Modelling Student Behavior in Open-Ended Learning Platforms

Randi Williams
6.862 Applied Machine Learning
Final Report
May 12, 2017

Introduction

The constructivist approach to learning focuses on enabling students learn through autonomous exploration rather than through prescribed curricula. Open-ended learning platforms are powerful tools that promote learning through discovery. These platforms are often used to learn skill sets like scientific inquiry where formal problem sets are less appropriate. The explorative nature of these platforms makes it difficult to track student progress. To remedy this, most explorative platforms periodically use exams to measure student knowledge. However, the use of exams can undermine the free nature of the platforms. Using a dataset of student interactions with an open-ended learning platform, I aim to group students by learner styles. By grouping students into different learner styles I hope to later be able to predict their knowledge and provide interventions to improve their learning.

Dataset

The dataset was collected using the INQPRO learning platform Figure 1, an interface that students can use to explore scientific concepts and develop scientific hypotheses. The platform has 6 modules: Scenario, Hypotheses Visualization, Self-Verification, Formula Investigation, Simulated Experiment, and Data Comparison. Each module contains a different components like a simulator, buttons, and track bars. The platform also includes an intelligent tutoring agent, a parrot called Peedy. Peedy occasionally asks questions to evaluate a student's understanding and can provide help to the student upon request. The Scenario module describes the context in which scientific inquiry skills will be developed and has students construct a hypothesis statement, identify variables, and describe the relationship between the variables. The Hypothesis Visualization module allows the student to test their hypothesis with a series of simulated experiments. The Self-Verification module asks the student to reflect on their understanding of the hypothesis and variables. The Formula Investigation interface presents the experiment as a formula and has the student reflect on the hypothesis again. Peedy asks questions to probe for student understanding of all of the concepts. In the Simulated Experiment module the student again interacts with the variables, but using real life data rather than hypothetical data. Finally, in the Data Comparison module the student compares the real data to the hypothetical data. Peedy asks questions about the conclusions that can be drawn from the graphs.

The dataset includes the interactions of 55 students with the INQPRO platform. Each student is separated into an individual file containing seven tables describing the interaction. The first table is a complete log of the interactions between the student and the interface including typing, pressing buttons, and questions asked by Peedy. A sample of this log of interactions is shown in Figure 2. The number of rows in the table

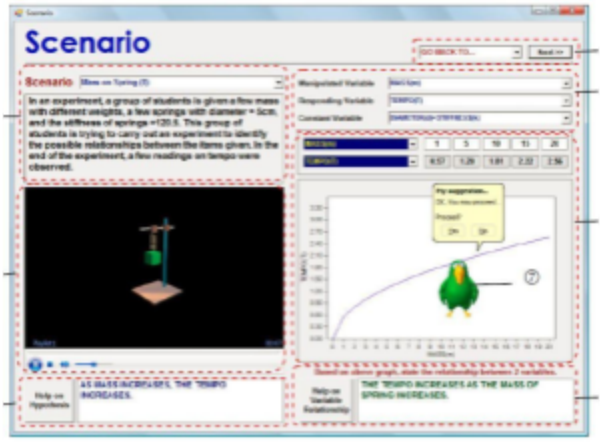

Figure 1: INQPRO learning environment, Scenario module

is equal to the number of interactions the student had with the platform during their entire interaction. Every interaction is timestamped. The interface component column refers to an object within a specific module that the student interacted with. Each module has its own set of interface components and each component has its own set of possible actions and values. Compared to similar datasets that only contain keypresses and mouse presses, this dataset provides a much richer set of information.

Student name	*Interface name*	*Interface component*	*Action*	*Value*	*Time*
Ting	Sce	Hypothesis	Type-in	If mass increases, tempo decreases.	10:20:23
Ting	Sce	Variable	Select	Manipulated—tempo Response—mass Constant—time	10:25:03
Ting	HypoViz	Simulation	Drag-and-drop	Drag—mass 50g	10:30:22
Ting	HypoViz	Graph	Mouse-move	–	10:31:41
Ting	Formula	Formula	Type-in	m = 2 k = 2	10:33:13
Ting	SimExp	Simulation	Drag-and-drop		10:36:21
Ting	SimExp	Result	Mouse-move	–	10:38:52
Ting	Sce	Hypothesis	Type-in	If mass increases, tempo increases.	10:40:12
Ting	Sce	Variable	Select	manipulated—mass response—tempo constant—k	10:42:17
Ting	Sce	Graph	Mouse-move	–	10:44:55
Ting	DC	Graphs	Mouse-move	–	10:45:02

Figure 2: Sample raw log data

The other six tables in the dataset contain a transformed log for each of the six modules in the INQPRO platform, as shown in Figure 3. The transformed log tables have a column for every possible action that can be performed in the module. The number of rows is equal to the number of visits to that module. This means that if Figure 1 represented the entire interaction of a student with the INQPRO platform, then the Scenario transformed log table would have two rows, one for each unique visit to the interface. The values in each column are dependent on the action. A value of "-" means that the student did not do that action during a particular visit or did not change their answer from a previous visit. Time nodes, indicated by a prefix of "t_" have values equivalent to the amount of time the student spent interacting with a component as determined by their mouse behavior. Student action nodes, indicated by a prefix of "SA_" or "SQ_" have values of "yes" if the student interacted with the component or chose the correct answer or "no" if the component was not interacted with. Agent action nodes are indicated by "AQ_" and represent

Peedy-initiated actions, usually a question checking in on understanding, where a “yes” indicates that the student answered correctly and “no” indicates that the student did not. Finally, the values of components in general, like sliding bars and text boxes, are recorded as well.

SQ_Definition	*SA_KeyInData*	*AQ_Concept*	*t_Hypo*	*t_variable*
Yes	Yes	No	60	30
–	–	Yes	50	20
–	–	–	10	10

Figure 3: Transformed Data Log Sample

With such a small dataset and no ground truth, my options for machine learning methods were limited to unsupervised learning methods. Also, every student interacted with the interface differently. This made it difficult identify metrics for analyzing the dataset.

Related Work

In 2015, Ting et. al released the INQPRO dataset. This included the interactions of 100 students with a scientific inquiry learning environment, INQPRO (Ting & Ho, 2015). This dataset allows students to learn about scientific inquiry and how to construct and test scientific hypotheses to learn. Ting et. al used this dataset to model students’ behavior using a Bayesian network to track knowledge states and learning policies (Ting, Cheah, & Ho, 2013). Another study modeled when conceptual change using Bayesian Networks, or the replacement of an old belief was replaced with a new belief (Ting, Sam, & Wong, 2013). In both of these cases, the Bayesian networks were designed by experts and not learned.

Within the space of educational data mining, there has been work on modelling student affect states from their actions in Intelligent Tutoring Systems (ITS). In 2010, Cetintas et. al used the amount of time it took to complete a task, the student’s previous performance, and the frequency of mouse movements to detect if a student was off-task. This successfully predicted 0.85 of off-task behaviors using a ridge regression algorithm. In 2015 Leong was able to predict student frustration from their keystrokes with an accuracy of 0.67 using logistic regression. In 2016, Klinger et. al emphasized the importance of not only recognizing student affect states, but recognizing that student interaction patterns evolve over time and can be clustered.

Problem Statement

In order to provide effective interventions, it is important to understand a student’s interaction patterns. The primary objective is to build an algorithm that clusters students into learner types. In completing this task I attempt to answer the following questions:

1. Which features of the interaction log can be used to group students
2. What kinds of characteristics do similar students share
3. Do the student groupings contain underlying information about how students are learning

Clustering algorithms seek to organize a set of data points $x_i \in \Re^d$ into k clusters $\{c_1, ..., c_k\}$ by minimizing an objective function J. Where i is the index of the data point and takes values 1 through n, the total number of data points and d is the length of the feature vector. The set of data points in this case is a feature vector describing each student. The objective function is a measure of the distance between points. It depends on the clustering algorithm.

Methodology

Clustering data was approached by first processing the dataset, selecting parameters and clustering algorithms, performing the actual clustering, and then interpreting the results of the clusters. Figure 4 shows all of the parts involved in the approach used.

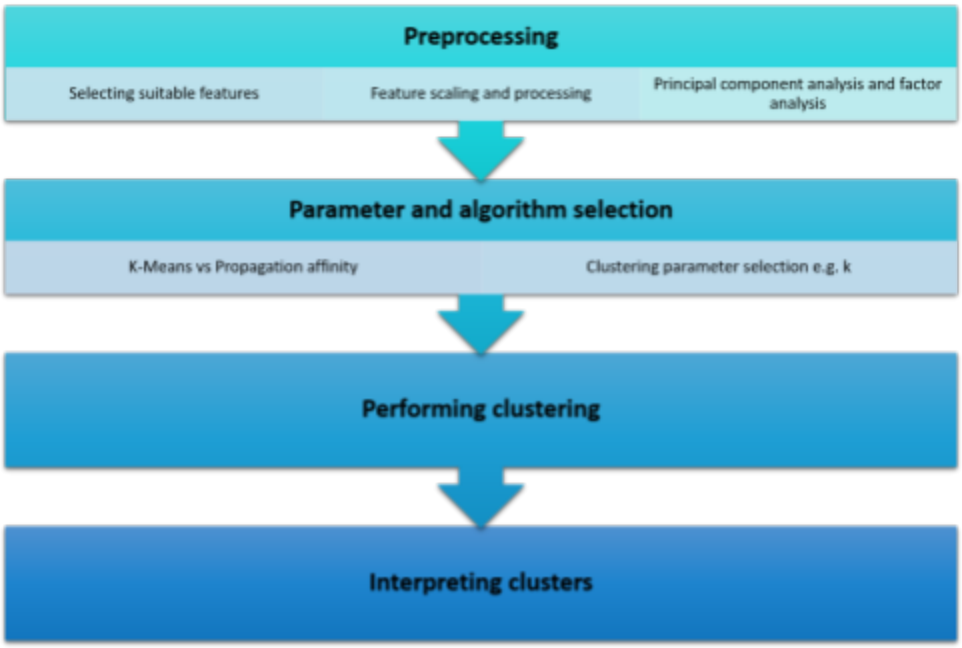

Figure 4: Methodology of the clustering approach

Preprocessing

Initially, the dataset was stored as 55 separate database files. The first step was extracting the features I wanted from each student and saving a single matrix of containing a feature vector for all students. I collected three feature sets.

The first feature set stored all interaction data found in the raw data log. First, I created a dictionary of all possible actions in the raw data log. The dictionary contains all of the names of actions from the "Action" column. First, all of the interactions are stored, then the interactions are stored again, but sorted into the module they occurred in. This was meant to capture information such as the total number of mouse moves compared to the number of mouse moves that happened in a particular module.

Feature dictionary: { All interactions } ∪ { Scenario interactions } ∪ { Hypothesis visualization interactions } ∪ { Self-verification interactions } ∪ { Formula investigation interactions } ∪ { Simulated experiment interactions } ∪ { Data comparison interactions }

The feature vector for each student was then evaluated as a count of the number of times a particular action occurred plus the total amount of time spent overall and in each module plus the total number of interactions done overall and in each module. Each feature vector had a length of 193. Since the feature vectors include counts, times, and frequencies the data is quite heterogeneous. To remedy this, I used scaling on the frequency data so that all the features were between 0 and 1. As for the counts of every action, some actions occurred more than other, mouse moves for example occurred an average of 1,317 times across the students. I used term frequency - inverse document frequency features so that more rare actions receive greater emphasis in the dataset. Finally, the large number of features made it necessary to use feature selection. For this, I compared principal component analysis to factor analysis. An example of the processed data is shown in Figure 5.

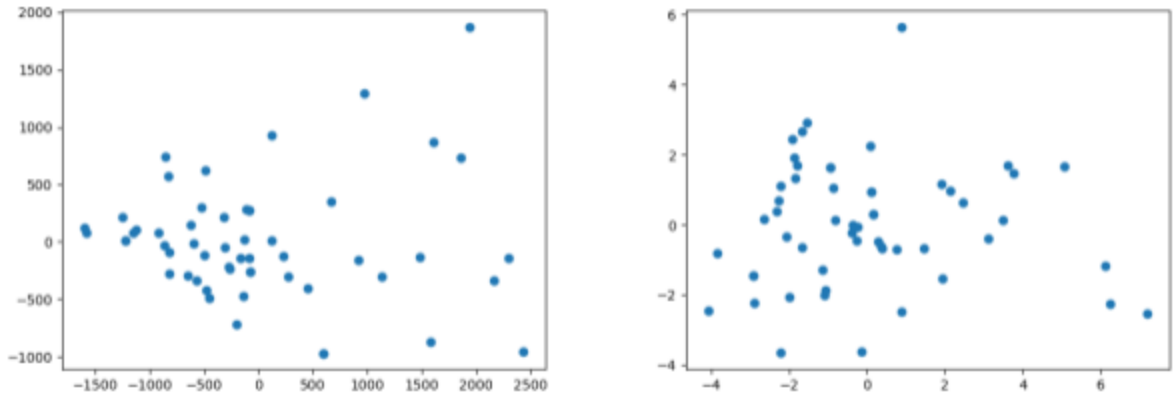

Figure 5: Raw data (left) and processed data (right) plotted with two component PCA

The second feature set records the number of times the student transitions between modules in the raw data log. This was meant to capture whether students moved through modules sequentially or jumped around. For each student, I created a 6 by 6 matrix of transitions say from the Scenario module to the Hypothesis Visualization Module. To account for the differences in the numbers of interactions of each student, I divided the counts by the total number of interactions done by each student. Finally, I flattened the transition matrix into a vector of length 36. A heatmap of the data is shown in Figure 6. The heatmap shows that some students jump around, but most work through one module before proceeding to the next.. It also shows that some students stay in particular modules for longer times.

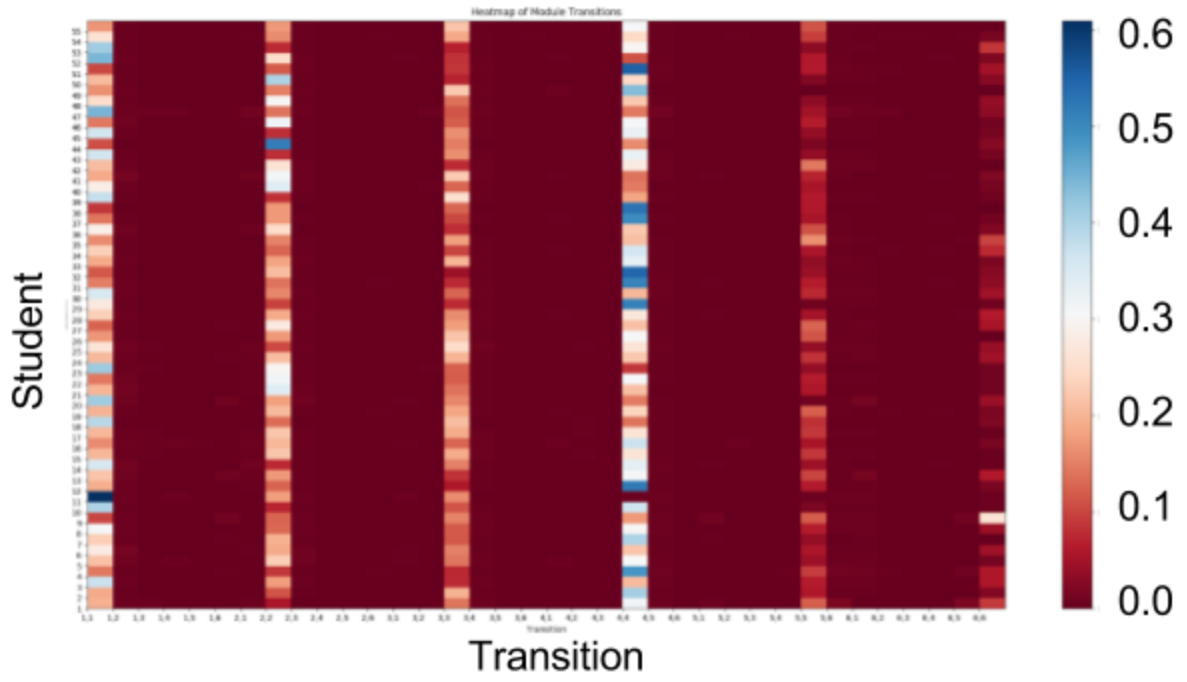

Figure 6: Heatmap of student transitions

The third feature set used the transformed data log to capture information about the data the student entered into the platform. The other two feature sets were meant to be inputs to the clustering algorithm, but this final feature set will be used to evaluate the clusters. In particular, this feature set focused on the questions that Peedy asked to determine student's understanding and platform components that asked the students how much they understood. In total, the feature vector contained 30 data features containing information like "-" "mastery" "non (mastery)" "partial (mastery)", "high", "low", "yes", and "no".

Parameter and algorithm selection

I tried two clustering algorithms, k-means and affinity propagation.

The K-means algorithm iteratively groups points into clusters by choosing k points as the center of a cluster and choosing all of the points closest to that centroid to be a part of the cluster. Closeness is defined as the Euclidean squared distance between points $d = \sum_{i=1}^{d}(x_i - y_i)^2$. Then, with the fully formed clusters the centroids are recomputed. This continues until the clusters converge or a maximum number of iterations is reached. In order to use this algorithm I empirically searched for a k that made sense given the data. This was done by calculating the clusters and evaluating them to see if they made sense.

The affinity propagation algorithm iteratively groups points into clusters by comparing pairs of data points. The similarity between two points $s = -||x_i - y_i||^2$ should be maximized between two points,

exemplars and a point in the exemplar's cluster. The algorithm searches for exemplars where the total similarity is greater than the previous total similarity. This algorithm is good for finding clusters in smaller datasets. The number of clusters is indirectly related to the preference parameter. This parameter was determined by empirical methods.

Performing clustering

Clustering was implemented using Python's scikit learn library. The clustering algorithm was performed on the processed data; its performance on raw data was also evaluated. The results of the clustering were printed and plotted on a two-component PCA.

Interpreting clusters

After clustering was performed, datasets were reunited with the data in the third feature set for comparison. The k-means clustering did not choose an exemplar from the dataset, so the average member of each cluster was calculated for this evaluation.

From the third feature set I derived six metrics to evaluate student understanding. The first two metrics are from the Scenario module, where students input their prior understanding of the variables in the scenario ("high" understanding or "low") and then are asked to construct a hypothesis statement and are asked how much they mastered the construction of the hypothesis ("mastery, "partial" mastery or "non"). The next two metrics come from the Self-verification module after students have evaluated their hypothesis through simulation. Students now reevaluate their understanding of the variables ("mastery", "partial, or "non") and the hypothesis ("mastery", "partial", "non"). Then, students are quizzed on the variables in the Formula module, so I calculate the number of students who get both questions right. Finally, students are quizzed in the last module, Data Comparison, on their understanding of everything and I calculate the number of students who get all of those questions right.

Results

Interaction Data

The fewest number of interactions was 153, the largest was 704. On average, every student did 339.6 interactions during their session. The least amount of time spent was 8 minutes, 3 second, and the most time spent during one session was 52:06. On average, students spent 27 minutes interacting with the interface.

In comparing PCA to FA I used the percent explained by variance and log likelihood as metrics. The result was that a 40-component PCA explained 0.99 of the variance. A 50 component FA resulted in a log likelihood of 1888, which is close to the asymptote. After clustering, the PCA data had more stable clusters and visually looked better.

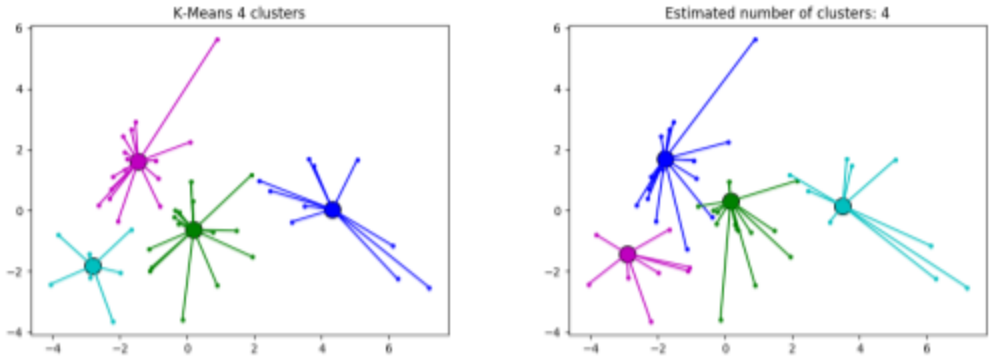

Figure 5: K-means (left) and affinity propagation (right) clusters for k=4

I settled on wanting four clusters to describe the interaction data. This meant setting *k=4* in the k-means algorithm and *preference=-80* in the affinity propagation algorithm. Four clusters was chosen because it resulted in the most reasonable clusters in the analysis step and in graphing. I also tried to use the silhouette coefficient as a metric for the appropriateness of the clustering. The silhouette coefficient is a value between -1 and 1 that describes how close clusters are to one another. If a data point is perfectly on the boundary between two clusters it will have a silhouette coefficient of 0. However, the silhouette coefficient is not a perfect measure. For sparse data, like what I am evaluating, the highest silhouette coefficients occur when there are 50 or more clusters and at 3 or fewer clusters. I decided that four clusters was reasonable and close enough to a high silhouette coefficient.

The resulting clusters of the k-means and affinity propagation algorithms are shown in Figure 5. The two clustering algorithms produced very similar clusters. The silhouette coefficient for the k-means clustering was higher because it was not restricted to using one of the data points as an exemplar.

Table 1: Interaction Characteristics of Cluster Exemplars, Interaction Data

Cluster	n	Total Time	Total Interactions
1	18	20:10	210
2	17	25:26	345
3	10	40:42	488
4	10	13:03	256
Avg	55	27:00	339

Table 2 shows the percent of students in each cluster who felt they mastered the topics or got all questions correct in the short quizzes. The only significant difference between the clusters was the prior mastery of variables exhibited by cluster 4. For all other variables, we do not see significant differences, but we

observe some trends. Overall, cluster 1 had the lowest prior knowledge and the lowest overall quiz score. This cluster did the fewest interactions in in the platform, perhaps this shows that their lack of initial knowledge led them to unsuccessful learning. Clusters 2 and 3 had similar levels of prior and post knowledge, but cluster 3 did many more interactions than cluster 2. Finally, cluster 4 exhibited the most prior knowledge out of all of the groups, but did the fewest interactions and had a drop in confidence at the end of the session. This perhaps suggests that these students became confused at some point while interacting with the platform and did not spend time to overcome their confusion, choosing instead of end the session more quickly.

Table 2: Evaluation Metrics for Clusters, Interaction Data
* $p<0.05$

Cluster	Prior Knowledge "Mastery"		Post Knowledge "Mastery"		Quizzes - All correct	
	Variables	Hypothesis	Variables	Hypothesis	Variables	Overall
1	0.44	0.44	0.56	0.44	0.80	0.39
2	0.47	0.59	0.71	0.71	0.71	0.62
3	0.5	0.7	0.8	0.8	0.65	0.7
4	0.8*	0.7	0.6	0.6	0.71	0.55

Transition Data

The fewest number of transitions between different modules was 4, this represented an incomplete interaction with the platform since there are 6 modules. The most number of transitions was 41, this student jumped around a lot especially to the first module. On average, students transitioned between modules 11.3 times, suggesting that they went through each module at least once and went back and forth between modules too. Students did the most interactions in the Scenario and Formula Investigation Modules. This data did not have to be scaled or processed like the former module since the information was more homogeneous. PCA showed that the first two components explain 0.8 of the variance.

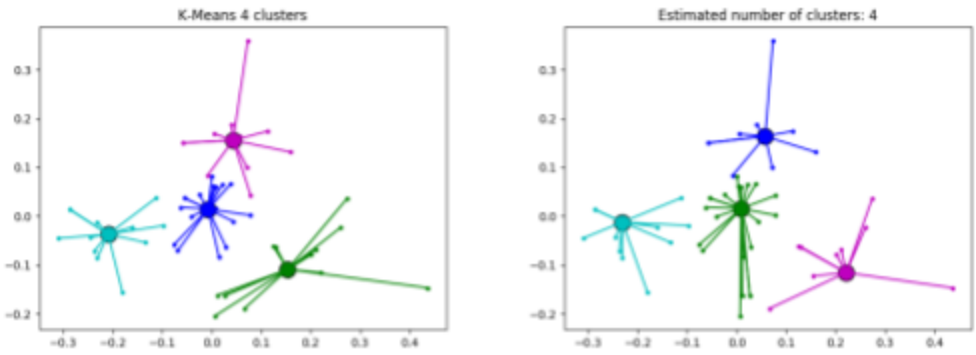

Figure 6: K-means (left) and affinity propagation (right) clusters for k=4

I compared k-means clusters to affinity propagation clusters. I used four clusters because it had the highest silhouette coefficient within the range of reasonable ks. This meant setting *k=4* for the k-means algorithm and *preference=-0.2* in the affinity propagation algorithm. The results are shown in Figure 6.

These clusters are also very similar to one another. The affinity propagation algorithm has a silhouette coefficient of 0.441 and k-means had a silhouette coefficient of 0.433. The only difference is that the cluster in the middle of the graph chose more points with affinity propagation.

Table 3: Transition Characteristics of Cluster Exemplars, Transition Data

Cluster	n	Total Time	Total Transitions
1	10	41:42	17
2	23	18:20	11
3	12	23:26	7
4	10	28:09	15
Avg	55	27:00	11.3

Table 4 show the evaluation metrics for each of the clusters. This clustering showed more significant differences between the clusters, in particular for people learning the material well and struggling with the material. Cluster 1 clearly represents the students who had prior experience with the material, spent sufficient time with the platform, and then retained their knowledge. Cluster 3 on the other hand clearly shows the people who had partial understanding of the material coming in and were clearly confused by the end of their session. These students also did very little jumping around between modules, suggesting that they did not exhibit self-regulation techniques.

Table 4: Evaluation Metrics for Clusters, Interaction Data

* p<0.05, **p<0.01

Cluster	Prior Knowledge "Mastery"		Post Knowledge "Mastery"		Quizzes - All correct	
	Variables	Hypothesis	Variables	Hypothesis	Variables	Overall
1	0.9**	0.7	0.9**	0.9**	0.8	0.75*
2	0.48	0.52	0.78	0.61	0.7	0.54
3	0.5	0.5	0.33**	0.42	0.7	0.46
4	0.5	0.5	0.5	0.6	0.7	0.45

Conclusions

The goal of this work was to explore three things: how should features be selected from a log of student interactions with an open-ended learning environment, what kinds of clusters of students naturally arise, and what underlying information might these clusters hold.

Compared to the feature vector of all actions, the transition matrix features produced more structured data. My intuition is that in this open environment there are so many possible actions that only a very large dataset would be able to capture an underlying structure amongst the interactions a student has with a platform. My datasets was simply insufficient. However, even from these few data points the transition feature set was able to capture a lot of the differences between students. This makes me hopeful because this is an easy set of features to extract.

In my analysis I used four clusters, guided by what seemed reasonable and the silhouette coefficient. In both the interaction and transition feature set there were clusters that were much larger than the others. Perhaps a fifth cluster would be a better representation of the data. That said, my preliminary analysis of the k=5 clusters did not show any signs of significantly different groups of students. In the case of the transition matrix the fifth cluster was a division between clusters 2 and 4.

While I can rely on the transition feature set to give underlying information about expert students and struggling students the differences between clusters 2, 4, and in a k=5 clustering, 5 are yet to be seen. Perhaps the differences between the clusters would be more clear with formal pre/post test data or affect data. The students in cluster 4 spent more time in the platform and did more transitions, so perhaps they actually have a better understanding of the material than students in cluster 2.

The natural next step in this work is to classify students into clusters while they are working through the platform. This would allow us to perform interventions on the student if, say, we know a student is struggling then we can encourage them to self-regulate and reflect more on previous modules.