Towards Predicting Future Transfer of Learning

Ryan S.J.d. Baker[1], Sujith M. Gowda[1], and Albert T. Corbett[2]

[1] Department of Social Science and Policy Studies, Worcester Polytechnic Institute
100 Institute Road, Worcester MA 01609, USA
{rsbaker,sujithmg}@wpi.edu
[2] Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Avenue,
Pittsburgh, PA 15213, USA
corbett@cmu.edu

Abstract. We present an automated detector that can predict a student's future performance on a transfer post-test, a post-test involving related but different skills than the skills studied in the tutoring system, within an Intelligent Tutoring System for College Genetics. We show that this detector predicts transfer better than Bayesian Knowledge Tracing, a measure of student learning in intelligent tutors that has been shown to predict performance on paper post-tests of the same skills studied in the intelligent tutor. 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 reach near-asymptotic predictive power.

Keywords: Transfer, Bayesian Knowledge Tracing, Educational Data Mining, Student Modeling, Robust Learning.

1 Introduction

Over the previous two decades, knowledge engineering and educational data mining (EDM) methods have led to increasingly precise models of students' knowledge as they use intelligent tutoring systems and other AIED systems. Modeling of student knowledge has been a key theme in AIED 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. 8, 14, 16, 19]. 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 future correctness within the tutoring software studied [cf. 12, 16].

However, the ultimate goal of AIED systems is not to promote better future performance within the system itself. Ideally, an intelligent tutoring system or other AIED system should promote "robust" learning [13] that is retained over time [15], transfers to new situations [20], and prepares students for future learning [6]. Historically, student modeling research has paid limited attention to modeling the robustness of student learning. Although studies have demonstrated that learning in intelligent tutors can be made robust [1, 7, 17], student models used in intelligent tutors have typically not explicitly modeled robustness, including whether knowledge will

G. Biswas et al. (Eds.): AIED 2011, LNAI 6738, pp. 23–30, 2011.

transfer. In fact, only a handful of studies have even attempted to predict immediate posttest performance on the same skills studied in a tutor [e.g., 3, 8, 10, 19], a very limited form of transfer. For instance, Bayesian Knowledge Tracing models of student knowledge have been shown to predict this type of post-test performance [8], but with a small but consistent tendency to overestimate students' average post-test performance, systematic error that can be corrected by incorporating pretest measures of students' conceptual knowledge into the knowledge tracing model [9]. Other student models have modeled the inter-connection between skills, within a tutor [cf. 14]. However, it is not clear whether this can in turn support prediction of transfer to different skills and situations outside of the tutor.

Within this paper, we present a model designed to predict student performance on a transfer post-test, a post-test involving related but different skills than the skills studied in the tutoring system, within a Cognitive Tutor for genetics problem solving [10]. This model is generated using a combination of feature engineering and linear regression, and is cross-validated at the student level. We compare this model to Bayesian Knowledge Tracing – a student model shown to predict post-test performance – as a predictor of transfer. As a student model predicting transfer 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 came from the Genetics Cognitive Tutor [10]. 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 [3]. 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: a straightforward problem-solving post-test discussed in detail in [3], a transfer test, a test of preparation for future learning, and a delayed retention test. Within this paper we focus on predicting performance on the transfer test of robust learning. The transfer test included two problems intended to tap students' understanding of the underlying processes. The first was a three-factor cross problem that could not be solved with the standard solution method and required students to improvise an alternative method.

The second problem asked students to extend their reasoning to four genes. It provided a sequence of four genes on a chromosome and asked students to reason about the crossovers that must have occurred in different offspring groups.

Students demonstrated good 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 transfer test performance of 0.85 (SD=0.18). The correlation between the problem-solving post-test and the transfer test was 0.590 suggesting that, although problem-solving skill and transfer skill were related, transfer 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 the transfer test, using a hybrid of data mining and knowledge engineering methods. Within this approach, a small set of features are selected based on theory and prior work to detect related constructs. These features are based on thresholds which are given initial values but are also optimized by grid search, using as goodness criterion the cross-validated correlation between an individual feature and each student's performance on the transfer test. Finally a model is trained on these features (using both the original and optimized thresholds) to predict each student's performance on the transfer test, and is cross-validated. We then compare this model to a baseline prediction of transfer, Bayesian Knowledge Tracing (BKT) [8] fit using brute force, which has been previously shown to predict student post-test problem-solving performance reasonably well within this lesson [3]. Recent work in other tutoring systems has suggested that other algorithms (BKT fit using Expectation Maximization; Performance Factors Analysis) may fit within-tutor performance slightly better than BKT fit using Brute Force [12, 16], but thus far no published studies have demonstrated that these algorithms fit post-test performance better. As BKT accurately predicts problem-solving post-tests, and the transfer test was reasonably correlated to the problem-solving post-test in this study, it should correlate reasonably well to transfer. Hence, a useful detector predicting transfer should perform better than BKT, under cross-validation.

3.1 Feature Engineering

The first step of our process was to engineer the feature set. As we were predicting performance on a measure external to the tutor, given after tutor usage, we focused on proportions of behavior across the full period of use of the tutoring system (e.g. what proportion of time a student engaged in each behavior). Our data features consisted of the following behaviors (the prime notation connotes a feature closely related to the previous feature): 1) Help avoidance [2]; 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; 2') Short pauses after receiving bug messages, indicating failure to self-explain; 3) Long pauses after reading hint messages; 4) Long pauses after reading hint message(s) and then getting the next action right [cf. 18]; 5) Off-task behavior;

5') Long pauses that are not off-task; 6) Long pauses on skills assessed as known; 7) Gaming the system [4]; 7') Fast actions that do not involve gaming; 8) Carelessness, detected as contextual slip [3]; 9) Learning spikes [5].

Three of these features were incorporated into the final model predicting transfer: 1, 2’ and 7’. We will discuss our model development process in a subsequent section, but in brief, no additional feature both achieved better cross-validated performance than zero on its own, and also improved cross-validated predictive power in a model already containing these three features. The exact operational 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 [8], does not ask for help, and makes an error on their first attempt. Initial value of N = 60% probability.
2': Proportion of actions where the student enters an answer labeled as a bug, and then makes their next action 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 a full year of data from high school algebra [4]. Initial value of N = 1 s.

Each of these three features depends on a threshold parameter, N; adjusting a feature’s parameter can result in very different behavior. In some analyses below, we used an arbitrary but plausible value of N chosen prior to optimization, as given above. Features were then optimized to select optimal thresholds, using grid search. Parameters involving probabilities were searched at a grid size of 0.05; parameters involving time were searched at a grid size of 0.5 seconds.

3.2 Detector Development

Our first step towards developing a detector was to fit a one-parameter linear regression model predicting transfer from each feature, using leave-out-one-cross-validation (LOOCV), in RapidMiner 4.6. LOOCV was conducted at the student level, the overall level of the analysis. The cross-validated correlations for single-feature regression models are shown in Table 1. This process was conducted for both original and optimized threshold parameters. Both help avoidance (1) and making fast responses after bugs (2') were found to be negatively associated with transfer. Fast non-gaming actions (7') were positively correlated with transfer, perhaps because these actions are a signal that the skill has been acquired very strongly (additionally, for low values of the threshold, very few fast non-gaming responses are help requests, which is some additional evidence for interpreting this feature in this fashion).

Table 1. Goodness of single-feature linear regression models at predicting transfer

Feature	Direction of relationship	Cross-validated r (orig. thresholds)	Cross-validated r (optimized thresholds)
1. Help Avoidance	Neg.	0.362	0.376
2'. Fast After Bugs	Neg.	0.167	0.269
7'. Fast Not Gaming	Pos.	0.028	0.189

Given each set of features, we developed linear regression models using RapidMiner 4.6. To find the set of parameters, Forward Selection was conducted by hand. In Forward Selection, the best single-parameter model is chosen, and then the parameter that most improves the model is repeatedly added until no more parameters can be added which improve the model. Within RapidMiner, feature selection was turned off, and each potential model was tested in a separate run, in order to determine how well a specific set of features predicts transfer. 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 transfer. The goodness metric used was the LOOCV correlation between the predictions and each student's performance on the transfer 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 model generalizability; comparing cross-validated correlations is a redundant test [cf. 11].

The cross-validated correlation of the model to the transfer test was 0.407, for the original thresholds, and 0.416 for the optimized thresholds. By comparison, the Bayesian Knowledge Tracing estimates of student knowledge achieved a cross-validated correlation of 0.353 to the transfer test. Hence, the transfer model appears to perform better than this reasonable baseline.

We then investigated the possibility that multiplicative interaction features (where one feature is multiplied with another feature) would lead to a better model. To reduce the potential for over-fitting, we restricted our analysis to multiplicative features consisting of the 3 features above, and the 3 original features. This model achieved a cross-validated correlation of 0.435 to the transfer test, for the original thresholds, and 0.428 for the optimized thresholds.

One question is whether the resultant models are better predictors solely of transfer or of student knowledge overall. This can be investigated by examining how well the transfer prediction models predict the regular problem-solving post-test, with no re-fitting. If we predict the problem-solving post-test using Bayesian Knowledge-Tracing, we obtain a correlation of 0.535. As seen in Table 2, each of the four transfer prediction models perform better than this at predicting the post-test, with the optimized model without multiplicative interactions performing best (r=0.633).

Table 2. Cross validated correlation between models and transfer test

Model	Cross-validated correlation to transfer test	Correlation to problem-solving test
Only BKT	0.353	0.535
Model with optimized features (no interactions)	0.416	0.633
Model with original features (no interactions)	0.407	0.546
Model with optimized features (multiplicative interactions)	0.428	0.615
Model with original features (multiplicative interactions)	0.435	0.598

4 Analysis of Model for Use in Running Tutor

One potential concern with models developed using proportions of behavior across entire episodes of tutor use is that the models may not be usable to drive interventions in a running tutor. If an entire tutor lesson worth of data is required for accurate inference, the detector may have low usefulness for intervention compared to approaches such as Bayesian Knowledge Tracing which make a prediction after each problem-solving step [8]. However, it is possible to make a version of the transfer detector that can be used in a running tutor. Specifically, it is possible to take the data up to a specific problem step, compute the model features using only the data collected up until that point, and make an inference about the probability of transfer. 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 20 subsets of data containing the first N percent of each student's interactions within the tutor, using every increment of 5% of the data. Our process for doing this does not take skills into account – e.g. data from some skills may not be present in the first 5%. We then compute the feature values for each data subset, using the optimized thresholds. Next, we apply the transfer prediction model generated using the full data set to the new data sets (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 transfer test. Cross-validation is not used, as the model is not being re-fit in either case.

Figure 1 shows the predictive performance of the transfer prediction model and BKT based on having the first N percent of the data. From the graph we can see that the transfer prediction model performs substantially better than BKT for small amounts of data. For instance, with only the first 20% of the data, the transfer prediction model achieves a solid correlation of 0.463 while the BKT model achieves a much weaker correlation of 0.254. These findings suggest that it may be possible to use the transfer prediction model to drive interventions, from very early in tutor usage.

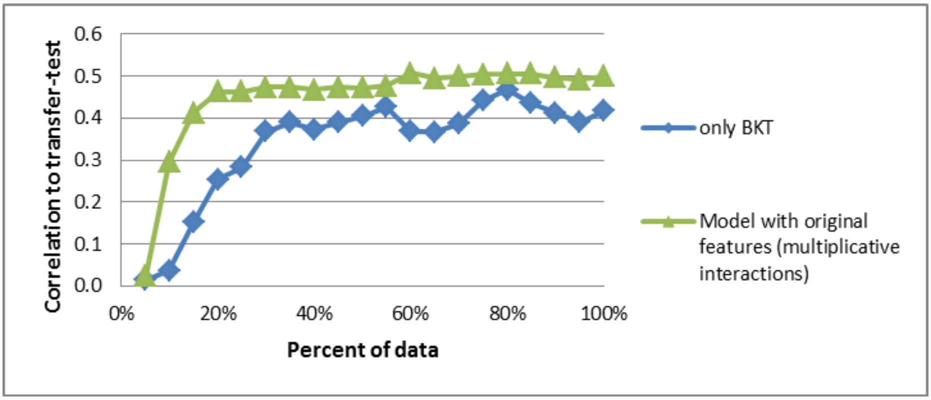

Fig. 1. Predicting transfer with first N percent of the data

5 Conclusions

Within this paper, we have presented a model which can predict with reasonable accuracy how well a student will perform on a transfer post-test, a post-test involving related but different skills than the skills studied in the tutoring system, within a Cognitive Tutor for College Genetics. This model is based on the percentage of student actions that involve help avoidance [2], fast actions which do not involve gaming the system [4], and fast responses after receiving a bug message. Interestingly, two of these features (help avoidance and fast responses after bugs) appear to reflect meta-cognitive behavior rather than reflecting what students know, at least according to prior theory that these behaviors are meta-cognitive in nature [e.g. 2,18]. The result is in line with theory that suggests a key role for meta-cognition in transfer [13].

We examine several variants of this model, and find that a variant of the model based on multiplicative interactions of non-optimized versions of these features achieves the best cross-validated prediction of the transfer test. This is substantially higher than the cross-validated correlation of Bayesian Knowledge Tracing, a measure of skill learning within the tutor software. Furthermore, we find that the transfer detector achieves near-asymptotic predictive power by the time the student has completed 20% of the tutor software, suggesting that the transfer detector can be used to drive intervention early enough to influence overall learning. Another potential use of future work is to investigate the degree to which the transfer detector correlates to other measures of robust learning, such as retention [cf. 15] and preparation for future learning [cf. 6], in order to improve understanding of how these constructs relate to one another. Overall, we view this detector as a potential early step towards intelligent tutors that can predict and respond automatically to differences in the robustness of student learning, an important complement to ongoing research on designing tutors that promote robust learning [e.g. 1, 7, 17].

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

1. Aleven, V., Koedinger, K.R.: An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. Cognitive Science 26, 147–179 (2002)
2. Aleven, V., McLaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. International Journal of Artificial Intelligence and Education 16, 101–128 (2006)
3. 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.: Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. In: Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization, pp. 52–63 (2010)

4. Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a Generalizable Detector of When Students Game the System. User Modeling and User-Adapted Interaction 18(3), 287–314 (2008)
5. Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T.: Detecting the moment of learning. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 25–34. Springer, Heidelberg (2010)
6. Bransford, J.D., Schwartz, D.: Rethinking transfer: A simple proposal with multiple implications. Review of Research in Education 24, 61–100 (1999)
7. Butcher, K.R.: How Diagram Interaction Supports Learning: Evidence from Think Alouds during Intelligent Tutoring. In: Goel, A.K., Jamnik, M., Narayanan, N.H. (eds.) Diagrams 2010. LNCS, vol. 6170, pp. 295–297. Springer, Heidelberg (2010)
8. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction 4, 253–278 (1995)
9. Corbett, A., Bhatnagar, A.: Student Modeling in the ACT Programming Tutor: Adjusting Procedural Learning Model with Declarative Knowledge. In: User Modeling: Proceedings of the 6th International Conference, pp. 243–254 (1997)
10. Corbett, A.T., Kauffman, L., MacLaren, B., Wagner, A., Jones, E.: A Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. Journal of Educational Computing Research 42(2), 219–239 (2010)
11. Efron, B., Gong, G.: A leisurely look at the bootstrap, the jackknife, and cross-validation. American Statistician 37, 36–48 (1983)
12. Gong, Y., Beck, J.E., Heffernan, N.T.: Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 35–44. Springer, Heidelberg (2010)
13. Koedinger, K.R., Corbett, A.T., Perfetti, C.: (under review) The Knowledge-Learning-Instruction (KLI) Framework: Toward Bridging the Science-Practice Chasm to Enhance Robust Student Learning (manuscript under review)
14. Martin, J., VanLehn, K.: Student Assessment Using Bayesian Nets. International Journal of Human-Computer Studies 42, 575–591 (1995)
15. Pavlik, P.I., Anderson, J.R.: Using a Model to Compute the Optimal Schedule of Practice. Journal of Experimental Psychology: Applied 14(2), 101–117 (2008)
16. Pavlik, P.I., Cen, H., Koedinger, J.R.: Performance Factors Analysis – A New Alternative to Knowledge Tracing. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 531–540 (2009)
17. Salden, R.J.C.M., Koedinger, K.R., Renkl, A., Aleven, V., McLaren, B.M.: Accounting for Beneficial Effects of Worked Examples in Tutored Problem Solving. Educational Psychology Review 22, 379–392 (2010)
18. Shih, B., Koedinger, K.R., Scheines, R.: A response time model for bottom-out hints as worked examples. In: Proc. 1st Int'l Conf. on Educational Data Mining, pp. 117–126 (2008)
19. Shute, V.J.: SMART: Student modeling approach for responsive tutoring. User Modeling and User-Adapted Interaction 5(1), 1–44 (1995)
20. Singley, M.K., Anderson, J.R.: The Transfer of Cognitive Skill. Harvard University Press, Cambridge (1989)