Generalizing Detection of Gaming the System Across a Tutoring Curriculum

Ryan S.J.d. Baker[1], Albert T. Corbett[2], Kenneth R. Koedinger[2], Ido Roll[2]

[1] Learning Sciences Research Institute, University of Nottingham, Nottingham, UK
Ryan.Baker@nottingham.ac.uk
[2] Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
{corbett, koedinger, iroll}@cmu.edu

Abstract. In recent years, a number of systems have been developed to detect differences in how students choose to use intelligent tutoring systems, and the attitudes and goals which underlie these decisions. These systems, when trained using data from human observations and questionnaires, can detect specific behaviors and attitudes with high accuracy. However, such data is time-consuming to collect, especially across an entire tutor curriculum. Therefore, to deploy a detector of behaviors or attitudes across an entire tutor curriculum, the detector must be able to transfer to a new tutor lesson without being re-trained using data from that lesson. In this paper, we present evidence that detectors of gaming the system can transfer to new lessons without re-training, and that training detectors with data from multiple lessons improves generalization, beyond just the gains from training with additional data.

1 Introduction

Developing models that can reliably detect differences in how students choose to use intelligent tutoring systems, and the attitudes and goals which underlie these decisions, has received considerable attention in recent years [1,3,4,7,8]. A number of models have been developed which can reliably detect specific student behaviors – from avoiding help [cf. 1], to gaming the system [4], to competing with other students [7]. These models have supported the development of systems that influence students to learn to use intelligent tutoring systems more effectively [2].

However, to be widely useful, detectors of student behaviors and motivation need to be generalizable. Thus far, most such detectors have been developed using data from individual lessons from a tutoring curriculum, or from fairly small-scale intelligent tutors. However, intelligent tutors are increasingly being used as major components in year-long curricula. A model of help-seeking behavior developed using only log file data has been shown to generalize effectively across lessons [11], but many of the models developed to detect student behaviors and attitudes have been trained using additional data such as human observations [4,8], improving accuracy [11]. Unfortunately, human observations are time-consuming to collect for an entire

year-long curriculum. Therefore, to be maximally useful – and used – detectors of behaviors and motivation need to be able to take advantage of observational data, while generalizing to new tutor lessons without the collection of additional data.

In this paper, we will discuss our work to generalize a behavior detector which detects whether a student is "gaming the system", attempting to succeed in an educational environment by exploiting properties of the system rather than by learning the material and trying to use that knowledge to answer correctly [6]. Within the set of intelligent tutor lessons that we will discuss in this paper, gaming behavior consists of systematic guessing and rapid-fire hint requests. Prior analyses have also found that gaming can be divided into two distinct categories of behavior: harmful gaming, which is associated with poor learning outcomes and appears to occur on the problem steps the student knows least well, and non-harmful gaming, which is not associated with poor learning outcomes and appears to occur on problem steps the student already knows [4].

Additionally, we will consider the question of what data is most useful for developing generalizable detectors. A considerable amount of machine learning research treats generalizability largely as a function of the sheer amount of data trained on, and the degree to which the training technique over-fits to that data. In this paper, we examine whether additional advantage can be gained by collecting a broader, more heterogeneous data set – in specific, presenting analyses suggesting that training on data from multiple tutor lessons improves generalizability more than would occur simply from increasing the sample size.

2 Methods

2.1 Data Sources

The first gaming detector [4] was developed using data from a tutor lesson on scatterplots, drawn from a middle-school Cognitive Tutor mathematics curriculum. In order to study issues of generalizability, we collected data from three additional lessons (on geometry, percents, and probability) from the same tutoring curriculum. All data came from classes in two school districts in suburban Pittsburgh. For the scatterplot lesson, we had data from classes in 2003, 2004, and 2005. For each of the

Table 1. Quantity of data obtained for each tutor lesson

Lesson	Number of students	Number of actions
SCATTERPLOT	268	71,236
PERCENTS	53	16,196
GEOMETRY	111	30,696
PROBABILITY	41	10,759

other lessons, we had data from a single year (2004 for geometry and probability, 2005 for percents). In total, we had data from 300 students (with 113 students represented in multiple lessons), with 128,887 actions across the 473 student/lesson pairs. Each student completed between 50 and 500 actions in the tutor.

For each lesson, we collected quantitative field observations (using the method in [6]), to estimate what percentage of time each student gamed the system. Pre-tests and post-tests were given for each lesson – in all cases, test items were counterbalanced across the pre-test and post-test. Data on learning gains enabled us to distinguish between harmful gaming and non-harmful gaming [cf. 4], both during training and when evaluating goodness-of-fit. In our analyses, we will refer to students who engaged in harmful gaming as "GAMED-HURT", and students who engaged in non-harmful gaming as "GAMED-NOT-HURT".

Finally, we obtained logs of each student's actions within the tutor. For each student action recorded in the logs, we distilled a set of 26 features (listed in [4 and 5]) describing that action, including information about the action itself (time taken, type of interface widget) and the action's historical context (for instance, how many errors the student had made on the same skill in past problems).

2.2 Modeling Framework

Using this combination of data, we trained a set of detectors to predict how frequently an arbitrary student gamed the system. Each detector of gaming, within our framework, is a hierarchical Latent Response Model [10] with one observable level and two hidden ("latent") levels. In a gaming detector's outermost/observable layer, the detector predicts how frequently each student is gaming the system, labeling these predictions $G'_0 \ldots G'_n$. These predictions can then be compared to the

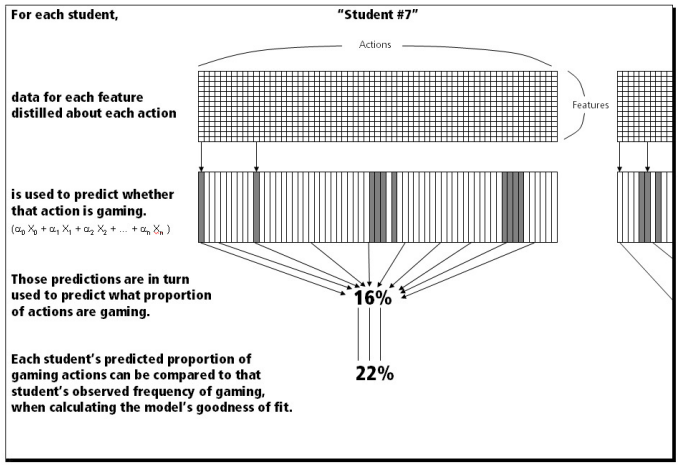

Fig. 1. The Gaming Detector

observed proportions of time each student spent gaming the system, $G_0 \ldots G_n$ (the metrics used will be discussed momentarily). The middle layer consists of a set of binary predictions as to whether each individual student action (denoted P'_m) is an instance of gaming. The observable predictions $G'_0 \ldots G'_n$ are derived by taking the percentage of actions which are predicted to be instances of gaming, for each student. The innermost layer is a function on features drawn from each action's characteristics, which are used to make the binary predictions in the middle layer. Each parameter in a model of gaming is either a linear effect on a feature (a parameter value α_i multiplied by the corresponding feature value $X_i - \alpha_i X_i$), a quadratic effect (parameter value α_i multiplied by feature value X_i, squared – $\alpha_i X_i^2$), or an interaction effect on two features (parameter value α_i multiplied by feature value X_i, multiplied by feature value $X_j - \alpha_i X_i X_j$).

A prediction P_m (in the innermost layer) as to whether action m is an instance of gaming is computed as $P_m = \alpha_0 X_0 + \alpha_1 X_1 + \alpha_2 X_2 + \ldots + \alpha_n X_n$, where α_i is a parameter value and X_i is the data value for the corresponding feature, for this action, in the log files. Each prediction P_m is then thresholded using a step function to form the binary predictions that form the middle layer, such that if $P_m \leq 0.5$, $P'_m = 0$, otherwise $P'_m = 1$. This gives us a set of classifications P'_m for each action within the tutor, which are then used to create the predictions of each student's proportion of gaming, $G'_0 \ldots G'_n$ which are compared to their observed frequency of gaming.

2.3 Detector Selection

Detectors are trained as follows: First, a set of single-parameter detectors are selected (using Fast Correlation-Based Filtering [13]) such that each single-parameter gaming detector is at least 60% as good as the best single-parameter detector found (in terms of linear correlation to the observed data). If two parameters have a closer correlation than 0.7 to each other, only the better-fitting single-parameter detector is used. Then, for each single-parameter detector, we repeatedly add the parameter that most improves the linear correlation between the detector's predictions and the original data, using Iterative Gradient Descent to find the best value for each candidate parameter. Generally, when selecting detectors, we continue adding parameters until the most recent parameter worsens the model's fit under Leave-One-Out-Cross-Validation (LOOCV); however, for the analyses in this paper, we stopped when a detector had six parameters, for tractability in training a large number of detectors. Generally, the detectors had very little absolute improvement in fit after the first three or four parameters, regardless of the results of LOOCV. This process resulted in a set of detectors with comparable correlation, from which the model with the best A' [1] is selected (averaging A' across the model's ability to distinguish GAMED-HURT students from non-gamers, and the model's ability to distinguish GAMED-HURT students from GAMED-NOT-HURT students).

[1] A' is both the area under the ROC curve, and the probability that the detector can successfully distinguish between an arbitrary student from each of the two groups being classified.

3 Detector Comparisons

3.1 Statistical Techniques for Detector Comparison

In the remainder of this paper, we will investigate how well gaming detectors transfer across different tutor lessons, examining detectors trained on single lessons, detectors trained on multiple lessons (but not all lessons), and a detector trained on all available lessons. Conducting these comparisons in a statistically appropriate fashion requires meta-analytic techniques, which we discuss in this section.

When comparing detectors to one another across multiple test lessons, the data from different test lessons cannot simply be collapsed into a single data set, since this will bias towards detectors that do best on the lesson with the most data; additionally, since gaming may occur with different frequency in different lessons, the A' value of a combined data set will be substantially lower than the A' values of the individual data sets, underestimating all detectors' effectiveness. Hence, we will in all cases determine our measures of interest for each test lesson individually, compare the detectors to each other within each test lesson, and then use meta-analytic techniques to combine these comparisons into a single statistical comparison.

In the analyses to follow, we will compare detectors to each other in terms of their A' and correlation. In order to use common meta-analytic techniques, we will convert these metrics to Z-scores. Two A' values can be compared to each other, giving a Z-score as the result, by using the standard Z-score formula in combination with Hanley and McNeil's technique for estimating the variance of an A' value [9]. Correlations can be compared to each other, giving a Z-score, by converting the correlations to Z-scores via the Fisher Zr transformation [12], and then comparing those Z-scores to one another.

Once all values are Z-scores, comparisons between results from different test lessons (for example, to estimate whether a detector performs significantly better than chance, across multiple test lessons) will be made using Stouffer's method [12] and denoted Z_s. Comparisons between results within the same test lesson (for example, to compare two detectors to each other) will be made using the mean Z-score method [12] and denoted Z_m. Comparisons of multiple detectors (such as the set of detectors trained using data from three lessons) across multiple test sets will be denoted Z_{ms}. In these cases, all within-lesson comparisons will be made before any between-lesson comparisons, in order to avoid comparing Z-scores estimated with methods which have different assumptions to each other. Z-scores derived without meta-analytic aggregations or comparisons will be denoted Z.

3.2 Transferring Models Trained on a Single Lesson

We begin our analysis by investigating how well a detector trained on a single tutor lesson will transfer to other tutor lessons. We trained four detectors – one on each of

the four lessons. We then tested how well each detector detected gaming within its training lesson, and within each of the 3 other lessons.

The four detectors trained on a single lesson had an average A' of 0.86, in the training lessons, at distinguishing GAMED-HURT students from non-gamers, significantly better than chance, Z_s=10.74, p<0.001. The detectors were significantly worse at making this same distinction in the transfer lessons (A' =0.71), Z_{ms} =3.63, p<0.001, though their performance in the transfer lessons was still better than chance, Z_m = 2.12, p=0.03. The detectors had an average A' of 0.79, in the training lessons, at distinguishing GAMED-HURT students from GAMED-NOT-HURT students, significantly better than chance, Z_s =5.07, p<0.001. The detectors were not significantly worse at making this distinction in the transfer lessons (A' =0.74), Z_{ms} =0.56, p=0.58, and were significantly better than chance, Z_m =2.86, p<0.01. The detectors had an average correlation of 0.57 between the observed and predicted frequencies of harmful gaming, in the training lessons, significantly better than chance, Z_s = 12.08, p<0.001. The detectors were significantly worse at making this same distinction in the transfer lessons (r=0.22), Z_{ms} =5.15, p<0.001, though their performance in the transfer lessons was still better than chance, Z_m =2.40, p=0.02.

Hence, a detector trained on one lesson performs significantly better than chance when transferred to other lessons. However, there is a significant and substantial drop in performance from training lessons to transfer lessons, on 2 of the 3 metrics of interest. The overall pattern of results from the comparisons is shown in Table 2.

3.3 Training a Detector on All Four Lessons

One potential explanation for the relatively poor transfer of detectors trained on single lessons is that it is simply not possible to develop a single gaming detector which is highly effective at detecting harmful gaming in multiple lessons, using our techniques. To investigate this possibility, we trained a detector on all four lessons together.

The detector trained on all four lessons had an average A' of 0.85, across the four lessons, at distinguishing GAMED-HURT students from non-gaming students. This was not significantly lower than the average A' (0.86) of the models trained on single lessons, when tested on the training lessons, Z_{ms} = 0.38, p=0.70. The detector trained

Table 2. Detectors trained on just one of the four lessons. Italics denotes when detectors were, in aggregate, statistically significantly better than chance. Boldface denotes when detectors were significantly better for training lessons than transfer lessons

Metric	Training lesson average	Transfer lesson average
A' (GAMED-HURT versus NON-GAMING)	***0.86***	*0.71*
A' (GAMED-HURT versus GAMED-NOT-HURT)	*0.79*	*0.74*
Correlation	***0.57***	*0.22*

Table 3. Comparing a detector trained on all four lessons to detectors trained on just one of the four lessons, within the training lessons. All detectors were statistically significantly better than chance, on each metric. There were no statistically significant differences between detectors, on any metric

Metric	Training on one lesson	Training on all lessons
A' (GAMED-HURT versus NON-GAMING)	0.86	0.85
A' (GAMED-HURT versus GAMED-NOT-HURT)	0.79	0.80
Correlation	0.57	0.60

on all four lessons had an average A' of 0.80, across the four lessons, at distinguishing GAMED-HURT students from GAMED-NOT-HURT students. This was also not significantly lower than the average A' (0.79) of the models trained on single lessons, when tested on the training lessons, $Z_{ms} = 0.12$, p=0.90. Finally, the model trained on all four lessons had an average correlation of 0.60, across the four lessons, between the observed and predicted frequencies of harmful gaming, in the training lessons. This was again not significantly different than the average correlation (0.57) of the models trained on single lessons, when tested on the training lessons, $Z_{ms} = 0.53$, p=0.60.

Hence, a model trained on all four lessons is equally as effective as four models trained on individual lessons, within the training lessons. This indicates that it is possible to develop a gaming detector which is effective in multiple lessons. The overall pattern of results from these comparisons is shown in Table 3.

3.4 Training a Detector on Three of Four Lessons

The next question to consider is whether we can develop a gaming detector which is not just effective across multiple lessons, but which can also transfer effectively to lessons it was not trained on. To this end, we trained a set of detectors on three of four of the lessons together, and then tested each of these detectors on the fourth, left-out, lesson.

We will compare these detectors' effectiveness at transferring to two other conditions. The first comparison condition is how well detectors perform when trained on a single lesson and then tested on the same lesson. We view this level of performance as a reasonable "gold standard" for how well a detector can do on any lesson. The second comparison condition is how well detectors perform when trained on a single lesson and then tested on the other lessons. Our goal is to obtain significant and substantial improvements on this level of performance.

The detectors trained on three lessons had an average A' of 0.84 at distinguishing GAMED-HURT students from non-gamers, in the training lessons, and an average A' of 0.80 at making the same distinction in the test lessons. The test set performance of detectors trained on three lessons (A'=0.80) was not significantly lower than the training set performance of detectors trained on one lesson (A'=0.86), $Z_{ms} = 1.36$, p=0.17. However, the test set performance of detectors trained on three lessons

(A'=0.80) *was* significantly higher than the test set performance of detectors trained on one lesson (A'=0.71), Z_{ms} = 1.98, p=0.05.

The detectors trained on three lessons had an average A' of 0.78 at distinguishing GAMED-HURT students from GAMED-NOT-HURT students, in the training lessons, and an average A' of 0.80 at making the same distinction in the test lessons. The test set performance of the detectors trained on three lessons (A'=0.80) was not significantly lower than the training set performance of the detectors trained on one lesson (A'=0.79), Z_{ms} = 0.67, p=0.50.

The detectors trained on three lessons had an average correlation of 0.55 between the observed and predicted frequencies of harmful gaming, in the training lessons, and an average correlation of 0.41 in the test lessons. In this case, the test set performance of the detectors trained on three lessons (r=0.41) was marginally significantly lower than the training set performance of the detectors trained on one lesson (r=0.57), Z_{ms} = 1.74, p=0.08. However, the test set performance of detectors trained on three lessons (r=0.41) was still significantly higher than the test set performance of detectors trained on one lesson (r=0.22), Z_{ms} = 2.46, p=0.01.

Overall, detectors trained on three lessons suffered considerably less degradation in performance when transferred to new lessons than detectors trained on a single lesson. Detectors trained on a single lesson had large and significant drops on 2 of 3 metrics when transferred to new lessons; the detectors trained on three lessons had much smaller and less significant drops in performance when transferred to new lessons. The overall pattern of results is shown in Table 4.

3.5 For a More Generalizable Detector, Should We Collect More Data or More Representative Data?

In the previous section, we showed that detectors trained on multiple lessons transfer better than detectors trained on a single lesson. While it is tempting to conclude that training on multiple lessons led to the better performance, it is also possible that the better performance came simply from training using more data. We developed linear regression models to distinguish between these hypotheses, predicting each detector's A' (GAMED-HURT vs non-gaming) and correlation to observed harmful gaming, within each lesson it was not trained on. These models can distinguish the relative contribution of sample size and number of lessons, because each of the four lessons had a different sample size (see Table 1). In these analyses, we define sample size as the number of observed gaming frequencies in the training set (for which there is one per student, per lesson), since this was the value correlated to during training.

A model which predicts A' using only the sample size (A' = α_0*SampleSize) achieves an r^2 of 0.02; a model which predicts A' using both the sample size and the number of lessons used in training (A' = α_0*SampleSize + α_1*Lessons) achieves an r^2 of 0.13. The model which includes the number of lessons is a significantly better predictor of A', F(1,13)=8.61, p=0.01, for an extra-sum-of-squares F-test. A model which predicts correlation to observed harmful gaming using only the sample size

Table 4. Comparing detectors trained on three of the four lessons to detectors trained on just one of the four lessons. All detectors were statistically significantly better than chance, on each metric. Grey boxes denote indicate when a detector was worse than the best detector for that metric (light grey=marginal significance, dark grey = significance)

Metric	Training on one lesson (training-set performance)	Training on 3 of 4 lessons (test-set performance)	Training on one lesson (test-set performance)
A' (GAMED-HURT versus NON-GAMING)	0.86	0.80	0.71
A' (GAMED-HURT versus GAMED-NOT-HURT)	0.79	0.80	0.74
Correlation	0.57	0.41	0.22

($r = \alpha_0$*SampleSize) achieves an r^2 of 0.22; a model which predicts correlation to observed harmful gaming using both sample size and the number of lessons used in training ($r = \alpha_0$*SampleSize + α_1*Lessons) achieves an r^2 of 0.26. The model which includes the number of lessons is a marginally significantly better predictor of a detector's correlation to observed harmful gaming, $F(1,13)=4.19$, $p=0.06$, for an extra-sum-of-squares F-test.

These results indicate that training with more lessons improves a detector's generalizability, even when we control for the size of the training set. This pattern is consistent, whether A' or correlation is the measure of interest.

4 Discussion and Conclusions

Our results show that detectors of harmful gaming trained on single tutor lessons perform well in the lesson they were trained on, but considerably more poorly on other lessons. However, if a detector is trained using data from multiple lessons, the detector is effective both within the lessons it was trained for, and on a new lesson that it was not trained for. We have also presented analyses which suggest that the improvement in transferrability arises not just from training on more data, but from training on a broader cross-section of data.

The general implication is that, for developing detectors of complex student behaviors, it is not optimal to use data from only one segment of a larger curriculum – even if it is possible to obtain a very large amount of student data from that curricular segment. Training on just one curricular section or tutor lesson risks over-fitting to the specific features of that tutor lesson. By training on a larger cross-section of data from a curriculum, a developer can develop a behavioral detector which will generalize better to the rest of the entire curriculum.

Often, it is assumed that the best way to improve a machine-learned detector is to collect more data. We do not question that more data can lead to better detectors; however, the results of our investigation suggest that if there is a choice between collecting more data from a single tutor lesson (or curricular section) or collecting data from a variety of lessons, it is preferable to collect the broader data set.

Acknowledgements

We would like to thank Tom Mitchell, Darren Gergle, Irina Shklovski, and David Andre for helpful suggestions and assistance. This work was funded by NSF grant REC-043779 to "IERI: Learning-Oriented Dialogs in Cognitive Tutors: Toward a Scalable Solution to Performance Orientation".

References

1. Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R. (2004) Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS 2004)*, 227-239.
2. Aleven, V., Roll, I.,, McLaren, B.M., Ryu, E.J., Koedinger, K.. (2005) An Architecture to Combine Meta-Cognitive and Cognitive Tutoring: Pilot Testing the Help Tutor. *Proceedings of the 12th International Conference on Artificial Intelligence in Education,* 17-24.
3. Arroyo, I., Woolf, B. (2005) Inferring learning and attitudes from a Bayesian Network of log file data. *Proceedings of the 12th International Conference on Artificial Intelligence in Education,* 33-40.
4. 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.
5. Baker, R.S., Corbett, A., Koedinger, K., Roll, I. (2005) *Detecting When Students Game The System, Across Tutor Subjects and Classroom Cohorts*. Proceedings of User Modeling 2005, 220-224.
6. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.
7. Conati, C., McLaren, H. (2005) Data-Driven Refinement of a Probabilistic Model of User Affect. *Proceedings of the Tenth International Conference on User Modeling (UM2005)*, 40-49.
8. de Vicente, A., Pain, H. (2002) Informing the detection of the students' motivational state: an empirical study. *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*, 933-943.
9. Hanley, J.A., McNeil, B.J. (1982) The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology,* 143, 29-36.
10. Maris, E. (1995) Psychometric Latent Response Models. *Psychometrika*, 60 (4), 523-547.
11. Roll, I., Baker, R.S., Aleven, V., McLaren, B.M., Koedinger, K.R. (2005) Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems. *Proceedings of User Modeling*, 379-388.
12. Rosenthal, R., Rosnow, R. (1991) *Essentials of Behavioral Research: Methods and Data Analysis.* Boston, MA: McGraw-Hill.
13. Yu, L., Liu, H. (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. *Proceedings of the International Conference on Machine Learning*, 856-863.