Differences Between Intelligent Tutor Lessons, and the Choice to Go Off-Task

Ryan S.J.d. Baker

rsbaker@cmu.edu

Human Computer Interaction Institute, Pittsburgh Science of Learning Center,

Carnegie Mellon University

Abstract. Recent research has suggested that differences between intelligent tutor lessons predict a large amount of the variance in the prevalence of gaming the system [4]. Within this paper, we investigate whether such differences also predict how much students choose to go off-task, and if so, which differences predict how much off-task behavior will occur. We utilize an enumeration of the differences between intelligent tutor lessons, the Cognitive Tutor Lesson Variation Space 1.1 (CTLVS1.1), to identify 79 differences between tutor lessons, within 20 lessons from an intelligent tutoring system for Algebra. We utilize a machine-learned detector of off-task behavior to predict 58 students' off-task behavior within that tutor, in each lesson. Surprisingly, the best model predicting off-task behavior from lesson features contains only one feature: lessons that involve equation-solving. We discuss possible explanations for this finding, and further studies that could shed light on this relationship.

1 Introduction

What underlies students' choices, while they use educational software? In particular, why do students choose to game the system or go off-task, while using educational software? Much of the research on these questions has focused on the role that stable or semi-stable student individual differences play in driving these types of behaviors [2, 3, 8, 9]. Take, for example, the case of gaming the system ("attempting to succeed in an interactive learning environment by exploiting properties of the system rather than by learning the material" [cf. 5]). Several studies have been published that attempt to explain gaming behavior in terms of stable or semi-stable individual differences between students, such as a student's attitude towards mathematics or goal orientation [2, 8, 9]. These studies have generally found statistically significant relationships. However, the relationships found in these studies only explain 5-9% of the variance in gaming behavior ($r^2 = 0.05$ to 0.09) [2,8], a relatively low degree of explanatory power.

By contrast, [7] found that the differences between intelligent tutor lessons predict a large proportion of the variance in gaming behavior. In an analysis of 58 students' behavior within 20 lessons in an intelligent tutor for algebra (corresponding to the majority of a year's curriculum), a combination of features of tutor lessons was found to predict 56% of the variance in gaming behavior ($r^2 = 0.56$). In particular, lessons that incorporated interest-increasing text into problem scenarios had significantly less gaming; lessons with various types of ambiguity had more gaming; lessons with ineffective hints had more gaming; and lessons based on equation-solving had less gaming. These results suggest that it may be possible to bypass the intrusiveness and high development costs of interactive responses to gaming [cf. 1, 4, 22] simply by altering these features of lessons,

designing lessons with less extraneous ambiguity and more attempts to increase student interest.

The discovery that gaming the system can be well predicted by small-scale differences in educational software design raises the question of whether other prominent learner behaviors are similarly associated with small-scale features of software design. In this paper, we investigate whether small-scale differences in software design can predict variance in off-task behavior. Off-task behavior shares many characteristics with gaming behavior. Both behaviors have been found to be associated with poorer learning in intelligent tutoring systems, although gaming the system's impact on learning is both larger and more immediate [6, 11]. Additionally, the two behaviors have each been found to be weakly associated with some of the same student individual differences [3], in particular negative attitudes towards computers and mathematics.

In this study, we apply a previously validated detector of off-task behavior [3] to data obtained from the PSLC DataShop [15], representing an entire school year of use of Cognitive Tutor Algebra, a widely used intelligent tutoring system. During the school year, students worked through a variety of lessons on different topics. These lessons had moderate variation in subject matter and considerable variation in design, making it possible to observe which differences in subject matter and/or design are associated with differences in how much off-task behavior occurs. We apply an existing taxonomy of the differences between tutor lessons [7] to these lessons, and investigate which lesson features are most strongly associated with off-task behavior.

2   Data and Models Applied

Data was obtained from the PSLC DataShop [15] (dataset: Algebra I 2005-2006 Hampton Only), for 58 students' use of Cognitive Tutor Algebra during an entire school year. The data set was composed of approximately 437,000 student transactions (entering an answer or requesting help) in the tutor software. All of the students were enrolled in algebra classes in one high school in the Pittsburgh suburbs. The school used Cognitive Tutors two days a week, as part of its regular mathematics curriculum. None of the classes were composed predominantly of gifted or special needs students. The students were in the 9th and 10th grades (approximately 14-16 years old).

The Cognitive Tutor Algebra curriculum involves 32 lessons, covering a complete selection of topics in algebra, including formulating expressions for word problems, equation solving, and algebraic function graphing. Three lessons from Cognitive Tutor Algebra are shown in Figure 1. Data from 8 lessons was eliminated from consideration, as taxonomy codings were not available for those lessons (these lessons were not coded in [7], due to having limited data from those lessons available for that paper's analyses of interest). On average, each student completed 10.7 tutor lessons (among the set of 24 lessons considered), for a total of 619 student/lesson pairs.

Figure 1. Three lessons from Cognitive Tutor Algebra. Top: The Equation-Solver. Middle: Story Problem with Worksheet. Bottom: Function Graphing.

To determine how often each student was off-task, in each lesson, each student's actions were labeled using Baker's [3] detector of off-task behavior. The detector was developed using data from 429 students' classroom use of three lessons from an intelligent tutor on middle school mathematics. Applying this detector makes it tractable to study off-task behavior across a wide variety of tutor lessons. By contrast, other well-known methods are intractable – for instance, conducting quantitative field observations on a similar number of tutor lesssons and students would involve sending out two or more research assistants to classrooms for an entire year.

The detector, under cross-validation, achieved a correlation of 0.55 to field observations of off-task behavior – hence, it can be considered reasonably reliable for these purposes. The detector is also able to distinguish off-task behavior from on-task conversation, by looking at the student actions that occur immediately before and after a seemingly idle pause. We show the model that predicts off-task behavior within the detector in Table 1. The detector makes a prediction as to whether each action is off-task, and then aggregates across actions to indicate what proportion of student actions was off-task (or, alternatively, what proportion of student time was off-task). Full details on this detector are available in [3]. Two features (F3 and F6) involved features that were not available for this data set (string and generally-known). However, F3 and F6 together accounted for only 4.4% of the cross-validated correlation accounted for by this model [3] – hence, this model can still be expected to be accurate even in the absence of these features.

Table 1. The model of off-task behavior (OT) used in this paper, from [3]. In all cases, param1 is multiplied by param2, and then multiplied by value. Then the six features are added together. If the sum is greater than 0.5, the action is considered to be off-task. Features that were not applicable to the current data set are indicated in gray. "Pknowretro", a feature found in many behavior detectors, refers to the probability the student knew the skill if the action was the first opportunity to practice the current skill on the current problem step, and is -1 otherwise.

	param 1	param 2	value	Interpretation
F1	timelast3SD	timelast5SD	-0.08	OT: Very fast actions immediately before or after very slow actions
F2	timeSD	timeSD	0.013	OT: Extremely fast actions or extremely slow actions
F3	string	pknowretro	-0.36	OT: Less likely on well-known string-input steps OT: More likely when inputting a string after error
F4	notfirstattempt	recent8help	-0.38	Not OT: Asking for a lot of help
F5	notright	pknowretro	-0.16	OT: Two errors or help-requests in a row Not OT: Errors or help requests on skills the student has already mastered
F6	pctwrong	generally-known	0.04	OT: Indicated by many errors on skills students generally know prior to starting this lesson

Table 2. The 79 features of the Cognitive Tutor Lesson Variation Space (CTLVS1.1) used in study. Features captured using data mining methods (as opposed to hand-coding) marked with *.

Difficulty, Complexity of Material, and Time-Consumingness	
1*. Avg. % error	2. Lesson consists solely of review of material encountered in previous lessons
3*. Avg. probability that student will learn a skill at each opportunity to practice skill [cf. 12]	4*. Avg. initial probability that student will know a skill when starting tutor [cf. 12]
5. Avg. # of "distractor" values per problem	6. % of problems where "distractor" values given
7. Max number of mathematical operators needed to give correct answer on any step in lesson	8. Maximum number of mathematical operators mentioned in hint on any step in lesson
9. Intermediate calculations must be done outside of software (mentally or on paper) for some problem steps (ever occurs)	10. % of hints that discuss intermediate calculations that must be done outside of software
11*. Total number of skills in lesson	12*. Avg. time per problem step
13. % of problem statements that incorporate multiple representations (ex: diagram and text)	14. % of problem statements that use same numeric value for two constructs
15. Avg. number of distinct/separable questions or problem-solving tasks per problem	16. Maximum number of distinct/separable questions or problem-solving tasks in any problem
17. Avg. # of numbers manipulated per step	18*. Avg. # of times each skill repeated per problem
19*. Number of problems in lesson	20*. Avg. time spent in lesson
21. Avg. number of problem steps per problem	22. Minimum number of answers or interface actions required to complete problem
Quality of Help Features	
23*. Avg. amount that reading on-demand hints improves performance on future opportunities to use skill [cf. 10]	24*. Avg. Flesch-Kincaid Grade Reading Level [16] of hints
25. % of hints using inductive support, going from example to abstract concept/principle	26. % of hints that explicitly explain concepts or principles underlying current problem-solving step
27. % of hints that explicitly refer to abstract principles	28. On average, # of hints must student request before concrete features of problems are discussed
29. Avg. number of hint messages per hint sequence that orient student to math sub-goal	30. % of hints that explicitly refer to scenario content (instead of solely math constructs)
31. % of hint sequences that use terminology specific to this software	32. % of hint messages which refer solely to interface features
33. % hint messages that teacher can't understand	34. % of hint messages with complex noun phrases
35. % of skills where the only hint message explicitly tells student what to do	
Usability	
36. First problem step in first problem of lesson is either clearly indicated, or follows established convention (such as top-left cell in worksheet)	37. % of steps where student must change a value in a cell that was previously treated as correct (example: self-detection of errors)
38. After student completes step, system indicates where in interface next action should occur	39. % of steps where it is necessary to request hint to figure out what to do next
40. Not immediately apparent what icons in toolbar mean	41. Screen cluttered with interface widgets; difficult to determine where to enter answers
42. Problem-solving task is not immediately clear	43. Format of answer changes between problem steps without clear indication
44. If student has skipped step, and asks for hint, hints refer to skipped step without explicitly highlighting in interface (ever seen)	45. If student has skipped step, and asks for hint, skipped step is explicitly highlighted in interface (ever seen)
Relevance and Interestingness	
46. % of problems which appear to use real data	47. % of problem statements with story content
48. % of problem statements with scenarios relevant to potential student careers	49. % of problem statements with scenarios relevant to students' current daily life
50. % of problem statements which involve fantasy (example: being a rock star)	51. % of problem statements which involve concrete details unfamiliar students (example: dog sleds)
52. % of problem statements which involve concrete people/places/things	53. % of problem statements with text not directly related to problem-solving task
54. Avg. number of person proper names in problem statements	

Aspects of "buggy" messages notifying student why action was incorrect	
55. % of buggy messages that indicate concept student demonstrated misconception in	56. % of buggy messages that indicate how student's action was result of procedural error
57. % of buggy messages that refer solely to interface action	58. Buggy messages given by icon, which can be hovered over to receive buggy message
Design Choices Which Make It Easier to Game the System	
59. % of multiple-choice steps	60. Avg. number of choices in multiple-choice
61. % of hint sequences with final hint that explicitly tells student what the answer is, but not what/how to enter it in the tutor software	62. Hint gives directional feedback (example: "try a larger number") (ever seen)
63. Avg. number of feasible answers for each problem step	
Meta-Cognition and Complex Conceptual Thinking **(or features that make them easy to avoid)**	
64. Student is prompted to give self-explanations	65. Hints ever give explicit metacognitive advice
66. % of problem statements that use common word to indicate mathematical operation to use (example: "increase")	67. % of problem statements that indicate math operation with uncommon terminology ("pounds below normal" for subtraction)
68. % of problem statements that explicitly tell student which math operation to use ("add")	
Software Bugs/Implementation Flaws (generally rare)	
69. % of problems where grammatical error is found in problem statement	70. Reference in problem statement to interface component that does not exist (ever occurs)
71. Student can advance to new problem despite still visible errors	72. Hint recommends student do something which is incorrect or non-optimal (ever occurs)
73. % of problem steps where hints are unavailable	
Miscellaneous	
74. Hint requests that student perform some action	75*. Avg. length of text in popup widgets
76. Value of answer is very large (over four significant digits) (ever seen)	77. % of problem statements which include question or imperative
78. Student selects action from menu, tutor software performs action (as opposed to typing in answers, or direct manipulation)	79. Lesson is an equation-solver lesson

Each tutor lesson's attributes was represented using the Cognitive Tutor Lesson Variation Space version 1.1 (CTLVS1.1) [7], an enumeration of how Cognitive Tutor lessons can differ from one another. The CTLVS1.1 was developed by a diverse design team, including cognitive psychologists, educational designers, a mathematics teacher, and EDM researchers. The CTLVS1.1, shown in Table 2, consists of 79 features for how cognitive tutors differ from each other. The CTLVS1.1 was labeled with reference to the 24 lessons studied in this paper by a combination of educational data mining and hand-coding by the educational designer and mathematics teacher.

3   Analysis Methods and Results

The goal of our analyses was to determine how well each difference in lesson features predicts how much students will go off-task in a specific lesson. To this end, we combined the labels of the CTLVS1.1 features for each of the 22 lessons in Cognitive Tutor Algebra, and the assessments of how often each of the 58 students in the data set were off-task in each of the 22 lessons.

Our first step in conducting the analysis was to determine if the 79 features of the CTLVS1.1 grouped into a smaller set of factors. We empirically grouped the 79 features of the CTLVS1.1 into 6 factors, using the implementation of Principal Component Analysis (PCA) given in SPSS. These same 6 factors were previously successful in discovering a factor that was statistically significantly associated with gaming the system [7].

We analyzed whether the correlation between any of these 6 factors and the frequency of off-task behavior was significant. However, none of the factors was statistically significantly associated with off-task behavior – the closest factor to significance had $F(1,21)= 0.37$, $p=0.55$.

Taking the 79 features individually, only two were found to be statistically significantly associated with the choice to go off-task. Using an (overly conservative) Bonferroni adjustment [20] to control for the number of statistical tests conducted, only one feature was still found to be statistically significant. This feature was whether the lesson was an equation-solver lesson (as opposed to other types of lessons, such as story problems). An equation-solver lesson is shown at the top of Figure 1. Students were statistically significantly less likely to go off-task within equation-solver lessons, $r^2 = 0.55$, $F(1, 21)=27.29$, $p<0.001$, Bonferroni adjusted $p<0.001$.

To put this relationship into better context, we can look at the proportion of time students spent off-task in equation-solver lessons as compared to other lessons. On average, students spent 4.4% of their time off-task within the equation-solver lessons, much lower than is generally seen in intelligent tutor classrooms [5,6] or, for that matter, in traditional classrooms [cf.17, 18]. By contrast, students spent 14.1% of their time off-task within the other lessons, a proportion of time-on-task which is much more in line with previous observations. The difference in time spent per type of lesson is, as would be expected, statistically significant, $t(22)=4.48$, $p<0.001$.

The other feature found to be statistically significantly associated with off-task behavior, prior to the Bonferroni adjustment, was the proportion of hints that are solely bottom-out hints (more bottom-out-only-hints, less off-task behavior). However, a model including both of these two features was not statistically significantly better than the model that only considered whether the lesson was an equation-solver lesson, $F(1, 21)=0.73$, $p=0.40$.

4   Discussion and Conclusions

The results found here suggest that differences between lessons explain a large proportion of the variance in how much off-task behavior occurs, just as with gaming the system. However, the nature of the models found is quite different. Whereas the model that best explains how much gaming occurs was a complex set of fine-grained features [7], the model that best explains off-task behavior consists of a single, very coarse-grained difference. This leaves us with a problem of interpretation. Why were students off-task so much less within these equation-solver lessons?

One hypothesis is that there is some combination of features distinct to equation-solver lessons that produce less off-task behavior, but only when the full combination is encountered. For example, it is possible that the combination of features found in the equation-solver lessons (such as less complex hints, in combination with direct interaction with the equations, in problems that are generally shorter), combine to produce a state of very positive continued engagement (e.g. flow [13]) that precludes off-task behavior. It may be that this positive engagement is promoted by a specific combination of features only found in these lessons, explaining why off-task behavior

was not associated with any of the finer-grained features in the CTLVS1.1, once the coarser feature of whether the lesson used the equation-solver was included. Relatedly, it might be that the task of equation-solving is somehow more engaging, in and of itself, than other mathematical problem-solving tasks, leading students to engage in a lower degree of off-task behavior.

A second hypothesis is that teacher behavior causes the lower off-task behavior within the equation-solver lessons. A conversation with a colleague with school teaching experience indicated that teachers in the United States are often particularly worried about students' performance on equation-solving on state standardized exams (personal communication, L.A. Sudol). This concern may lead teachers to monitor a student more closely, if the student is working through an equation-solver lesson. This hypothesis could be tested through observing teachers' behavior with quantitative field observations [cf. 5], as students use either equation-solver lessons or other lessons. It is worth noting that this hypothesis may also help explain the lower incidence of gaming the system in equation-solving lessons [e.g. 7].

Determining which of these hypotheses best explains the lower incidence of off-task behavior in equation-solver lessons has the potential to help us understand this behavior better. In turn, this knowledge has the potential to aid us in developing learning software that students engage with to a greater degree. In doing so, it is essential to avoid decreasing off-task behavior in ways that could increase the prevalence of other behaviors associated with poorer learning, such as gaming the system. It is also essential to avoid reducing off-task behavior in ways that would make instruction generally less effective – a potential danger in many visions of educational games in the classroom.

More broadly, we believe that the methods used in this paper point to new opportunities for the field of educational data mining. The creation of taxonomies such as the CTLVS1.1 will enable an increasing number of data mining analyses about how differences in educational software concretely influence student behavior. In turn, these analyses can inform a deeper scientific understanding of the interactions between students and educational software.

Acknowledgements

References

[1] Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., Fisher, D., Barto, A., Mahadevan, S., Woolf. B.P. Repairing Disengagement with Non-Invasive Interventions. *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED-2007)*, p. 195-202.

[2] Arroyo, I., & Woolf, B. Inferring learning and attitudes from a Bayesian network of log file data. *Proceedings of the 12th International Conference on Artificial Intelligence in Education,* 2005, p. 33-40.

[3] Baker, R.S.J.d. Modeling and understanding students' off-task behavior in intelligent tutoring systems, *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, 1059-1068.

[4] 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. Adapting to When Students Game an Intelligent Tutoring System. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 2006, p. 392-401.

[5] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". *Proc. ACM CHI 2004*, p. 383-390.

[6] Baker, R.S.J.d., Corbett, A.T., Roll, I., Wagner, A.Z., Koedinger, K.R. The Relationship Between Gaming the System and Learning in Cognitive Tutor Classrooms. Manuscript under review.

[7] Baker, R.S.J.d., de Carvalho, A.M.J.B., Raspat, J., Aleven, V., Corbett, A.T., Koedinger, K.R. Educational Software Features that Encourage and Discourage "Gaming the System". *Proceedings of the International Conference on Artificial Intelligence in Education,* in press.

[8] Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., Koedinger, K. Why Students Engage in "Gaming the System" Behavior in Interactive Learning Environments. *Journal of Interactive Learning Research*, 2008, 19(2), p. 185-224.

[9] Beal, C.R., Qu, L., Lee, H. Mathematics motivation and achievement as predictors of high school students' guessing and help-seeking with instructional software. *Journal of Computer Assisted Learning,* 2008, 24 (6), p. 507-514.

[10] Beck, J.E. Using Learning Decomposition to Analyze Student Fluency Development. *Proceedings of the Educational Data Mining Workshop, at the 8th International Conference on Intelligent Tutoring Systems*, 2006, p. 21-28

[11] Cocea, M., Hershkovitz, A., Baker, R. The Impact of Off-Task and Gaming on Learning: Immediate or Aggregate? *Proceedings of the International Conference on Artificial Intelligence in Education,* in press.

[12] Corbett, A.T., Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction,* 1995, 4, p. 253-278.

[13] Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience,* 1990. New York: Harper and Row

[14] Inkpen, K.M., Ho-Ching, W., Kuederle, O., Scott, S.D., Shoemaker, G.B.D. This is fun! We're all best friends and we're all playing: supporting children's synchronous collaboration. *Proceedings of the 1999 Conference on Computer Support for Collaborative Learning.*

[15] Koedinger, K., Cunningham, K., Skogsholm A., Leber, B. An open repository and analysis tools for fine-grained, longitudinal learner data. *Proceedings of the First International Conference on Educational Data Mining,* 2008, p. 157-166.

[16] Kincaid, J. P., Fishburne, R. P., Rogers, R. L., Chissom, B. S. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel. Research Branch Report 8-75, Naval Technical Training, 1975.

[17] Lee, S.W., Kelly, K.E., Nyre, J.E. Preliminary Report on the Relation of Students' On-Task Behavior With Completion of School Work. *Psychological Reports,* 1999, 84, p. 267-272.

[18] Lloyd, J.W., Loper, A.B. Measurement and Evaluation of Task-Related Learning Behavior: Attention to Task and Metacognition. *School Psychology Review,* 1986, 15 (3), p. 336-345.

[19] Rodrigo, M.M.T., Baker, R.S.J.d., Lagud, M.C.V., Lim, S.A.L., Macapanpan, A.F., Pascua, S.A.M.S., Santillano, J.Q., Sevilla, L.R.S., Sugay, J.O., Tep, S., Viehland, N.J.B. Affect and Usage Choices in Simulation Problem Solving Environments. *Proceedings of Artificial Intelligence in Education 2007*, p. 145-152.

[20] Rosenthal, R., Rosnow, R.L. *Essentials of Behavioral Research: Methods and Data Analysis: 3rd Edition*, 1991. Boston, MA: McGraw-Hill.

[21] Skinner, E.A., Kindermann, T.A., Furrer, C. A motivational perspective on engagement and disaffection: Conceptualization and assessment of children's behavioral and emotional participation in academic activities in the classroom. *Educational and Psychological Measurement,* in press.

[22] Walonoski, J.A., & Heffernan, N.T. Prevention of off-task gaming behavior in intelligent tutoring systems. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems,* 2006, p. 722-724.