Towards Automatically Detecting Whether Student Learning is Shallow

Sujith M. Gowda, Ryan S.J.d. Baker, Albert T. Corbett, Lisa M. Rossi

Abstract. Recent research has extended student modeling to infer not just whether a student knows a skill or set of skills, but also whether the student has achieved robust learning – learning that enables the student to transfer their knowledge and prepares them for future learning (PFL). However, a student may fail to have robust learning in two fashions: they may have no learning, or they may have shallow learning (learning that applies only to the current skill, and does not support transfer or PFL). Within this paper, we present automated detectors which identify shallow learners, who are likely to need different intervention than students who have not yet learned at all. These detectors are developed using K* machine learned models, with data from college students learning introductory genetics from an intelligent tutoring system.

Keywords. robust learning, student modeling, educational data mining, intelligent tutoring system

INTRODUCTION

In recent years, there has been increasing interest in developing learning systems which promote not just learning of the domain skills being taught directly by the system, but also "robust learning," (Koedinger, Corbett, & Perfetti, 2012) – learning that enables students to transfer their knowledge (Singley & Anderson, 1989; Fong & Nisbett, 1991), prepares them for future learning (Bransford & Schwartz, 1999; Schwartz & Martin, 2004), and leads to retention of knowledge over the long-term (Schmidt & Bjork, 1992; Bahrick, Bahrick, Bahrick, & Bahrick, 1993). There is considerable evidence that students with insufficiently deep conceptual understanding may develop "shallow learning," which may focus on the surface features of a learning situation rather than on the deeper conceptual features of a domain that make learning robust (Chi, Feltovich, & Glaser, 1981; Rittle-Johnson & Siegler, 1998). Avoiding the outcome of shallow learning has been an increasing focus of research in the AIED community in recent years.

Researchers in the AIED community have attempted to increase the robustness of learning through a variety of approaches. One line of research attempts to develop feedback that supports students in developing robust understanding, including early work to incorporate conceptual explanations into "bug messages" (Anderson et al., 1995), followed by work to provide conceptually-meaningful graphical feedback (Corbett & Trask, 2000; Butcher, 2010) and integration of natural language discussion of difficult conceptual material (Graesser et al., 2004; Katz, Connelly, & Wilson, 2007). A second line of work gives student feedback on domain-independent strategies in order to improve the robustness of learning, including strategies for problem-solving (Chi & VanLehn, 2007), meta-cognition (Chin et al., 2010; Tan & Biswas, 2006), and help-seeking (Roll et al., 2011). A third line of work has attempted to support robust learning through encouraging and scaffolding students in conducting self-explanation (Aleven & Koedinger, 2002; Corbett et al., 2011; Hausmann & VanLehn, 2007; McLaren, Lim, & Koedinger, 2008; Schwonke et al., 2009).

One approach that has been proposed in recent years is to conduct individualization based on assessment of the robustness of student learning. For example, the FaCT system optimizes the order of practice of Chinese-language items based on prediction of the eventual retention of items that will occur depending on the schedule of practice chosen (Pavlik et al., 2008). This approach has considerable potential; past work to optimize practice based on assessments of student knowledge within intelligent tutoring systems has been shown to lead to better learning (Corbett, 2001), as well as more efficient learning (Cen, Koedinger, & Junker, 2007).

Along these lines, an increasing amount of recent work has attempted to assess the robustness of student learning, in various fashions. Beyond the work to assess retention in (Pavlik et al., 2008), there has also been work to assess retention in an intelligent tutoring system (ITS) teaching flight skills (Jastrzembski, Gluck, & Gunzelmann, 2006). Computational modeling work has been conducted to analyze the mechanisms leading to accelerated future learning (Li, Cohen, & Koedinger, 2010). Research has been conducted on the inter-connections between skills during learning, providing a way to infer how student knowledge transfers within an intelligent tutoring system (cf. Martin & VanLehn, 1995; Pavlik et al., 2008). Additionally, Baker, Gowda, & Corbett (2011a, 2011b) have developed models that can predict whether a student will eventually transfer their knowledge or be prepared for future learning outside the learning software, based on a set of features of student meta-cognitive behavior within a Cognitive Tutor for college Genetics.

Each of these projects is a step towards the long-term vision of modeling and supporting the acquisition of robust learning. All of this work has a common characteristic: it is focused on identifying students who will obtain robust learning, differentiating them from all other learners. However, this previous work does not explicitly distinguish shallow learners, students who may learn the exact skills presented in the tutor, but who do not learn in a robust fashion. In specific, work that simply identifies if a student has robust learning may be unable to distinguish between students who have not yet learned a skill, and students who have learned a skill shallowly. In this paper, we propose instead to build models that specifically identify the students with shallow learning, in order to make it possible to provide differentiated support tailored to this group's needs. A student who is on track to achieve robust knowledge but who has not yet fully acquired the skill may simply need more tutor practice (cf. Corbett, 2001); by contrast, a student who has shallow learning may need support in building from their procedural skill to deeper conceptual understanding. There are now interventions which have been shown to help students acquire robust learning, as discussed above, but not all students may need such interventions. A detector which can identify a student who has shallow learning, when combined with such interventions, may have the potential to enable richer intervention and better learner support than is currently possible.

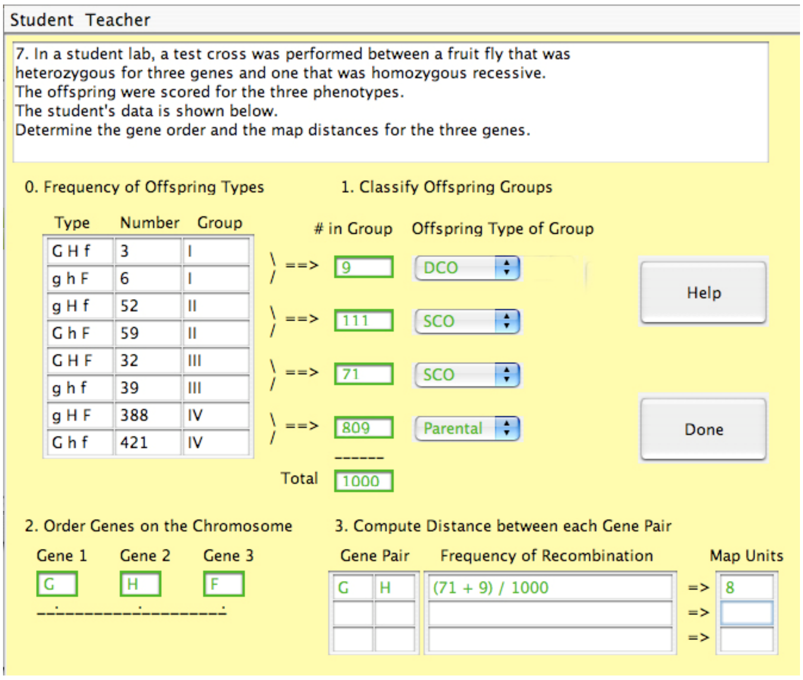
Hence, in this paper, we attempt to go beyond existing approaches that either identify learning (not considering whether it is robust) or identify robust learning (not considering the differences between students with shallow learning and students with no learning), to specifically identify shallow learners. By adding this type of detector to previously developed type of detectors, we will be able to effectively distinguish students with robust learning, students with shallow learning, and students with neither type of learning, supporting more differentiated learning support.

We conduct this research in the context of a Cognitive Tutor for Genetics problem-solving (Corbett, McLaren, Kauffman, Wagner, & Jones, 2010). As in Baker, Gowda, & Corbett (2011a, 2011b), we engineer a set of features of student learning and meta-cognitive behavior. We then use these features to predict whether students demonstrate shallow learning, which is operationalized as the students who perform better on tests of the exact material covered in the tutor, than on a test of robust learning. We create two variants of the shallowness detector, one for transfer and another for PFL. We report these detectors' effectiveness at identifying shallow learners when cross-validated at the student level (repeatedly trained on one group of students and tested on other students), and analyze the detectors' internal features, comparing their features to features previously used to predict transfer and PFL. We also study the degree to which the two detectors of shallow learning agree with one another.

DATA SET

Cognitive Tutors are a type of interactive learning environment which use cognitive modeling and artificial intelligence to adapt to individual differences in student knowledge and learning (Koedinger & Corbett, 2006). Within a Cognitive Tutor, Bayesian Knowledge Tracing (Corbett & Anderson, 1995) is used to determine how well the student is learning the skills taught in the tutor. Bayesian Knowledge Tracing calculates the probability that the student knows each skill based on that student's history of responses within the tutor. Using these estimates of student knowledge, the tutoring system gives each student problems that are relevant to the skills which he or she needs to learn, attempting to bring the student to mastery (i.e., 95% probability of knowing each skill) on all skills covered in the current tutor lesson.

The data analyzed in this paper come from undergraduates using the Genetics Cognitive Tutor (Corbett et al., 2010). The Genetics Cognitive 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 the data from a tutor module that employs a gene mapping technique called *three-factor cross* (3FC) in which students reason about recombination in meiosis to infer the order of three genes on a chromosome based on offspring phenotype frequencies. The tutor interface for this reasoning task is displayed in Figure 1. The 3FC technique is used to determine both the order of three genes, (F, G, and H in this example), which lie on one chromosome, and to find the relative distances between the pairs of genes. In this technique two organisms are crossed (two fruit flies in the example) and the resulting distribution of offspring phenotypes is analyzed to infer the arrangement of the three genes on the chromosome. In Figure 1 the student has almost finished the problem. The student has summed the number of offspring in each of four phenotype groups that appear in the offspring table, and has categorized each group (as "parental" (no crossovers occurred in meiosis), "single crossover" during meiosis, or "double crossover" during meiosis). The student has compared the phenotype patterns in the offspring groups, to identify the middle of the three genes and entered a gene sequence below the table. Finally, in the lower right the student has calculated the crossover frequency between two of the genes, G and H, and the distance between the two genes. The student will perform the last two steps for the other two gene pairs. Within this lesson, on-demand help was available, and bug messages were given when students' actions indicated a specific lack of understanding of the learning task or the process or order of completing the problem.



7. In a student lab, a test cross was performed between a fruit fly that was heterozygous for three genes and one that was homozygous recessive.
The offspring were scored for the three phenotypes.
The student's data is shown below.
Determine the gene order and the map distances for the three genes.

0. Frequency of Offspring Types

Type	Number	Group
G H f	3	I
g h F	6	I
g H f	52	II
G h F	59	II
G H F	32	III
g h f	39	III
g H F	388	IV
G h f	421	IV

1. Classify Offspring Groups

# in Group	Offspring Type of Group
9	DCO
111	SCO
71	SCO
809	Parental
Total 1000	

2. Order Genes on the Chromosome

Gene 1	Gene 2	Gene 3
G	H	F

3. Compute Distance between each Gene Pair

Gene Pair		Frequency of Recombination		Map Units
G	H	(71 + 9) / 1000	=>	8
			=>	
			=>	

Fig. 1. The Three-Factor Cross lesson of the Genetics Cognitive Tutor.

The data used in this paper, first published in (Baker, Gowda, & Corbett, 2011a, 2011b), were produced by 71 students who were enrolled in genetics or introductory biology classes at Carnegie Mellon University. These students used Cognitive Tutor-supported activities in two one-hour laboratory sessions, on successive days. In each session, students completed standard three-factor cross problems. During the first lab session, some students piloted 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 actions across 10,966 problem steps in the tutor. Four paper-and-pencil post-tests followed the tutor activities (cf. Baker et al., 2011). 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 (PFL). A retention test was administered one week later; this retention test correlated with substantially fewer behavioral indicators (as discussed below) than the transfer and PFL detectors, making a shallowness detector less potentially powerful than shallowness detectors based on transfer and PFL. As such, the retention test will not be analyzed in detail within this paper.

Within this paper we focus analysis on the immediate problem-solving post-test, the transfer test of robust learning, and the PFL test. The problem-solving post-test consisted of two problems, and had two test forms, counterbalanced with the pre-test. Each of the two problems on each test form consisted of 11 steps involving 7 of the 8 skills in the three-factor cross tutor lesson, with two skills applied twice in each problem and one skill applied three times. The transfer test included two problems intended to tap students' understanding of the underlying processes of three-factor cross. 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. In the preparation for future learning test, students were asked to solve parts of a four-factor cross problem. The reasoning is related to solving a three-factor cross problem, but sufficiently more complicated that a student could not be expected to invent a solution method by direct transfer, and certainly not in a short period of time. Consequently, this PFL test presented a 2.5-page description of the reasoning in a four-factor cross experiment, then asked students to solve some elements of a four-factor cross problem: identifying the middle genes, identifying all the offspring groups with a crossover between two specific genes and to find the map distance between those two genes. Each test was graded in terms of how many skills were demonstrated by the student, with partial credit allowed in cases where a response was correct once a previous response's incorrectness was taken into account.

Students demonstrated successful learning in this tutor, with an average pre-test performance of 0.31 (SD = 0.19), an average post-test performance of 0.81 (SD = 0.18). Only one student performed more poorly on the post-test than the pre-test. No students achieved a perfect score on the pre-test, while 18 students achieved a perfect score on the post-test. Students were also successful on the transfer test, with an average score of 0.85 (SD = 0.18), and on the PFL test, with an average score of 0.89 (SD=0.15).

The average scores on the basic problem-solving post-test and transfer tests were similar, with some students scoring higher on the transfer test than the post-test, because the two transfer test problems were somewhat easier than the basic problems, *if* a student was able to transfer a deep understanding to the novel problems. More specifically, the basic problem-solving test asked students to solve two conventional abductive 3FC problems. The first transfer problem was an abductive three-factor cross problem that could not be solved with the standard solution, but if students understood the standard solution well enough to improvise an alternative method, the resulting arithmetic steps were somewhat easier. The second transfer problem asked students to extend their reasoning to four genes. Students were not asked to solve an abductive problem, but instead were provided a genotype (the order of four genes on a chromosome) and asked to reason about the genotypes that result from various crossovers between genes. Again, if students can transfer their understanding of gene crossovers to sets of four genes, this forward modeling problem is somewhat easier to solve than the standard abductive problems. Similar considerations govern the PFL test, which gave the student explicit support in learning to solve the new type of problem.

The correlation between the problem-solving post-test and the transfer test was 0.59 suggesting that, although problem-solving skill and transfer skill were related, transfer may be predicted by more than simply skill at problem-solving within this domain. The problem-solving post-test and PFL test were also correlated, at a level of 0.41. Transfer and PFL were also correlated, with a positive correlation of 0.52.

SHALLOWNESS DETECTOR

Label Generation

The first step towards developing a data-mined model to predict which students have shallow learning is to create an operational definition of shallow learning that can be used as a training label (i.e., a "ground truth" label of the construct being predicted) for our shallowness detector. We employed data from the problem-solving post-test, the transfer post-test, and the PFL post-test to do this. We consider shallow learning in two fashions; learning that the student cannot transfer ("no-transfer-shallow learning") and learning that does not prepare the student for future learning ("no-PFL-shallow learning"). We operationalize no-transfer-shallow learning as the difference between a student's problem-solving test score and their transfer test score, and no-PFL-shallow learning as the difference between a student's problem-solving test score and their PFL test score. Better performance on the problem-solving test than on one of the measures of robust learning indicates the student has acquired basic problem-solving knowledge, but in a shallow fashion, without the deep understanding that enables the application of that knowledge in novel situations.

Given the approximately equal average performance on the three tests, we can directly compare the percent correct on each test to assess whether a student is a shallow learner or not (if the tests had radically different average performance, it might be better to use percentile rank on each test, or to look for natural breakpoints in the difference in scores between the two tests). As such, the present analysis treats students who achieve higher scores on the problem-solving post-test than on a robustness measure as having shallow learning.

According to this operational definition, 25 of the 71 students in this study are labeled as no-transfer-shallow learners. Of the remaining 46 students, treated as not having no-transfer-shallow learning, 10 had perfect scores on both the transfer test and post-test. No other students had the same score on the two tests. The other 36 students had higher scores on the transfer test than the post-test. Among the 25 students labeled as no-transfer-shallow learners, there was an average of a 0.11 point difference between performance on the two tests (standard deviation = 0.11), with an average score of 0.85 on the problem-solving post-test, and an average score of 0.74 on the transfer test.

For no-PFL-shallowness, 20 of the 71 students are labeled as no-PFL-shallow learners. Of the remaining 20 non no-PFL-shallow learners, 8 had perfect scores on both the PFL test and post-test. Among the 16 students labeled as no-PFL-shallow learners, there was an average of a 0.17 point difference between performance on the two tests (standard deviation = 0.10), with an average score of 0.92 on the problem-solving post-test, and an average score of 0.75 on the PFL test.

The full pattern of performance on the tests can be seen in Table 1.

Table 1
The pattern of performance on the tests. Mean and SD (in parens) given for each group.

Post	Transfer	PFL	Count	Pre-test	Post-test	Transfer	PFL
Perfect	Deep	Deep	7	38.3% (20.3%)	100% (n/a)	100% (n/a)	100% (n/a)
Perfect	Deep	Shallow	4	37.5% (15.4%)	100% (n/a)	100% (n/a)	83.3% (6.4%)
Perfect	Shallow	Deep	1	31.8% (n/a)	100% (n/a)	95.7% (n/a)	100% (n/a)
Perfect	Shallow	Shallow	6	37.9% (15.1%)	100% (n/a)	86.2% (9.3%)	81.5% (13.5%)
Not Perfect	Deep	Deep	31	26.4% (18.9%)	70.1% (17.4%)	88.6% (15.7%)	92.1% (10.8%)
Not Perfect	Deep	Shallow	3	34.8% (21.5%)	86.4% (12.0%)	94.2% (6.6%)	74.1% (17.0%)
Not Perfect	Shallow	Deep	11	32.2% (9.8%)	76.9% (16.7%)	68.8% (20.1%)	96.0% (7.5%)

Not Perfect	Shallow	Shallow	7	31.8% (25.3%)	82.5% (11.6%)	67.1% (13.9%)	65.1% (19.7%)

Data Features

The next step in our process of developing a model that can automatically identify shallow learning was to identify properties of students' problem-solving actions in the Cognitive Tutor that may be hallmarks of shallow learning. Towards this end, we selected a set of action-level features based on a combination of theory and prior work to model and detect related constructs. In particular, prior research on detectors of transfer (Baker, Gowda, & Corbett, 2011b) and PFL (Baker, Gowda, & Corbett, 2011a) influenced our design of features. As in that work, we can infer which students had shallow learning using the method discussed in the previous section, but we do not know exactly what actions are associated with the shallow learning in advance. Hence, we take features calculated at the level of actions, and aggregate them across actions. We do so using two kinds of computations: the proportion of time specific behaviors occurred, and a verage quantitative values across actions. The 25 features used in this analysis included two categories of basic features, and two categories of complex features.

The first category of basic features focused on overall response time and time spent processing tutor-provided assistance, including:

(1) Average response time per student response in the tutor software for first attempts at a problem step.

(2) The average unitized response time (in standard deviations above or below the mean for students on the current skill) for first attempts at a problem step.

(3) The proportion of actions that involved a lengthy pause after the student received a bug message, which may indicate self-explanation (cf. Chi, Bassok, Lewis, Reimann, & Glaser, 1989) of the bug message.

(4) The proportion of fast responses after a bug message, which may indicate a failure to self-explain.

(5) The proportion of long pauses after reading on-demand help messages (potentially indicating self-explanation).

(6) The proportion of short pauses after reading on-demand help messages.

(7) Long pauses after reading an on-demand help message and getting the current action right (cf. Shih, Koedinger, & Scheines, 2008), thought to be a particularly clear indicator of self-explanation.

(8) Short pauses after reading an on-demand help message and getting the current action.

The second category of basic features focused on aggregate indicators of correctness and help use:

(9) The proportion of student answers that were correct.

(10) The proportion of student attempts that were help requests.

(11) The proportion of incorrect answers that received bug message.

The first category of complex features involved Bayesian Knowledge Tracing estimates of the student's knowledge of relevant skills and estimates of the probability of correct performance coming from this model (Corbett & Anderson, 1995), as well as additional features leveraging this information:

(12) The average probability the student knew the skill, according to Bayesian Knowledge Tracing.

(13) The average probability of a correct answer, according to Bayesian Knowledge Tracing.

(14) The proportion of fast actions on well-known skills, potentially indicating fluency (cf. Mattler, Massey, & Kellman, 2011).

(15) The proportion of slow actions on well-known skills, potentially indicating that the student is continuing to think through the material even after achieving high accuracy.

The second category of complex features focused on features derived from previous research on meta-cognition and disengagement:

(16) The proportion of problem steps where the student engaged in help avoidance (Aleven et al., 2006), not requesting help on poorly known skills (on the student's first attempt at a specific problem step),

(17) The proportion of problem steps where the skill was known and help was not sought (considered an appropriate behavior in Aleven et al., 2006's model of meta-cognition in Cognitive Tutors)

(18) The proportion of fast actions not involving gaming the system (which may indicate a very well-known skill). This feature is computed using an automated detector of gaming the system (Baker, Corbett, Roll, & Koedinger, 2008).

(19) The proportion of slow actions not involving off-task behavior. This feature is computed using an automated detector of off-task behavior (Baker, 2007).

(20) The student's average probability of contextual slip/carelessness on errors, making an error when the student is assessed to know the relevant skill (which is known to predict post-test problem-solving performance – Baker et al., 2010). This feature is computed using an automated detector (Baker et al., 2010).

(21) The certainty of slip, the average contextual probability of slip among actions with over 50% probability of being a slip (Baker et al., 2010).

(22) The student's average probability of contextual guess on correct responses, giving a correct response despite having a low probability of knowing the skill (cf. Baker, Corbett, & Aleven, 2008)

(23) The certainty of guess, the average contextual probability of guess among actions with over 50% probability of being a guess.

(24) The student's average learning per learning opportunity using the moment-by-moment learning model, which estimates the probability that the student learned a relevant skill at each step in problem solving (Baker, Goldstein, & Heffernan, 2011).

(25) The average probability that the student was gaming the system, computed using an automated detector of gaming the system (Baker, Corbett, Roll, & Koedinger, 2008).

Some of these features relied upon cut-offs; in these cases, an optimized cut-off was chosen using a procedure discussed in the next section.

Detector Development

We fit detectors of shallowness using step regression and K*, two machine learning algorithms that fit fairly distinct patterns. Step regression (not the same as stepwise regression) involves fitting a linear regression model to predict the labels of shallowness using the features of student behavior in the tutor, and then thresholding that model's predictions with a pre-chosen cut-off, in this case 0.5. Within this statistical framework, all students for whom the linear regression predicted values of 0.5 or higher are assessed to have non-shallow learning, whereas all students for whom the linear regression predicted values below 0.5 are assessed to have shallow learning. Hence, this framework takes numerical predictions of shallowness and transforms them into a binary prediction of whether the student's learning is shallow or not, which can be compared to the labels initially derived from the two tests. Step regression achieves a general goal similar to the better-known logistic regression; logistic regression is generally preferred by statisticians due to the relatively high interpretability of model odds ratios and the ease of computing statistical significance.

The other algorithm which was used is K* (Cleary & Trigg, 1995), which is an instance-based learning algorithm. Instance-based learners classify an instance based on the assumption that similar instance will have similar classification. K* uses entropy as a distance measure in identifying the similarity between the instances. Using entropy as the distance measure, K* handles both categorical and real valued attributes and is robust to missing values in the data set. Step regression and K* are used because these algorithms often obtain good cross-validated performance on data sets similar to the data studied here; furthermore, these two algorithms are very different than each other; step regression finds linear patterns, while K* finds clumps of similar students.

These detectors of shallowness are assessed using 10-fold student-level cross-validation. In 10-fold cross-validation (Efron & Gong, 1983), the data points are divided into ten groups, each of which serves successively as a test set. That is, for each of the ten groups, the other nine groups are used to produce a model, and then the tenth group is used to test that model. In this case, we cross-validate at the level of students (i.e., each student is either in the test fold or the training fold at a given time), in order to validate model generalizability to new students. Hence, each model's goodness is never tested on the same students it was trained on, but each model is tested on every student. Because this process does not exclude any data points (students) from the modeling process, cross-validation is typically preferred to holding out a test set that is entirely excluded from model development. Cross-validated performance does not assess whether predictive performance is probabilistically better than chance, but whether the model can generalize, a subtle difference from traditional statistical significance tests.

Four metrics were used as the assessment of goodness for each model: (1) A' (Hanley & McNeil, 1982), (2) Cohen's (1960) Kappa, or κ, (3) Precision (Davis & Goadrich, 2006), and (4) Recall (Davis & Goadrich, 2006). A' is the probability that if the detector is comparing two students, one labeled as having shallow learning and the other one not labeled as having shallow learning, it will correctly identify which student is which. A', also called W (the Wilcoxon statistic), closely approximates the area under the ROC curve, also called AUC ROC (Hanley & McNeil, 1982). A model with an A' of 0.5 performs at chance, and a model with an A' of 1.0 performs perfectly. In these analyses, A' was computed using software at http://www.columbia.edu/~rsb2162/computeAPrime.zip which computes

A' directly rather than integrating to estimate the area under the ROC curve (many implementations in standard packages, which use integration, currently give moderately incorrect estimations of AUC ROC for some special cases). Cohen's Kappa (1960) assesses whether the detector is better than chance at identifying the correct action sequences as involving the category of interest. A Kappa of 0 indicates that the detector performs at chance, and a Kappa of 1 indicates that the detector performs perfectly. A' and Kappa both compensate for the possibility that successful classifications can occur by chance (cf. Ben-David, 2008). A' can be more sensitive to uncertainty in classification than Kappa, because Kappa looks only at the final label whereas A' looks at the classifier's degree of confidence in classifying an instance. Models were also evaluated using Precision and Recall, which indicate, respectively, how good the model is at avoiding false positives (measured by the number of true positives detected divided by the sum of true and false positives detected), and how good the model is at avoiding false negatives (measured by the number of true positives detected divided by the sum of true positives and false negatives).

We fit two detectors for each algorithm. The first detector uses only the individual features discussed above in section 3.2. Some of the features, involving proportions of specific types of actions, depend on a threshold parameter (such as how many seconds differentiates a "long pause" from a "short pause"). These parameters were optimized by computing the single-feature step regression model for a range of potential thresholds (see Baker, Gowda, & Corbett (2011b) for more details) and selecting the threshold with the best A' value. The second detector also includes multiplicative interactions between the individual features. In order to reduce the potential for over-fitting (a common outcome for overly-complex models, where a set of features does not generalize well to data from new students – cf. Caruana & Niculescu-Mizil, 2006), we reduce the parameter space of both models prior to fitting full models. The individual feature model is limited to considering features for which a single-feature model has cross-validated kappa above 0, reducing the data space considerably. The multiplicative interaction model only considers the interactions of that subset of features, and furthermore discards features that fail the same test of having cross-validated kappa above 0. However, original features that pass the test are investigated in the multiplicative-interactions model, as well as multiplicative features.

We used Forward Selection to find the best model for each one of the two feature sets. 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. In this case, the goodness criterion for model selection was Kappa. This simple method is chosen to avoid over-fitting.

RESULTS AND DISCUSSION

No-transfer-shallowness

The best-fitting K* and step regression models for each feature set are as follows:

Table 2

Models to detect no-transfer-shallowness with student-level cross-validated A' and Kappa (sorted on Kappa)

(For step regression higher values of model coefficients correspond to non-shallow learners)

(For K*, there are no model coefficients; the best feature set is given)

Model Type	Model/Model Features	A'	Kappa	Precision	Recall
K* No-Interactions	Hint Then Pause With Time < 1 (F6) No Off-task and Time > 13 (F19) Gaming the System (F25) Help Avoidance When Knowledge < 0.55 (F16)	0.766	0.550	0.873	0.815
K* Multiplicative interactions	Hint Then Pause With Time < 1 (F6) * No Off-task and Time > 13 (F19) Hint Then Pause With Time < 1 (F6) * Average Gaming (F25)	0.791	0.527	0.850	0.860
Step Regression Multiplicative-Interactions	- 1.0870 * Average Certainty of Slip (F21) - 335.5212 * Avg Unitized First Action Time(F2) - 2,731.11 * Average Certainty Of Slip (F21) * Avg Unitized First Action Time (F2) - 419.38 * Hint Then Correct Then Pause With Time>14 (F7) squared + 1.0499	0.728	0.419	0.756	0.689
Step Regression No-Interactions	- 1.25 * Average Certainty of Slip (F21) - 1,079.84 * Avg Unitized First Action Time (F2) + 1.07	0.725	0.294	0.730	0.635

As can be seen in Table 2, the K* no-interactions model achieves a better cross-validated Kappa and better precision than all other models, and the K* multiplicative interactions model achieves better A' and better recall than all other models. The K* model with multiplicative interactions achieved an acceptable cross-validated kappa of 0.550 (55% better than chance). It is worth noting that kappa values typically achieved in data mining tend to be lower than kappa values achieved in inter-rater reliability checks among human coders; the standards are different because the goals are different. The agreement between a data-mined model and a construct which is itself noisy will inherently be lower than human agreement on a tightly-defined construct. The A' value for the K* multiplicative-interactions model is 0.766, which indicates that the model can differentiate a student who performs better on the problem-solving test than the transfer test from a student who does not perform better on the problem-solving test than the transfer test, 76.6% of the time. This level of performance on the A' metric is typically considered to be sufficient to enable fail-soft intervention, where the interventions given are not particularly problematic if administered when not needed. Precision and recall were both generally acceptable, with precision at 0.873 and recall at 0.815 for the K* multiplicative-interaction model (for shallowness).

The overall success of the K* models suggests that the relationships between these variables and whether a student has shallow learning will be non-linear. This can be seen in figures 2 through 6, which show the data for pairs of features. Figures 5 and 6 show the features used by the K* multiplicative-interactions model; Figures 2-4 show pairs of the features used by K* no-interactions, but the actual decisions by the algorithm are made on the basis of a four-dimensional feature space, which is difficult to visualize in 2 dimensions. As each of these figures shows, there are clear regions where one class or the other is more common, but it is difficult to characterize these relationships as linear, or even as being characterized by straight lines in this space dividing the regions.

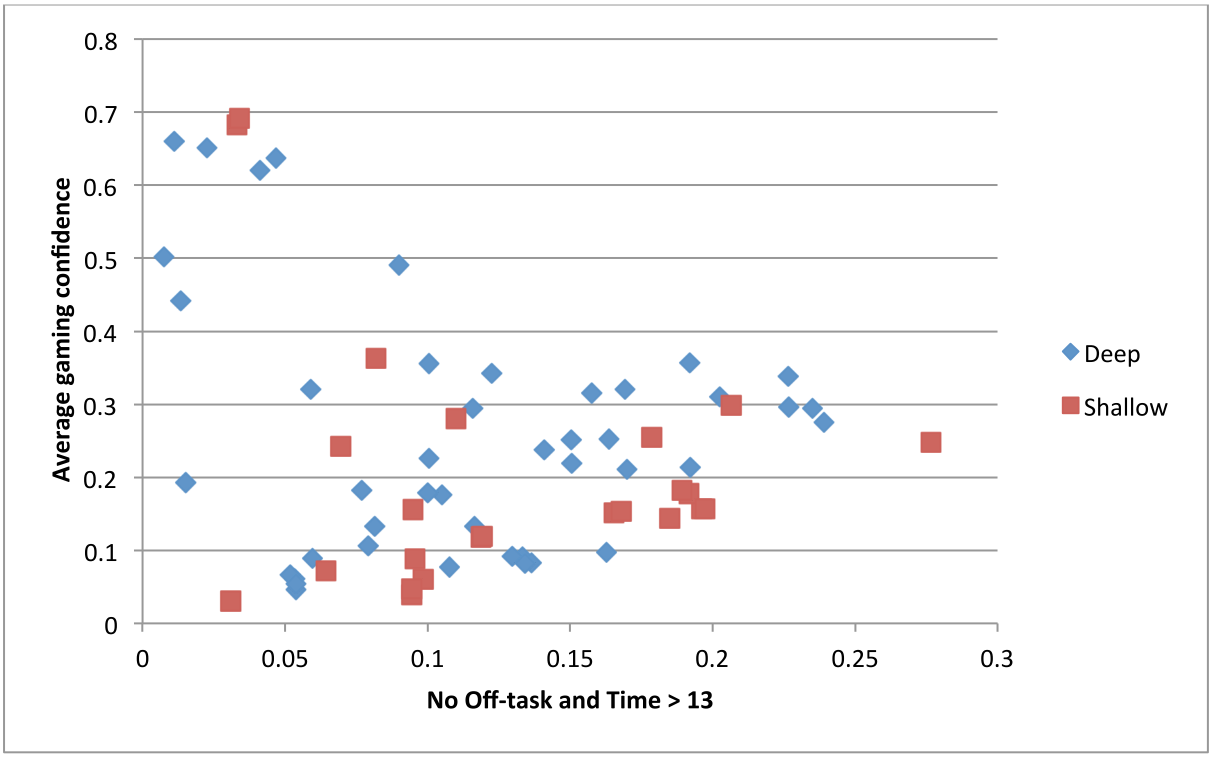


Fig. 2. The relationship between 2 variables used in the K* no-interactions model of transfer-shallowness.

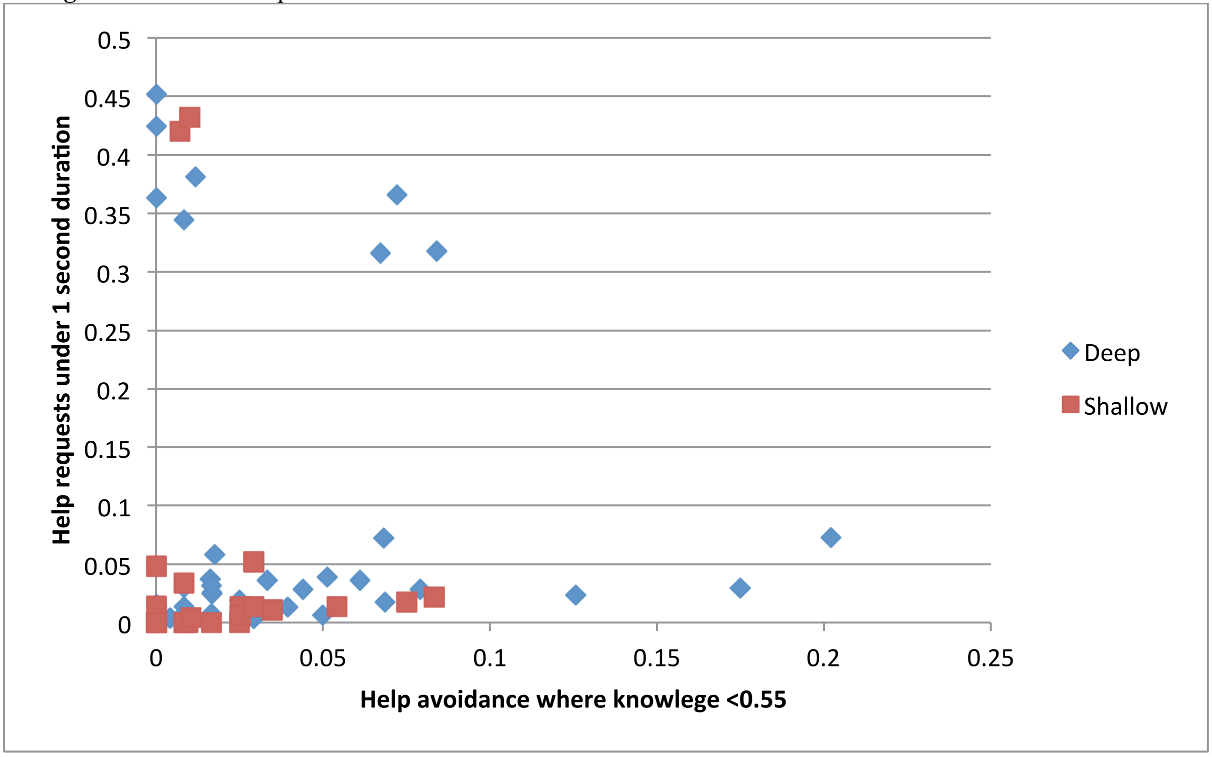


Fig. 3. The relationship between 2 other variables used in the K* no-interactions model of transfer-shallowness. (whole graph)

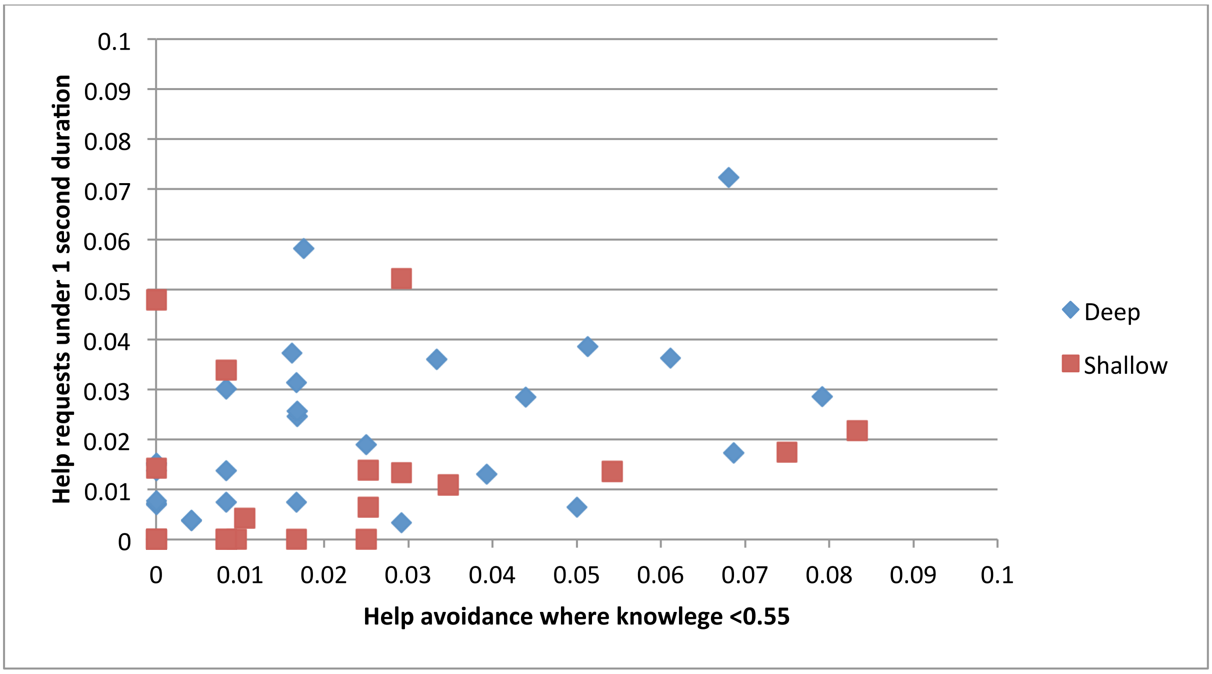


Fig. 4. The relationship between the same two variables used in Figure 3 (focusing on the bottom-left section of the graph)

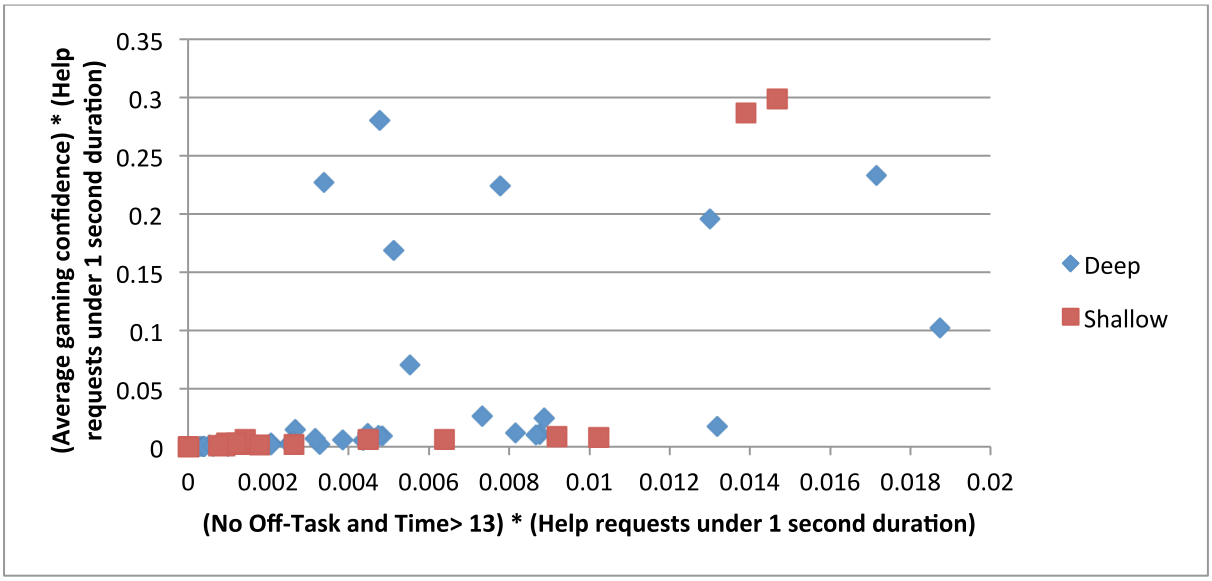


Fig. 5. The relationship between the 2 variables used in the K* mutliplicative-interactions model of transfer-shallowness. (whole graph)

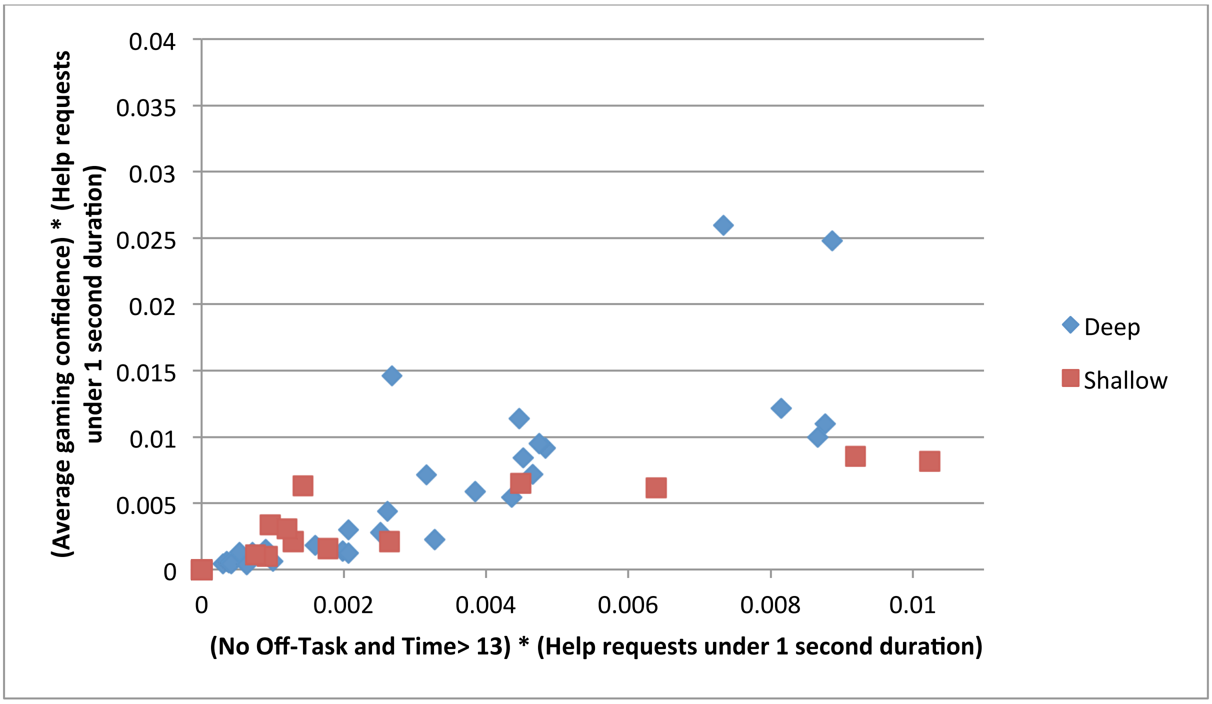


Fig. 6. The relationship between the 2 variables used in the K* mutliplicative-interactions model of transfer-shallowness, as in Figure 5. (focusing on the bottom-left section of the graph)

One key feature in the K* models is whether the student paused only very briefly after requesting a hint (1 second), a behavior that overlaps closely with gaming the system (Baker, Corbett, Roll, & Koedinger, 2008). As figure 5 shows, students who frequently engage in this behavior in a fashion detected as gaming (or frequently engage in this behavior and other forms of gaming such as systematic guessing – cf. Baker, Corbett, Roll, & Koedinger, 2008), or students who frequently engage in this behavior and long pauses that are not off-task, are more likely to be shallow learners; but this behavior is not indicative of shallow learning on its own (see figure 4). At the same time, figure 6 shows that students who frequently engage in this behavior and long pauses that are not off-task are more likely to be shallow learners if they either game frequently or not at all (the middle group may be students who game on easier material – cf. Baker, Corbett, Roll, & Koedinger, 2008 -- among other possibilities).

Aside from these behaviors, a fourth category of behavior that shows up in the K* models is help avoidance. As shown in Figure 4, help avoidance is only a meaningful predictor of shallowness among students who do not read help overly quickly; if the student reads help too quickly, the help is unlikely to do him or her much good (cf. Aleven et al., 2006).

A different set of features is prominent in the step regression models. In both such models, the first feature is the average certainty of slip. The average certainty of slip has been previously shown to predict final tutor knowledge, even after controlling for predictions from Bayesian Knowledge Tracing (Baker et al., 2010); as such, it makes sense that this feature may be related to the depth of learning. While the conventional interpretation of a slip is carelessness, an alternative interpretation is that a slip indicates imperfect acquisition of skill, where a student's skill knowledge works on some problems but not on others (cf. Baker, Corbett, & Aleven, 2008). Such lack of transfer even within basic problem solving is, of course, consistent with shallow learning. Shallow students have an average certainty of slip of 29.53%, compared to non-shallow students, who have an average certainty of slip of 25.3%.

The second feature in both step-regression models is the average unitized first action response time (i.e., the speed with which the student makes his or her first attempt on a problem step, in standard deviations below or above the mean across students). This feature has a negative coefficient in both models, indicating that in general slow response times are associated with shallow learning. Shallow learners are not characterized by fast guesses (which may lead to no learning at all), but just the opposite – they seem rather to be struggling compared to students achieving robust learning. One additional feature that is found in the step regression multiplicative-interactions model is the square of the proportion of time that the student requests a hint, enters a correct answer, and then pauses before entering the next answer. The behavior is rare in general: shallow students engage in this behavior 0.57% of the time, whereas non-shallow students engage in the behavior 0.43% of the time. However, this may not be surprising, as one or two instances of this type of self-explanation across the course of an entire educational activity may have a disproportionate impact. This feature switches directions (e.g. shallow students engage in this behavior less), when considered in the context of a model incorporating the other features.

No-PFL-shallowness

We followed the same procedure to detect no-PFL shallowness, as we had used to detect no-transfer shallowness, fitting both step regression and K* models to detect No-PFL-shallowness.

In the case of the step regression model, no individual feature had cross-validated Kappa over 0 for no-PFL-shallowness. As a result, no step regression model was built for no-PFL-shallowness. Several features had cross-validated Kappa over 0 for K*, so models could still be built in this case.

In the case of K*, successful models were possible. The multiplicative-interactions model achieved a better A' and Kappa than the no-interactions model, achieving an A' of 0.790, and a Kappa of 0.55. Excellent precision and recall (for shallowness) were also achieved. Model goodness is shown in Table 3.

Table 3

K* models to detect no-PFL-shallowness with student-level cross-validated A' and Kappa

Model Type	Model Features	A'	Kappa	Precision	Recall
K* Multiplicative interactions	No Off-task and Time > 17 (F19) No Off-task and Time > 17 (F19) squared Avg Response Time (F1) squared	0.790	0.55	0.891	0.887
K* No interactions	No Off-task and Time > 17 (F19) Avg Response Time (F1)	0.758	0.36	0.857	0.907

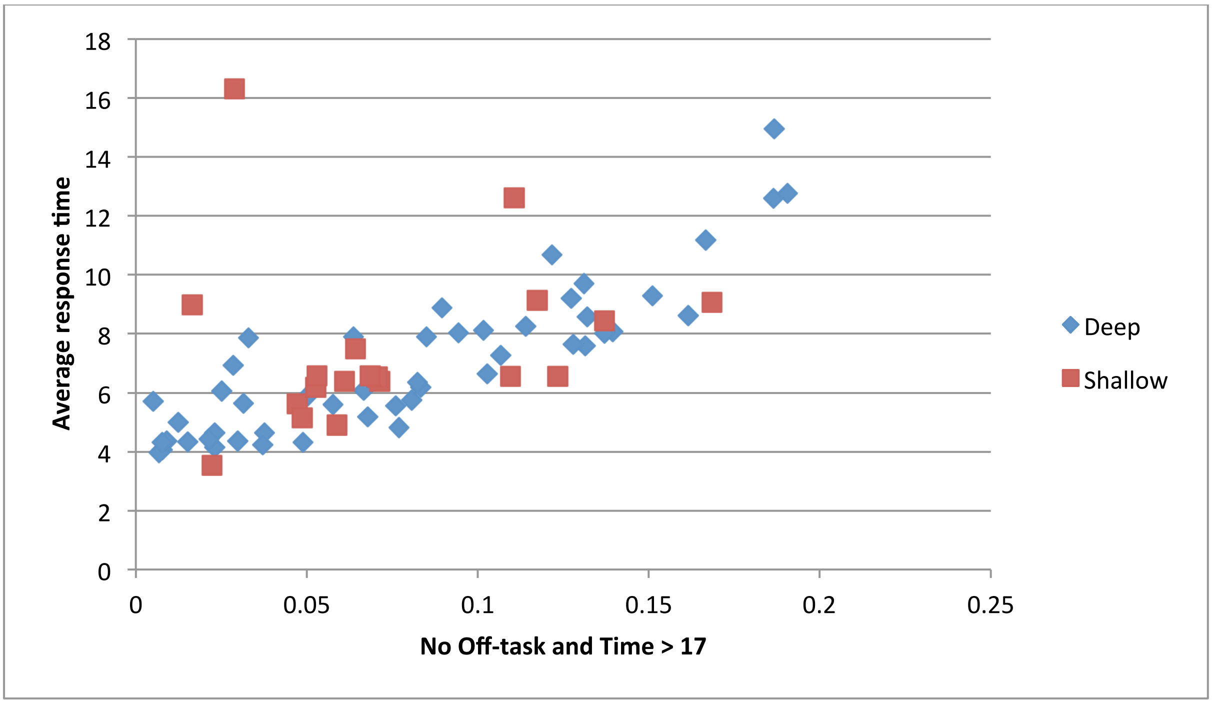


Fig. 7. The relationship between the 2 variables used in the K* mutliplicative-interactions model, as in Figure 5. (focusing on the bottom-left section of the graph)

As can be seen in Figure 7, there is a general trend towards students with longer average response time also having more lengthy pauses that are not off-task. Students with average response time well above this trend line are more likely to be shallow learners. Students on the trend line with very high or very low values for both variables are more likely to be deep; but there is a cluster about 1/3 of the way along the trend line that contains many of the shallow students. Looking at the scatterplot, it is clear that deep and shallow learners generally dominate different regions of the graph, but the relationship is not linear in nature.

Commonality Between Constructs

Transfer and PFL are moderately correlated to each other, with a positive correlation of 0.52. We know that automated detectors of transfer and PFL can successfully predict the other construct, with minimal degradation (Baker, Corbett, & Gowda, in press). As such, it is worth investigating whether detectors of transfer-shallow can predict PFL-shallow, and whether detectors of PFL-shallow can predict transfer-shallow.

The features used to predict transfer-shallow and PFL-shallow have some overlap, with lengthy non-off-task pauses showing up in both models. When the models built to predict transfer-shallow are applied to predict PFL-shallow, they have very minimal degradation in terms of A', achieving an A' of 0.75, almost as good as a model trained to predict PFL-shallow. More degradation is seen for Kappa, with Kappa dropping to 0.27, still substantially better than chance, but about halfway between the PFL-shallow detector's performance and chance performance.

When the models built to predict PFL-shallow are applied to predict transfer-shallow, they again have very minimal degradation in terms of A', achieving an A' of 0.73, almost as good as a model trained to predict transfer-shallow. Relatively less degradation is seen for Kappa than was seen for the transfer-shallow detector, with Kappa only declining to 0.41.

Overall, these results show that the detectors have fairly good portability between transfer-shallow and PFL-shallow, suggesting that the two constructs are related in a substantial fashion.

CONCLUSIONS

Within this paper, we have presented models that can distinguish with reasonable effectiveness whether a student has acquired shallow learning, operationally defined as performing better on a test of the material learned in the tutor than on a test of the ability to transfer that skill for related problems or on a preparation for future learning (PFL) test. We present two sets of models – one for students who are shallow in terms of transfer, and another for students who are shallow in terms of PFL. We also use two distinct algorithms to build these models – step regression, which looks for cut-offs in linear patterns, and K*, which looks for local regions of density of one category or the other.

These models are fairly successful at distinguishing shallow learners, in terms of each operationalization of shallow. The best model of transfer-shallow learning can distinguish a shallow learner from a non-shallow learner 79% of the time, performance that is 53% better than chance. The best model of PFL-shallow learning can distinguish a shallow learner from a non-shallow learner 79% of the time, performance that is 55% better than chance. In both cases, K* algorithms performed better than step regression. Detectors trained using one operationalization of shallowness could predict the other operationalization with relatively minimal degradation, showing that there are commonalities between these constructs. As such, these models serve as evidence that it is possible to identify shallow learners during online learning, a type of model that could potentially be applied to a range of learning environments. Investigating the general applicability of this approach will be an important area for future research.

A range of features were used in these models, but centered around three types of behavior: slow actions, both non-off-task and off-task, very rapid actions, including gaming the system and very fast help requests, and help avoidance. These same types of features have been found to be correlated to robust learning (Baker, Gowda, & Corbett, 2011a, 2011b), providing further evidence that the types of meta-cognition involved in appropriate help-seeking are essential for robust learning, and that disengaged behaviors play an important role in avoiding shallow learning.

One of the principal uses of detectors such as the one presented here is to support more intelligent remediation. Students who have learned the exact skills taught in the tutor but who have not achieved robust learning are a group especially in need of remediation. Traditional student modeling methods are likely to fail to provide them any remediation, as they have learned the skills being taught by the tutor and can demonstrate that skill. A detector of shallow learning can identify these students and offer them remediation specific to their needs, helping them to build on their procedural knowledge to achieve the conceptual understanding necessary for future use of their knowledge. One can imagine a detector of shallow learning being used in two fashions. First, by identifying these students, the students can be tracked towards activities that improve their conceptual grounding, perhaps then returning to procedural practice to solidify the conceptual gains made. Second, specific behaviors can be identified and responded to by the learning software in real time. For example, if a student fails to pause and self-explain after seeking a hint, the software can provide meta-cognitive feedback on that decision (Roll et al., 2011). To give another example, a student who games the system can be given exercises that help the student learn the material bypassed by gaming (cf. Baker et al., 2006). can be asked to reflect on why the answer is correct after eventually providing the correct answer, or can be provided with an explanation as to why the answer is correct. Building this type of remediation into the Genetics Tutor will be a potentially important step for future work.

Thus, we view this detector as a second step – building on the first step of transfer and PFL detectors – 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 robust learning (cf. Butcher, 2010; Chin et al., 2010; Roll et al., 2007, 2011; Salden et al., 2008; Tan & Biswas, 2006).

Acknowledgements. This research was supported by the National Science Foundation via the grant "Empirical Research: Emerging Research: Robust and Efficient Learning: Modeling and Remediating Students' Domain Knowledge", award number DRL0910188.

.

REFERENCES

Aleven, V., & Koedinger, K.R. (2002). An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, 26, 147-179.

Aleven, V., McLaren, B.M., Roll, I., & Koedinger, K.R. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence and Education*, 16, 101-128.

Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R. (1995) Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences, 4* (2), 167-207.

Bahrick, H.P., Bahrick, L.E., Bahrick, A.S., & Bahrick, P.E. (1993). Maintenance of Foreign Language Vocabulary and the Spacing Effect. *Psychological Science*, 4(5), 316-321.

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., Koedinger, K.R., Evenson, S.E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., Beck, J. (2006) Adapting to When Students Game an Intelligent Tutoring System. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401.

Baker, R.S.J.d., Corbett, A.T., Roll, I., & Koedinger, K.R. (2008). Developing a generalizable system to detect when students game the system. *User Modeling and User-Adapted Interaction*, 18(3), 287-314.

Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T. (2011) Detecting Learning Moment-by-Moment. *International Journal of Artificial Intelligence in Education*, 21 (1-2), 5-25.

Baker, R.S.J.d., Gowda, S., & Corbett, A.T. (2011a). Towards predicting future transfer of learning. *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, 23-30.

Baker, R.S.J.d., Gowda, S., & Corbett, A.T. (2011b). Automatically Detecting a Student's Preparation for Future Learning: Help Use is Key. *Proceedings of the 4th International Conference on Educational Data Mining*, 179-188.

Ben-David, A. (2008). About the Relationship between ROC Curves and Cohen's Kappa. *Engineering Applications of Artificial Intelligence*, 21, 874-882.

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

Butcher, K.R. (2010). How Diagram Interaction Supports Learning: Evidence from Think Alouds During Intelligent Tutoring. In A. Goel, M. Jamnik & N.H. Narayanan (Eds.), *Diagrammatic Representation and Inference: Lecture notes in computer science,* vol. 6170 (pp. 295-297). Berlin/Heidelberg: Springer.

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the International Conference on Machine Learning*, 161-168. Pittsburgh, PA, USA.

Cen, H., Koedinger, K.R., Junker, B. (2007). Is Over Practice Necessary? Improving Learning Efficiency with the Cognitive Tutor. *Proceedings of the 13th International Conference on Artificial Intelligence and Education,* 511-518. Marina del Rey, CA, USA.

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.

Chi, M.T.H, Feltovich, P.J. & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 18, 439-477.

Chi, M., & VanLehn, K. (2007). Domain-Specific and Domain-Independent Interactive Behaviors in Andes. In R. Luckin, K. R. Koedinger & J. Greer (Eds.) *Proceedings of the International Conference on Artificial Intelligence in Education* (pp. 548-550). Amsterdam, Netherlands: IOS Press.

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.

Cleary, J.G., Trigg, L.E. (1995) K*: An Instance- based Learner Using an Entropic Distance Measure, *Proceedings of the 12th International Conference on Machine learning,* 108-114.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46.

Corbett, A. (2001). Cognitive computer tutors: Solving the two-sigma problem. *UM2001, User Modeling: Proceedings of the Eighth International Conference*, 137–147.

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. A. (2010). Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. *Journal of Educational Computing Research*, 42(2), 219-239.

Corbett, A., MacLaren, B., Wagner, A., Kauffman, L., Mitchell, A., Baker, R., & Gowda, S. (2011). Preparing students for effective explaining of worked examples in the Genetics Cognitive Tutor. In *Proceedings of the Thirty-third Annual Meeting of the Cognitive Science Society,* 1476-1481. Boston, MA, USA.

Corbett, A. T., & Trask, H. (2000). Instructional interventions in computer-based tutoring: differential impact on learning time and accuracy. In *Proceedings of the ACM CHI '2000 Conerence on Human Factors in Computing Systems*, 97-104.

Davis, J., Goadrich, M. (2006) The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning,* 233-240. Pittsburgh, PA, USA.

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

Fong, G. T., & Nisbett, R. E. (1991). Immediate and delayed transfer of training effects in statistical reasoning. *Journal of Experimental Psychology: General*, 120, 34-45.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H.H.,Ventura, M., Olney, A., & Louwerse, M.M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Reseach Methods, Instruments and Computers*, 36, 180-193.

Hanley, J., & McNeil, B. (1982). The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143, 29-36.

Hausmann, R., & VanLehn, K. (2007). Explaining Self-Explaining: A Contrast Between Content and Generation. *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 417-424.

Jastrzembski, T. S., Gluck, K. A., & Gunzelmann, G. (2006). Knowledge tracing and prediction of future trainee performance. In *Proceedings of the 2006 Interservice/Industry Training, Simulation, and Education Conference*, 1498-1508.

Katz, S., Connelly, J., & Wilson, C. (2007). Out of the lab and into the classroom: An evaluation of reflective dialogue in ANDES. In *Proceedings of the 2007 conference on Artificial Intelligence in Education* (pp. 425-432). Amsterdam, The Netherlands: IOS Press

Koedinger, K. R., & Corbett, A. T. (2006) Cognitive tutors: Technology bringing learning sciences to the classroom. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*, 61-77. New York, NY: Cambridge University Press

Koedinger, K.R., Corbett, A.T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) Framework: Toward Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science*, 36(5), 757-798.

Li, N., Cohen, W.W., & Koedinger, K.R. (2010). A computational model of accelerated future learning through feature recognition. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, 368-370.

Martin, J., & VanLehn, K. (1995). Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, 42, 575-591.

Mattler, E., Massey, C.M., & Kellman, P.J. (2011). Improving Adaptive Learning Technology through the Use of Response Times. *Proceedings of the Annual Meeting of the Cognitive Science Society,* 2532-2537.

McLaren, B.M., Lim, S., & Koedinger, K.R. (2008). When and how often should worked examples be given to students? New results and a summary of the current state of research. *Proc. of the 30th Annual Conf. of the Cognitive Science Society*, 2176-2181.

Pavlik, P.I., Presson, N., Dozzi, G., Wu, S-m., MacWhinney, B., Koedinger, K. (2007) The FaCT (Fact and Concept Training System): A new tool linking cognitive science with educators. *Proceedings of the 29th Annual Conference of the Cognitive Science Society,* 397-402.

Rittle-Johnson, B., & Siegler, R.S. (1998). The relation between conceptual and procedural knowledge in learning mathematics: A review. In C. Donlan (Ed.), *The development of mathematical skill* (pp. 75-110). Hove, UK: Psychology Press.

Roll, I., Aleven, V., McLaren, B.M., & Koedinger, K.R. (2007). Can help seeking be tutored? Searching for the secret sauce of metacongitive tutoring. *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 203-210.

Roll, I., Aleven, V., McLaren, B.M., & Koedinger, K.R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, 21, 267-280. Marina del Rey, CA, USA.

Salden, R. J. C. M., Aleven, V., Renkl, A., & Schwonke, R. (2008). Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support? *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, 589-594.

Schmidt, R.A., & Bjork, R.A. (1992). New conceptualizations of practice: common principles in three paradigms suggest new concepts for training. *Psychological Science*, 3(4), 207-217.

Schwartz, D.L., & Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction*, 22, 129-184.

Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computer in Human Behavior*, 25, 258–266.

Shih, B., Koedinger, K.R., & 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. (2006). The Role of Feedback in Preparation for Future Learning: A Case Study in Learning by Teaching Environments. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 370-381.