Generation of Models for Detecting Off-task Behavior While Using an Intelligent Tutor for Algebra

Allan Edgar C. BATE  
Department of Information Systems and Computer Science  
Ateneo de Manila  
Quezon City, Philippines  
allbate@gmail.com

Ma. Mercedes T. RODRIGO  
Department of Information Systems and Computer Science  
Ateneo de Manila  
Quezon City, Philippines  
mrodrigo@ateneo.edu

Abstract— As more and more modern classrooms use Intelligent Tutoring Systems, it becomes imperative for our educators to determine whether these systems are being used properly. It is possible for students to engage in off-task behavior, defined as behavior that results from the students’ lack of motivation for learning. Off-task behavior can range from resting one's eyes, to talking to one's seatmate, to "gaming the system" defined as attempting to advance through the curriculum by abusing regularities in the system. Gaming is an operational systematic guessing or trial and error. These behaviors constitute time away from the learning task and are therefore considered detrimental to learning. By analyzing low-fidelity playbacks in action clips of interactions recorded by the tutor, we determine off-task behavior using text-action replays. Using clips labeled by experts, we use machine learning techniques to detect automatically off-task behavior. This automatic detection can lead to interventions that can retain student attention and increase learning.

Keywords- Affective Computing, Intelligent Tutoring Systems, Machine-learning, Aplusix, Off-task behavior

I. INTRODUCTION

This paper begins with a review of preliminary work on this project, conducted and presented in [8]. After this review, we will discuss further the findings from our completed experiment.

Intelligent tutoring systems (ITSs) are a subtype of computer-based learning system that makes use of artificial intelligence to increase teaching effectiveness. ITSs contain a pedagogical model that is designed to teach and thus provide ample explanations and exercises for the students to learn the ITS’ domain [5]. Interactivity and appropriate challenge levels are the key aspects that allow these tutors to provide a positive effect to the learning environment for students [6]. The use of ITSs has been found to increase student motivation [4].

A. Statement of the Problem

Off-task behavior is a symptom of disengagement from a learning experience [7] and is associated with poor learning.

Support for this project was provided by the Science Education Institute - Department of Science and Technology (SEI-DOST) through the Engineering Research and Development for Technology (ERDT) program.

Baker et al [1] found that the students who engaged in off-task behavior during the use of an intelligent tutor learned two-thirds of the subject matter compared to students who used the tutor properly.

B. Goal

In order to prevent the loss of learning opportunities for the student while using ITSs, we will attempt to create a model that will be able to detect off-task behavior during the students’ use of the ITS by analyzing data logs that records actions taken by the student during the use of the ITS.

C. Research Questions

1. What information do we need to have a significantly valid low-fidelity playback of the use of the ITS?

2. What are the different patterns of behavior that display off-task behavior of the student?

D. Significance

In traditional classrooms, teachers are able to identify when students start to lose interest and when to intervene in correcting them. For ITSs, we believe that it is important to be able to detect when students begin to lose interest and start engaging in off-task behavior. Automatic detection enables ITS designers to devise interventions when faced with a student who is off-task. As Baker mentioned in [1], how an ITS reacts to off-task behavior should be carefully studied as these responses may greatly affect learning, most especially of students who are classified as false positives. By preventing students from being off-task on long periods can increase the learning gained and the efficiency and effectiveness of ITSs.

II. THEORETICAL FRAMEWORK

A. Log File Analysis

Log file analysis is the systematic approach to examining and interpreting the content of behavioral data [3]. Log file analysis approaches include:

   Transition analysis refers to the analysis the changes in behavior.
**Frequency analysis** refers to the tallying of frequencies of actions and computing for different statistics such as averages, and standard deviations.

**Learning-indicator approach**, similar to frequency approach, consists of clustering actions that have close-to-similar frequencies and determine groups in a global coverage.

**Sequence analysis** pays more attention to the belief that actions are the results of the actions before it and the reasons for the actions after it.

Log file analysis works with low-fidelity data as compared to live observations. This makes conducting tests faster and economical. Low-fidelity playbacks have been shown to be sufficient enough to make accurate inferences to off-task behavior during the use of the ITS [2].

## III. METHODOLOGY

Aplusix\(^2\) is an ITS for Algebra. Aplusix presents students with exercises on varying math topics and levels of difficulties. Students solve these problems in a stepwise fashion, as they would paper (See Figure 1). Aplusix records any interaction the student makes, from starting the exercise to ending the exercise by either solving the problem or abandoning it.

### A. Data Distillation from Aplusix

We obtained the log files Aplusix outputted from the exercises of a previous research by Rodrigo et al. [6]. We then selected six rows of parameters from the log files: **turn**, **time**, **action**, **step**, **expression**, and **status**. Actions were grouped into 20-second clips. Clips were identified by the classification and difficulty of the problem they were found being done in and the original state of the problem. We then displayed each clip to our experts (Figure 2) for them to classify. From a population of 11,220 clips, we randomly selected a sample of 391 clips for labeling.

### B. Labeling

We asked Dr. Cornelia Soto of the Ateneo de Manila University’s Education department and Mrs. Ria Arespacochaga of the Ateneo High School Math Department to serve as our experts. Both had extensive knowledge of math education and off-task behavior in classrooms. They identified which behavioral patterns will tell us if the student is being off-task.

### C. Machine Learning Using WEKA

Our classified clips were summarized into vectors containing the labeling supplied by our experts and the features concerning each clip. Feature reduction was performed to further optimize the machine learning. The J48 algorithm supported by WEKA, was used and gave us an output of a C4.5 decision tree, and was validated using the 10-fold cross validation.

## IV. DISCUSSION

### A. Feedback from our Experts

As our experts went about classifying our clips, we documented some of the more relevant conversation that occurred during the process. From this, we picked up some insight on the train of thought our experts used in labeling off-task behavior.

1) **Identifying “Thinking”**

During experimentation, our experts found that one of the biggest discerning points of determining off-task behavior was that if the students’ actions correspond to the proper way of finding the solution. If the numbers reasonably resembled a number that was expected to come out given a problem, they would deem the student to be thinking about the lesson and thus be on-task. One problem in discernment using this method was that it was difficult for us to replicate this way of thinking operationally without having our model solve the problem each time. Not only was it difficult to determine if the student was in the correct path among the many paths in solving the problem, it was also difficult to determine if the student is also merely being careless or over-looking the simple mistakes they made, which in this state, the students were still considered on-task but confused. For these cases, it was not enough to simply detect if the students had the problem partially solved, or had step equivalence, which were what Aplusix mainly provided us as feedback.

2) **Interface-related mistakes of Aplusix**

---

Many students were found to be on-task and not-confused up to the point where they arrived at the solution. However, they were unable to “declare” the problem as solved using Aplusix’s convention. When this happened, students would perform random actions such as deleting their answer, perhaps thinking that it was wrong, or that the problem was not completely simplified. It may be argued that students should be classified as off-task at this point due to the nature of their actions. However, our experts considered these students to be truly at a loss because of the Aplusix interface. Therefore, students exhibiting these behaviors were regarded as on-task but confused.

3) Time as a Factor
Mrs. Arespacochaga said that one of the main factors she used in determining on-task behavior was time. If a student paused at the start of the exercise, the student is regarded as “thinking” but if a student paused at the end then the student was found confused and resulted in being off-task. Upon further analysis, we found that majority of the clips classified as on-task resulted from three main attributes, two of which were time-related:

- The average time of each action across all actions performed is greater than 0.45 seconds.
- The total time of actions before the student becomes inactive for the rest of the 20-second clip is greater than 10.7 seconds.

These first two features reflected Mrs. Arespacochaga’s thought-process of looking at when a student pauses. If a student paused at the end, it lessened their actions taken within 20 seconds and thus reduced action time and students who paused at the start and generally raised the average time across all actions taken.

B. Building the Model
As mentioned in the methodology, we will now discuss the process of building our model. From what our experts said, the main criterion that determines whether a student is on-task was “thinking”. The experts examined whether the numbers that the students typed were correct or if they were wrong due to carelessness. We attempted to perform this operationally using string parsing and simple logic.

1) Identifying the Feature for “Thinking”
In attempting to simulate our experts’ criteria for “thinking”, we parsed each equation, starting from the original problem and identified the numbers present, whether they were coefficients, addends, or factors, and so on. For this explanation, let us use the example of:

\[ 4(-x - 7) + 2(-9x + 4) + 6.2(3x + 9) \]

Using our algorithm, we will be able to identify the numbers: 4, 7, 2, 9, 6.2, and 3. Our parser is designed to get unique positive numbers. Negative numbers are treated as positive for our purposes and fractions are only known for their separate numbers as well. Since these set of numbers belong to the original problem, they will be given a score of 1, which means it does not necessarily mean the student is progressing if these numbers come up during the solution.

Our algorithm then performs simple operations, namely addition, subtraction, multiplication, and division, between these numbers and these operations were done to match each numbers in permutation. This will give us a list of predicted numbers. A sample of these predicted numbers would be: 11, from 4 + 7; 28, from 4 * 7; 5, from 9 - 4; and a 6 again, from 4 + 2. This time, these numbers are given a score of 5. This means that if any of these numbers appear, we can assume that a student has made some progress in solving the problem. Take note that we do not actually compute for the solution based on the original problem. We simply try to perform simple possible operations that can occur in between numbers to take into consideration careless mistakes made by the students in performing the wrong operation.

During the process of reading the clip line by line, the algorithm compares the numbers that appear in the clip with the algorithm’s list of numbers. If a similar number is found, a progression score is incremented with the score assigned to specified number on the list. That number’s score will then be set to 0, stating that the predicted number has been found. If the number in the clip was not found on the algorithm’s list, the new number will be added to the list and is given a score of 0. The algorithm then generates a new set of predicted numbers, incrementing existing ones by 1 and creating new ones with a score of 3.

The progression score is a quantitative measure for how reasonable the student’s solution is based on the original problem and the successive steps taken by the student. In the case of clips that are found to be in the middle of the exercise, it emphasizes on the usage of predicted numbers rather than dwelling on the original problem. The scores were arbitrarily assigned to emphasize the importance of predicted numbers based on the original problem, over the predicted numbers based on the previous equations typed by the student, which in turn is also given a higher importance over numbers only found in previous equations. To further explain the relevance of this score, during our sessions with our experts, one of their main points in determining on-task behavior was when they find a number relevant to the original problem, they would deem the student on-task. Examples of cases would be if the problem contains:

\[ 6x + 3(5x - 4) \]

And the student would carelessly answer:

\[ 6x + 8x - 4 \]

The student was regarded to be on-task and they would just write him / her off as being careless. On a similar note, if a student carelessly typed 18 and later corrected it with 16, which is part of the correct answer, the student is still considered on-task from the moment they typed 18. Originally, we believed that Aplusix’ feedback of equivalence in between steps would be enough but because of the numerous instances of carelessness from our clips we believe that at least this feature could capture those instances.

We have defined the patterns to identify that students are thinking. We also defined some behavior that we had look out
for particularly from our sample set. From the information we have gathered through our sessions with our experts, we are now ready to generate our models based on their classifications. The results of these will be discussed in the next section.

V. Results

We manually removed some features based on our experts’ feedback and then used the attribute selection algorithm provided by WEKA to reduce the feature space. We shall briefly discuss our sample sets and its characteristics based on the statistics of its features. We will then present the models we generated using WEKA³.

A. Features Used and Describing our Sample Population

In implementing our model based on the feedback from our experts, we extracted information from the clips as our experts analyzed them. The features used for our machine learning are as follows:

1) Problem Difficulty and Complexity

One of the more basic features required by our experts was what type of problem and how difficult the student was trying to solve. This is usually the bases on how “reasonable” the pauses the student made were. Problem difficulty alone was not sufficient since more than 80% of the problems were of B1 – Expansion and Simplification. Problem complexity gives a numerical rating on how complicated the original problem appears to be as to possibly confuse the student. Because of this, our data will most likely represent behavior students who are answering relatively simple problems.

2) Starting Turn

Clips do not necessarily begin at the start of the exercise and sometimes contain actions that already find the students in the middle of solving problems. In conjunction with the problem difficulty, how reasonable the actions of the students are depends on this feature.

The majority of our sample population is composed of clips starts out after students have done 44 or less actions. This means that students generally solve exercises without making too many unnecessary actions. A few students, however, took a lot of time and effort on the problem. Some clips start after a student has performed 575 actions.

3) Action Count and Time

These are the two basic pieces of information about the clip: how many actions the student performed within clip and the time from the first action to the last action.

From the statistics, the majority of our clips show that students are active throughout the 20-second intervals. However, from the statistics for action count we see from Figure 3 that the graph is skewed to the left. This means that most students’ actions performed done at a mean of 5.5 – 10 actions of the 20 second time window.

4) Deletion

In keeping track of trial-and-error, we kept track of the deletion activity the students made and the activity of other actions in between deletions. Students who performed trial-and-error would have bursts of deletion with little activity in between bursts.

Students solve Aplusix by first duplicating the problem equation. From here, they edit the problem. It is not unusual to find students deleting either single characters or a whole selection of a string. These set of features were supposed to identify trial-and-error but as we saw during classification, students would part-by-part edit the equation. They perform deletes and inputs in alternating patterns, as is the definition of “trial-and-error”.

However, the sign of trial-and-error can be later contradicted when we check for average action time, cursor inputs and our progression feature. Trial-and-error is usually performed quickly since the student has a set pattern of inputs i.e. consecutive numbers and usually by changing the same number [1]. By moving the cursor, we can assume that the student is actually editing the equation part by part. The progression feature will also register low given the number actions performed since only a few of the numbers they entered will give them a score.

5) Keyboard Inputs and Interaction

This set of features constitutes the statistics of number of each type of input the student made during the exercise. These inputs include number inputs, symbol inputs, letter inputs, cursor movement, editing functions such as cut and paste, comments made by students, and miscellaneous functions such as declaration of problem solved or abandonment.

From our data, we can see in Figure 4 that there was a good balance of activity between deleting, number inputs, letter inputs, and symbol inputs. Although all of the input counts are skewed to the left, meaning that there are only a few of such inputs within the 20-seconds.

³ http://www.cs.waikato.ac.nz/ml/weka/
6) Solution status

Solution status refers to the correctness or wrongness of the student’s solutions. We kept track of the number of help requests made, if the solution was abandoned, was the student able to solve it or partly solved it, and if the student came across equivalences in between steps, and finally how many steps the student went through within the time span of the clip.

Many of the clips did not capture the end part of an exercise where a student either solved the problem or abandoned it. This was evident from the fact that there was only one instance of abandonment and 22 instances of a clip ending in being solved. There were, however, numerous clips found to be “partly solved”. This meant that the student was able to get the problem solved but was not able to declare it solved, probably because there may have been still some inequivalence in some of the steps they have written.

7) Progression

This feature was the score rating of each equation after each action of the student based on the relationships of the numbers found within to the original problem. As explained earlier, this feature was derived from an algorithm we created in order to simulate the thought-process of our experts to take notice of the numbers the students are working with in relation to the original problem. The clips were given a higher progression score if the students were found to be working on predicted numbers, which were based on original numbers with some sort of mathematical operation performed on them, rather than numbers that had no relation to the original problem or the original numbers themselves.

A. Ms. Arespacochaga’s Model

Out of the 391 samples, our expert, Mrs. Arespacochaga classified 283 clips as being on-task, 80 as off-task, and 28 as unknown. With this model, the algorithm is able to classify 80% of our samples correctly and after performing a 10-fold cross-validation, we got a result of a Kappa statistic of 0.4848. Figure 5 shows the decision tree for the model generated.

B. Dr. Soto’s Model

Out of the 391 samples, our expert, Dr. Soto classified 319 clips as being on-task, and 72 as off-task. With this model, the algorithm is able to classify 87.7% of our samples correctly and after performing a 10-fold cross-validation, we got a result of a Kappa statistic of 0.5477. Figure 6 shows the decision tree for the model generated.

Both models use action count as its first criteria in the decision-making process and close to it would be action time. These results may reflect what Arespacochaga mentioned during classification about checking if the student pauses early or late. It is possible that short clips tend to provide insufficient information on behavior such that the few random actions that is done within the 20 seconds could already give a sign that the student is confused with the work and tend to become off-task.

VI. Future Work

The models created generated a moderate value of Kappa, giving us satisfactory results that it is possible to automatically detect off-task behavior using text-action logs from an intelligent tutor for Algebra. If we were to improve the results of the experiment, a research that focuses on the features
themselves could further refine the model generated from the algorithm. Our features were derived from the feedback of two experts. It is possible that opinions of other people will open up other ideas on the features we can extract from action clips. Although we believe that our sample set was sufficient as a representation of the whole population of clips, it is also possible that classifying more clips could level out any inconsistencies our experts may have accidentally shown in these few 391 clips and thus provide WEKA with more information to work with and further increase the reliability of the model.

ACKNOWLEDGMENT

We would like to thank Dr. Cornelia Soto and Mrs. Ria Arespacochaga for their cooperation in this research as our experts.

REFERENCES


