Walonoski, J., Heffernan, N.T. (accepted). Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In Ikeda, Ashley & Chan (Eds.). *Proceedings of the Eight International Conference on Intelligent Tutoring Systems*. Springer-Verlag: Berlin. pp. 382-391. 2006

Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems

Jason A. Walonoski and Neil T. Heffernan

Worcester Polytechnic Institute, Computer Science Department, 100 Institute Rd, Worcester, MA 01601 USA
{jwalon, nth}@wpi.edu

Abstract. A major issue in Intelligent Tutoring Systems is off-task student behavior, especially performance-based gaming, where students systematically exploit tutor behavior in order to advance through a curriculum quickly and easily, with as little active thought directed at the educational content as possible. The goal of this research was to explore the phenomena of off-task gaming behavior within the *Assistments* system. Machine-learned gaming-detection models were developed to investigate underlying factors related to gaming, and an analysis of gaming within the Assistments system was conducted to compare some of the findings of prior studies.

1 Introduction

Intelligent Tutoring Systems (ITS) have been shown to have a positive effect on student learning [1], however these effects may be negated by a lack of student motivation or student misuse. One area of research examining these issues involves studying student "gaming" of the system, especially recognition of gaming behavior [2]. A student is gaming if they are attempting to systematically use the tutor's feedback and help methods as a means to obtain a correct answer with little or no work, in order to advance through the curriculum as fast or easily as possible. Student gaming has been correlated with substantially less learning [3]; therefore it is of particular importance to understand in order to maximize tutor effectiveness.

One objective of this research was to apply existing methodologies of gaming behavior detection to the *Assistments* mathematics ITS [4]. These methods involved the construction of machine-learned models to identify gaming behavior. Although gaming behavior has only two hallmark appearances (help abuse and systematic guessing and checking), there may be various hidden factors at work: some students are harmed by their gaming while others are not. Machine learning has been shown to be able to differentiate between these two types of gamers [2].

Additionally, previous work by Baker et al [2][3] has resulted in documentation of the phenomenon of gaming within ITS and theories about why students game. Our second objective was to corroborate or contradict those findings by profiling the typical gaming student in the *Assistments* system and exploring the relation of gaming behavior with student learning, and then comparing the results of the studies.

2 Detection of Gaming

If tutoring software is outfitted with a model that can reliably identify gaming behavior, then intervention strategies can be developed and deployed with reasonable assurance that they are invoked under appropriate circumstances. Rather than manually constructing a model by authoring rules based on the surface features of gaming (systematic guessing and checking or requesting help until answers are directly supplied), machine-learning methods were employed to identify the underlying hidden variables that lead students to game and illustrate how they are affected by their behavior. A prior study by Baker et al has shown that gaming behavior can be reliably detected with machine-learned models [2], and in the course of constructing a model for the *Assistments* system we adapted those methods for general verification of their findings, to determine if gaming behavior has the same causes, appearances, and resulting effects in different tutoring systems.

The methodology used was essentially a four-step process: (1) classroom observation of students using the tutoring software, (2) dataset creation based upon those observations to be used by machine-learning algorithms, (3) the construction of classifiers (prediction models) using the datasets, and (4) analysis of the results.

2.1 Classroom Observation

Students were observed using the *Assistments* ITS in real classroom sessions. Each observation was a triplet of observation time, student identity (alias), and recorded behavior. Observation was conducted as unobtrusively as possible, with students unaware that surveillance was taking place (students treated observers as assistant teachers and displayed no knowledge of being systematically observed). Observations were taken from a modest distance to (1) minimize student self-consciousness and awareness of being watched, (2) allow the observer flexibility in positioning themselves within the environment to maximize line of sight, and (3) allow the simultaneous observation of groups of students. For consistency purposes, each observer was given an instruction sheet on how to conduct observations. Groups of students were observed for approximately 20 to 30 seconds per student; so a group of 3 students would have been observed for 60 to 90 seconds. The possible variation of observation times was left to the observer depending on the consistency or deviation in the students' behavior in order to get a representative measurement. A numerical code from Table 1 (adapted from measurements in [3]) was recorded for each student per observation period. During a given observation period, a student might exhibit multiple behaviors. In that case, rather than record all the behaviors, observers were instructed to give priority where gaming behavior was given the highest priority, followed by the three on-task behaviors (sorted by least engaged to most engaged), followed by the remaining two off-task behaviors (sorted by most active to least active).

To ensure that this methodology was not subjective to observer bias, an inter-rater reliability study was performed. Two observers (one of which was the first author) were provided with the observation instructions and then observed two classes, with students observed in the exact same order and at the exact same time. The two observers made 71 observations each. The consistency across all six numerical behaviors was 77% (57 out of 71 observations matched). The consistency across the three

Table 1. Measurement Definitions

Category	Observation	Definition	Priority
A - On Task	1	On Task with the Tutor	4
A - On Task	2	On Task with Paper or Teacher (including talking about the problem)	3
A - On Task	3	On Task, but talking while working (subject matter of conversation is irrelevant)	2
B - Off Task	4	Off Task and Talking	5
B - Off Task	5	Off Task and Inactive (including web-surfing, staring into space, sleeping, et cetera)	6
C - Gaming	6	Gaming (guessing-and-checking or bottom-out-hinting)	1

alpha-encoded categorical behaviors was 97% (69 out of 71 observations matched), while there was 100% agreement in the identification of gaming behavior. Since our classifier was aimed purely at the identification of gaming behavior, as opposed to distinction between all behaviors, these results have a suitable level of consistency.

Overall, 850 observations were recorded, spanning 8 classes that lasted approximately 50 minutes each. Those 8 classes consisted of experienced users of the *Assistments* system, who had been using it biweekly for the entire 2004-2005 academic year.

2.2 Dataset Creation

The *Assistments* system automatically logs all user actions and events except mouse movements. From these logs we can distill information such as a student's number of attempts (including whether the attempt was correct or incorrect, or if it was the first attempt on a given problem), numbers of hint requests (including bottom-out hint requests that directly supply the correct answer), and action response time in milliseconds. Actions were joined to the recorded classroom observations by user identification and time to create training instances for the machine-learning algorithms.

Given the length of time spent observing particular students, it is not clear which actions should be matched with a particular observation. To resolve this issue, actions were joined to observations using an "unsupervised action filter" based on a variable "time window." Informally, a time window is defined as a dilation of time around a recorded observation time. Not being sure what size time windows are reasonable, five sizes were utilized: 30 seconds, and 1, 2, 4, and 6 minutes. For example, given a 2-minute time window, all actions made between 1 minute before and after each observation were included in the generation of the training instances. The filter is considered "unsupervised" because no attempt is made to filter in or out actions based on their applicability to the recorded observed behavior.

Using the observations and action logs, a number of datasets were generated via unsupervised action filtering using time windows. The datasets had 1430 attributes and 1 classification value (gaming, *true* or *false*). The six observation types in Table 1 were rolled up into either gaming is *true* (observation #6) or gaming is *false* (all other observations). Machine-learning algorithms are dependent on relevant attributes, so the selection of attributes is an important exercise. We adapted the attributes of Baker et al [2] to the particulars of the *Assistments* system and variable time windows. For an observation within a particular time window, the attributes were as defined:

- Actions: the total number of all actions.
- Attempts: six attributes for the total number of all attempts, correct attempts, incorrect attempts, correct first attempts, and incorrect first attempts.
- Attempt Time: five attributes for the sum, minimum, maximum, average, and standard deviation of all attempt times in milliseconds. Also four Boolean attributes were included indicating whether attempt times were slow, extra-slow, quick, or extra-quick, which were calculated by comparing the student response time with the average response time of all students on the given problems (and plus or minus the standard deviation of all student response times for the extra-slow and extra-quick attributes).
- Hints: two attributes for the total number of hint requests and bottom-out hint requests.
- Hint Time: five attributes for the sum, minimum, maximum, average, and standard deviation of all hint request times in milliseconds. Also four Boolean attributes were included indicating whether hint request times were slow, extra-slow, quick, or extra-quick, which were calculated by comparing the student response time with the average response time of all students on the given problems (and plus or minus the standard deviation of all student response times for the extra-slow and extra-quick attributes).
- Problems: two attributes for the total number of top-level problem questions, and the total number of follow-through helping questions.
- User-Interfaces: two attributes for the total number of questions that featured a multiple-choice user-interface and another for the total number of questions that featured a textbox user-interface.
- Replays: the total number of times a problem was "replayed" by the *Assistments* tutor runtime [5] (this generally indicates that the student tried to exit the system and the runtime had to "replay" the students actions on a given problem to reconstruct the tutors agenda *exactly* for the given problem).
- *pmpKnow*: "poor man's prior knowledge," the probability that the student possesses the prior knowledge required to answer the given question correctly. Prior knowledge in ITS is often determined by knowledge tracing, however the *Assistments* system currently lacks dynamic knowledge model tracing, so as a substitute we use the poor man's version: the student's percent correct across all previous problems. Also four Boolean attributes were included indicating whether the prior knowledge was high, extra-high, low, or extra-low in comparison to the average prior knowledge of all students and in combination with the standard deviation of that average (for the extra-high and extra-low variables).

- Problem-Difficulty: four attributes for the minimum, maximum, average, and standard deviation of problem difficulties for all problems encompassed within the observation time window. Problem difficulty is a number between 0 (easy) and 1 (hard) that is the percent correct on first attempt by all previous students. The combination of these values would hopefully represent the range of difficulty during the observation.
- Ratios: six attributes representing the following ratios: the number of attempts per problem, the number of correct attempts per problem, the number of incorrect attempts per problem, the number of hints per problem, the number of bottom-hints per problem, and the number of replays per top-level problem.
- Pair-wise Interaction Effects: 1378 attributes representing the quadratic effects between any two of the attributes listed above. For example, the total number of hints times the average problem difficulty. The list of pair-wise interaction effect attributes is comprehensive (all the original attributes have a pair-wise interaction effect attribute with every other original attribute, including itself).

2.3 Machine-Learning Algorithms

We used 12 different algorithms from the WEKA machine-learning system [6] on our datasets to generate models including decision tree methods, lazy methods (k-nearest neighbors), locally weighted learning, Bayesian methods, a neural network, a propositional-logic rule learning algorithm (PRISM) [7], as well as logistic regression. A large number of algorithms were used out of curiosity because each has advantages and disadvantages (which are outside of the scope of this document) that could potentially reveal different kinds of relationships within the data. Some of the algorithms generate human-readable rules while others produce mathematical models that are often difficult to interpret by humans. The results were then examined for (1) classification accuracy, (2) accuracy of the confusion matrices, and (3) reasonable rules, especially those that might corroborate or contradict expected findings based on previous studies, or other interesting results.

2.4 Results and Discussion

None of the algorithms used significantly outperformed any of the others (according to statistical tests automatically performed by WEKA). Therefore, choosing a final model rested on a selecting a classifier that generated reasonable rules that corroborated both the surface-level hallmark characteristics of gaming and the findings of previous studies.

The classifier that was ultimately selected as our preferred model was generated using the J48 decision tree algorithm (based on Quinlan's C45 algorithm [8]). Although other algorithms had faster classification times or higher accuracies, this model was chosen because across all training and testing folds it produced reasonably clean confusion matrices, generated human-readable rules that upon interpretation seemed to corroborate findings from past studies.

Several of our resulting rules offered further support to the hypotheses of Baker et al [2][3] that suggest that one cause of gaming is low prior knowledge combined with problem difficulty. Other rules could be interpreted in such a way as to identify the class of "gamed-not-hurt" students, which supports the Baker et al distinction between students whose learning is affected by gaming and those who are not. Finally, results at the four and six minute intervals suggests that using longer time windows does not adversely effect the detection of gaming, and in fact improves as those students who game tend to make a habit of it and identifying them becomes easier and easier as they continue their off-task behavior.

After being selected as the preferred model, the J48 algorithm was rerun using leave-one-out cross-validation (LOOCV) as the testing method. This involved generating our model 850 times, each time using 849 of the 850 instances for training purposes and leaving out one different instance for testing, and then using the model to predict whether the 850th instance was gaming or not. This process was repeated for each of the datasets (one for each time window).

While most of the models resulting from LOOCV had 100% classification accuracy, averaging out the results of all models results in about 96% accuracy. Given the low rate of observed gaming (19 out of 850 observations, ~2.2%), the effectiveness of the models becomes questionable. Analysis of the confusion matrices helps our understanding of how the models perform. On average, the models tend to correctly identify non-gaming instances about 98% of the time, while correctly identifying gaming instances only about 19% of the time. Although this is not ideal, if we consider that gaming is much more harmful to learning than other behaviors [3] and it is such an infrequent behavior, then 19% of gaming instances may seem better than what might be expected from chance alone. So, while the model accuracy leaves something to be desired, we are at least satisfied in the general reasonability of the resulting "rules" given what is known about gaming behavior.

Ultimately, results of our final model were satisfactory since construction of the datasets and models verified some of the underlying hidden variables that lead students to game (e.g. low prior knowledge), and the generated rules were human-readable and reasonably captured the hallmark surface-level characteristics and other known causes of gaming behavior. Although we would like to improve the accuracy and strength of our final model, it could be outfitted as-is into the *Assistments* system to dynamically detect gaming behavior and trigger various intervention strategies, as a post-tutoring reporting device, or as an objective evaluator of various intervention strategies within controlled experiments.

3 Gaming Within the Assistments System

The last portion of this research was a general examination of gaming behavior within the *Assistments* system. This examination made use of a *prima facie* algorithm (as opposed to the machine-learned model) that calculates how frequently individual students gamed based on surface-level features of hint abuse and guessing-and-checking only. If a student asks for a hint on, or answers incorrectly (possible guess), any step within a given problem three consecutive times, then they are assumed to be gaming that problem. When a problem is gamed, a *possibly-gaming* index is increased by one. If an entire problem is completed without any step being gamed, then the *possibly-gaming* index is reduced by one. If at any time the *possibly-gaming* index is above three, any further identified gaming increases a student's *total-gaming-score* by one.

3.1 Assistments System Survey Responses

A survey was administered to students who had been using the *Assistments* system throughout the 2004-2005 academic year on a biweekly basis. The survey consisted of 32 Likert-scale questions and some open response items. Gaming scores were calculated for those students who completed the entire survey using the *prima facie* algorithm. Depending on where a students average score fell in relation to the overall average and the overall standard deviation, they were classified as very-high, above-average, below-average, or very-low gamers. Out of 365 students, 53 were very-high gamers, 91 were above-average, 179 were below-average, and 42 were very-low gamers. By analyzing the distribution of responses by gaming-classification we constructed the following profile:

- Mathematics: Students who gamed were more likely to believe that they were not good at math and less likely to believe they could do well at math if they worked hard. Students who gamed often said they were less likely to do homework in math class, and the more students gamed, the less they said they liked math class. The less a student gamed the more they were likely to strongly agree that they would use math in a job when they grew up. Students who gamed often were much more likely than other students to strongly agree that their parents thought it important for them to do well in math, which may explain the performance-based motivation behind some gaming.
- Computers: Even though students who gamed often were less likely to have a computer at home, they were also less likely to report having trouble concentrating on the computer. Students who gamed often agreed more often and more strenuously that they liked learning from a computer than those who gamed very little.
- Educational Medium: Students who tended not to game were more likely to say that they preferred using the *Assistments* system to doing homework. In a similar question, there were no differences between the groups when asked if they would prefer to use the tutor rather than take a test – they mostly all strongly agreed that they would. The less a student gamed, the more strongly they would prefer using the *Assistments* system to normal classroom activity.
- Help Seeking: Students who gamed very little were more likely to strongly agree that they would seek help when they didn't understand something. The more a student gamed, the more they thought that being told the answer was more helpful than reading the hints. The more a student gamed, the more they agreed that the hints aided in their understanding of similar problems.
- Problem Difficulty: Students who gamed often tended to strongly agree that the items were frustrating because they were too hard, while students who gamed very little were more likely to disagree. This is probably partially

related to student prior knowledge. Students who gamed often were more likely to agree or strongly agree that they tried to get through difficult problems as quickly as possible.

- Goals: Students who gamed often tended more than other students to say that their goal was to get through as many items as possible. Interestingly, the more students gamed, the more they also tended to strongly agree that their goal was to learn new things.
- Students who gamed often had a slight tendency to say that they prefer facts and data to concepts and ideas more than other students.

Some of these results are interesting merely because they either corroborate or disagree with past findings. For example, Baker et al have reported that students who game do not like computers [9], while our survey suggests that those students who appeared to be heavily gaming *prima facie*, agreed more often and more strenuously that they liked learning from a computer than those who gamed very little. However, our survey results show that gamers were less likely to own a computer at home, and were more likely to dislike math class (also inconsistent with previous findings).

3.2 Gaming and Learning

Off-task gaming behavior has been correlated with substantially less learning in several prior studies [3]. In an attempt to validate those findings, learning rates that had been previously calculated for [10] using traditional methods as well as longitudinal data analysis, were grouped by the very-high, above-average, below-average, and very-low gaming categories. The results are summarized in Table 2.

These results seem to corroborate previous findings that indicate that off-task gaming behavior is correlated with substantially less learning. However, a few statistical tests were run to examine the significance of these results. Before those tests were run, students were classified as being gamers or not. If a student's score put them at the level of very-high gaming, then they were a considered a gamer, and otherwise they were not. This was done to simplify the results and make them easier to interpret.

Table 2. Learning Rates by Gaming Category

Category	Traditional Slope	Traditional Intercept	SW Slope	SW Intercept	Actual MCAS	Scaled Ans.
Very Low Gaming	1.68	26.92	0.34	24.04	35.17	0.76
Below Avg Gaming	1.62	19.53	0.33	19.31	30.56	0.80
Above Avg Gaming	1.29	15.07	0.33	14.23	24.24	0.70
Very High Gaming	0.95	11.99	0.26	11.71	19.66	0.77
Overall Average	1.44	17.88	0.32	17.25	27.69	0.76

The first test was an analysis of variance (ANOVA) of learning (SW slope) by gaming status. The results suggest that the learning rates of students who game and those that do not are reasonably different than mere chance alone, and they show that gaming behavior is correlated with less learning ($p < 0.19$).

The second test was an ANOVA of knowledge (SW intercept) by gaming status. The results very strongly indicate that students who engage in gaming behavior are more likely to come to the *Assistments* system with lower prior knowledge than other students ($p < 0.0001$).

One last test examined the correlation of gaming with a students actual MCAS score via Bartlett's Test of Sphericity, which very strongly showed that gamers do not perform well on the actual MCAS state administered mathematics exam ($p < 0.0001$).

These three tests show that *prima facie* gamers start with less knowledge, learn less, and perform worse on the actual MCAS examination.

4 Conclusions

Off-task gaming behavior is a major issue within the field of ITS, since it has been correlated with poor learning. The goal of this research was to explore this important phenomenon within the *Assistments* system. A machine-learned decision-tree model for gaming detection was developed, and while the practicality of this model was questionable, the resulting rules corroborated the connection of low prior knowledge and problem difficulty with gaming. Further analysis of gaming and its effects within the *Assistments* system was undertaken, via student survey responses and student learning-rates. The survey results provide some agreement and disagreement with previous studies about the nature of gaming, and the learning rates corroborated findings that indicate that off-task gaming behavior is correlated with lower prior knowledge and less learning.

Acknowledgements

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. All the opinions in this article are those of the authors only. This work would not have been possible without the assistance of the 2004-2005 WPI/CMU *Assistments* team including Mingyu Feng, Andrea Knight, Ken Koedigner at CMU, Abraao Lourenco, Michael Macasek, Goss Nuzzo-Jones, Kai Rasmussen, Leena Razzaq, Steven Ritter at Carnegie Learning, Carolyn Rose at CMU, Terrence Turner, and Ruta Upalekar.

References

1. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). *Intelligent tutoring goes to school in the big city*. International Journal of Artificial Intelligence in Education, 8, 30-43.
2. 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.

3. 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.
4. Razzaq, L, Feng, M., Nuzzo-Jones, G., Heffernan, N.T. et. al (2005). *The Assistment Project: Blending Assessment and Assisting*. Proceedings of the 12th Annual Conference on Artificial Intelligence in Education, Amsterdam.
5. Nuzzo-Jones, G., Walonoski, J.A., Heffernan, N.T., Livak, T. (2005). *The eXtensible Tutor Architecture: A New Foundation for ITS*. Proceedings of the 12th Annual International Conference on Artificial Intelligence in Education, Amsterdam.
6. Ian H. Witten and Eibe Frank (2005). "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
7. J. Cendrowska (1987). *PRISM: An algorithm for inducing modular rules*. International Journal of Man-Machine Studies. Vol.27, No.4, 349-370.
8. JR Quinlan. "C4. 5: Programs for Machine Learning." The Morgan Kaufmann Series in Machine Learning, San Mateo, 1993.
9. Baker, R.S., Roll, I., Corbett, A.T., Koedinger, K.R. (2005) *Do Performance Goals Lead Students to Game the System?* Proceedings of the 12th International Conference on Artificial Intelligence and Education, Amsterdam.
10. Feng, M., Heffernan, N.T, Koedinger, K.R. (2006) *Addressing the Testing Challenge with a Web-Based E-Assessment System that Tutors as it Assesses*, Proceedings of WWW2006, Edinburgh, Scotland.