Looking for Sources of Error in Predicting Student's Knowledge

Mingyu Feng, Neil T. Heffernan, Kenneth R. Koedinger

Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA
mfeng@cs.wpi.edu, nth@wpi.edu, koedinger@cmu.edu

Abstract

Recent research has focused on detecting the "gaming" behavior of students while using an intelligent tutoring system. For instance, Baker, Corbett & Koedinger (2004) reported detecting "gaming" by students, and argued that it explained lower learning results for these students. In this paper, we report that while our computer system's correlation with a student's actual state test score is well correlated (r=.7), we found that we were systematically under-predicting their scores. We wondered if that under-prediction had to do with students engaging in some form of gaming. In this paper, we look to see if some of the online metrics (e.g. rate of asking for hints) Baker et al reported correlated with our under-prediction of student's scores. We report results from the Assistment Project's data set of about 70 students collected from May, 2004. We performed a stepwise regression to predict what metrics help to explain our poor prediction of state exam scores. We conclude that while none of the metrics we used were statistically significant, several of the metrics were correlated with our under-prediction, suggesting that there is information in these signals but that it might be too weak for the small sample size we have. For future work, we need to replicate this method with the dataset we are collecting this year that has 600 students using the system for 10 times as long.

Introduction

As part of the "Assistment" Project (Razzaq et al, 2005), so named because we blend assessment reporting to teachers with instructional assistance to student (i.e., tutoring), we have developed a reporting system (Feng, & Heffernan 2005) that gives teachers live feedback about how their students are doing as the students are working individually on their own computers. Figure 1 shows part of the report called "Grade book", in which each line shows information about one student. Currently, we can tell teachers about how many problems their students have done, students' percent correct and their predicted MCAS (the state 8th grade math test in Massachusetts is called the MCAS, which is a graduation requirement which all students educated with public funds in the tested grades are required to participate) score. Our teachers like to be able to hit the "Refresh" button and get instantaneous feedback. Our teachers will notice that a student is asking for too many hints [typically questions have 2-5 hints, with the final hint telling the students exactly what to do] and will call the students over to confront them with the evidence by showing them log transcripts that might show where they did not even bother to read a hint and instead asked for a more detailed hint. While our teachers like our system, sometimes students should be asking for lots of hints if they don't know a topic, so our teachers are requesting a column to be added to this report that gives a score on their seriousness of effort (which we think of as the opposite of a gaming index). If we can quickly tell teachers who is gaming, then they can go speak to the student. The other problem Figure 1 brings up is, as we report in this paper, these reports are under-predicting students' MCAS scores. Some of this under-prediction is probably due to students gaming and or guessing behavior. We would like to be able to correct our predictions of their MCAS score to be more robust even in the presence of students' occasional "gaming".

Therefore, while some intelligent tutoring systems show impressive results in both student learning (Morgan and Ritter, 2002) and motivation (Schofield, 1995) some research coming out now is focusing on detecting this when a student's motivation is flagging. Recently Arroyo, Murray and Woolf (2004) have reported their attempts to diagnose flagging motivation, as well as their attempts to respond to this. An early ITS, by del Soldato and du Boulay (1995), asked students to self-report their motivation, which the system used to do problem selection differently based upon their motivational state. Others, such as de Vicente and Pain (2003), have developed models that classify students' motivation state into many fine-grained categories. For this work, we will only think about the grossest of models of motivation, where students are scored on their gamingness along a single continuum. Closest to our work is work done by Baker, Corbett and Koedinger (2004) who studied gaming in an intelligent tutoring system that is also similar to our system. This paper attempts to replicate some of the results from Baker et al. Baker, Corbett and Koedinger looked to see what online metrics could be correlated with gaming as indicated by classroom observations. Baker had classroom

This research was made possible by the US Dept of Education, Institute of Education Science, "Effective Mathematics Education Research" program grant #R305K03140, the Office of Naval Research grant # N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation.

observation data on two class periods for 70 students that first asked to attempt the item with no assistance. Only if

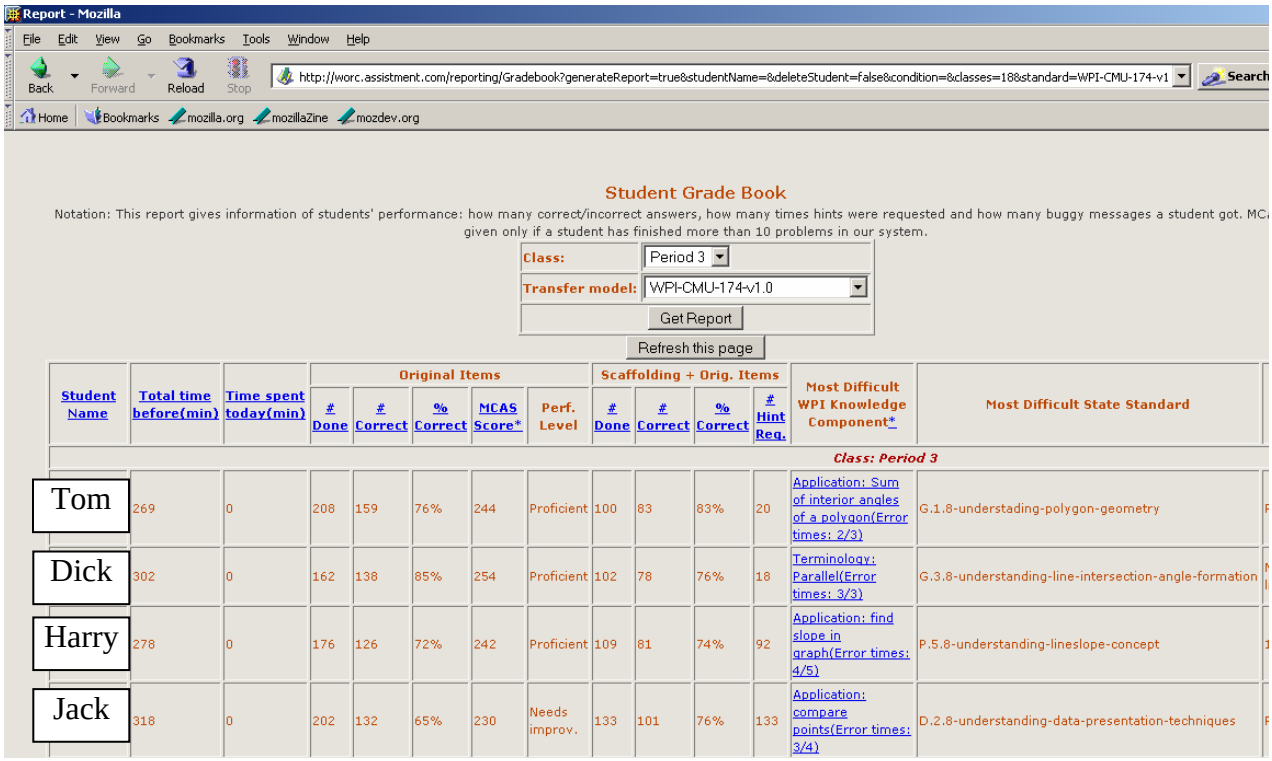

Figure 1: Part of the Grade book report for one of our teachers' class

indicated which students seemed to exhibit gaming behavior.

For this paper, we are not using classroom observation data like Baker et al did (we have found the systemic collection of this more difficult than we had anticipated.) Instead, we arrive at this problem by assuming that some gaming is happening, and further assume that students that were gaming would be correlated with the degree of our system's under-prediction.[1].

So our research plan was to develop a series of metrics, [that were similar but not identical to Baker et al.] and to see if we could predict gaming by predicting when our system was drastically under-predicting students' real scores.

Methods

Data Sources

Though similar, the Assistment System differs from many of the Carnegie Mellon related tutors in that for each original item (which is generally the text of a specified MCAS test item) presented to the student, the student is the student makes an error, or asks for a hint, do we provide 2-5 "scaffolding" questions that attempt to walk the students through the problem. Some students who game seem to try to hurry through the item by either 1) asking for hints until they reach the "bottom out hint" that will tell them exactly what to type, or 2) to guess on the item (which Baker said seemed to be relevant particularly for multiple choice items). Figure 2 shows an Assistment we built for item 19 from the year 2003 MCAS. In this case, the original question has been broken into 5 scaffolding questions. The scaffolding questions appear only if the student gets the original item wrong. Figure 2 shows that the student typed "23" (which happened to be the most common wrong answer for this item from the data we have collected). After an error, students are not allowed to try the item further, but instead must now answer a sequence of scaffolding questions (or "scaffolds") presented one at a time[2]. Students work through the scaffolding questions, possibly with hints, until they eventually get the problem correct. If the student presses the hint button while on the first scaffold, the first hint is displayed, which is the definition of congruence in this example. If the student hits the hint button again, the hint

[1] Baker hypothesizes the many different reasons that students might be "gaming". We will not discuss the reasons in this paper but instead just assume it is some form of laziness in which guessing or asking for a hint is easier than making an earnest attempt to answer the problem.

[2] As future work, once we have built a predictive model and are able to reliably detect students trying to "game the system" (e.g., just clicking on an answer without reading the question) we may allow students to re-try a question if they do not seem to be "gaming". Thus, studious students may be given more flexibility.

that is shown in upper right corner of Figure 2 appears, which describes how to apply congruence to this problem. If the student asks for another hint, the answer is given. Once the student gets the first scaffolding question correct (by typing "AC"), the second scaffolding question appears. If the student selects ½ * 8x when confronted with the second scaffold, the *buggy message* shown would appear, suggesting that it is not necessary to calculate area. (*Hints* appear on demand, while *buggy messages* are responses to a particular student error). Once the student gets the second question correct, the third appears, and so on. Figure 2 shows the state of the interface when the student is done with the problem as well as a hint for the 4th scaffolding question.

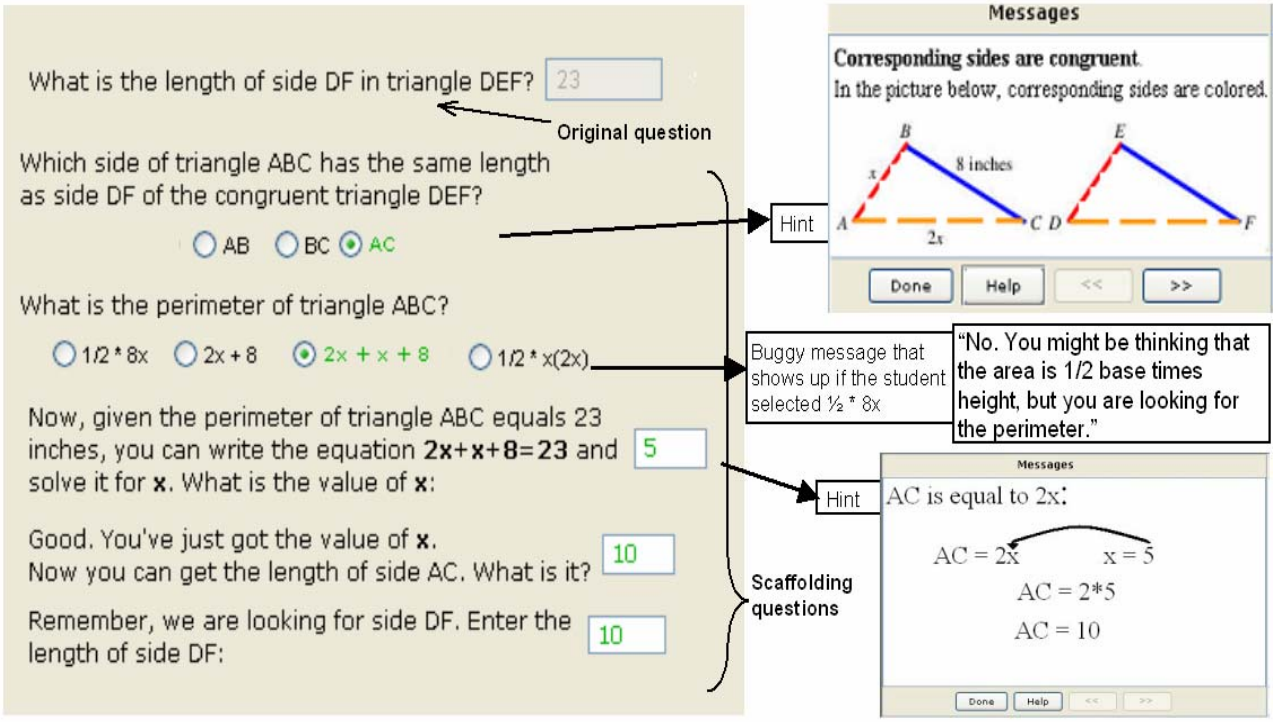

Figure 2: An Assistment shown just before the student hits the "done" bottom, showing two different hints and one buggy message that can occur at different points.

The Assistment items are different than most other CMU related tutors that tend to present the student with complicated workspace/tools up front with which to use to answer the problem. Because the Assistment system always asks the student to first attempt the original question we can use their percent correct to calculate a predicted MCAS score by multiplying their percent correct by the total number of points available on an MCAS score, which is 54 points.[3] To be clear, we are predicting their MCAS scores by looking to see if a student gets the original question correct, and ignoring how they do on the "scaffolding" questions. However, many of the online metrics we propose try to garner information from the scaffolding questions, as well.

Our data set for this paper consisted of 68 students who used our system in May of 2004 and for whom we later received real state test scores. Each of these students used our system for a mean length of time of 40 minutes each, doing on average 10 original MCAS test items and each of them had finished at least 5 items. If a student got an item wrong, they spent on average 3 minutes to answer the scaffolding questions (i.e., they got tutored). To calculate a student's raw MCAS score, which ranges from 0 to 54 points, the state adds one point for each of the 29 multiple choice items, one point for each of the short answer questions and up to 4 points for each of the approximately 5 human scored open-response (i.e., essay) questions. We calculated what we call a student's predicted raw MCAS score (referred to as **PredictedMCASScore** later) by taking their percent correct on the original items (referred to as **%CorrectOnOriginal** later) and multiplying a student's average percent correct by 54. We found that on average students predicted scores were 9 points lower than their real MCAS score.

We then calculated for each student the students' **DiffScore** by subtracting a student's predicted MCAS scores from his or her actual MCAS score. Large positive numbers indicated students whose performance we drastically underestimated. To review, we suspect that this underestimate is partially due to students adopting

[3] It turns out a student's MCAS scale score is simply a function of the items the student got correct. (2002 MCAS Technical Report, p33-p34).

undesirable gaming behavior. We wanted to find online metrics that would predict the students "difference" scores. We will now describe the metrics we created that we hypothesized will be correlated with gaming. We associated the first two metrics with good behavior (labeled "Good"), while metrics 3-8 are marked as Bad, based upon the hypothesis that they would be positively correlated with gaming. Baker suggested 24 online features while here we calculate 9 metrics, which we hypothesize would be closely correlated to student gaming. These metrics were:

1. AttemptAfterHint – Good – How many times a student made an attempt after asking for a hint (not bottom-level hint which almost always reveals the correct answer to ensure that students can finish a whole assignment and won't get stuck on one problem). We encourage students to ask for help when they get stuck on a question, try to learn from the given hint messages and attempt again. Thus students, who made attempts after reading hint messages instead of asking for the next hint, are anticipated to be more serious learners, while student with lower scores are more likely to be gaming.
2. ReadingHint – Good – How many times a student spent time reading (or thinking about) a hint message. Students who spend a very short period of time reading hints are probably those students that are just clicking on the "More" button to get to the bottom out hint.
3. HintBeforeAttempt – Bad – How many times a student asked for hint even before giving any answers. There are reasons why students would like to ask for a hint. It could be that the problem is so hard for the student that he didn't have any idea how to solve it; or it could be that the question text is long, so the student won't bother reading the question; or the student was just lazy.
4. FastConsecutiveAttempt – Bad – How many times a student made quick actions after giving an answer (intervals between consecutive attempts are less than 4 seconds). We noticed that students tend to try "FastConsecutiveAttempt" if the question is hard and shown as a text field. Some students would rather quickly try all choices of a multiple choice questions instead of seeking a solution on their own or asking for hints. We assume a student is less likely to be gaming if he slows down during subsequent actions after making an error, but it could be that good students often can quickly correct their answers so the directness of this metric is not clear
5. ConsecutiveBottomHint – Bad – How many times a student went to the bottom level hint without making any attempts before reaching it. As mentioned above, the bottom level hint almost always reveals the correct answer. Students who figured this out, or had been told about this trick, could just go all the way down to get the bottom level hints and finish the current problem.
6. WeightedHintCount – Bad – How many hints a student received while working on the Assistments. Given that some questions had anywhere from 1-5 hints we wanted to scale this number into a range of zero to 1. If a student received the bottom hint they got a 1. If three hints were possible and they requested to see the second hint, they got 2/3. If they never asked for a hint they got a zero for that question. The final metric was divided by the total number of original and scaffolding questions.
7. QuickDoneWithError – Bad – We hypothesize that those students that were quickly answering were more likely to be guessing. Of course, some items take longer than others on average so we did this analysis on a per original question basis. We also did what is standard in psychology and calculated, only for those students that got the item correct, the average time it took. We then decided quick students were those that answered in less than 20% of the time it normally took. It is known to researchers that work on Item Response Theory for Computer Adaptive Testing (e.g., Wise & Kong) that students who answer problems very quickly tend to be guessing.
8. AverageAttempts – Bad – If a student gets an original item incorrect, we then counted the total number of attempts the student had to make before finishing the item. We then average the number of attempts by dividing by the number of original items. This metric is a little suspicious since if you were unlucky and got items wrong that had many scaffolding questions even if you got every scaffolding question correct, you would have a higher number of attempts then a student who got an item wrong that had say only two scaffolding questions and got them both correct on her first attempt.
9. %CorrectOnOriginal – Neutral – This metric differs from all the rest and was only included so that we could look at the interaction of this metric with the other metrics. Ideally, it would be nice to have had a probability that a student would get that particular skill correct (Baker et al had reasonable predictions for each skill from the knowledge tracing algorithm) but we were assessing all of 8th grade math so could not reliably predict finer grained skills given the small amount of time relative to the large number of skills. Instead, we used the students' percent correct on the original questions as a surrogate for their overall knowledge.

It should be noted that we calculated these metrics on a question-by-question basis and the first six metrics are averaged over the sum of the original questions and scaffolding questions. The 7th, 8th and 9th metrics apply

only to the original question, and therefore are averaged over the number of original questions answered and they do not apply to scaffolding questions. Following Baker et al, we planned to use these 9 metrics and do a stepwise linear regression where we added all the factors as well as quadratic terms for each metric and all two by two interactions.

Data Analysis and Modeling

To begin our analysis, we had 68 students who had done at least 5 Assistments (we would not expect a great fit with their MCAS scores since the MCAS test has many more items, i.e., 39 items per test).

The first thing we did was to see if our computer based score (i.e., PredictedMCASScore) is well correlated with their real MCAS raw score. We found a strong correlation (r = 0.7) but we also saw that we were under-predicting many students as shown in Figure 3 with a oval around a group of students that were poorly fit. We want to find out what online metrics are correlated with our under-prediction. So we subtracted the computer based score from their real MCAS score to get Diff score (we will refer to this score simply as DiffScore) and our overall goal is to figure out when we are doing a poor job of predicting MCAS scores.

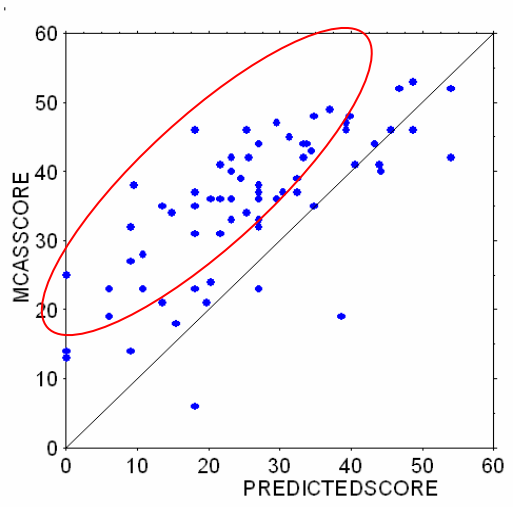

Figure 3: Bivariate Scattergram for PredictedMCASScore and real MCAS raw score

Given that some students had done only 5 items while others had done 35 items, we suspected there might be a relationship between DiffScore and the number of items done. However, after doing a linear regression, we found there was not much correlation given that the number of items done was a poor predictor of DiffScore (R^2 = .003)

So our intent was to simply build a model to predict DiffScore based upon those online metrics mentioned before. However we noticed that after we did a simple linear regression on the computer-based predicted score to predict DiffScore, we saw that lower performing students had higher DiffScores (see Figure 4, R = .59 and p <0.0001)

Therefore, it was necessary to modify our goal and instead try to predict DiffScore *after* taking into account that lower scoring students would have higher DiffScore ranking. We will call this term ModifiedDiff (modified in that we took into account that lower scoring students

Regression Summary
DIFF vs. PREDICTEDSCORE

Count	68
Num. Missing	0
R	.593
R Squared	.352
Adjusted R Squared	.342
RMS Residual	7.447

Regression Coefficients
DIFF vs. PREDICTEDSCORE
Row exclusion: FinalModelStepwise.svd

	Coefficient	Std. Error	Std. Coeff.	t-Value	P-Value
Intercept	27.019	1.309	27.019	20.645	<.0001
PREDICTEDSCORE	-.475	.046	-.869	-10.366	<.0001

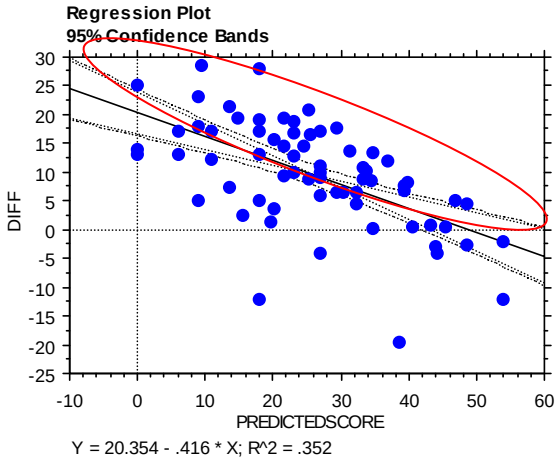

Figure 4: Result of simple regression to predict DiffScore using PredictedMCASScore

would have a high DiffScore on average.) Strictly speaking, ModifiedDiff is the residual after having done a linear regression to predict DiffScore using predicted score. Furthermore, because we assume gaming is more likely to result in under-prediction of state test scores, we want to focus our attention on those students that are being under-predicted, which is the group of 38 students in the oval above the regression line.

So we looked to see what online metrics were correlated with ModifiedDiff. Our planned analysis was to do a forward stepwise regression (using a "F-to-Enter" value of 4) and consider all 9 metrics, as well as quadratic terms for each metric, as well as all two by two interactions, giving us 54 (9+9+36) factors to consider adding to the model.

Table 1: Correlation of 9 Metrics with ModifiedDiff and DiffScore

Metric	Expected Effect	Correlation with ModifiedDiff	Correlation with DiffScore
AvgAttempts	Bad	.25	.62
QuickDoneWithError	Bad	.20	.33
AttemptAfterHint	Good	**.23**	.25
ReadingHint	Good	-.11	.21
HintBeforeAttempt	Bad	**-.05**	.14
FastConsecutiveAttempt	Bad	.06	.02
WeightedHintCount	Bad	**-.18**	.18
ConsecutiveBottomHint	Bad	**-.14**	-.07

Results

To summarize the results of this model, we need to remind ourselves that we are looking for factors that predict the ModifiedDiff for the students who were being under predicted more than expected value (even after we took into account that the percent correct was a predictor of DiffScore). The result of this Stepwise regression was that NO factors were considerably statistically significant enough to explain ModfiedDiff to be added to the model. So instead we report just the correlations between each of the 8 metrics and the ModifiedDiff and original DiffScore. See table 1.

We observed that the correlations were rather small, and 4 of them in bold were in the opposite direction to what we had hypothesized. For instance, AttemptAfterHint, which is the percent of time students did not ask for a second hint and instead attempted the item, was something that we thought would be associated with "good" students, and thus should be negatively correlated with ModifiedDiff, was positively correlated here.

AvgAttempts was the best correlated metric, suggesting that students that make more attempts are students that have higher ModifiedDiff. We were curious to know if we were being too strict in ensuring statistically significant so we did another forward stepwise regression, but one in which we lower the threshold for including factors into the model (F-to-Enter = 3). When we did this many of our interaction and quadratic terms showed up but were considered overfitting. So we dropped all factors except the original nine online metrics to see how big, in practical terms, the effect of those metrics were. The regression went one step and the result is shown in Figure 5 in which Residual DIFF refers to what we called ModifiedDiff here. As we have expected, AvgAttempts is the only one entering the model while the other metrics were not selected into the model. The coefficient on AvgAttempts would lead one to conclude that if a student were to instead make one more attempt per item than they had really done, our ModifiedDiff prediction would go up .4 points.

Stepwise Regression Summary
Residual DIFF vs. 9 Independents
Row exclusion: FinalModelStepwise.svd

F-to-Enter	3.000
F-to-Remove	3.000
Number of Steps	1
Variables Entered	1
Variables Forced	0
Stepwise Procedure	Forward

Variables Not In Model
Residual DIFF vs. 9 Independents
Step: 0
Row exclusion: FinalModelStepwise.svd

	Partial Cor.	F-to-Enter
PREDICTEDSCORE	-.214	1.675
AVGATTEMPTS	.286	3.127
ATTEMPTAFTERHINT	.244	2.208
READINGHINT	-.110	.430
HINTBEFOREATTEMPT	-.091	.290
QuickAttempt/Scaffod	.015	.008
QUICKDONEWRONG/item	.243	2.192
WEIGHTEDHINTCOUNT	-.189	1.296
CONSECUTIVEBOTTOMHINT/scaffold	-.180	1.176

Variables In Model
Residual DIFF vs. 9 Independents
Step: 1
Row exclusion: FinalModelStepwise.svd

	Coefficient	Std. Error	Std. Coeff.	F-to-Remove
Intercept	3.183	1.218	3.183	6.834
AVGATTEMPTS	.440	.249	.286	3.127

Figure 5: Result of stepwise regression to predict ModifiedDiff using 9 online metrics

Summary and Discussion

Fundamentally, we have a null result which leaves us merely speculating as to why.

One area to look is the metrics themselves. For instance, ReadingHint, QuickDoneWithError and FastConsecutiveAttempt are the three metrics that have parameters to set to try to determine what constitutes a students being quick or slow. It could be that we have chosen poorly in setting these parameters. For instance the ReadingHint metric was true if students spent more than 10 seconds reading the hint. Two ideas for suggestions are

that that number might be too high, and furthermore, some hints are longer than others, so we should expect to see difference between difference questions

We speculate that our two major problems are 1) our metrics might not be as informative as we would hope and 2) that we need a strong signal to learn from, and feel that using classroom observation data to be a much stronger signal.

Limitations and Future Work

There are many more issues to deal with, and we believe we are just touching the tip of the iceberg in using online metrics. We will want to err on the side of not blaming students for gaming so further work will need to be done to minimize the errors of accusing someone of gaming when they are not.

One of the weaknesses of our approach is that we did not have a strong model of the difficulty of each item while it seems that students more likely game with hard or easy but calculation intensive problems. We have evidence of this with this current year's data because we have students that have done hundreds of items but have a relatively smaller number of skills. Another weakness to our approach is that we do not take into account the fact that the subject should be learning as they are proceeding, and having a strong cognitive model will help make this possible. Yet a third weakness in our approach is that we assume gaming independently with respect to time, but it seems likely that students game in streaks, so we will want to look at this by adding a metric that tries to track over periods of time. As future work we are also working on collecting classroom observation data on gaming similar to the way Baker et al did.

Another limitation is that this method tells us who is gaming, but does not identify the particular actions that the system thinks are gaming actions. It would take some effort to present this sort of data to teachers so that they can best confront their students with this data.

To further refine our models it would be helpful to collect classroom observations of gaming behaviors. Also another factor that we have left out that should be factored into determining an effort score is whether students are showing evidence of learning.

In conclusion, we can neither replicate nor contradict Baker et al's findings. Nevertheless we have replaced Baker et al's method, and plan to continue to try to use it to detect gaming. We conclude that we are cautiously optimistic that these techniques described above should help us 1) in meeting our teachers goals of giving them an easy way to flag students that might be gaming, as well as 2) helping us correct our predictions of MCAS scores to take into account that some students are not trying as hard as we might like. We are planning on implementing this method into the teacher's "Grade Book" report that we showed at the beginning of this paper to see if teachers agree with the system predictions.

Reference

Arroyo, I., Murray, T., Woolf, B. P., Beal, C. R. (2004). Interring Unobservable Learning Variables from Students Help Seeking Behavior. James C. Lester, Rosa Maria Vicari, Fábio Paraguaçu (Eds.): *Intelligent Tutoring Systems*, 7th International Conference, ITS 2004, Maceiò, Alagoas, Brazil, Proceedings.

Baker, R.S., Corbett, A.T., Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 531-540.

Baker, R.S., Designing Intelligent Tutors That Adapt to Student Misuse, Ph.D. dissertation proposal

Cognitive tutors (see http://pact.cs.cmu.edu/)

Feng, M., Heffernan, N.T., (2005). Informing Teachers Live about Student Learning: Reporting in the Assistment System. *Presentation at 2005 AERA Annual Meeting*. (http://www.cs.wpi.edu/~mfeng/pub/AERA-Reporting-Scandura.pdf)

Fogarty, J., Baker, R., Hudson, S. (to appear) Case Studies in the use of ROC Curve Analysis for Sensor-Based Estimates in Human Computer Interaction. *To appear at Graphics Interface 2005*

de Vicente, A., Pain, H. (2003) Validating the Detection of a Student's Motivational State. In A. Mendez Vilas, J. A. Mesa Gonzalez, J. Mesa Gonzalez, editors, *Proceedings of the Second International Conference on Multimedia Information & Communication Technologies in Education (m-ICTE2003)*, number 15 of *"Sociedad de la Informacion" Series*, Volume III, pages 2004-2008, Merida, Junta de Extremadura

del Soldato, T. & du Boulay, B. (1995). Implementations of motivational tactics in tutoring systems. *Journal of Artificial Intelligence in Education*, 6 (4), 337-378.

Razzaq, L, Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T., Upalekar, R., Walonoski, J., Macasek, M., Rasmussen, K., Koedinger, K., Junker, B., Knight, A., Ritter, S. (2005). The Assistment Project: Blending Assessment and Assisting. *Submitted to the 12th Annual Conference on Artificial Intelligence in Education* 2005, Amsterdam

Schofield, J., 1995, *Computers and Classroom Culture*, Cambridge University Press, USA.

Wise, S. L., & Kong, X. (2004). Response time effort: A new measure of examinee motivation in computer-based tests. Manuscript submitted for publication.

Morgan, P., & Ritter, S.(2002). An experimental study of the effects of Cognitive Tutor Algebra Ion student knowledge and attitude. (Available from Carnegie Learning, Inc., 1200 Penn Avenue, Suite 150, Pittsburgh, PA 15222)