Identifying Learning Trajectories in an Educational Video Game

Deirdre Kerr
UCLA/CRESST
Peter V. Ueberroth Building (PVUB)
10945 Le Conte Ave., Suite 1400
Los Angeles, CA 90095-7150
dkerr@gseis.ucla.edu

Gregory K.W.K. Chung
UCLA/CRESST
Peter V. Ueberroth Building (PVUB)
10945 Le Conte Ave., Suite 1355
Los Angeles, CA 90095-7150
greg@ucla.edu

Abstract

Educational video games and simulations hold great potential as measurement tools to assess student levels of understanding, identify effective instructional techniques, and pinpoint moments of learning because they record all actions taken in the course of solving each problem rather than just the answers given. However, extracting meaningful information from the log data produced by educational video games and simulations is notoriously difficult. We extract meaningful information from the log data by first utilizing a logging technique that results in a far more easily analyzed dataset. We then identify different learning trajectories from the log data, determine the varying effects of the trajectories on learning, and outline an approach to automating the process.

1. INTRODUCTION

Computer games and simulations hold great potential as measurement tools because they can measure knowledge that is difficult to assess using paper-and-pencil tests or hands-on tasks (Quellmalz & Pellegrino, 2009). These measures can then be used to support diagnostic claims about students' learning processes (Leighton & Gierl, 2007), provide detailed measures of the extent to which players have mastered specific learning goals (National Science and Technology Council, 2011), and generate information that can be used to improve classroom instruction (Merceron & Yacef, 2004).

Log files from games can store complete student answers to the problems (Merceron & Yacef, 2004), allowing the researcher to record unobtrusively (Kim, Gunn, Schuh, Phillips, Pagulayan, & Wixon, 2008; Mostow, Beck, Cuneao, Gouvea, Heiner, & Juarez, 2011) the exact learning behavior of students (Romero & Ventura, 2007) that is not always captured in written or verbal explanations of their thought processes (Bejar, 1984).

Though log data is more comprehensive and more detailed than most other forms of assessment data, analyzing such data presents a number of problems because the log files typically include thousands of pieces of information for each student (Romero, Gonzalez, Ventura, del Jesus, & Herrera, 2009) with no known theory to help identify which information is salient (National Research Council, 2011). Additionally, the specific information stored in the log files is not always easy to interpret (Romero & Ventura, 2007) as the responses of individual students are highly context dependent (Rupp, Gusta, Mislevy, & Shaffer, 2010) and it can be very difficult to picture how student knowledge, learning, or misconceptions manifest themselves at the level of a specific action taken by the student in the course of the game. Due to these difficulties, there is currently no systematic approach to extracting relevant data from log files (Muehlenbrock, 2005). The interpretation of the rich stream of complex data that results from the tracking of in-game actions is one of the most serious bottlenecks facing researchers examining educational video games and simulations today (Mislevy, Almond, & Lukas, 2004).

1.1 RELATED WORK

Due to the difficulty involved in analyzing log data of students' in-game performance, educational researchers occasionally analyze student in-game performance by hand, despite the size of the data. Trained human raters have been used to extract purposeful sets of actions from game logs (Avouris, Komis, Fiotakis, Margaritis, &

Voyiatzaki, 2005) and logs of eye-tracking data (Conati & Merten, 2007). One study hand-identified student errors in log files from an introductory programming environment (Vee, Meyer, & Mannock, 2006) and another examined behavior patterns in an exploratory learning environment by hand to categorize students into learning types (Amershi & Conati, 2011). Another had the teacher play the role of a game character to score student responses and provide live feedback to the students (Hickey, Ingram-Goble, & Jameson, 2009).

Other studies avoided hand-coding log data by using easily extracted in-game measures such as percent completion or time spent on task to measure performance. The number of activities completed in the online learning environments *Moodle* (Romero, Gonzalez, Ventura, del Jesus, & Herrera, 2009) and *ActiveMath* (Scheuer, Muhlenbrock, & Melis, 2007) have been used to predict student grades. The time spent in each activity in an online learning environment has been used to detect unusual learning behavior (Ueno & Nagaoka, 2002). Combinations of the total time spent in the online environment and the number of activities successfully completed have been used to predict student success (Muhlenbrock, 2005) and report student progress (Rahkila & Karjalainen, 1999).

1.2 OUR CONTRIBUTION

In this study, we identify learning trajectories from information stored in log data generated by an educational video game. We do this by extracting the number of attempts required to solve each level (rather than the time spent or the number of levels completed) and then hand clustering the individual learning trajectories that result from plotting the attempts over time. We show that this process results in the identification of substantively different types of learning trajectories that differ on a variety of measures. We also discuss the benefits of our logging, preprocessing, and exploratory analysis techniques in regards to ease of interpretation and potential use in data mining techniques.

1.3 SAMPLE

This study uses data from 859 students who played an educational video game about identifying fractions called *Save Patch* in their classrooms for four days as part of a larger study. These students were given a paper-and-pencil pretest to measure their prior knowledge of fractions. After they played the game, students were given both an immediate posttest and a delayed posttest. The immediate posttest was computerized and was given on the last day of game play. The delayed posttest was a paper-and-pencil test that was given a few weeks later. All three tests consisted of both a set of content items and a set of survey items. In addition, the game generated log data consisting of each action taken by each student in the course of game play. The resulting dataset consisted of 1,288,103 total actions, 17,685 of which were unique.

2. DATA PREPARATION

The *Data Preprocessing and Intelligent Data Analysis* article (Famili, Shen, Weber, & Simoudis, 1997) lists eleven problems with real-world data that should be addressed in preprocessing. Our data comes from a single source, so we do not have to worry about merging data from multiple sources or combining incompatible data. The nine remaining problems and how they are applicable to our data are shown in Table 1.

Table 1: Potential Problems with *Save Patch* Data

PROBLEM	DESCRIPTION
Corruption and noise	Interruptions during data recording can lead to missing actions
Feature extraction	Important events must be identified from sets of individual actions
Irrelevant data	Not all actions taken in the game are meaningful
Volume of data	Hundreds or thousands of actions are recorded for each student
Missing attributes	Logs can fail to capture all relevant attributes
Missing attribute values	Logs can fail to record all values for all captured attributes
Numeric and symbolic data	Data for each action contains both numeric and symbolic components
Small data at a given level	We only have data for 859 students
Multiple levels	Data are recorded at multiple levels of granularity for each action

Our approach to minimizing the impact of these problems is explained in the following sections. Missing attributes are addressed in Section 2.1 (Game Design) and Section 2.2 (Logging). Corruption and noise, missing attribute values, numeric and symbolic data, and multiple levels are addressed in Section 2.2 (Logging). Feature extraction is addressed in Section 2.3 (Preprocessing), irrelevant data is addressed in Section 2.3.1 (Data Cleaning), and volume of data and small data at a given level are addressed in Section 3.1 (Exploratory Analysis).

2.1 GAME DESIGN

The educational video game used in this study is *Save Patch*. The development of *Save Patch* was driven by the findings that fluency with fractions is critical to performance in algebra (U.S. Department of Education, 2008), and that the understanding of fractions is one of the most difficult mathematical concepts students learn before algebra (Carpenter, Fennema, Franke, Levi, & Empson, 2000; McNeil & Alibali, 2005; National Council of Teachers of Mathematics, 2000; Siebert & Gaskin, 2006).

Once fractions concepts were identified as the subject area for the game, the most important concepts involved in fractions knowledge were analyzed and distilled into a set of knowledge specifications delineating precisely what students were expected to learn in the game (Vendlinski, Delacruz, Buschang, Chung, & Baker, 2010). These knowledge specifications, in turn, drove game design.

Because the game was designed specifically to measure student understanding of a predetermined set of knowledge specifications, both game mechanics and level design reflected those knowledge specifications and helped assure that all important attributes were measured in the game and recorded in the log files.

Figure 1: Example Level from *Save Patch*

In *Save Patch*, students must identify the fractional distances represented in each level, break ropes into pieces representing that distance, and place the correct number of rope pieces on each sign on the game grid to guide the puppet to the cage containing the prize. Units are represented by dirt paths and large gray posts, and small red posts break the units into fractional pieces. The level in Figure 1 is two units wide and one unit tall, and each unit is broken into thirds. To solve the level correctly, students must place four thirds on the first sign, one third on the second sign, and change the direction on the second sign so that it points down.

Save Patch is broken into stages based on content. All levels in a given stage represent the same fractions content. The game starts with whole number representations so that students can learn how to play, and then advances to unit fractions, whole numbers and unit fractions, proper fractions, and mixed numbers. After the fractions content stages, the game contains a test stage that was intended to be an in-game measure of learning and a series of challenge levels. The test stage includes an exact replica of one level from each of the previous stages and the challenge levels provide complicated combinations of the earlier material. This study focuses on the mixed numbers stage, because it contains the most complex representation of fractions in the game.

2.2 LOGGING

The data from *Save Patch* was generated by the logging technique outlined in Chung and Kerr (2012). As opposed to most log data from educational video games that consists of only summary information about student performance, such as the number of correct solutions or a probability that the content is known, the log data from this system consists of each action taken by each student in the course of game play.

However, such actions are not fully interpretable without relevant game context information indicating the precise circumstances under which the action was taken (Koedinger, Baker, Cunningham, Skogsholm, Leber, & Stamper, 2011). For this reason, each click that represented a deliberate action was logged in a row in the log file that included valuable context information such as the game level in which the action occurred and the time at which it occurred, as well as both general and specific information about the action itself.

As shown in Table 2, general information is stored in the form of a Data Code that is unique to each type of action (e.g., Data Code 3000 = selecting a rope piece from the Path Options). Each Data Code has a unique Description, for human readers and for documentation purposes, that identifies the action type and lists the interpretation of the following three columns. Data_01, Data_02, and Data_03 contain specific information about each action in the form of values that correspond to the bracketed information in the Description. For example, the third row in the table indicates that a rope was added (Data Code 3010) to the first sign (1/0 in Data_01), that the rope was a 1/3 piece (1/3 in Data_02), and that the resulting value on the sign was 1/3 (1/3 in Data_03). Additionally, the Gamestate records the values already placed on all signs in the level at the time of each action.

Logging the data in this manner allows for the easy interpretation of numeric and symbolic data because all comparable data is stored in the same format (e.g., 1/3 rather than .33) and because different representations of the same values have different interpretations in the game (e.g., 1/3 differs from 2/6). Additionally, the redundancy of carrying down each level of granularity (e.g., storing student ID and Level Number in each action) allows data to be recorded and analyzed at multiple levels without having to combine different datasets. This also reduces the negative effects of corruption and noise stemming from interruptions during data recording, because each action can be interpreted independently. Even if a given action is corrupted, all other actions in the level are still recorded correctly and each action contains all the information necessary for interpretation. While data corruption may result in missing attribute values in many

Table 2: Example Log Data from *Save Patch*

ID	Level	Game Time	Data Code	Description	Data_01	Data_02	Data_03	Gamestate
1115	14	3044.927	2050	Scrolled rope from [initial value] to [resulting value]	1/1	3/3		0/0_on_Sign1
1115	14	3051.117	3000	selected coil of [coil value]	1/3			0/0_on_Sign1
1115	14	3054.667	3010	added fraction at [position]: added [value] to yield [resulting value]	1/0	1/3	1/3	0/0_on_Sign1
1115	14	3058.443	3000	selected coil of [coil value]	1/3			1/3_on_Sign1
1115	14	3064.924	3010	added fraction at [position]: added [value] to yield [resulting value]	1/0	1/3	2/3	1/3_on_Sign1
1115	14	3088.886	3020	Submitted answer: clicked Go on [stage] – [level]	2	3		2/3_on_Sign1
1115	14	3097.562	3021	Moved: [direction] from [position] length [value]	Right	1/0	2/3	2/3_on_Sign1
1115	14	3106.224	4020	Received feedback: [type] consisting of [text]	Success	Congratulations!		2/3_on_Sign1
1115	14	3108.491	5000	Advanced to next level: [stage] – [level]	2	4		2/3_on_Sign1

other logging techniques, this is rarely the case with data logged in this manner because attribute values are recorded at the action level rather than calculated over time.

2.3 PREPROCESSING

The game design and logging techniques addressed a number of potential issues with the data, but it was still necessary to extract relevant features from the data.

In this study we were interested in examining student performance over time. In order to create these learning trajectories, we needed to identify a measure of performance in each level of the mixed numbers stage. Simply calculating whether students had correctly solved the level was insufficient, because students could replay a level as many times as was necessary and students could not advance to the next level without solving the current one. Therefore, we determined that the number of attempts it took a student to solve each level was the best measure of performance.

Attempts were not an existing feature of the log data, so each new attempt had to be calculated from existing information. We defined an attempt as all actions from the start of a level to either a reset of that level or advancing to the next level. The start of each attempt was identified using the following SPSS code, wherein Data Code 4010 indicates a reset:

If $casenum = 1 attempt = 1.
If id < > *lag*(id, 1) attempt = 1.
If curr_level < > *lag*(curr_level, 1) attempt = 1.
If *lag*(data_code, 1) = 4010 attempt = 1.

The first action in each attempt was then numbered consecutively using the following SPSS code:

Sort Cases By attempt(D) id curr_level uber_sn.
If id = *lag*(id,1) *and* attempt = 1
and curr_level = *lag*(curr_level, 1)
attempt = *lag*(attempt, 1) + 1.

Finally, the following SPSS code propagated the attempt number to all subsequent actions in that attempt:

Sort Cases By id curr_level uber_sn.
If attempt = 0 attempt = *lag*(attempt, 1).

2.3.1 Data Cleaning

Given the game design, logging technique, and preprocessing, little additional data cleaning was required after the attempts were calculated. However, irrelevant data still needed to be identified.

Irrelevant data in this analysis were defined as *invalid attempts*, which were attempts wherein students made no meaningful actions. In *Save Patch*, invalid attempts occurred largely because the student clicked reset twice in a row (either accidentally or due to impatience with the speed of the avatar) or because the student accidentally clicked “Go” immediately after a new level loaded (due to the initial location of the cursor directly above the “Go” button). If left in the dataset, these invalid attempts would artificially inflate the number of attempts those students required to solve each level and thereby indicate a greater level of difficulty than was actually the case.

Invalid attempts were identified and dropped using the following SPSS code, wherein Code_3000 was a count of the number of times a rope was selected in that attempt:

> *Calculate* DropAttempt = 0.
> *If* Code_3000 = 0 DropAttempt = 1.
> *Select If* DropAttempt = 0.

Remaining attempts were renumbered after all invalid attempts were dropped.

Additionally, a small number of students had not reached the portion of the game being analyzed. Approximately five percent of the students were dropped from the analysis because they had not reached the mixed numbers levels and therefore their learning trajectories for this content area could not be calculated.

3. EXPLORATORY ANALYSIS

Extracting the number of attempts each student required to solve each level reduced the dataset from over a million rows to only 21,713 rows of data (2,316 of which belonged to the subsample of students in the first 10% of the dataset, 413 of which occurred in the levels of interest). While this is too large of a volume of data for standard educational statistics, the data is also too small at this level for unsupervised, exploratory data mining techniques. Therefore, we decided to run some exploratory analyses to give us the information we would need to run a supervised data mining analysis.

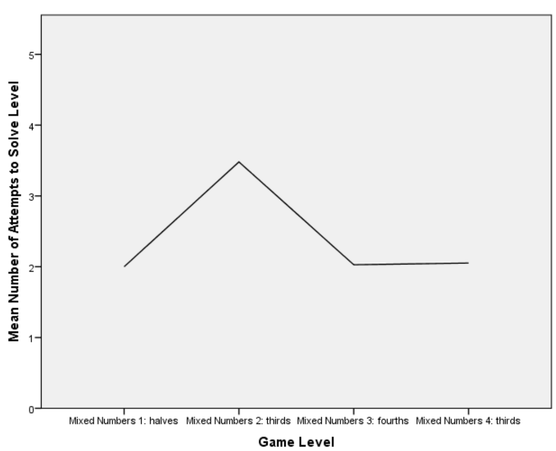

Figure 2: Mean Number of Attempts Per Level

An initial plot of the mean number of attempts students required to solve each of the mixed numbers levels is shown in Figure 2. This graph seems to indicate that the second level is more difficult than the other three levels, but does not otherwise seem to indicate any change in student performance as they move through the stage. Even given that the first level in the stage was designed as a training level and was intended to be much easier than other levels in the stage, it is difficult to make any claims about increased performance over time that might indicate student learning occurred. However, when examining performance curves over time, examining only mean values can hide more meaningful differences in learning trajectories between individuals (Gallistel, Fairhurst, & Balsam, 2004). Therefore, we decided to examine the individual learning trajectories of each of the students in our subset by hand.

3.1 IDENTIFYING LEARNING TRAJECTORIES

Only the first 10% of students in the sample was selected for the hand clustering dataset. The remaining 90% of the data was retained for subsequent data mining techniques. The individual learning trajectories for each of these 78 students were printed out. Similar to a hierarchical agglomerative clustering approach, we started with the first student's trajectory in a single cluster. Each subsequent student's trajectory was added to an existing cluster if it appeared substantively similar, or placed in a separate pile forming a new cluster if it appeared substantively different.

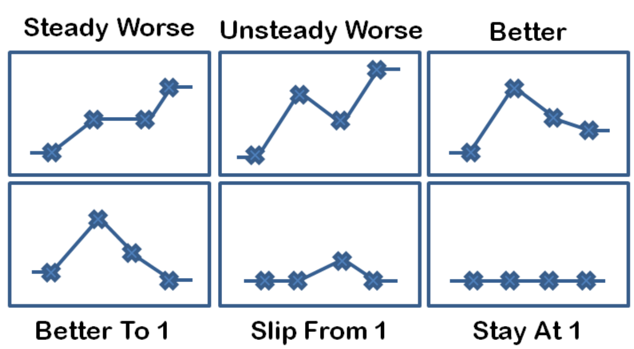

Figure 3: Identified Types of Learning Trajectories

The hand clustering resulted in six different groups of students, corresponding to six different types of learning trajectories (see Figure 3). The first type of learning trajectory demonstrated increasingly worse performance throughout the stage. In each consecutive level, these students (Steady Worse) took as many or more attempts to solve the level than they had required to solve the previous level. The second type of learning trajectory (Unsteady Worse) also demonstrated poorer performance later in the stage, but performed better on the third level in the stage than they had on the second level in the stage, resulting in a more ragged uphill trajectory.

The third type of learning trajectory (Better) performed consistently better on each of the last three levels of the stage, and the fourth type of learning trajectory (Better To

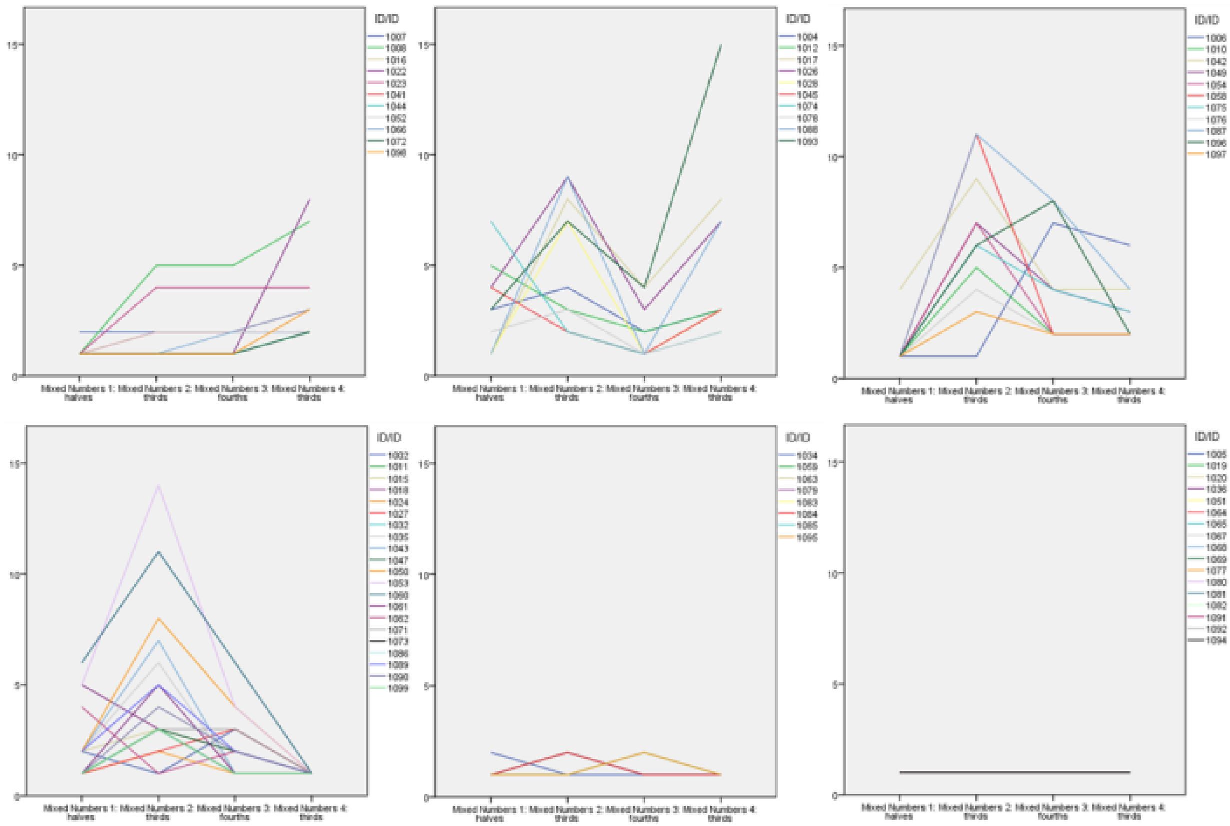

Figure 4: Student Learning Trajectories by Type

1) improved consistently better on the last three levels to the point that they solved the final level in their first attempt. The fifth type of learning trajectory (Slip From 1) solved all levels in the stage on their first attempt, except for one level which they took two attempts to solve. The sixth type of learning trajectory (Stay At 1) solved all levels in the stage on their first attempt, making no mistakes at all.

The individual learning trajectories for each student are plotted in Figure 4. The top three graphs represent (from left to right) students in the Steady Worse, Unsteady Worse, and Better learning trajectory types. The bottom three graphs represent students in the Better To 1, Slip From 1, and Stay At 1 learning trajectory types.

3.2 FINDING DIFFERENCES

In order to determine whether the learning trajectories were substantively different, and therefore worth further analysis, a number of exploratory ANOVAs were run

Students in the six different learning trajectory types differed significantly on both prior knowledge measures: the pretest score ($p < .001$) and prior math grades ($p = .024$). Slip From 1 and Stay At 1 had the highest mean pretest scores (4.42 and 4.22 respectively) and Unsteady Worse had the lowest (1.17). Similarly, Slip From 1 had the highest mean prior math grades (1.0 where 1 is an A) and Unsteady Worse and Better had the lowest (2.17 and 2.50 respectively). See Table 3 for results.

The learning trajectory types also differed significantly on in-game performance measures. There were significant differences between types in the percent of game levels completed ($p < .001$), but not the time they spent playing ($p = .889$), with Slip From 1 and Stay At 1 having the highest mean percentage of levels completed (84% and 87% respectively) and Better having the lowest (65%).

There were also significant differences in the percentage of students in the group solving the mixed numbers test level in their first attempt ($p < .001$) and in improvement between their performance on the corresponding level in the mixed numbers stage and the test level ($p < .001$). All students in Slip From 1 solved the mixed numbers test level on their first attempt (as did 82% of Stay At 1 students). Only 20% of Unsteady Worse, and none of the Better students, solved the mixed numbers test level on their first attempt. However, the Better, Better To 1, and

Table 3: ANOVA Results

MEASURE	**SIGNIFICANCE**	**BEST MEANS**	**WORST MEANS**
Pretest Score	$p < .001$	Slip From 1 (4.42) Stay At 1 (4.22)	Unsteady Worse (1.17)
Prior Math Grades	$p = .024$	Slip From 1 (1.0)	Unsteady Worse (2.17) Better (2.50)
Number of Game Levels Completed	$p < .001$	Stay At 1 (87%) Slip From 1 (84%)	Better (65%)
Time Spent Playing	$p = .889$	*no difference*	*no difference*
Solved Test Level on First Attempt	$p < .001$	Slip From 1 (100%) Stay At 1 (82%)	Unsteady Worse (20%) Better (0%)
Improve on Test Level	$p < .001$	Better (3.18) Unsteady Worse (2.80) Better To 1 (2.48)	Know (-0.18) Worse (-0.80)
Immediate Posttest	$p = .012$	Stay At 1 (5.78) Slip From 1 (5.50)	Unsteady Worse (2.71)
Delayed Posttest	$p = .010$	Stay At 1 (5.84) Slip From 1 (4.75)	Unsteady Worse (2.82)
Self-Belief in Math Before the Game	$p = .221$	*no difference*	*no difference*
Self-Belief in Math After the Game	$p = .022$	Stay At 1 (3.44) Slip From 1 (3.21)	Steady Worse (2.57) Unsteady Worse (2.33)

Unsteady Worse students all showed improvement between the corresponding level in the stage and the test level, taking an average of 3.18, 2.48, and 2.80 fewer attempts respectively to solve the test level.

Students in the different learning trajectory types also differed significantly on the immediate posttest ($p = .012$) and delayed posttest ($p = .010$), retaining most of the significant differences present in the pretest measures. As with the pretest, Slip From 1 and Stay At 1 had the highest mean immediate posttest scores (5.50 and 5.78 respectively) and delayed posttest (4.75 and 5.84), and Unsteady Worse had the lowest immediate posttest (2.71) and delayed posttest (2.82). However, the learning trajectory types also differed in their self-belief in math after the game ($p = .022$), though there was no significant difference before the game ($p = .221$). Slip From 1 and Stay At 1 had highest self-belief in math after the game (3.21 and 3.44 respectively), followed by Better and Better To 1 (3.07 and 2.82 respectively), with Steady Worse and Unsteady Worse having the lowest self-belief in math (2.57 and 2.33 respectively).

4. NEXT STEPS

Now that the six different learning trajectory types have been identified and evidence exists that the differences between the groups are substantive, the next step in our research is to test different cluster analysis techniques to determine which one best classifies students into these groups.

However, the accuracy of a cluster analysis technique depends, at least in part, on the appropriateness of the attributes used to create the distance matrix it operates on. There are three possible sets of attributes that might be used. First, the learning trajectories could be seen as splines. In this case, the attribute set would consist of the spline values, initial values, and ending values of each trajectory.

On the other hand, it might be more appropriate to treat the learning trajectories as a series of connected line segments. In this case, the attribute set would consist of the initial value, slope, and ending value of each line segment in each learning trajectory.

However, examination of the learning trajectories plotted in Figure 4 indicate that the value of each point may not be as important in determining which cluster a given learning trajectory falls in as the general shape of the trajectory. In this case, the attribute set would consist of a binary indicator of whether or not the initial value of each line segment was 1 or more than 1, a binary indicator of whether or not the ending value of each line segment was 1 or more than 1, and a set of binary indicators of whether the slope of each line segment was positive, negative, or neutral. These three options are summarized below.

1. Splines: initial value, spline values, ending value
2. Line Segments: initial value, slope, ending value
3. Binary Line Segments: initial value of 1 or more than 1, positive, negative, or neutral slope, ending value of 1 or more than 1

The distance matrix created from each of these three attribute sets will be fed into a hierarchical, partitioning, and fuzzy clustering algorithm. This will result in nine clustering techniques. Each of these clustering techniques will be run over the 10% of students whose learning trajectories have already been hand clustered in order to determine which technique best classifies the students.

Once the best clustering technique has been identified, it will be used to classify the remaining 90% of students in the sample into the learning trajectory type which best describes their in-game performance. Then a MANOVA will be run to determine which learning trajectory types differ on which measures across the entire sample (as opposed to the 10% reported in Table 3). If differences are found, the clustering technique could then be used (without requiring additional manual analysis) on attempt data from other stages in *Save Patch*, other *Save Patch* data collections, or other stages in similar games.

5. DISCUSSION

The logging technique used in this study resulted in a dataset that eased preprocessing and feature extraction. Additionally, the hand clustering led to the identification of six different types of learning trajectories who differed substantively on measures of prior knowledge, in-game performance, and posttest performance.

Perhaps the most interesting types of learning trajectories are the Better To 1, Better, and Unsteady Worse types. These trajectories appear to identify the potential learners for a given game, students who don’t know the material but are capable of learning from the game play. In contrast, the Stay At 1 and Slip From 1 trajectory types seem to identify students who already know the material and the Steady Worse trajectory type seems to identify students who do not know the material and are not learning from the game.

The results of this study seem to indicate that using data mining techniques to cluster learning trajectories would be a worthwhile endeavor, as the different clusters appear to correspond to substantively different groups of students. If the data mining results support the results of this study, it would not only support claims that educational video games and simulations can be used as stand-alone measures of student knowledge, but also provide the designers of those games with the information about which students’ needs are being met by the game.

However, it is possible that the findings of this study will not be supported by the data mining. This is only partially because the data mining might classify students differently than the hand clustering, and is mostly due to the fact that the small sample size in the hand clustered subset combined with the use of multiple ANOVAs rather than a single MANOVA might have identified some differences between learning trajectory types that occurred merely by chance. Currently, this study represents a promising process for analyzing data from educational video games, but the specific findings about performance should not be considered definitive without support from further studies.

Acknowledgements

The work reported herein was supported under the Educational Research and Development Centers Program, PR/Award Number R305C080015. The findings and opinions expressed here do not necessarily reflect the positions or policies of the National Center for Education Research, the Institute of Education Sciences, or the U.S. Department of Education.

References

Amershi, S., & Conati, C. (2011). Automatic recognition of learner types in exploratory learning environments. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.), *Handbook of Educational Data Mining* (pp. 389-416). Boca Raton, FL: CRC Press.

Avouris, N, Komis, V., Fiotakis, G., Margaritis, M., & Voyiatzaki, E. (2005). Logging of fingertip actions is not enough for analysis of learning activities. In *Proceedings of the Workshop on Usage Analysis in Learning Systems at the 12th International Conference on Artificial Intelligence in Education*.

Bejar, I. I. (1984). Educational diagnostic assessment. *Journal of Educational Measurement*, *21*(2), 175-189.

Carpenter, T. P., Fennema, E., Franke, M. L., Levi, L. W., & Empson, S. B. (2000). *Cognitively Guided Instruction: A Research-Based Teacher Professional Development Program for Elementary School Mathematics*. Madison, WI: National Center for Improving Student Learning and Achievement in Mathematics and Science.

Chung, G. K. W. K., & Kerr, D. (2012). *A primer on data logging to support extraction of meaningful information from educational games: An example from Save Patch* (CRESST Report 814). Los Angeles, CA: University of California, National Center for Research on Evaluation, Standards, and Student Testing.

Conati, C., & Merten, C. (2007). Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge Base Systems*, *20*(6), 557-574.

Famili, F., Shen, W. M., Weber, R., & Simoudis, E. (1997). Data pre-processing and intelligent data analysis. *International Journal on Intelligent Data Analysis*, *1*(1), 3-23.

Gallistel, C. R., Fairhurst, S., & Balsam, B. (2004). The learning curve: Implications of a quantitative analysis. *Proceedings of the National Academy of Sciences*, *101*(36), 13124-13131.

Hickey, D. T., Ingram-Goble, A. A., & Jameson, E. M. (2009). Designing assessments and assessing designs in virtual educational environments. *Journal of Science Education and Technology, 18*, 187-208.

Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B. C., Pagulayan, R. J., & Wixon, D. (2008). Tracking real-time user experience (TRUE): A comprehensive instrumentation solution for complex systems. In *Proceedings of the 26th annual SIGCHI Conference on Human Factors in Computing Systems* (pp. 443-452).

Koedinger, K. R., Baker, R. S.J.d., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2011). A data repository for the EDM community: The PSLC DataShop. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.) *Handbook of Educational Data Mining* (pp. 43-55). Boca Raton, FL: CRC Press.

Leighton, J. P., & Gierl, M. J. (2007). Defining and evaluating models of cognition used in educational measurement to make inferences about examinees' thinking processes. *Educational Measurement: Issues and Practice*, *26*(2), 3-16.

McNeil, N. M., & Alibali, M. W. (2005). Why won't you change your mind? Knowledge of operational patterns hinders learning and performance on equations. *Child Development*, *76*(4). 883-899.

Merceron, A., & Yacef, K. (2004). Mining student data captured from a web-based tutoring tool: Initial exploration and results. *Journal of Interactive Learning Research*, 15, 319-346.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2004). *A brief introduction to evidence centered design* (CSE Report 632). Los Angeles, CA: University of California, National Center for Research on Evaluation, Standards, and Student Testing.

Mostow, J., Beck, J. E., Cuneao, A., Gouvea, E., Heiner, C., & Juarez, O. (2011). Lessons from Project LISTEN's session browser. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S.J.d. Baker (Eds.), *Handbook of educational data mining* (pp. 389-416). Boca Raton, FL: CRC Press.

Muehlenbrock, M. (2005) Automatic action analysis in an interactive learning environment. In Choquet, C., Luengo, V. and Yacef, K. (Eds.), *Proceedings of the workshop on Usage Analysis in Learning Systems at AIED-2005*.

National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA.

National Research Council. (2011). *Learning science through computer games and simulations*. Washington, DC: The National Academies Press.

National Science and Technology Council (2011). *The federal science, technology, engineering, and mathematics (STEM) education portfolio*. Washington, DC: Executive Office of the President.

Quellmalz, E. S., & Pellegrino, J. W. (2009). Technology and testing. *Science*, *323*, 75-79.

Rahkila, M., & Karjalainen, M. (1999). Evaluation of learning in computer based education using log systems. In *Proceedings of 29th ASEE/IEEE Frontiers in Education Conference (FIE '99)* (pp. 16-22).

Romero, C., Gonzalez, P., Ventura, S., del Jesus, M. J., & Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: A practical application using Moodle data. *Expert Systems with Applications*, *39*, 1632-1644.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, *35*, 135-146.

Rupp, A. A., Gushta, M., Mislevy, R. J., & Shaffer, D. W. (2010). Evidence centered design of epistemic games: Measurement principles for complex learning environments. *The Journal of Technology, Learning, and Assessment*, *8*(4).

Scheuer, O., Muhlenbrock, M., & Melis, A. (2007). Results from action analysis in an interactive learning environment. *Journal of Interactive Learning Research, 18*(2), 185-205.

Siebert, & Gaskin (2006). Creating, naming, and justifying fractions. *Teaching Children Mathematics*, *12*(8), 394-400.

U.S. Department of Education (2008). *Foundations for success: The final report of the National Mathematics Advisory Panel*. Washington, DC.

Ueno, M. & Nagaoka, K. (2002). Learning log database and data mining system for e-learning: On line statistical outlier detection of irregular learning processes. In *Proceedings of the International Conference on Advanced Learning Technologies* (pp. 436-438).

Vee, M. N., Meyer, B., & Mannock, M. L. (2006). Understanding novice errors and error paths in Object-oriented programming through log analysis. In *Proceedings of the Workshop on Educational Data Mining* (pp. 13-20).

Vendlinski, T. P., Delacruz, G. C., Buschang, R. E., Chung, G. K. W. K., & Baker, E. L. (2010). *Developing high-quality assessments that align with instructional video games* (CRESST Report 774). Los Angeles, CA, University of California, National Center for Research on Evaluation, Standards, and Student Testing.