Automatically Detecting a Student's Preparation for Future Learning: Help Use is Key

RYAN S.J.D. BAKER, SUJITH M. GOWDA
Worcester Polytechnic Institute
AND
ALBERT T. CORBETT
Carnegie Mellon University

We present an automated detector that can predict a student's later performance on a paper test of preparation for future learning, a post-test involving learning new material to solve problems involving skills that are related but different than the skills studied in the tutoring system. This automated detector operates on features of student learning and behavior within a Cognitive Tutor for College Genetics. We show that this detector predicts preparation for future learning better than Bayesian Knowledge Tracing, a widely-used measure of student learning in Cognitive Tutors. We also find that this detector only needs limited amounts of student data (the first 20% of a student's data from a tutor lesson) in order to achieve a substantial proportion of its asymptotic predictive power.

Categories and Subject Descriptors: I 2.7 **[Artificial Intelligence**]

General Terms: Preparation for Future Learning, Cognitive Tutor, Educational Data Mining

Additional Key Words and Phrases:

1. INTRODUCTION

Over the previous two decades, knowledge engineering and educational data mining (EDM) methods (cf. Baker & Yacef, 2009; Romero et al., 2010) have led to increasingly precise models of students' knowledge as they use intelligent tutoring systems and other types of interactive learning environments. Modeling student knowledge has been a key theme in research in intelligent tutoring systems from its earliest days. Models of student knowledge have become successful at inferring the probability that a student knows a specific skill at a specific time, from the student's pattern of correct responses and non-correct responses (e.g. errors and hint requests) up until that time (cf. Corbett & Anderson, 1995; Martin & VanLehn, 1995; Pavlik et al., 2009). In recent years, the debate about how to best model student knowledge has continued, with attempts to explicitly compare the success of different models at predicting students' future correctness within the tutoring software (cf. Pavlik et al., 2009; Gong et al., 2010; Baker, Pardos, et al., in press).

However, the ultimate goal of tutoring systems is not to improve future performance within the system itself but to improve unassisted performance outside the system. Ideally, interactive learning environments should promote "robust" learning (Koedinger et al., under review) that is retained (better remembered) over time (Pavlik & Anderson, 2008), transfers to new situations (Singley & Anderson, 1989), and prepares students for future learning (termed "PFL") (Bransford & Schwartz, 1999). The difference between transfer and PFL is whether a student has the ability to use their existing knowledge in

Authors' addresses: Ryan S.J.d. Baker, Sujith M. Gowda, Department of Social Science and Policy Studies, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA USA 01609. E-mail: rsbaker@wpi.edu, sujithmg@wpi.edu; Albert T. Corbett, Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh PA USA 15213; E-mail : corbett@cmu.edu

new situations or new fashions (transfer), or whether a student acquires new knowledge more quickly or effectively, using their existing knowledge (PFL).

Historically, student modeling research has paid limited attention to modeling the robustness of student learning. To the extent that there has been attention to modeling the robustness of learning, it has focused on retention and transfer. For example, Pavlik and Anderson (2008) predict how long knowledge will be retained after learning within an ILE teaching foreign language vocabulary. Martin and VanLehn (1995) and Desmarais et al. (2006) predict whether student knowledge of one skill will transfer to another skill. Baker, Gowda, and Corbett (in press) predict student performance on a paper post-test of transfer. However, it can be argued that the most important form of robust learning is the ability to apply learned skills and concepts to support future learning outside of the context where those skills and concepts were learned (Bransford & Schwartz, 1999). Though studies have demonstrated that learning from some types of interactive learning environments can prepare students for future learning (Tan & Biswas, 2006; Chin et al., 2010) student models have not yet explicitly modeled PFL.

Within this paper, we present a model designed to predict student performance on a paper post-test of PFL, a post-test where the student reads instructional text to learn new problem-solving skills related but different to those in the tutor, and then applies those skills in problem-solving. Such a model could be used both to understand the conditions of robust learning, and to drive interventions designed to increase the robustness of learning for students who are learning the skills in the tutor, but in a shallow fashion.

This model is generated using a combination of feature engineering and linear regression, and is cross-validated at the student level (e.g. trained on one group of students and tested on other students). We compare this model to Bayesian Knowledge Tracing – a student model shown to predict post-test performance – and to a model trained to detect transfer, in order to see how well each model can predict preparation for future learning. As a student model predicting PFL will be most useful if it can be used to drive interventions fairly early during tutor usage, we also analyze how much student data is needed for the model to be accurate.

2. DATA SET

The data set used in the analyses presented here came from the Genetics Cognitive Tutor (Corbett et al., 2010). This tutor consists of 19 modules that support problem solving across a wide range of topics in genetics. Various subsets of the 19 modules have been piloted at 15 universities in North America. This study focuses on a tutor module that employs a gene mapping technique called *three-factor cross*, in which students infer the order of three genes on a chromosome based on offspring phenotypes, as described in (Baker, Corbett, et al 2010). In this laboratory study, 71 undergraduates enrolled in genetics or in introductory biology courses at Carnegie Mellon University used the three-factor cross module. The students engaged in Cognitive Tutor-supported activities for one hour in each of two sessions. All students completed standard three-factor cross problems in both sessions. During the first session, some students were assigned to complete other cognitive-tutor activities designed to support deeper understanding; however, no differences were found between conditions for any robust learning measure, so in this analysis we collapse across the conditions and focus solely on student behavior and learning within the standard problem-solving activities. The 71 students completed a total of 22,885 problem solving attempts across 10,966 problem steps in the tutor.

Post-tests, given by paper-and-pencil, consisted of four activities (cf. Baker, Corbett, et al., 2010). Three tests were given immediately after tutor usage: a straightforward problem-solving post-test, a transfer test, and a test of preparation for future learning. The fourth test was a delayed retention test. Within this paper we focus on predicting performance on the test of preparation for future learning, requiring the student to learn new skills after using the tutor. The PFL test consisted of 2½ pages of instruction on the reasoning needed for an analogous, but more complex, four-factor cross gene mapping task, followed by a single four-factor cross problem for students to solve.

Students demonstrated successful learning in this tutor, with an average pre-test performance of 0.31 (SD=0.18), an average post-test performance of 0.81 (SD=0.18), and an average PFL performance of 0.89 (SD=0.15). The correlation between the problem-solving post-test and the PFL test was 0.41, suggesting that, although problem-solving skill and preparation for future learning were related, PFL may be predicted by more than just simply skill at problem-solving within this domain.

3. ANALYSIS OF MODEL USING CROSS VALIDATION

In this paper, we introduce a model that predicts each student's performance on a test of preparation for future learning (PFL), using a hybrid of data mining and knowledge engineering methods. Within this approach, a small set of features is selected based on past literature. Each feature is defined as the proportion of times a specific student behavior occurs in the log files. Within feature selection, the goodness criterion is the cross-validated correlation between an individual feature and each student's performance on the PFL test. Cross-validated correlation is computed between the predictions from each of the test folds and the actual PFL test scores; positive cross-validated correlation indicates the relationship is consistent between the training and test folds (but has no implication about the relationship's direction). By contrast, negative cross-validated correlation indicates that the models obtained from the training folds fail to predict the actual scores in the test folds. . Finally a model is trained on these features to predict each student's performance on the PFL test, using cross-validation.

We then compare this model to a baseline prediction of PFL, Bayesian Knowledge Tracing (BKT) (Corbett & Anderson, 1995) fit using brute force. Bayesian Knowledge-Tracing fit in this fashion has been previously shown to predict student post-test problem-solving performance reasonably well within the Genetics Tutor (Baker et al., 2010). As BKT accurately predicts problem-solving post-tests, and the PFL test was reasonably correlated to the problem-solving post-test in this study, BKT should correlate reasonably well to PFL. But the goal of a robust learning test such as PFL is to measure depth of understanding that may not be reflected solely in basic problem-solving skill, which is tracked by BKT. Hence, it may be possible to develop a detector based on other performance features that predicts PFL better than BKT, under cross-validation.

3.1 FEATURE ENGINEERING

The first step of our process was to engineer a set of features based on a combination of theory and prior work detecting related behaviors. Since we were predicting post-test performance, we focused on proportions of behavior across the period of use of the tutoring system (e.g. what proportion of time a student engaged in behavior N). Many of the features below depend on a continuous variable, such as pause duration (2, 3, 4) or probability of knowing a skill (1,6). For each such feature, we used a cut-off value to indicate the presence or absence of a behavior, in order to identify the incidence of specific behaviors hypothesized to be associated with robust learning. That is, we empirically determined (see section 3.2) a cut-off value that indicates the student behavior occurred (e.g. a long pause or low probability), rather than averaging the actual values (pause durations or probabilities). We tested the following features:

1. Help avoidance (Aleven et al, 2006), not requesting help on poorly known skills, and its converse, feature 1', requesting help on relatively poorly known skills

2. Long pauses after receiving bug messages (error messages given when the student's behavior indicates a known misconception) which may indicate self-explanation (cf. Chi et al., 1989) of the bug message, and its converse, 2', short pauses after receiving bug messages (indicating a failure to self-explain)
3. Long pauses after reading on-demand hint messages (potentially indicating deeper knowledge or self-explanation), and a related feature, 3', short pauses after reading the on-demand hint message
4. Long pauses after reading an on-demand hint message and getting the current action right (cf. Shih, Koedinger, & Scheines, 2008), and a related feature, 4', short pauses after reading an on-demand hint message and getting the current action right. Features 4 and 4' can be seen as sub-sets of features 3 and 3'.
5. Off-task behavior (Baker, 2007), and a related feature, 5', long pauses that are not off-task (may indicate self-explanation, or asking teacher for help – cf. Schofield, 1995)
6. Long pauses on skills assessed as known (may indicate continuing to self-explain even after proceduralization), and a related feature, 6', short pauses on skills assessed as known
7. Gaming the system (Baker et al., 2008), and a related feature, 7', fast actions that do not involve gaming
8. Contextual slip/carelessness (known to predict post-test problem-solving performance – Baker et al, 2010)
9. The presence of spikes during learning using the moment-by-moment learning model, which estimates the probability that the student learned a relevant skill at each step in problem solving (spikes in this model have been found to predict final knowledge in the tutor – cf. Baker, Goldstein, Heffernan, in press).

Five of these features showed positive cross-validated correlations between the individual feature and the students' performance on the PFL test: 1 (failing to request help on poorly-known skills), 3 (long pauses after reading hint messages), 6' (short pauses on skills assessed as known), 7' (fast actions that do not involve gaming), and 9 (spikiness in the moment-by-moment learning model). The exact definition of these features was:

1: Proportion of actions where the student has a probability under N of knowing the skill, according to Bayesian Knowledge Tracing (Corbett & Anderson, 1995), does not ask for help, and makes an error on their first attempt. Initial cut-off value of N = 60% probability.
3: Proportion of actions where the student asks for hint, and then makes their next action in over N seconds. Initial value of N = 5 seconds.
6': Proportion of actions where the student has a probability over 0.95 of knowing the skill, according to Bayesian Knowledge Tracing (Corbett & Anderson, 1995), and applies the skill in under N seconds. Initial value of N = 5 seconds.
7': Proportion of actions where the student enters an answer or requests a hint in under N seconds, but the action is not labeled as gaming, using a gaming detector previously trained on data from a high school algebra course (cf. Baker & de Carvalho, 2008 – where a single detector was trained on all lessons to maximize detector generalizability – cf. Baker et al., 2008). This detector has previously been shown to achieve a correlation over 0.3 to the post-test within another dataset from the same lesson in the Genetics Tutor. Initial value of N = 5 second.
9: Highest value of moment-by-moment learning model estimate (cf. Baker, Goldstein, & Heffernan, in press) for each skill, divided by the average moment-by-moment learning estimate for that skill, averaged across skills, for the student. This model infers student

learning at each problem step, and is initially trained using data from future student correctness within the tutor, but the model itself uses only data from the past.

As can be seen, four of these features depend on a threshold parameter, N; adjusting this parameter can result in very different behavior. In all three cases, we started with an initial plausible value of N, as given above. The following section discusses how these features were optimized later in the modeling process.

3.2 FEATURE OPTIMIZATION

We used brute-force grid search to find an optimal cut-off level for four of the above mentioned features (in grid search, values are tried for every step at the same interval – for instance 0.5 seconds, 1 second, 1.5 seconds, 2 seconds, etc.). Variables involving probabilities were searched at a grid size of 0.05; variables involving time were searched at a grid size of 0.5 seconds with the exception of feature 6', which was searched at a grid size of 5 seconds. After the generation of the features at different grids, we built one-parameter linear regression models predicting PFL from each feature using leave-out-one-cross-validation, in RapidMiner 4.6 (Mierswa et al., 2006). Cross-validated correlation was used as the goodness measure. Single-feature regression models fit on the whole data set and their associated cross-validated correlations are shown in Table I.

Many of the features, subsequent to optimization, changed meaning. For instance, feature 1, initially conceptualized as Help Avoidance, achieved optimal performance when help avoidance occurred on skills for which the probability the student knew the skill was <=1 – that is to say, on all help. So feature 1 can be re-conceptualized as the help/error ratio, and specifically as MoreErrorsLessHelp. Similarly, feature 6' was initially conceptualized as short pauses on mastered skills, but the optimal performance was achieved when the cut-off for shortness was set to 55 seconds. Hence, feature 6' can be re-conceptualized as pauses which are not long, on mastered skills.

The feature most strongly associated with PFL was making errors rather than requesting help, which was negatively associated with PFL (cross-validated r=0.356). This feature may have multiple interpretations; for instance, it may be that these students learned less by avoiding help (cf. Aleven et al., 2006), perhaps learning less at a conceptual level as the tutor hints are fairly conceptual in nature. This may in turn have made these students less prepared for future learning in the same domain. Alternatively, it may be that these students are less successful at learning from text (or less motivated to learn from text), causing them both to avoid hints in the tutor, and to perform less well as reading the text needed to succeed on the future learning test. Distinguishing these two hypotheses is challenging, but may be a productive avenue for future research.

A second feature individually associated with PFL is spending less than a minute on skills assessed as known, which achieved a cross-validated r of 0.340. This feature suggests that relatively quick performance on known skills is indicative of robust learning. Similarly, non-gaming actions taking less than 4 seconds were positively correlated with PFL (cross-validated r=0.166).

Additionally, the spikiness of the moment-by-moment learning model, is positively associated with PFL, achieving a cross-validated r of 0.286. This finding suggests that if a student's learning more frequently occurs in relatively sudden "aha" moments, as compared to occurring more gradually, deeper learning is occurring.

Finally, spending more than 5.5 seconds to answer after receiving a hint was negatively correlated with PFL (cross-validated r=0.230). This result is unexpected, as pauses after reading hints have previously been shown to be positively correlated with post-test performance (e.g. Shih, Koedinger, & Scheines, 2008). One possible explanation is that pausing after reading help is generally beneficial, but that the students

Table I. Linear regression model predicting the PFL test using single optimized features.

Feature	PFL=	Cross-validated *r*
1. MoreErrorsLessHelp	-0.980 * MoreErrorsLessHelp + 1.01	0.356
3. Long pauses After Hint with Time > 5.5	-2.445 * LongPausesAfterHint + 0.912	0.230
6'. Non-Long Mastered Skill with Time < 55	+ 0.768 * Non-LongMasteredSkill + 0.296	0.340
7'. Fast Not Gaming with Time < 4	+0.340 * FastNotGaming + 0.739	0.166
9. Spikiness	+0.0083 * spikiness + 0.7773	0.286

who do so with high frequency are the students who are struggling. Hence, it may be an interesting question for future research to examine whether there is a non-linear relationship between lengthy pauses after reading hints and PFL (and learning in general).

3.3 DETECTOR DEVELOPMENT

Given the set of optimal features, we developed linear regression models in RapidMiner 4.6 (Mierswa et al., 2006) using Forward Selection (Ramsey & Schafer, 1997), conducted by hand. In Forward Selection, the best single-parameter model is chosen, and then the parameter that most improves correlation (cross-validated in this case) is repeatedly added until no more parameters can be added which improve the correlation.

Within RapidMiner, feature selection was turned off, and each potential model was tested in a separate run – while this creates some risk of over-fitting (even given the use of cross-validation), it enables us to determine how well a specific set of features predicts PFL. Keeping feature selection on would result in some features being filtered out for some sub-sets of the data, making it harder to infer how well a specific set of features predicts PFL. As before, Leave-One-Out Cross-Validation (LOOCV) was used to reduce the risk of over-fitting, and the goodness metric used was the Pearson correlation between the predictions and each student's performance on the PFL test. In addition, as an additional control on over-fitting, we did a first pass where we eliminated all features that, taken individually, had cross-validated correlation below zero. We give differences in cross-validated correlation rather than statistical significance tests, as a measure of generalizability; differences in non-cross-validated correlations of non-nested models have low statistical power – (Cohen, 1988) – and comparing cross-validated correlations is a redundant test – (cf. Efron & Gong, 1983).

The best model, using the optimal feature cut-offs, and fit to all data (not cross-validated; cross-validation produces one model per each of the 71 training sets) was as follows:

*PFL = -0.7837 * MoreErrorsLessHelp(1) -1.3042 * LongPausesAfterHints(3) + 0.9936*

3.4 DETECTOR GOODNESS

The overall correlation of this model to the PFL test was 0.360, only very slightly better than feature 1 alone (0.356). By comparison, fitting a baseline model consisting of Bayesian Knowledge Tracing post-test predictions (using brute force – cf. Baker et al., 2010) to the PFL test results, under LOOCV, achieved a correlation of 0.285 to the PFL test. This is a reasonable baseline, as Bayesian Knowledge-Tracing has previously been shown to predict the post-test well in this tutor (Baker et al., 2010) as well as in general (e.g. Corbett & Anderson, 1995; Corbett & Bhatnagar, 1997), and performance on the PFL test was correlated reasonably well to performance on the post-test in this data set (non-cross-validated r=0.41). Hence, the optimal feature model appears to perform substantially better at predicting PFL than this reasonable baseline, although there is still likely to be substantial room for improvement.

Interestingly, if the post-test and the PFL detector are used together in linear regression to predict the PFL test, the cross-validated correlation is 0.391. However, if the Bayesian Knowledge Tracing estimates and the PFL detector are used together in this way to predict the PFL test, the cross-validated correlation drops to 0.309. This result suggests that, despite the PFL detector's reasonable effectiveness at detecting PFL, the paper post-test still captures a small amount of variance in students' preparation for future learning which is not yet detectable from student behavior in the tutor software. Furthermore, this additional predictive power is separate from the assessment of student knowledge made by Bayesian Knowledge Tracing (since combining BKT with the PFL detector does not lead to better prediction).

As another test of the PFL detector's unique predictive power, we can compare it to another model of robust learning. In past work, we have developed a model that predicts transfer (i.e., the application of problem-solving knowledge to novel but related problems without additional instruction), using the same overall method that was used to develop the PFL detector (Baker et al., in press). That model is:

*Transfer = –1.20 * HelpAvoidance(1) – 14.764 * FastAfterBugs(2') + 0.234 * FastNotGaming(7') + 0.832*

Though there is considerable correlation (0.520) between the transfer test and the PFL test, the transfer detector did not do as well as the PFL detector at predicting PFL. With no re-fitting, the transfer detector achieved a cross-validated correlation of 0.273 to the PFL-test, comparable to BKT's performance at predicting PFL, and lower than the PFL detector's cross-validated performance of 0.360. This result suggests that PFL, to at least a moderate extent, is associated with different student behavior in the tutor than transfer is. These results, though somewhat small in absolute terms, are fairly large in relative terms (0.360 is 32% higher than 0.273), suggesting that it is unlikely this difference is solely due to noise in the two test measures.

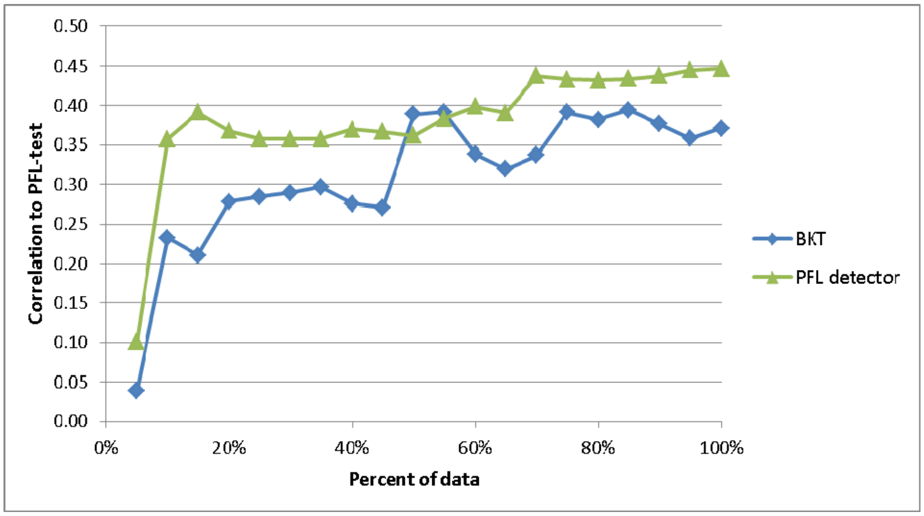

Fig.1 Predicting PFL with first N percent of the data.

4. ANALYSIS OF MODEL FOR USE IN RUNNING TUTOR

One potential criticism of models developed using proportions of behavior across entire episodes of tutor use is that the models, in their initial form, may not be usable in a running tutor. Bayesian Knowledge Tracing makes a prediction after each problem-solving step, which can be used to drive Cognitive Mastery Learning (Corbett & Anderson, 1992). If an entire tutor lesson worth of data is required for accurate inference, the detector may have low usefulness for intervention.

However, it is possible to make a version of the model that can be used in a running tutor. Similar to the way that Bayesian Knowledge Tracing makes a prediction after each problem-solving step, it is possible to take the data up to a specific problem step and attempt to make an overall inference about the probability of PFL, using only the data collected up until that point. In other words, the features used in the model can be computed at any time, using the data collected so far. In this section, we investigate how much data is needed for the model to make accurate predictions within this data set, comparing our model's predictive power to Bayesian Knowledge-Tracing, when both are given limited data.

Our first step in this process is to construct subsets of data containing the first N percent of each student's interactions within the tutor. We use every increment of 5% -- e.g. a subset with the first 5% of each student's data (not taking skills into account – e.g. data from some skills may not be present in the first 5%), a subset with the first 10% of each student's data, a subset with the first 15% of each student's data, up to 100%. This gives us 20 data sets. We then compute the optimal features discussed in section 3 for each subset of data. Next, we apply the PFL prediction model generated using the full data set (e.g. we do not refit the models for the new data sets). We also apply Bayesian Knowledge Tracing on the limited data sets without re-fitting the BKT parameter estimates. After obtaining the predictions we compute the correlation between each of the predictions and each student's performance on the PFL test. Cross-validation is not used, as the model is not being re-fit in either case.

Figure 1 shows the graph with x-axis as percent of data and y-axis as the correlation to the PFL test. The graph depicts the predictive performance of the PFL prediction model and BKT based on having the first N percent of the data. From the graph we can see that the PFL prediction model performs substantially better than BKT for small amounts of data. For instance, with only the first 20% of the data, the PFL prediction model achieves a solid correlation of 0.368 to the PFL test, while the BKT model achieves a weaker correlation of 0.278. These findings suggest that it may be possible to use the PFL prediction model to drive interventions, from very early in tutor usage.

5. DISCUSSION AND CONCLUSIONS

Within this paper, we have presented a model which can predict with reasonable accuracy how well a student will perform on a post-test measuring how well the student is prepared for future learning (PFL), within a Cognitive Tutor for College Genetics. We find that this model achieves decent cross-validated prediction of this PFL post-test, and achieves better cross-validation prediction than Bayesian Knowledge Tracing, a measure of skill learning within the tutor software, or a detector trained to detect transfer. Furthermore, we find that the PFL detector achieves a large proportion of its predictive power by the time the student has completed 20% of the tutor software, suggesting that the PFL detector can be used to drive intervention early enough to influence overall learning. Overall, we view this detector as a potential step towards educational software that can predict and respond automatically to differences in the robustness of student learning, an important complement to ongoing research on designing educational software that promotes preparation for future learning (Tan & Biswas, 2006; Chin et al., 2010).

This model is based on the proportion of time the student engages in long pauses after requesting help (cf. Shih et al., 2008), and the ratio of help requests to errors, with more help use associated with better preparation for future learning, but lengthy pauses after help requests associated with poorer preparation for future learning. Both of these features indicate that the use of help is particularly essential for preparing students for future learning. Past studies have found mixed relationships between help and domain learning (cf. Aleven et al., 2003, 2006; Beck et al., 2008); this analayis, however, suggests that help use may under certain conditions lead to robust forms of learning that are not captured by typical metrics of in-tutor performance (e.g. Beck et al., 2008) and problem-solving post-tests (e.g. Aleven et al., 2006). We recommend that future research on help-seeking and learning consider measures of preparation for future learning of new skills and concepts to a greater degree.

ACKNOWLEDGEMENTS

This research was supported via grant “Empirical Research: Emerging Research: Robust and Efficient Learning: Modeling and Remediating Students’ Domain Knowledge”, National Science Foundation award number #DRL-0910188, via grant “Promoting Robust Understanding of Genetics with a Cognitive Tutor that Integrates Conceptual Learning with Problem Solving”, Institute of Education Sciences award number #R305A090549, and by the Pittsburgh Science of Learning Center, NSF award number #SBE-0836012.

REFERENCES

ALEVEN, V., MCLAREN, B., ROLL, I., & KOEDINGER, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. International Journal of Artificial Intelligence and Education, 16, 101-128.

BAKER, R.S.J.D. (2007) Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems.Proceedings of ACM CHI 2007: Computer-Human Interaction, 1059-1068.

BAKER, R.S.J.D., CORBETT, A.T., ALEVEN, V. (2008) More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. Proceedings of the 9th International Conference on Intelligent Tutoring Systems, 406-415.

BAKER, R.S.J.D., CORBETT, A.T., GOWDA, S.M., WAGNER, A.Z., MACLAREN, B.M., KAUFFMAN, L.R., MITCHELL, A.P., GIGUERE, S. (2010) Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization, 52-63

BAKER, R.S.J.D., CORBETT, A.T., ROLL, I., KOEDINGER, K.R. (2008) Developing a Generalizable Detector of When Students Game the System. User Modeling and User-Adapted Interaction, 18, 3, 287-314.

BAKER, R.S.J.D., DE CARVALHO, A. M. J. A. (2008) Labeling Student Behavior Faster and More Precisely with Text Replays. Proceedings of the 1st International Conference on Educational Data Mining, 38-47.

BAKER, R.S.J.D., GOLDSTEIN, A.B., HEFFERNAN, N.T. (in press) Detecting Learning Moment-by-Moment. To appear in International Journal of Artificial Intelligence in Education.

BAKER, R.S.J.D., GOWDA, S.M., CORBETT, A.T. (in press) Towards predicting future transfer of learning. To appear in the Proceedings of the 15th International Conference on Artificial Intelligence in Education.

BAKER, R.S.J.D., PARDOS, Z.A., GOWDA, S.M., NOORAEI, B.B., HEFFERNAN, N.T. (in press) Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems. To appear in *Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization.*

BRANSFORD, J.D., SCHWARTZ, D.L. (1999) Rethinking transfer: a simple proposal with multiple implications. *Review of Research in Education, 24*, 61-100.

CHI, M.T.H., BASSOK, M., LEWIS, M.W., REIMANN, P., GLASER, R. (1989) Self-explanations: how students study and use examples in learning to solve problems. Cognitive Science, 13, 145-182.

CHIN, D.B., DOHMEN, I.M., CHENG, B.H., OPPEZZO, M.A., CHASE, C.C., SCHWARTZ, D. L. (2010) Preparing Students for Future Learning with Teachable Agents. Educational Technology Research and Development, 58 (6), 649-669.

COHEN, J. (1988) Statistical Power Analysis for the Behavioral Sciences, 2nd Edition. Hillsdale, NJ: LEA.

CORBETT A., BHATNAGAR A. (1997). Student Modeling in the ACT Programming Tutor: Adjusting Procedural Learning Model with Declarative Knowledge. User Modeling: Proceedings of the 6th International Conference, 243-254.

CORBETT, A.T., ANDERSON, J.R. (1992) Student modeling and mastery learning in a computer-based programming tutor. Proceedings of the International Conference on Intelligent Tutoring Systems, 413-420.

CORBETT, A.T., ANDERSON, J.R. (1995) Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction, 4, 253-278.

CORBETT A.T., MACLAREN, B., KAUFFMAN, L., WAGNER, A., & JONES, E. (2010). A Cognitive Tutor for genetics problem solving: Learning gains and student modeling. Journal of Educational Computing Research, 42, 219-239.

DESMARAIS, M. C., MESHKINFAM, P., AND GAGNON, M. Learned Student Models with Item to Item Knowledge Structures. User Modeling and User-Adapted Interaction, 16 5 (2006), 403-434

EFRON, B. & GONG, G. (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. American Statistician, 37, 36–48

GONG, Y, BECK, J. E., HEFFERNAN, N. T. (2010). Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. Proceedings of the 10th International Conference on Intelligent Tutoring Systems, 35-44.

MARTIN, J., VANLEHN, K. (1995) Student assessment using Bayesian Nets. *International Journal of Human-Computer Studies*, *42*, 575-591.

MIERSWA, I., WURST, M., KLINKENBERG, R., SCHOLZ, M., EULER, T. (2006) YALE: rapid prototyping for complex data mining tasks. Proceedings of the 12th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, 935-940.

PAVLIK, P.I., ANDERSON, J.R. (2008). Using a Model to Compute the Optimal Schedule of Practice. Journal of Experimental Psychology: Applied, 14 (2), 101-117.

PAVLIK, P.I., CEN, H., KOEDINGER, J.R. (2009) Performance Factors Analysis – A New Alternative to Knowledge Tracing. Proceedings of the 14th International Conference on Artificial Intelligence in Education, 531-540.

RAMSEY, R.L., SCHAFER, D.W. (1997) The Statistical Sleuth. Belomnt, CA: Wadsworth Publishing.

SCHOFIELD, J. W. (1995). Computers and Classroom Culture. Cambridge, UK: Cambridge University Press.

SHIH, B., KOEDINGER, K.R. AND SCHEINES, R. (2008) A response time model for bottom-out hints as worked examples. Proceedings of the 1st International Conference on Educational Data Mining, 117-126.

SINGLEY, M.K., ANDERSON, J.R. (1989) The Transfer of Cognitive Skill. Cambridge, MA: Harvard University Press.

TAN, J, BISWAS, G: The Role of Feedback in Preparation for Future Learning: A Case Study in Learning by Teaching Environments. Intelligent Tutoring Systems 2006: 370-381