problems was to use a similar approach as knowledge tracing by splitting the training data apart into multiple copies of the item (one copy for each skill). This approach makes sense in the context of facilitating a proper comparison with KT since the train and test sets should be equal across techniques. However, we found that this methodology costs PFA some accuracy as it is naturally able to model problems with multiple skills. Researchers considering using PFA should not simply mimic the data preparation required for KT, and should take advantage of this aspect of PFA.

We also extended PFA by considering new aspects. PFA ignores the order of a student’s performance, which is unfortunate as the order provides a clue about student knowledge. We found that aging old observations resulted in a marginal improvement in performance. We also explored a version of PFA that ignored the transfer model for the problem and instead relied on all 104 skills to predict student
performance on each question. Surprisingly, this version of PFA had the highest accuracy. Researchers using PFA should consider experimenting with both data aging and bypassing their tutor’s transfer model and predicting directly using all of the skills to see if those techniques improve accuracy on their data sets.

This work leaves several interesting questions unaddressed. The two major ones are possibly related:

1. Why are there divergent results for PFA vs. KT when compared to an analysis of Cognitive Tutor data?
2. Does ignoring the transfer model really result in more accurate predictions?

For the first question, one possible reason is the ASSISTments system has a poorer transfer model than the Cognitive Tutors. Another possibility is that the difference is due to some difference between ASSISTments and the Cognitive Tutors with respect to the domain, types of problems presented, or the students using the tutor. A fruitful area of future work is to rerun these experiments across multiple datasets—ideally from different research groups as tutors created by a particular group are likely to share similar properties and have similar inherent biases (Ioannidis, 2005).

For the second question, the reason the version of PFA that ignored the transfer model performed the best could also be explained by having a weak transfer model. Another possibility is that there is considerable useful information contained in performance on “irrelevant” skills; in fact it was this intuition that prompted the first author to conduct the experiment. Interestingly, an attempt to replicate this modelling approach using the 2010 KDD Cup dataset from the Algebra Cognitive Tutor found that ignoring the transfer model resulted in markedly inferior performance relative to standard PFA and was about as accurate as a model that simply predicted the mean (Gong & Beck, 2011). Understanding the reason for these divergent results, and under what circumstances ignoring the transfer model is justified, is a useful avenue of future work.

Finally, the approach of just using the transfer model or just using all of the skills equally can be thought of as endpoints in a continuum. Perhaps a hybrid model which separated out the impact of the skills (believed to be) exercised by this problem and the impact of all of the other skills would be even more accurate?

A corollary to the above is better understanding the sensitivity of KT and PFA to various transfer models. Presumably, PFA will be more robust since it directly estimates an item parameter, providing solid baseline performance independent of the accuracy of the transfer model. It would be interesting to see how the techniques compare when evaluated with an automatically derived transfer model.

One concern the authors have is that the field will move to adopt PFA, as there is a clear reason to use KT in some circumstances: it provides a model of learning. PFA includes a parameter called learning, but it only reflects getting an item correct (and the technique is quite properly named Performance Factors Analysis). Getting an item wrong is represented by a separate parameter, which is usually negative. Clearly this value does not represent “anti-learning,” but instead reflects the effect of the evidence of getting an item incorrect. The clearer semantics of KT are an advantage for certain educational data mining activities, such as investigating which items result in the most learning. Finding a method of improving the semantics of models such as PFA, so we could have the advantages of model accuracy and model interpretability, would be a major step forward.

The marginal results for data aging point to another possible line of inquiry: considering time in the model. At present, our aging scheme simply ages data at each practice opportunity of a skill. Such an approach captures ordering effects, but neglects that a student’s knowledge changes when not using the tutor. If three months have elapsed between practicing a certain skill, the student might have forgotten, or learned, something about that skill in the meantime. Thus, estimating the effect of time directly in the model would nicely augment the aging approach.

Finally, one commonality across all of the models tested is they are much more accurate in making predictions when students respond to items correctly than when students respond incorrectly. Understanding
the reason for this phenomenon, and developing models that do a better job at handling these cases, is a major open challenge for the student modeling community.

In conclusion, this paper compared PFA and KT as approaches for student modeling, and found that, for our data, PFA is a stronger approach for accuracy both of predictions and for parameter plausibility. Since accuracy at predicting student performance is a good test of the accuracy of a student modeling system, system developers should consider using PFA. Educational data mining researchers should consider what it is they wish to do with the model before deciding: The use of the model, rather than the task being modeled, should drive the decision making. Also note that this issue of clear model semantics is related to, but quite distinct from parameter plausibility—an area where PFA did quite well.

Overall, we found that many decisions were not crucial. For KT users, EM vs. BF is a tossup; most researchers will probably do fine using whatever software is more easily available. Similarly, PFA’s negative learning rates seem to be a minor issue and can be ignored or patched in a variety of ways. Handling of items with multiple skills was a common issue for both PFA and KT, and there are clear recommendations for both techniques there. However, for accuracy in modeling, PFA seems to be the stronger choice.

ACKNOWLEDGMENTS

This research was made possible by the US Dept. of Education, Institute of Education Science, “Effective Mathematics Education Research” program grant #R305A070440, NSF CAREER award to Neil Heffernan, the Spencer Foundation, and a Weidenmeyer Fellowship from WPI.

REFERENCES


