Modeling and Studying Gaming the System with Educational Data Mining

7

Ryan S.J.d. Baker, A.T. Corbett, I. Roll, K.R. Koedinger, V. Aleven, M. Cocea, A. Hershkovitz, A.M.J.B. de Caravalho, A. Mitrovic, and M. Mathews

Abstract

In this chapter, we will discuss our work to understand why students game the system. This work leverages models of student gaming, termed "detectors", which can infer student gaming in log files of student interaction with educational software. These detectors are developed using a combination of human observation and annotation, and educational data mining. We then apply the detectors to large data sets, and analyze the detectors' predictions, using discovery with models methods, to study the factors associated with gaming behavior. Within this chapter, we will discuss the work to develop these detectors, and what we have discovered through these analyses based on these detectors. We will discuss evidence for how gaming the system impacts learning and evidence for why students choose to game. We will also discuss attempts to address gaming the system through adaptive scaffolding.

R.S.J.d. Baker (✉) • A. Hershkovitz
Columbia University Teachers College,
New York, NY, USA
e-mail: ryan@educationaldatamining.org

A.T. Corbett
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Pittsburgh Advanced Cognitive Tutor Center, Carnegie Mellon University, Pittsburgh, PA, USA

I. Roll
Carl Wieman Science Education Initiative, University of British Columbia, Vancouver, BC, Canada

Pittsburgh Science of Learning Center,
Pittsburgh, PA, USA

K.R. Koedinger
Pittsburgh Science of Learning Center,
Pittsburgh, PA, USA

Human-Computer Interaction and Psychology,
Carnegie Mellon University, Pittsburgh, PA, USA

V. Aleven
Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Pittsburgh Science of Learning Center,
Pittsburgh, PA, USA

M. Cocea
School of Computing, University of Portsmouth,
Portsmouth, Hampshire, UK

A.M.J.B. de Caravalho
Human-Computer Interaction Institute, CMU Carnegie Mellon University, Pittsburgh, PA 15213, USA

A. Mitrovic • M. Mathews
Intelligent Computer Tutoring Group (ICTG),
Department of Computer Science and Software Engineering, University of Canterbury,
Christchurch, New Zealand

R. Azevedo and V. Aleven (eds.), *International Handbook of Metacognition and Learning Technologies*,
Springer International Handbooks of Education 28, DOI 10.1007/978-1-4419-5546-3_7,

Introduction

In recent years, there has been increasing awareness that students using interactive learning technologies often "game the system," defined as attempting to succeed in an educational task by systematically taking advantage of properties and regularities in the system used to complete that task, rather than by thinking through the material (Baker, Corbett, Koedinger, et al., 2006). Examples of gaming the system include misusing help features of educational software to obtain answers (Aleven, McLaren, Roll, & Koedinger, 2006), systematic guessing (Baker, Corbett, Koedinger, & Wagner, 2004), intentional rapid mistakes (Murray & VanLehn, 2005), spam postings in graded newsgroups (Cheng & Vassileva, 2005), and point cartels in collaborative games (Magnussen & Misfeldt, 2004). Analogous behaviors also occur within wholly human classrooms, where students ask teachers and teachers-aides repeatedly for answers (Nelson-Le Gall, 1985).

Gaming the system occupies an interesting place within self-regulated learning (SRL) and metacognition. In some ways, it can be considered as a behavior that requires sophisticated metacognition, involving—to quote Hacker (1999) discussing Flavell (1976)—"active monitoring and consequent regulation and orchestration of cognitive processes to achieve cognitive goals." Qualitative and quantitative analysis has suggested that students actively choose which problem steps to game on, with some students explicitly gaming specific poorly known material and other students gaming well-known material (cf. Baker, Corbett, & Koedinger, 2004). To the degree that students game the system precisely on the material that they do not know, the choice to game appears to explicitly involve "knowledge of one's knowledge" (cf. Hacker, 1999). Gaming clearly involves a substantial degree of self-regulation (Zimmerman, 2000) as well, inasmuch as the student appears to consciously choose to game as opposed to other strategies, such as attempting to seek help or answer using their knowledge (cf. Aleven et al., 2006).

However, while gaming appears to involve self-regulation, it is open to question whether gaming can be considered a strategy within SRL (cf. Butler & Winne, 1995). Many students who game the system appear not to be trying to learn at all during their gaming behavior (there are exceptions, which are discussed in this chapter). Hence, gaming the system could potentially be viewed as self-regulated behavior with the goal of avoiding learning, rather than SRL. There are several forms of self-regulation driven towards avoiding learning or effort, including self-handicapping (Midgley & Urdan, 2002) and off-task behavior (cf. Fisher & Ford, 1998). It is not clear that gaming is a form of self-handicapping, and gaming and off-task behavior appear to emerge from different motivation, at least in part (Baker, 2007b). Nonetheless, it may be valuable to conceptualize gaming in this fashion—as a self-regulated behavior but not as a strategy for SRL. Alternatively, gaming the system could be viewed as a tactic or a strategy emerging from low motivation during self-regulation, a possibility implicit within models of SRL that incorporate motivation (e.g., Winne & Hadwin, 1998). Interestingly, the one model of metacognition or SRL which explicitly incorporates gaming behaviors is Aleven and colleagues' (2006) model of help-seeking within tutors. Within this model, gaming is conceptualized as a "metacognitive bug," a cognitive rule that represents an ineffective or maladaptive form of help-seeking.

However, there is increasing evidence that gaming is more than simply an ineffective or maladaptive form of help-seeking. First of all, as we discuss in this chapter, there are multiple ways that students game. For instance, some students appear to game on time-consuming steps that they already know, potentially to spend more time on what they need to learn (Baker, Corbett, & Koedinger, 2004). Other students game in order to obtain answers more quickly, and then self-explain those answers (Shih, Koedinger, & Scheines, 2008). Gaming in these fashions may therefore be a strategy within sophisticated self-regulatory behavior (cf. Winne & Hadwin, 1998). Secondly, there is recent evidence that the triggers of gaming the system include features of the design of intelligent tutors (discussed in this chapter), and the student emotion immediately prior to gaming (Baker, D'Mello, Rodrigo, & Graesser, 2010). As such, it appears that gaming emerges from relatively complex self-regulatory processes, involving several factors, including an assessment of the current situation and the student's emotion.

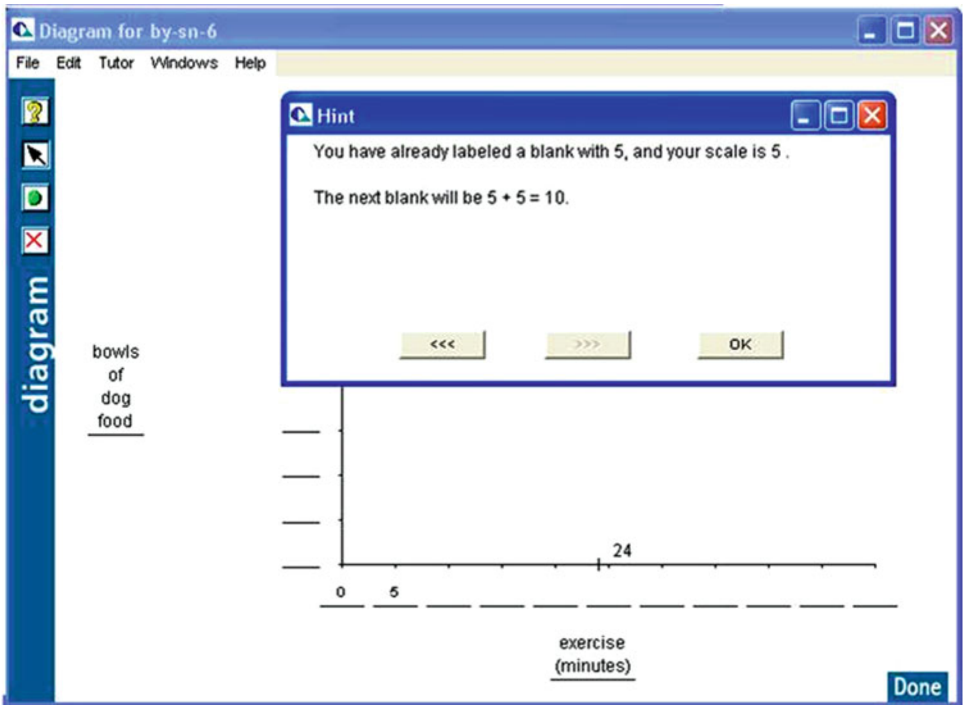
In this chapter, we discuss our work to understand why students game the system. This work leverages models of student gaming, termed "detectors," which can infer student gaming in log files of student interaction with educational software. These detectors are developed using a combination of human observation and annotation, and educational data mining (Baker & Yacef, 2009; Romero & Ventura, 2010). We then apply the detectors to large data sets, and analyze the detectors' predictions, using discovery with model methods (Baker & Yacef, 2009), to study the factors associated with gaming behavior. Within this chapter, we discuss the work to develop these detectors, and what we have discovered through these analyses based on these detectors. We discuss evidence for how gaming the system impacts learning and evidence for why students choose to game. We also discuss attempts to address gaming the system through adaptive scaffolding.

Contexts of Detector Development and Use

Gaming the system has been studied in a variety of learning systems (Baker, Corbett, Koedinger, & Wagner, 2004; Baker, D'Mello, et al., 2010; Baker, Mitrovic, & Mathews, 2010; Beal, Qu, & Lee, 2006; Beck, 2005; Gobel, 2008; Johns & Woolf, 2006; Muldner, Burleson, Van de Sande, & VanLehn, 2011; Murray & VanLehn, 2005; Walonoski & Heffernan, 2006a). In this chapter, we focus on the research into gaming the system within Cognitive Tutors, though we briefly discuss research in other learning systems as well. A key advance that has supported research on gaming the system in recent years has been the advent of models that assess whether a student is gaming, often termed "detectors" (e.g., Aleven et al., 2006; Baker, Corbett, & Koedinger, 2004; Baker, Corbett, Roll, & Koedinger, 2008; Baker, Mitrovic et al., 2010; Beal et al., 2006; Beck, 2005; Johns & Woolf, 2006; Muldner et al., 2011; Walonoski & Heffernan, 2006a). Cognitive Tutors were the first type of learning environment for which gaming detector development occurred; they are also the environment for which gaming detectors have been most thoroughly validated, and for which gaming detectors have been used in the largest number of "discovery with models" analyses.

Cognitive Tutors are a popular type of interactive learning environment now used by around half a million students a year in the USA, in particular for high school Algebra and Geometry (Koedinger & Corbett, 2006). Cognitive Tutor curricula combine conceptual instruction delivered by a teacher with problem-solving where each student works one on one with a cognitive tutoring system which chooses exercises and feedback based on a running model of which skills the student possesses (Koedinger & Corbett, 2006). Within this chapter, we focus on students' online problem-solving. We discuss results from the Middle School Mathematics Cognitive Tutor, shown in Fig. 7.1, and the Algebra Tutor, shown in Fig. 7.2. In its original version, the Middle School Tutor was used by the U.S. middle school students, who are typically between approximately 11 and 14 years old. The Middle School Tutor has become Bridge to Algebra, and is now in use in high schools and middle schools across the USA (we refer to it by its original name, as this was the version used in the research presented in this chapter). The Algebra Tutor is typically used in the U.S. high schools, where students typically range from 14 to 18 years old.

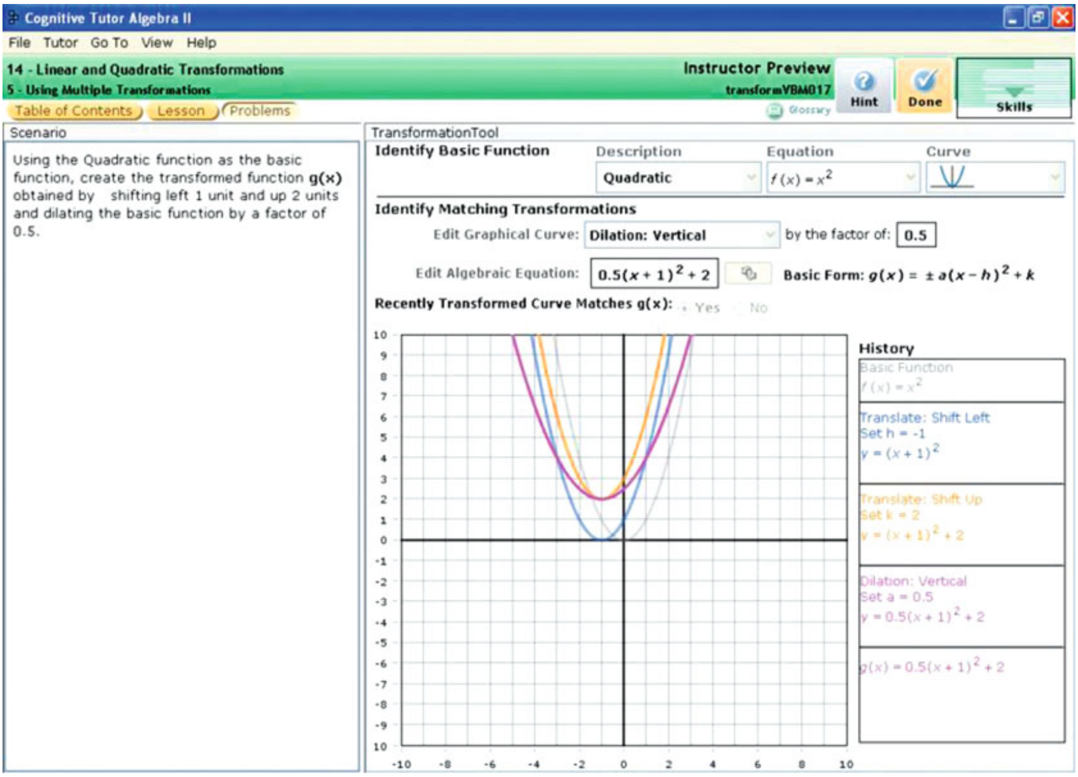
Cognitive Tutor learning environments are designed to promote learning by doing. Within the Cognitive Tutor environments discussed within this chapter, each student individually completes mathematics problems. The Cognitive Tutor environment breaks down each mathematics problem into the steps of the process used to solve the



You have already labeled a blank with 5, and your scale is 5 .

The next blank will be 5 + 5 = 10.

Fig. 7.1 A screenshot from the Cognitive Tutor for Middle School Mathematics



Cognitive Tutor Algebra II
File Tutor Go To View Help
14 - Linear and Quadratic Transformations
5 - Using Multiple Transformations
Instructor Preview
transformVBM017
Hint
Done
Skills
Glossary
Table of Contents
Lesson
Problems
Scenario
TransformationTool

Using the Quadratic function as the basic function, create the transformed function g(x) obtained by shifting left 1 unit and up 2 units and dilating the basic function by a factor of 0.5.

Identify Basic Function

Description: Quadratic

Equation: $f(x) = x^2$

Curve

Identify Matching Transformations

Edit Graphical Curve: Dilation: Vertical by the factor of: 0.5

Edit Algebraic Equation: $0.5(x+1)^2+2$ Basic Form: $g(x) = \pm a(x-h)^2+k$

Recently Transformed Curve Matches g(x): ○ Yes ○ No

History

Basic Function
$f(x) = x^2$

Translate: Shift Left
Set h = -1
$y = (x+1)^2$

Translate: Shift Up
Set k = 2
$y = (x+1)^2 + 2$

Dilation: Vertical
Set a = 0.5
$y = 0.5(x+1)^2 + 2$

$g(x) = 0.5(x+1)^2 + 2$

Fig. 7.2 A screenshot from the Algebra Cognitive Tutor

problem, making the student's thinking visible. As a student works through a problem, a running cognitive model assesses whether the student's answers map to correct understanding or to a known misconception (cf. Anderson, Corbett, Koedinger, & Pelletier, 1995). If the student's answer is incorrect, the answer turns red; if the student's answers are indicative of a known misconception, the student is given a "buggy message" indicating how their current knowledge differs from correct understanding. Cognitive Tutors also have multistep hint features; a student who is struggling can ask for a hint. He or she first receives a conceptual hint, and can then request further hints, which become more and more specific until the student is given the answer. The hints are context-sensitive and tailored to the exact problem step the student is working on. As the student works through the problems in a specific curricular area, the system uses Bayesian Knowledge-Tracing (Corbett & Anderson, 1995) to determine which skills that student is having difficulty with, calculating the probability that the student knows each skill based on that student's history of responses within the tutor. Using these estimates of student knowledge, the tutoring system gives each student problems which are relevant to the skills which he or she is having difficulty with. Cognitive Tutor material is typically structured into independent lessons, each of which covers a set of related skills and concepts. Year-long courses are composed of sequences of lessons, where the knowledge in later lessons generally builds upon the knowledge in previous lessons.

Detector Development

Detectors of gaming the system can be developed in several ways. While many researchers have utilized knowledge engineering to develop detectors of gaming the system (cf. Aleven et al., 2006; Gong, Beck, Heffernan, & Forbes-Summers, 2010; Johns & Woolf, 2006; Muldner et al., 2011), our research group has emphasized machine learning/data mining approaches, in order to support more thorough model validation. We believe that comprehensive validation is essential when using detectors to support research in the complex phenomena found in metacognition and SRL; without high confidence in a detector's validity and generalizability, it is difficult to have confidence in the results obtained from analyzing a detector's output. Within this section, we present our work to develop and validate detectors of gaming the system. A fuller discussion of the trade-offs between machine learning and knowledge engineering approaches for modeling student behaviors, such as gaming the system, can be found in Baker (2010).

Our approach to developing gaming detectors is as follows. We first use human labeling methods to gather "ground truth" labels of students or actions judged to be gaming the system. We then use data mining methods to distill these labels into reusable detectors of gaming. We then validate these models at multiple levels, including generalizability to new students and lessons, and temporal precision. We discuss these steps, as well as some challenges that need to be met for these detectors to be maximally useful for the field.

Human Labeling Methods

Within our research, we have used two methods for humans to label gaming the system. The first is *quantitative field observations* (Baker, Corbett, Koedinger, & Wagner, 2004; Karweit & Slavin, 1982). Quantitative field observations are repeated observations of students (in this case, whether they are gaming the system or not), conducted according a predefined coding scheme and observation method. Within our observations of gaming the system, each observation lasted 20 s, and was conducted using peripheral vision. That is, the observers stood diagonally behind or in front of the student being observed and avoided looking at the student directly (cf. Baker, Corbett, Koedinger, & Wagner, 2004), in order to make it less clear when an observation was occurring. If two distinct behaviors were seen during an observation, only the first behavior observed was coded. Any behavior by a student other than the student currently being observed was not coded. Observations are in some cases carried out by single observers and other times by observational pairs. Inter-rater reliability on assessments of gaming using this method have been calculated at over 0.7 across several studies involving different coders (Baker, Corbett, & Wagner, 2006; Baker, D'Mello, et al., 2010; Rodrigo et al., 2008). Another benefit of quantitative field observation is that it can be used for a variety of constructs (including affect as well as behavior—cf. Baker, D'Mello, et al., 2010; Rodrigo et al., 2008). The method's key disadvantages are that it is time-consuming, and it has historically been challenging to synchronize exactly between field observations and log files. (Our research group has recently developed a handheld observation application which synchronizes to the same time server as the software logs; we believe that this will substantially reduce challenges to synchronization.)

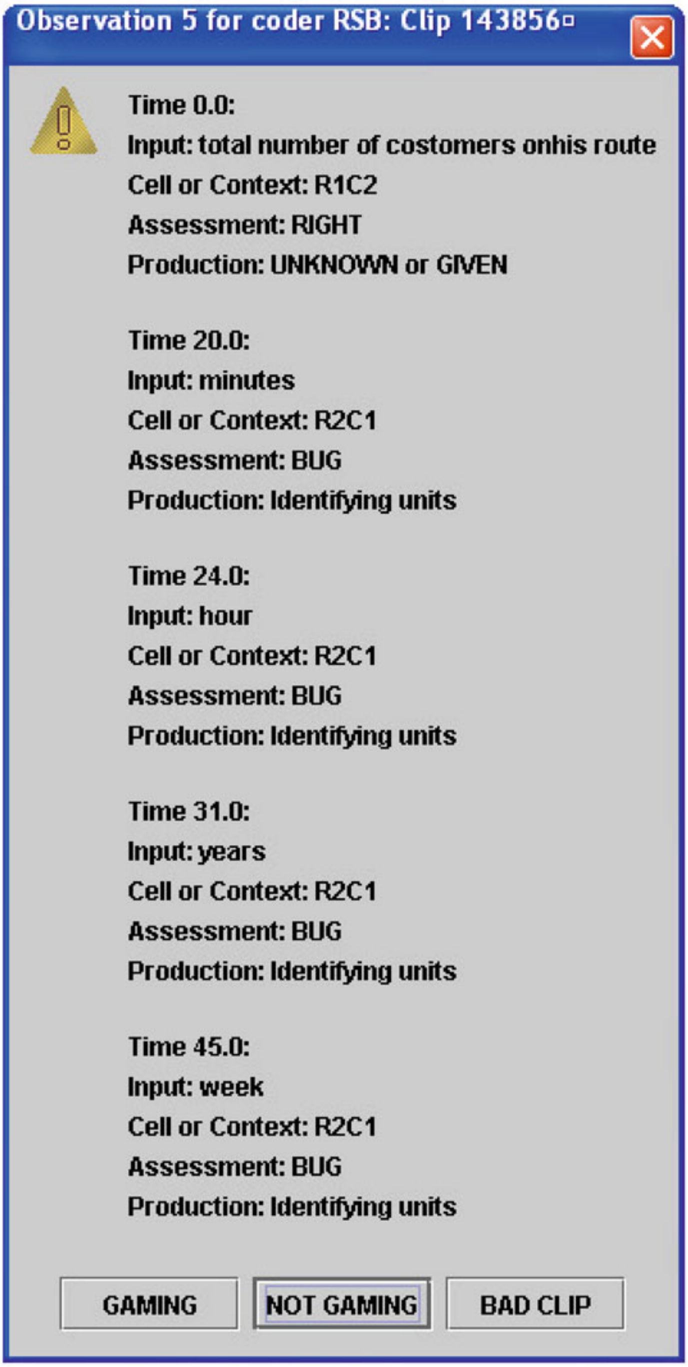
The second method we have used for humans to label gaming the system is *text replays* (Baker, Corbett, & Wagner, 2006; Baker & de Carvalho, 2008; Baker, Mitrovic, et al., 2010). Text replays represent a segment of student behavior from the log files in a textual ("pretty-printed") form. A sequence of actions of a preselected duration (in terms of time or length) is shown in a textual format that gives information about the actions and their context. In the example shown in Fig. 7.3, the coder sees each action's time (relative to the first action in the clip), the problem context, the input entered, the relevant skill (production), and how the system assessed the action (correct, incorrect, a help request, or a "bug"/misconception). The coder can then choose one of a set of behavior categories (in this study, gaming or not gaming), or indicate that something has gone wrong, making it impossible to code the clip. Text replays give relatively limited information, compared to quantitative field observations; however, text replays are very quick to classify, between two and ten times faster than quantitative field observations (Baker, Corbett, & Wagner, 2006; Baker & de Carvalho, 2008), and can be generated automatically from existing log files, enabling retrospective analysis. Inter-rater reliability has been found to be comparable to quantitative field observations, ranging between 0.58 and 0.80 (Baker, Corbett, & Wagner, 2006; Baker, D'Mello, et al., 2010), though it typically requires multiple rounds of training to get convergent categorization (Baker, D'Mello, et al., 2010; Sao Pedro, Baker, Montalvo, Nakama, & Gobert, 2010).

Educational Data Mining Methods Used

All of our detectors of gaming are based upon a distillation of features of students' actions within the tutoring software. For Cognitive Tutors, for each student action recorded in the log files, a set of 26 features describing that student action were distilled. These features included the following (an exhaustive list is given in Baker, Corbett, et al., 2008):

- Details about the action
  - The tutoring software's assessment of the action—Was the action correct, incorrect and indicating a known bug (procedural misconception), incorrect but not indicating a known bug, or a help request?
  - The type of interface widget involved in the action.
  - Was this the student's first attempt to answer or obtain help on this problem step?
- Knowledge assessment
  - The tutor's assessment, after the action, of the probability that the student knows the skill involved in this action, derived using the Bayesian knowledge tracing algorithm in Corbett and Anderson (1995).
  - Whether the action involved a skill which students, on the whole, knew before starting the tutor lesson, or failed to learn during the tutor lesson.
- Time
  - How long the action took, both in absolute time and in standard deviations faster or slower than the mean time taken by all students on this problem step, across problems (e.g., unitized time).
  - Unitized time across the last 3, or 5, actions.

Fig. 7.3 A text replay of student gaming behavior



Observation 5 for coder RSB: Clip 143856▫

Time 0.0:
Input: total number of costomers onhis route
Cell or Context: R1C2
Assessment: RIGHT
Production: UNKNOWN or GIVEN

Time 20.0:
Input: minutes
Cell or Context: R2C1
Assessment: BUG
Production: Identifying units

Time 24.0:
Input: hour
Cell or Context: R2C1
Assessment: BUG
Production: Identifying units

Time 31.0:
Input: years
Cell or Context: R2C1
Assessment: BUG
Production: Identifying units

Time 45.0:
Input: week
Cell or Context: R2C1
Assessment: BUG
Production: Identifying units

- Previous interaction
  - The total number of times the student has gotten this specific problem step wrong or asked for help, across all problems (includes multiple attempts within one problem).
  - How many recent actions involved this problem step, help requests, or errors?

Our research group has used two primary methods to develop detectors of gaming the system for Cognitive Tutors: Latent Response

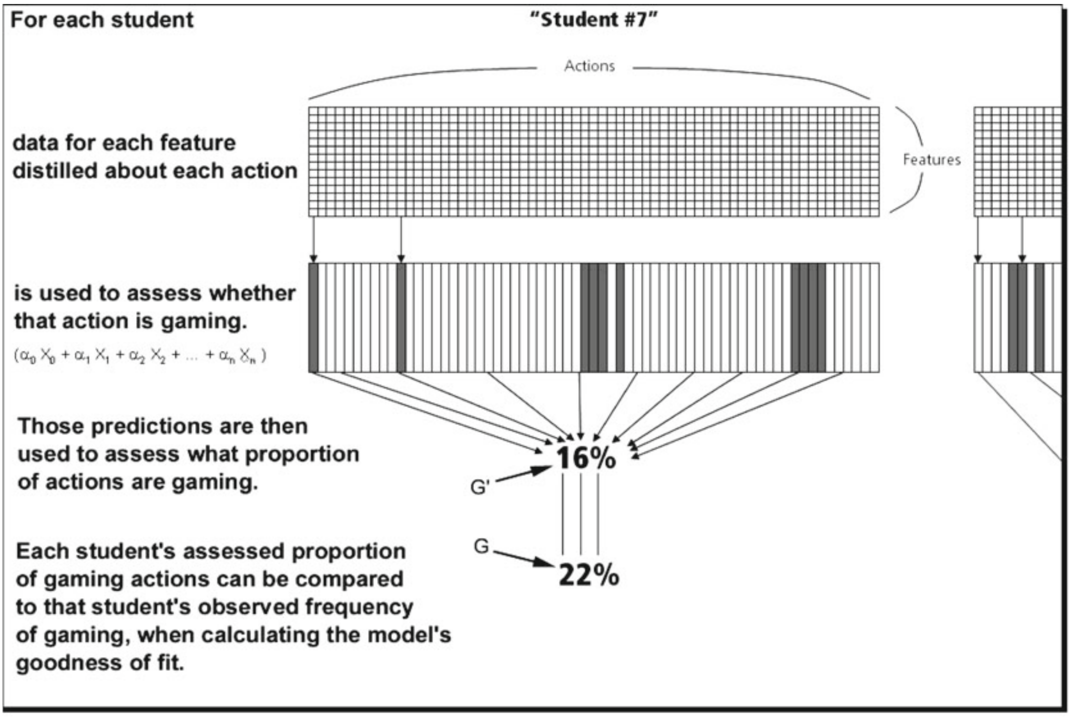

For each student
"Student #7"
Actions
data for each feature distilled about each action
Features
is used to assess whether that action is gaming.
$(\alpha_0 X_0 + \alpha_1 X_1 + \alpha_2 X_2 + \ldots + \alpha_n X_n)$
Those predictions are then used to assess what proportion of actions are gaming.
G'
16%
Each student's assessed proportion of gaming actions can be compared to that student's observed frequency of gaming, when calculating the model's goodness of fit.
G
22%

Fig. 7.4 The architecture of a gaming detector based on a Latent Response Model

Models (Maris, 1995) and J48 Decision Trees (an open-source variant published by Witten and Frank, 2005, of the C4.5 algorithm developed by Quinlan, 1993). J48 Decision Trees were first used to detect gaming the system by Walonoski and Heffernan (2006a). Recent work on detecting gaming the system in SQL-Tutor has also used step regression (Baker, Mitrovic, et al., 2010), an approach similar to the internal step function in our Latent Response Model approach discussed below.

Latent Response Models have the advantage of easily and naturally integrating multiple data sources, at different grain-sizes, into a single model. They can be used when data is not well synchronized. A detector of gaming, in the framework used here, has one observable level and two hidden ("latent") levels. The model's overall structure is shown in Fig. 7.4. In a gaming detector's outermost/observable layer, the gaming detector assesses how frequently each of n students is gaming the system; those assessments are labeled G 0, …, G n. The gaming detector's assessments for each student can then be compared to the observed proportions of time each student spent gaming the system, G0, …, Gn (the metrics used will be discussed within the next section). The proportion of time each student spends gaming is assessed as follows: First, the detector makes a (binary) assessment as to whether each individual student action (denoted P m) is an instance of gaming. From these assessments, G 0, …, G n are derived by taking the percentage of actions which are assessed to be instances of gaming, for each student. An action is assessed to be gaming or not, by a function on parameters composed of the features drawn from each action's characteristics. An assessment Hm as to whether action m is an instance of gaming is computed as Hm = a0·X0 + a1·X1 + a2·X2 + … + a n·Xn, where ai is a parameter value and Xi is the data value for the corresponding parameter, for this action, in the log files. The value given by the linear combination is the first hidden level and top layer in Fig. 7.4. Each assessment Hm is then thresholded using a step function, such that if Hm £ 0.5, $H\,m=0$; otherwise $H\,m=1$. The set of thresholded values makes up the second hidden level and middle layer in Fig. 7.4. This gives us a set of classifications $H\,m$ for each action within the tutor, which are then used to create the assessments of each student's proportion of gaming, $G\,0, \ldots, G\,n$. These assessments of each student's proportion of gaming, which make up the observable level of the model (the bottom layer in Fig. 7.2), are compared to the observed values of gaming during model fitting and validation. Within the model framework, the best model is selected out of the large space of possible models, using a combination of Fast Correlation-Based Filtering (Yu & Liu, 2003) and Forward Selection (Ramsey & Schafer, 1997).

J48 decision trees, the second method used to detect gaming in Cognitive Tutors, are a standard data mining method. As such, J48 has a single level of hierarchy and can be used when specific actions are known to involve gaming the system or to not involve gaming the system, requiring text replays or good synchronization of field observations. In the case of text replays, we label segments of behavior as gaming or not gaming; if a segment is labeled as involving gaming, every action in the segment is labeled as gaming. It is worth noting that these labels of individual actions cannot be considered perfectly accurate, since the observer labeled a clip as "gaming" if any of the actions in the clip involved gaming. Therefore, actions at the beginning or end of clips may not in all cases be instances of gaming. This suggests that, within text replay data, a 100% perfect match between our classifier's labels of individual actions and those actions' labels is not necessary (or desirable). This limitation could be addressed by having observers explicitly label which actions in a clip are gaming, but would have the cost of reducing the method's speed. J48 decision trees are a good approach for noisy data of this nature, as the pruning step of this algorithm addresses noise in the data and reduces over-fitting.

Validation Methods and Effectiveness

In order to validate the effectiveness of a detector of the types discussed here, and its appropriateness for different types of use, it is important to analyze its generalizability at multiple levels. Four types of generalizability are particularly important for a detector that will be used in "discovery with models" analyses. First, a detector should be able to accurately determine which students game, even for entirely new students. This is important, because it enables the detector to be used with new students, for instance at run-time, or in larger data sets than the original training set. In order to do this, it is necessary to train a detector with one group of students and test it with a different group of students. Cross-validation is a systematic method for splitting up a data set into groups and testing model generalizability across groups (Efron & Gong, 1983). However, one limitation is that many existing tools for data mining, such as Weka (Witten & Frank, 2005), do not support student-level cross-validation, only supporting cross-validation at the grain-size of individual data points. Another tool, RapidMiner (Mierswa, Wurst, Klinkenberg, Scholz, & Euler, 2006), does not directly support student-level or lesson-level cross-validation, but its "batch cross-validation" functionality makes it possible to conduct student-level or lesson-level cross-validation through predefining student batches outside of the data mining software. Student-level cross-validation has been conducted for gaming detectors based on Latent Response Models (e.g., Baker, Corbett, et al., 2008) and J48 Decision Trees. Latent Response Models appear to achieve the goal of detecting which students game more successfully than J48 Decision Trees (Baker & de Carvalho, 2008).

Second, a detector should be able to accurately determine exactly when a student games. This is important, because it enables inference about the context and antecedents and immediate consequences of gaming behavior. Determining exactly when each student games is not possible without synchronized observations or text replays, since exact labels are needed. In Baker and de Carvalho (2008), Latent Response Models were compared to J48 Decision Trees in terms of ability to determine this, and J48 Decision Trees performed significantly better. There was evidence that the Latent Response Model identified gaming on the correct skills, but identified gaming the system later than it actually occurred.

Third, a detector should be able to transfer to new classrooms and schools. This is important because some behaviors may differ in important ways between students across different classroom cultures. This type of validation is generally rare because of the difficulty of collecting and labeling data sets that span significant numbers of classrooms. Latent Response Models of gaming the system have been validated in this fashion (Baker, Corbett, Koedinger, & Roll, 2005).

Fourth, a detector should be able to transfer to new tutor lessons or related tutors. This is important because many modern intelligent tutoring systems, including Cognitive Tutors, cover a significant number of topics that necessarily differ in presentation and user interaction, over the course of a semester or a year (cf. Koedinger & Corbett, 2006). It is important for analyses spanning across these topics to be based on detectors validated to be accurate across all of the interaction contexts where the detectors are applied. Latent Response Models of gaming the system have been validated in this fashion (Baker, Corbett, et al., 2008).

"Harmful" Gaming and "Non-harmful" Gaming

One finding in the development of detectors of gaming the system which has not been fully explained is the possible split between "harmful" and "non-harmful" forms of gaming, defined as forms of gaming behavior associated with differential learning outcomes. In specific, "harmful" forms of gaming can be conceptualized as gaming associated with poor performance on the posttest (e.g., the failure to learn), whereas "non-harmful" forms of gaming are not associated with poor posttest performance (Baker, Corbett, & Koedinger, 2004). Within a Cognitive Tutor for middle school mathematics, a replicable split (replicable across lessons) has been found between gaming students who perform poorly on the posttest, and gaming students who nonetheless still perform well on the posttest. This split is sufficiently strong that detectors can be trained to detect students in either category, not detecting students in the other category (Baker, Corbett, et al., 2008); in at least one data set the attempt to detect both groups only succeeded in detecting gaming students who perform poorly on the posttest (Baker, Corbett, & Koedinger, 2004). This split appears to be between gaming that occurs on poorly known skills (harmful gaming) and gaming that occurs on well-known skills (non-harmful gaming), thus far, this split has failed to replicate within other systems and populations, including middle school students using Math ASSISTments (Walonoski & Heffernan, 2006a), high school students using a Cognitive Tutor for Algebra (Baker & de Carvalho, 2008), and college students using SQL-Tutor (Baker, Mitrovic, et al., 2010). It is not clear what aspect of the middle school mathematics Cognitive Tutor or its population leads to the split in types of gaming, but it is an important area of future research.

A third type of gaming, not explicitly studied in our research, is the beneficial form of gaming discovered by Shih et al. (2008) in high school students using a Cognitive Tutor for Geometry. In this behavior, a student clicks through hints in order to receive the answer to a problem step, but then stops and self-explains the step before proceeding. This behavior is associated with positive learning gains, and is likely to be a way of turning tutoring into a worked example. We view this behavior as a positive metacognitive strategy that is only related to gaming the system at a surface level.

Challenges

One of the key challenges to studying gaming (or metacognitive behavior in general) at scale is generalizability. Even though the generalizability of gaming detectors has been validated across students, and across tutor lessons, all validation has been within the context of specific intelligent tutors. Gaming detectors have been developed through the process discussed here for multiple intelligent tutors, including Cognitive Tutors for Algebra (Baker & de Carvalho, 2008) and middle school mathematics (Baker, Corbett, et al., 2008; Baker, Walonoski, et al., 2008), and a constraint-based tutor for SQL (Baker, Mitrovic, et al., 2010). However, the detectors developed for Cognitive Tutors have had relatively little obviously in common, feature-wise, with the detectors of gaming for SQL-Tutor. The features distilled from log data have themselves had fairly little in common between tutors. This lack of commonality limits the broader generalization of gaming research, as the entire process of labeling data, distilling data features, developing a detector, and validating generalizability must be undertaken for any new learning system. Our research group is currently attempting to address this limitation, by building gaming detectors for multiple learning systems for which there exists data in a standardized format in the Pittsburgh Science of Learning Center DataShop (Koedinger et al., 2010). The hope is that by studying generalizability across learning systems within data collected in the same standardized format, we can learn whether gaming the system has common features across learning systems that can be used as the basis of gaming detection that generalizes across learning systems.

Another key challenge is balancing between detecting exactly when a student is gaming, and detecting which students game. Our initial investigations (Baker & de Carvalho, 2008) appear to suggest that J48 Decision Trees are more successful at detecting the exact moment of gaming, while Latent Response Models are more successful at detecting exactly which students game. Both of these goals are clearly important. One immediate takeaway message is that the selection of algorithm should be based upon which of these goals is more important for model usage. For instance, analyzing the different rates of gaming across schools depends on higher accuracy as to which students game, whereas analyzing the antecedents and consequents of gaming behavior depends upon higher accuracy as to exactly when students game. Interventions, in general, are probably more important to target towards the right students than towards the right moments. In the long term, it will be valuable to develop modeling approaches that optimize simultaneously on both of these goals, or at least balance between accuracy on the two goals.

Use in "Discovery with Models" Analyses

In this section we discuss the utilization of gaming detectors in "discovery with models" analyses. Discovery with models is defined as taking a model of a phenomenon developed via prediction, clustering, or knowledge engineering, and then using this model as a component in another type of analysis (Baker & Yacef, 2009). We will present two discovery with models analyses, which establish the potential of this class of research method to support the development of future models and theories of SRL and metacognition.

Studying Why Gaming Leads to Poorer Learning

A negative association between gaming the system and learning has been seen in most of the studies investigating this relationship (Aleven et al., 2006; Baker, Corbett, & Koedinger, 2004; Baker, Corbett, Koedinger, & Wagner, 2004; Baker, Corbett, Koedinger, et al., 2006; Walonoski & Heffernan, 2006a), though exceptions exist (e.g., Gobel, 2008). However, up until the publication of a discovery with models analysis of this relationship (e.g., Cocea, Hershkovitz, & Baker, 2009), it was not clear what mechanism might be leading to this relationship. Cocea et al. (2009) examined whether this relationship was the result of gaming leading to less learning within individual problem steps, an immediate harmful impact due to gaming. In order to analyze these possibilities, a validated Latent Response Model of gaming was applied to data from four tutor lessons (scatterplots, geometry, percents, and probability), drawn from a middle school Cognitive Tutor mathematics curriculum (Koedinger & Corbett, 2006).

We assessed whether gaming the system was associated with immediate poorer learning, by setting up a logistic regression model similar to the approach in Beck's (2006) learning decomposition method, where learning over time is assessed in terms of events that occur in the student's learning process. Performance on a given skill at a given time was predicted based on the number of steps on this skill where the student previously engaged in gaming behavior; we distinguish between "harmful gaming" (HG) steps and "non-harmful gaming" (NHG) steps. Within the model, harmful gaming was statistically significantly associated with less learning ($p<0.01$), at the step-by-step grain-size. Surprisingly, NHG was also associated with less learning at the step-by-step grain-size, though only to about half the degree of harmful gaming, and only marginally significantly ($p=0.054$). In other words, student performance improves less over time if the student games the system, as compared to the other potential learning strategies the student could have used. Off-task behavior, by contrast, was not associated with poorer immediate performance improvement. Complete details on this analysis are given in Cocea and colleagues (2009).

Studying Why Students Game the System

Discovery with models methods were also used to study why students game the system. Broadly, two classes of hypothesis have been advanced for why students game the system. First, researchers have hypothesized that some individual difference leads students to game the system (Arroyo & Woolf, 2005; Baker, Walonoski, et al., 2008; Beal, Qu, & Lee, 2008; Martínez Mirón, du Boulay, & Luckin, 2004). Second, researchers have hypothesized that aspects of software design lead students to game the system (Magnussen & Misfeldt, 2004; Baker et al., 2009). Discovery with models analyses have been used to study both of these possibilities within Cognitive Tutors.

Baker and colleagues (Baker, Walonoski, et al., 2008) applied gaming detectors to two data sets of usage of the middle school mathematics Cognitive Tutor. The students in these data sets had also completed questionnaires measuring a range of moti-vational and attitudinal constructs, including grit (Duckworth, Peterson, Matthews, & Kelly, 2007), performance goals (Dweck, 2000), anxiety, negative attitudes towards mathematics, and negative attitudes towards computers. Though some constructs were statistically significantly associated with gaming the system (specifically, grit, negative attitudes towards mathematics, and negative attitudes towards computers), none accounted for more than 5% of the variance in how much a student gamed ($r^2<0.05$). A similar pattern, with significant but weak correlations between learner characteristics and gaming frequency, was found in other learning systems (Arroyo & Woolf, 2005; Baker, Walonoski, et al., 2008). Beal and colleagues (2008) also reported statistically significant relationships between learner characteristics and gaming frequency, but did not report the magnitude of the correlations or other measures of effect size.

Following on this research, Baker (2007a) attempted to determine whether these prior results were the result of investigating the wrong learner characteristics, by assessing the overall predictive power of knowing which student was gaming the system. In doing so, this analysis treated the student as a proxy for the combination of all explanations stemming from learner characteristics. This analysis applied the Latent Response Model gaming detector validated to transfer across students and tutor lessons (Baker, Corbett, et al., 2008) to every action by a set of students during the use of the middle school mathematics Cognitive Tutor, a data set of 240 students using 35 Cognitive Tutor lessons during the course of a school year. Within this data set, the student predicted 16% of the variance in gaming whereas the lesson predicted 55% of the variance in gaming. Recent results within the

Andes system and ASSISTments attempting to predict gaming the system by student and problem have obtained a strong opposite result, with student predicting gaming significantly better than problem (Gong et al., 2010; Muldner et al., 2011). It is not yet clear why such contradictory results have been obtained; in particular, it is possible that the difference stems from the difference between the learning systems or the difference in the definition of gaming (the definitions of gaming used in the Gong et al., 2010 and Muldner et al., 2011 analyses were knowledge-engineered, and to the best of our knowledge have not yet been validated against human labels of gaming).

Following up on the apparent strong relationship between the lesson and the amount of gaming in Cognitive Tutors, Baker and colleagues (2009) investigated which specific differences between lessons predicted gaming. In this case, automated gaming detection was not used, in case specific lessons might be mis-predicted, biasing the model. Although overall generalizability of the gaming detector across lessons has been validated (e.g., Baker, Corbett, et al., 2008), it is still possible that generalization might fail for a specific lesson. If that lesson exemplified a specific set of rare lesson features, those features could be spuriously predicted to lead to gaming (or to reduce gaming). Hence, instead, text replay labels of gaming were used. A set of 79 features of tutor lessons were developed and applied to 22 lessons in a Cognitive Tutor for Algebra. Then, Principal Component Analysis was used to group the 79 features into six components. One component was predictive of gaming, predicting 29% of the variance in gaming. Two additional features were added through forward selection. The eventual best model predicting gaming through lesson features predicted 56% of the variance in gaming, roughly five times the degree of variance in gaming predicted by any prior study predicting gaming with specific student individual differences. The lesson features that predicted gaming the system, either as part of the component or as individual features, included the following:

- The same number is used for multiple constructs [more gaming].
- Hints do not lead to better future performance [more gaming].
- Hints are abstract [more gaming].
- Toolbar icons are unclear [more gaming].
- Lack of interest-increasing text in problem statements [more gaming].
- Lack of problem statement [less gaming].
- Directional feedback given [less gaming].
- Hints request that student perform some action [more gaming].
- Location of the first problem step is not directly indicated and does not follow standard conventions (e.g., being the top-left cell of a worksheet) [more gaming].

Overall, many of these lesson features can be interpreted in the following fashion: lesson features that could be expected to cause boredom or confusion are associated with more gaming. This finding accords with work studying the affective antecedents of gaming behavior (e.g., Baker, D'Mello, et al., 2010). However, many other features that also might have been expected to cause boredom or confusion were not associated with more gaming (a full list of the lesson features can be found in Baker et al., 2009). Hence, the factors mediating the relationship between lesson features and gaming are still not fully understood. However, the relationship between gaming and specific lesson features seems established, at least within Cognitive Tutors.

Potential to Contribute to Future Models and Theories of Self-Regulated Learning and Metacognition

This work has the potential to contribute to future models and theories of SRL and metacognition in at least two ways.

First, this work establishes key findings about gaming the system, a behavior that appears to involve sophisticated metacognition and self-regulation (as discussed in the introduction), but which appears to have the goal of avoiding learning rather than being an SRL behavior. Research in the last 5 years has indicated that gaming behaviors are found in a wide variety of learning systems, and analogues can also be seen even in wholly human classrooms as well (e.g., Nelson-Le Gall, 1985). Depending on how and when students game, the impacts on learning appear to differ. Current theories of metacognition and SRL do not explicitly incorporate gaming the system and similar disengaged behaviors (e.g., off-task behavior and carelessness), with the exception of Aleven and colleagues' (2006) help-seeking model. That model does an excellent job of integrating gaming into consideration of complex phenomena. However, that model's conceptualization of gaming as metacognitive bugs does not appear to fully represent the complex self-regulation and metacognition that appear to be associated with gaming, including consideration of the current learning situation (inferable from the relations between tutor design features and gaming), and the student's current emotions. As such, models of SRL and motivation which incorporate gaming will need to explicitly model the motivation, affective, and situational factors which precede gaming behavior, as well as how gaming (in its various forms) influences learning. This type of linkage is present, at a high level, in existing models of SRL (cf. Winne & Hadwin, 1998). The work presented here represents a step towards making these links concrete and specific, towards models of SRL are increasingly precise.

Second, this work serves as an example for how educational data mining methods can be integrated into future research in SRL and metacognition. Increasingly, research into metacognition and motivation in interactive learning environments leverages models of student behavior (examples relevant to gaming behavior include Aleven et al., 2006; Beal et al., 2008; Beck, 2005; Gong et al., 2010; Muldner et al., 2011; Shih et al., 2008). However, the work presented here goes to a further degree than most other work in attempting to validate construct validity (through connecting to a significant volume of human labels of the constructs of interest) and generalizability (through cross-validating models across contexts as well as students). A fuller discussion of the benefits of using human labels and generalizability analysis in development of student metacognitive models is out of the scope of this chapter, but one such discussion can be found in Baker (2010). In general, the endeavor of using student models to computationally study student metacognition will be facilitated by improving the reliability and validity of our models.

Design Implications: How to Reduce Gaming

As we improve our understanding of why students game the system, we can begin to think about developing learning environments that adapt in a relevant and purposeful way to gaming when it occurs. In recent years, there have been a number of attempts to develop systems that adapt to gaming in a productive and constructive fashion, or to address gaming by preventing it from ever occurring.

The first way that developers of educational software attempted to address gaming was by attempting to eliminate gaming by making it more difficult to game. For instance, both Cognitive Tutors and AnimalWatch adopted the strategy of putting delays between hint messages (e.g., Beck, 2005; Murray & VanLehn, 2005). Each time a student received a hint, the option to request the next hint was grayed out for several seconds. However, Murray and VanLehn (2005) found that students simply found alternate ways to game the system.

A second approach towards reducing gaming was to give students feedback on the metacognition associated with gaming (according to the model in Aleven et al., 2006), as soon as gaming behavior was recognized (Roll, Aleven, McLaren, & Koedinger, 2007). This feedback suggested that a student who games help should slow down and read the hints more carefully, and that a student who responds too quickly (a proxy for guessing behaviors) should slow down and either request a hint or try to figure out the answer. This system was successful at reducing students' degrees of these behaviors as they used the tutoring system, and led to long-term positive changes in help-seeking behavior (Roll, Aleven, McLaren, & Koedinger, 2011), but had no impact on domain learning (Roll et al., 2007, 2011).

Based on the low success of this generation of gaming interventions at improving domain learning (despite the success of both interventions at changing student behavior), a second generation of gaming interventions attempted to address gaming through introducing more complex interactions intended to impact students' awareness of gaming by communicating gaming's prevalence via visualizations or attempting to mitigate its effects through cognitive interventions.

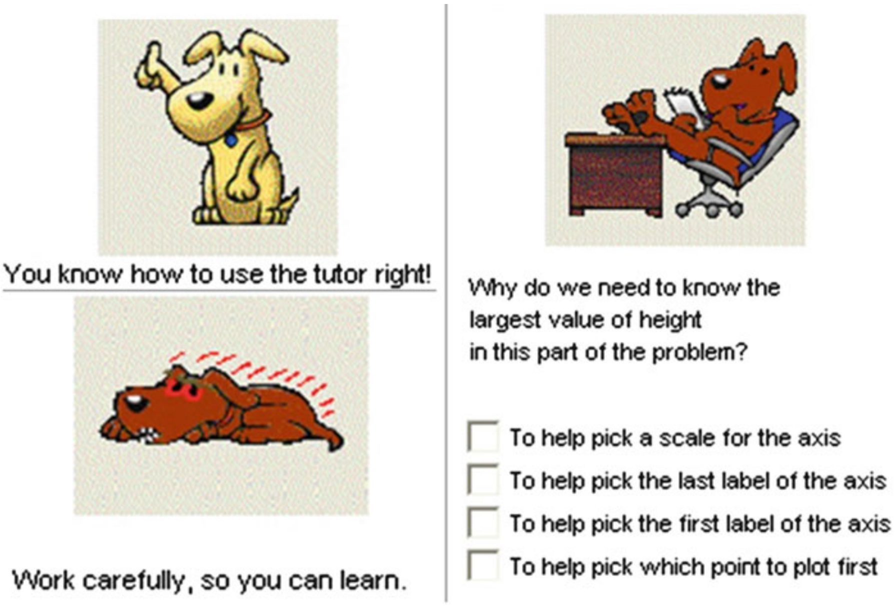
Within this second generation of gaming interventions, Walonoski and Heffernan (2006b) placed visualizations about gaming behavior over the last 20 min on-screen, for viewing by students and teachers. These visualizations had a bar travelling from left to right; the visualizations indicated the passage of time from left to right, gaming behavior by color (red indicating certain gaming, yellow indicating possible gaming, and green indicating no gaming), and the correctness (at the cognitive level) of the action from top to bottom. Placing the mouse pointer over a point on the graph gave greater detail about that action. This system was successful at reducing students' degrees of gaming behavior as they used the tutoring system; domain learning was not measured.

A second project, Arroyo and colleagues (2007), gave visualizations of student correctness between problems rather than during problems. These visualizations showed overall correctness rather than directly showing gaming (the two constructs are, of course, closely related). Along with the visualizations were textual messages (when gaming had occurred) about how correctness could be raised by avoiding gaming strategies. This system was successful at reducing students' degrees of gaming behavior as they used the tutoring system, and was also found to improve domain learning.

A third project, Baker and colleagues (Baker, Corbett, Koedinger, et al., 2006), combined feedback on how to use the software appropriately (as in Roll et al., 2007, but the feedback was substantially less sophisticated), with an attempt to give students another way to learn material missed by gaming. This intervention involved a pedagogical agent named Scooter the Tutor, shown in Fig. 7.5. When students did not game, Scooter remained in the background, occasionally giving a positive message; when the student gamed the system, Scooter first displayed negative emotion and gave metacognitive messages similar to those in Roll and colleagues (2007), and then gave supplementary exercises which involved using the same skills or concepts bypassed via gaming. Scooter was successful at reducing students' degrees of gaming behavior as they used the tutoring system, and was also found to improve domain learning. However, the very students who benefitted from Scooter's interventions reported strongly disliking Scooter.

Each of these interventions was successful in reducing gaming, and two were successful in improving domain learning. However, none of these interventions were successful in a broader sense: none were adopted and applied at a wider scale by software developers, even within the three projects that originally developed them. One possible explanation for this puzzling lack of uptake is that all three of these interventions required significant development and made the interaction between the student and the educational software substantially more complex. This may be a general limitation for interventions intended to solve single problems in metacognition or address single problematic behaviors: the intervention cannot be larger in scope and complexity than the problem's perceived level of importance justifies to software developers.

The recent research on which features of intelligent tutoring systems lead to gaming, described earlier in this chapter, provides a possible avenue for addressing gaming the system in a more lightweight fashion. Knowing the features that predict gaming creates the possibility that changing these features will reduce students' propensity to game the system (this is not guaranteed, of course, as correlation does not imply causation), and perhaps also improve learning. Research into this possibility is an important area of future works.



You know how to use the tutor right!

Work carefully, so you can learn.

Why do we need to know the
largest value of height
in this part of the problem?

□ To help pick a scale for the axis
□ To help pick the last label of the axis
□ To help pick the first label of the axis
□ To help pick which point to plot first

Fig. 7.5 Scooter the Tutor—looking happy when the student has not been gaming harmfully (*top-left*), giving a supplementary exercise to a gaming student (*right*), and looking angry when the student is believed to have been gaming heavily, or attempted to game Scooter during a supplementary exercise (*bottom-left*)

Conclusions

In this chapter, we have talked about our work to model and study gaming the system using educational data mining methods (Baker & Yacef, 2009; Romero & Ventura, 2010). Our work has leveraged the development of automated detectors of gaming behavior for Cognitive Tutors and other interactive learning environments. Our detector development has relied upon first using human labeling methods to gather "ground truth" labels of students or actions judged to be gaming the system, then using data mining methods to distill these labels into reusable detectors of gaming, and finally validating these models at multiple levels.

We then discuss two "discovery with models" analyses where these detectors are leveraged in order to analyze research questions of interest. Gaming detectors have supported the analysis of why students game the system, and how gaming the system impacts learning. In specific, these analyses show that gaming the system is associated with less learning, in an immediate fashion—a different pattern than was found for off-task behavior, where learning was only reduced in the aggregate. In addition, these analyses discover a set of nine features of tutor lessons that are associated with differences in the prevalence of gaming the system.

We also discuss ongoing work, both in our research group and other research groups, to develop software that remediates gaming the system. Thus far, this work has had only partial success. We discuss how the discovery with models analyses presented earlier in the chapter may have the potential to influence the design of educational software that effectively prevents gaming in a nonintrusive fashion. If successful, this program of research will form a key example of how to design for effective student behavior, in a fashion that either stimulates metacognition which leads to more effective learning strategies or alternatively by addressing the negative learning outcomes potentially stemming from students' ineffective or counterproductive self-regulation during learning.

In the long term, studying gaming and related phenomena using discovery with models methods has the potential to significantly improve our field's understanding of the metacognitive and motivational processes that occur during learning with interactive learning technologies, in turn leading to software more effectively tuned to students' educational needs.

References

Aleven, V., McLaren, B. M., Roll, I., & Koedinger, K. R. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education, 16*, 101–130.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences, 4*(2), 167–207.

Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., & Fisher, D. (2007). Repairing disengagement with non-invasive interventions. In J. Greer, R. Luckin, & K. Koedinger (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (pp. 195–202). Amsterdam, Netherlands: Ios Press.

Arroyo, I., & Woolf, B. (2005). Inferring learning and attitudes from a Bayesian Network of log file data. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education* (pp. 33–40). Amsterdam, Netherlands: Ios Press.

Baker, R. S. J. d. (2007a). Is gaming the system state-or-trait? Educational data mining through the multi-contextual application of a validated behavioral model. In R. Baker, J. Beck, B. Berendt, A. Kroner, E. Menasalvas, & S. Weibelzahl (Eds.), *Complete on-line Proceedings of the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling* (pp. 76–80). Pittsburgh, PA: International Working Group on Educational Data Mining.

Baker, R. S. J. d. (2007b). Modeling and understanding students' off-task behavior in intelligent tutoring systems. In M. Rosson & D. Gilmore (Eds.), *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems* (pp. 1059–1068). Washington, DC: Association for Computing Machinery.

Baker, R. S. J. d. (2010). Mining data for student models. In R. Nkmabou, R. Mizoguchi, & J. Bourdeau (Eds.), *Advances in intelligent tutoring systems* (pp. 341–356). Secaucus, NJ: Springer.

Baker, R. S., Corbett, A. T., & Koedinger, K. R. (2004a). Detecting student misuse of intelligent tutoring systems. In J. Lester, R. Vicari, & F. Paraguacu (Eds.), *Proceedings of the 7th International Conference on Intelligent Tutoring Systems* (pp. 531–540). Heidelberg, Germany: Springer.

Baker, R. S. J. d., Corbett, A. T., Koedinger, K. R., Evenson, S. E., Roll, I., & Wagner, A. Z. (2006). Adapting to when students game an intelligent tutoring system. In M. Ikeda, K. Ashley, & T. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 392–401). Heidelberg, Germany: Springer.

Baker, R. S., Corbett, A., Koedinger, K., & Roll, I. (2005). Detecting when students game the system, across tutor subjects and classroom cohorts. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *Proceedings of the 10th International Conference on User Modeling* (pp. 220–224). Heidelberg, Germany: Springer.

Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004b). Off-task behavior in the cognitive tutor classroom: When students "game the system". In E. Dykstra-Erickson & M. Tscheligi (Eds.), *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems* (pp. 383–390). Washington, DC: Association for Computing Machinery.

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

Baker, R. S. J. d., Corbett, A. T., & Wagner, A. Z. (2006). Human classification of low-fidelity replays of student actions. In C. Heiner, R. Baker, & K. Yacef (Eds.), *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems* (pp. 29–36). Pittsburgh, PA: International Working Group on Educational Data Mining.

Baker, R. S. J. d., & de Carvalho, A. M. J. A. (2008). Labeling student behavior faster and more precisely with text replays. In R. Baker, T. Barnes, & J. Beck (Eds.), *Proceedings of the 1st International Conference on Educational Data Mining* (pp. 38–47). Pittsburgh, PA: International Working Group on Educational Data Mining.

Baker, R. S. J. d., de Carvalho, A. M. J. A., Raspat, J., Aleven, V., Corbett, A. T., & Koedinger, K. R. (2009). Educational software features that encourage and discourage "gaming the system". In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 475–482). Amsterdam, Netherlands: Ios Press.

Baker, R. S. J. d., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human Computer Studies, 68*(4), 223–241.

Baker, R. S. J. d., Mitrovic, A., & Mathews, M. (2010). Detecting gaming the system in constraint-based tutors. In P. de Bra, A. Kobsa, & D. Chin (Eds.), *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization* (pp. 267–278). Heidelberg, Germany: Springer.

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research, 19*(2), 185–224.

Baker, R. S. J. d., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining, 1*(1), 3–17.

Beal, C. R., Qu, L., & Lee, H. (2006). Classifying learner engagement through integration of multiple data sources. In Y. Gil & R. Mooney (Eds.), *Proceedings of the 21st National Conference on Artificial Intelligence* (pp. 2–8). Washington, DC: Association for the Advancement of Artificial Intelligence.

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

Beck, J. (2005). Engagement tracing: Using response times to model student disengagement. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)* (pp. 88–95). Amsterdam, Netherlands: Ios Press.

Beck, J. E. (2006). Using learning decomposition to analyze student fluency development. In C. Heiner, R. Baker, & K. Yacef (Eds.), *Proceedings of the workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems* (pp. 21–28). Pittsburgh, PA: International Working Group on Educational Data Mining.

Butler, D. L., & Winne, P. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Resarch, 65*(3), 245–281.

Cheng, R., & Vassileva, J. (2005). Adaptive reward mechanism for sustainable online learning community. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education* (pp. 152–159). Amsterdam, Netherlands: Ios Press.

Cocea, M., Hershkovitz, A., & de Baker, R. S. J. (2009). The impact of off-task and gaming behaviors on learning: Immediate or aggregate? In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 507–514). Amsterdam, Netherlands: Ios Press.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*, 253–278.

Duckworth, A. L., Peterson, C., Matthews, M. D., & Kelly, D. R. (2007). Grit: Perseverance and passion for long-term goals. *Journal of Personality and Social Psychology, 92*, 1087–1101.

Dweck, C. S. (2000). *Self-theories: Their role in motivation, personality, and development*. Philadelphia, PA: Psychology Press.

Efron, B., & Gong, G. (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician, 37*(1), 36–48.

Fisher, S. L., & Ford, J. K. (1998). Differential effects of learner effort and goal orientation on two learning outcomes. *Personnel Psychology, 51*(2), 397–420.

Flavell, J. H. (1976). Metacognitive aspects of problem solving. In L. B. Resnick (Ed.), *The nature of intelligence*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gobel, P. (2008). Student off-task behavior and motivation in the CALL classroom. *International Journal of Pedagogies and Learning, 4*(4), 4–18.

Gong, Y., Beck, J., Heffernan, N. T., & Forbes-Summers, E. (2010). The fine-grained impact of gaming (?) on learning. In V. Aleven, J. Kay, & J. Mostow (Eds.), *Proceedings of the 10th International Conference on Intelligent Tutoring Systems* (pp. 194–203). Heidelberg, Germany: Springer.

Hacker, D. J. (1999). Definitions and empirical foundations. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 1–24). Mahway, NJ: Erlbaum.

Johns, J., & Woolf, B. (2006). A dynamic mixture model to detect student motivation and proficiency. In Y. Gil & R. Mooney (Eds.), *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)* (pp. 163–168). Washington, DC: Association for the Advancement of Artificial Intelligence.

Karweit, N., & Slavin, R. E. (1982). Time-on-task: Issues of timing, sampling, and definition. *Journal of Experimental Psychology, 74*(6), 844–851.

Koedinger, K. R., de Baker, R. S. J., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the EDM community: The PSLC DataShop. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. de Baker (Eds.), *Handbook of educational data mining* (pp. 43–56). Boca Raton, FL: CRC Press.

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

Magnussen, R., & Misfeldt, M. (2004). Player transformation of educational multiplayer games. In M. Sicart & J. Smith (Eds.), *Proceedings of other players*. Copenhagen, Denmark: IT University of Copenhagen.

Maris, E. (1995). Psychometric latent response models. *Psychometrika, 60*(4), 523–547.

Martínez Mirón, E. A., du Boulay, B., & Luckin, R. (2004). Goal achievement orientation in the design of an ILE. In C. Frasson & K. Porayska-Pomsta (Eds.), *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments* (pp. 72–78). Maceio, Brazil: Federal University of Alagoas.

Midgley, C., & Urdan, T. (2002). Academic self-handicapping and achievement goals: A further examination. *Contemporary Educational Psychology, 26*(1), 61–75.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. In L. Ungar, M. Craven, & D. Gunopulos (Eds.), *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 935–940). Washington, DC: Association for Computing Machinery.

Muldner, K., Burleson, W., Van de Sande, B., & VanLehn, K. (2011). An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impact. *User Modeling and User-Adapted Interaction, 21*(1–2), 99–135.

Murray, R. C., & VanLehn, K. (2005). Effects of dissuading unnecessary help requests while providing proactive help. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education* (pp. 887–889). Amsterdam, Netherlands: Ios Press.

Nelson-Le Gall, S. (1985). Help-seeking behavior in learning. *Review of Research in Education, 12*, 55–90.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufmann.

Ramsey, F. L., & Schafer, D. W. (1997). *The statistical sleuth: A course in methods of data analysis*. Belmont, CA: Duxbury Press.

Rodrigo, M. M. T., Rebolledo-Mendez, G., de Baker, R. S. J., du Boulay, B., Sugay, J. O., & Lim, S. A. L. (2008). The effects of motivational modeling on affect in an intelligent tutoring system. In Y. Yano (Ed.), *Proceedings of 16th International Conference on Computers in Education*. Jhongli, Taiwan: Asia-Pacific Society for Computers in Education.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2007). Designing for metacognition—Applying cognitive tutor principles to the tutoring of help seeking. *Metacognition and Learning, 2*(2), 125–140.

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

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state-of-the-art. *IEEE Transactions of Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40*(6), 601–618.

Sao Pedro, M. A., de Baker, R. S. J., Montalvo, O., Nakama, A., & Gobert, J. D. (2010). Using text replay tagging to produce detectors of systematic experimentation behavior patterns. In R. Baker, A. Merceron, & P. Pavlik (Eds.), *Proceedings of the 3rd International Conference on Educational Data Mining* (pp. 181–190). Pittsburgh, PA: International Working Group on Educational Data Mining.

Shih, B., Koedinger, K., & Scheines, R. (2008). A response time model for bottom-out hints as worked examples. In R. Baker, T. Barnes, & J. Beck (Eds.), *Proceedings of the 1st International Conference on Educational Data Mining* (pp. 117–126). Pittsburgh, PA: International Working Group on Educational Data Mining.

Walonoski, J. A., & Heffernan, N. T. (2006a). Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In M. Ikeda, K. Ashley, & T. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 382–391). Heidelberg, Germany: Springer.

Walonoski, J. A., & Heffernan, N. T. (2006b). Prevention of off-task gaming behavior in intelligent tutoring systems. In M. Ikeda, K. Ashley, & T. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 722–724). Heidelberg, Germany: Springer.

Winne, P., & Hadwin, P. (1998). Studying as self-regulated learning. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 277–304). Mahway, NJ: Erlbaum.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. San Francisco, CA: Morgan Kaufmann.

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: a fast correlation-based filter solution. In T. Fawcett & N. Mishra (Eds.), *Proceedings of the 20th International Conference on Machine Learning* (pp. 856–863). Washington, DC: Association for the Advancement of Artificial Intelligence.

Zimmerman, B. J. (2000). Attaining self-regulation: A social cognitive perspective. In M. Boekarts, P. R. Pintrich, & M. Zeidner (Eds.), *Handbook of regulation* (pp. 13–39). Amsterdam, Netherlands: Elsevier.