Semester Project - CSCI 5622
Woot Math Unsupervised Learning Task

Shane Lockwood
shane.lockwood@colorado.edu

Brian Lubars
brian.lubars@colorado.edu

Brian McKean
brian.mckean@colorado.edu

Carl Mueller
carl.mueller@colorado.edu

Geoffrey Sutcliffe
geoffrey.sutcliffe@colorado.edu

December 15, 2017

Abstract

Elementary school students are at a critical point in their mathematical education wherein basic intuitions are formed. Identifying conceptual gaps related to these mathematical concepts, and fixing them as soon as possible, is clearly valuable. Machine learning techniques can help identify such student errors, but require relevant data to operate. Fortunately, as a maker of educational software teaching math to elementary school students, Woot Math is uniquely positioned to provide such a dataset. In this paper, we explore two unsupervised learning approaches to discover 'behavioral traits' that may indicate such conceptual errors. The first approach samples a subset of the entire dataset, converts the student responses from JSON to a text document, and then does k-means clustering on TF-IDF vector of the document text. The second approach selects only the incorrect answers from a specific question, extracts all relevant features, and runs k-modes or k-means to group incorrect responses. Both approaches yielded meaningful results. The text k-means found existing traits in clusters and identified clusters of lessons with no previously identified traits. The single-question k-modes found clusters of specific ways students incorrectly solved problems. Woot Math asked for details on lessons grouped in 'trait-less' clusters from the first approach so they can look for possible new traits. From the second approach Woot Math was able to identify a new trait instantly from one cluster of the data we presented to them.

1 Introduction

Woot Math is a Boulder-based education software startup founded in March, 2013. The company focuses on helping teachers customize learning for students who are struggling with core math concepts, beginning with rational numbers. It is a browser and tablet-based software that is designed to be used independently by students in the classroom or at home as a supplement to the core curriculum. Woot Math delivers a personalized progression of interleaved video instruction and scaffolded problems to mimic the natural give and take between a student and a tutor[1].

1.1 Project Motivation

Students grades 3-8 are in the midst of learning the core mathematical concepts that provide the foundation necessary to future academic success. Students that fail to reach an adequate mastery of these topics exhibit a measurable deficiency in their future academic performance. Woot Math strives to improve outcomes for students of all ability leveles through adaptive presentation of difficulty levels in lessons.

Woot Math's software responds to the progress of the student as they work through topics and lessons. In an attempt to further refine the adaptive capabilities of the software, Woot Math has identified a large number of behavioral mistakes and/or misunderstandings they have labeled 'behavioral traits'. These traits indicate common misconceptions and misuse of software tools, and this data spans a variety of settings and grades. Some examples of these traits are

'incorrect modeling', 'recognizing a whole fraction', and 'simplifying mixed numbers'. Woot Math will assist the student exhibiting these traits with hints or supplementary lessons.

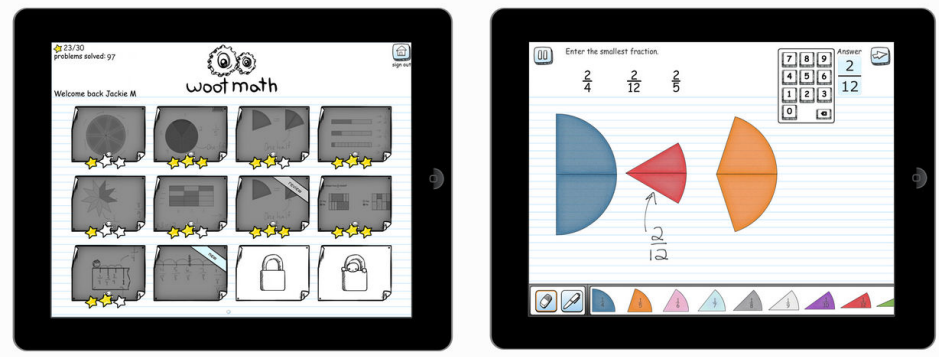


Figure 1: The Woot Math software interface. The right image shows the student workbench which the dataset most represents.

One goal is to automatically identify these misconceptions or traits as students exhibit them based on their problem submissions. Such identification could help in two major ways. First, the software can immediately provide assistance to a student who would otherwise spend unnecessary time struggling with the problem – for example, providing an instructional session addressing the specific 'behavioral_trait' identified. Second, teachers would benefit immensely from the ability to identify conceptual problem areas of their students. Instant feedback on areas that need clarification could help teachers to improve their lesson plans or teaching styles.

Currently, while Woot Math provides instructional videos to address common incorrect answers, there is room for improvement. Oftentimes their instructional videos are too general to provide the student the insight into why their approach is incorrect. Providing effective learning through Woot Math's novel technology is mission-critical for both Woot Math as a company and for their desired goal to improve the education of essential mathematical topics necessary for students to excel in their future academic careers.

1.2 Project Task

The project goal was to use unsupervised machine learning to help identify common student misconceptions or gaps in a student's knowledge of grade-level mathematics.

Woot Math tasked human experts to label a significant percentage of a 10 million submission dataset with behavioral traits. With the given dataset, our team hoped to apply machine learning to both validate these behavioral traits and to potentially reveal new patterns that point to new or existing classes of traits. This is an unsupervised learning problem, since we are attempting to discover latent structure in the dataset as it relates to behavioral traits. We tried a variety of feature extraction methods and unsupervised approaches, specifically K-means and K-modes, and analyzed their effectiveness in both identifying existing behavioral traits and in identifying potential new ones.

2 Dataset

Woot Math provided a 23GB bson-format dataset stored in a MongoDB database for this project. The database contains over 10 million attempts by students to solve over 27,000 unique math problems. The attempts are recorded in the form of variables tracking final canvas states at the point when the answer was submitted to the database, specifically in the 'response' field. This 'response' field contains over 3000 possible subfields, which vary from problem-to-problem and response-to-response.

A subset of the attempts were tagged with behavioral traits as discussed in the previous section. There are 62 expert-identified traits assigned to approximately 20% of the problems.

The dataset was not designed for machine learning, but rather is a dump from their product database, used for the day-to-day operations of the software. The dataset was anonymized by Woot Math to remove any identifying student information before we received it.

3 Model Choice and Background

3.1 Algorithm Choice

We opted for two major unsupervised learning algorithms: k-means & k-modes. As the data can be represented as feature vectors and categorical data, both these algorithms are well suited to cluster the data.

3.1.1 K-means

K-means is an unsupervised clustering approach that groups vectorized data around centroid points. These centroid points in the feature space are generated through iterative averaging of centers of mass of the dataset into k-Veronoi cells i.e. k-partitions of the hyperspace. See **Algorithm 1** for the pseudocode of this algorithm.

input : $D = \{d_1, d_2, d_3, ...\}$ vectors each of size n (D-documents, n-features)
k (number of clusters)
m (Maximum Number of Iterations)
output : $C = \{c_1, c_2, ..., c_k\}$ (set of cluster centroids)
$L = \{\ell(d) | d = 1, 2, .., n\}$ (set of cluster labels of D)

foreach $c_i \epsilon C$ **do**
  $c_i := d_j \epsilon D$ *(select initial centroids based on kmeans++ algorithm)*
end
foreach $d_i \epsilon D$ **do**
  $\ell(d_i) := argminDistance(d_i, c_j)\ j \epsilon \{1..k\}$ *(assign each* $\ell(d_i)$ *cluster* c_j *based on minimum distance)*
end
unchanged := true;
iter := 0;
while *unchanged = false and iter* $\leq m$ **do**
  foreach $c_i \epsilon C$ **do**
    $c_i := averageCenterOfCurrentCluster(\{d_1, d_2, ..., d_j\} | \ell(d_j) = c_i)$
  end
  foreach $d_i \epsilon D$ **do**
    $\ell(d_i)_{new} := argminDistance(d_i, c_j)\ j \epsilon \{1..k\}$;
    if $\ell(d_i)_{new} \neq \ell(d_i)$ **then**
      $\ell(d_i) := \ell(d_i)_{new}$;
      unchanged := false;
    end
  end
  iter++;
end

Algorithm 1: K-means Algorithm

3.1.2 K-modes

K-modes is an unsupervised clustering approach that groups categorical data located around centroid modal points. Each centroid represents a set of feature values from a document in the dataset. Each document is assigned to the centroid with the most similar feature values. See **Algorithm 2** for the pseudocode of this algorithm.

input : $D = \{d_1, d_2, d_3, ...\}$ count vectors each of size n (D-documents, n-categories)
k (number of clusters)
m (Maximum Number of Iterations)
output : $C = \{c_1, c_2, ..., c_k\}$ (set of cluster centroids)
$L = \{\ell(d) | d = 1, 2, .., n\}$ (set of cluster labels of D)

foreach $c_i \epsilon C$ **do**
  $c_i := d_j \epsilon D$ *(assign initial centroids based on density)*
end
foreach $d_i \epsilon D$ **do**
  $\ell(d_i) := argminDistance(d_i, c_j)\ j \epsilon \{1..k\}$ *(assign each $\ell(d_i)$ cluster c_j based on minimum distance)*;
  // Distance is calculated based on agreement of categories. How many categories of document are in agreement with centroid document.
end
changed := true;
iter := 0;
while *changed = true and iter* $\leq$ *m* **do**
  foreach $c_i \epsilon C$ **do**
    $c_i := modalCenterOfCurrentCluster(\{d_1, d_2, ..., d_j\} | \ell(d_j) = c_i)$ // Cluster is assigned to the document label with the highest count of a given category.
  end
  foreach $d_i \epsilon D$ **do**
    $\ell(d_i)_{new} := argminDistance(d_i, c_j)\ j \epsilon \{1..k\}$;
    if $\ell(d_i)_{new} \neq \ell(d_i)$ **then**
      $\ell(d_i) := \ell(d_i)_{new}$;
      changed := false;
    end
  end
  iter++;
end

Algorithm 2: K-modes Algorithm

4 Feature Engineering

Feature engineering took two distinct approaches that dictated parallel modeling paths. Each approach generally pulled from the same data field, making heavy use of the 'response' field, but were distinct in their subset of data and model application of k-means versus k-modes.

4.1 Fields of Interest

Woot Math provided an outline describing each field of the submission .bson data. They placed extra emphasis on the 'response' field as well as the 'description' and 'txt' fields. The 'response' field is a deeply nested key-value dictionary

indicating the aggregate arrangement and type of software elements a student used at the time of submission. The 'description' is an internal corporate description of the lesson. The 'txt' field is the text that the student sees associated with the lesson. The 'qual_id' field was identified as a potential partitioner of the dataset as it identified specific problem types.

4.2 Approach 1

In the first approach, we created a bag of words vector for each document derived from the 'response' key-value bson, 'description' and 'txt' fields. The response data for each student session includes the segments of the learning tool that the student used in solving the lesson. This includes the tools used, such as 'fraction_circle_groups' and also specific instantiation of the tool, such as 'pieces: 1/8' indicating that the fraction-circle piece used was 1/8 of a circle.

In order to generalize the responses and cover all variations, keys, values, and key-value pairs (joined with underscores) were concatenated into strings. In addition, any URLs in the fields are decomposed as they appear to have information. Each document therefore consisted of a bag-of-words for which Term Frequency - Inverse Document Frequency (TF-IDF) feature reduction was performed during vectorization processing.

Table 1: Example of json String Concatenation for Bag-of-Words Corpus Generation

Response Data	Generated Documents
'radio_group_mc1': {'choice': 'B', 'text': 'No'}, 'radio_group_mc2': {'choice': 'A', 'text': 'Yes'}	radio_group_mc2_choice_A radio_group_mc2_text_Yes radio_group_mc1_choice_B radio_group_mc1_text_No

This form of vectorization provided vectors mapping to a continuous vector space in which we could perform k-means clustering. We chose this approach since the key data in the 'reponse' field was extremely varied between types of problems and document/text/topic techniques easily handly this variety.

4.3 Approach 2

The second approach ran clustering on incorrect student attempts within a specific problem (qual_id in Woot Math's database terminology). For each unique problem, we identified all of the student attempts on that problem which were incorrectly answered. Each student submission for that problem becomes a document. By partitioning the data this way, clustering techniques can reveal common ways that students incorrectly answer each specific question. A human can then look at the discovered clusters and intuit the reasons behind the students' mistakes (be they conceptual errors on the students' part, or issues with the question or software).

Using all of those documents, we generated a custom feature vector using the union of all keys from all documents in that problem. This way, we have a common feature vector for every document's 'response' field. A category based vectors represented whether or not one of the categories were present or not. Any key-value pair feature that a specific document is missing from the problem's global set of features is recorded as a 0.

This unique set of features made it difficult to compare clusters across different problems. Therefore, we were only interested in identifying the latent structure within specific problems. Given a common feature set across all problems, this approach may be extended in the future.

5 Modeling

5.1 Approach 1

For our first approach, we used the k-means clustering algorithm. Based on time performance constraints, we chose 100,000 random samples from the 10 million sample set. The approach did not perform any partitioning or filtering of

the training set. By using bag-of-words vectorization we were able cluster in a word count vectorization hyperspace. k = 100 was chosen as our number of clusters for this approach.

The goal of this approach was two-fold. First, we wanted to try to reproduce the trait mapping supplied to us as part of the data set. Second, we wanted to identify clusters of samples that did not have associated traits. By exploring clusters that did not map strongly to a trait or set of traits, we might be able to discover new patterns amongst the documents in those clusters. Additionally, any cluster that contained multiple traits, we hoped to identify important relationships between student submissions.

5.2 Approach 2

We approached clustering of student attempts with several models, and compared their results:

1. String-based Discrete Clustering: K-modes
2. Numeric Clustering: K-means, DBSCAN

Our hope with this approach is to identify common incorrect-answer-signatures that students make which indicate a specific misconception or gap in knowledge. In a complicated problem, we see a typical pattern emerge. A noticeable portion of students (commonly anywhere from 15% to 20%) of students will have nearly-identical canvas states when they missed the problem, which we infer to mean that they made the same mistake. Then there will be four or five smaller but still significant groups of canvases (2% to 10% of students). By clustering on these canvas states and identifying the most populous clusters, we hope to provide the teachers or Woot Math with a powerful tool to identify new types of mistakes. By saving the centroids from k-means or k-modes, the software could even identify these mistakes in real-time when they reoccur!

6 Results

6.1 Approach 1

Running the k-means algorithm on the 100,000 sample subset generated clusters that appear meaningful. Some clusters contained numerous traits, some clusters only contained one trait, while others had very little clustering included.

While we did not achieve perfect one-to-one mapping between clusters and traits, the traits that were grouped heavily in clusters shared some obvious commonalities. For example, cluster 34 in figure 2 contains a series of labels associated with decimal comparisons which can be seen in table 2.

Table 2: Trait-Document Count within Cluster 34

Trait	Document Count Within Cluster
deci_add_to_model	113
deci_break_tenths_hundredths	34
deci_building_tenths	46
deci_compare_no_models	7
deci_forgot_decimal_point	18
deci_hops_instead_ticks	2
deci_incorrect_inequality	30
deci_placing_decimal_points	3
deci_point_location_correct	2
deci_tens_vs_tenths	317
deci_understanding_gt_lt	1425

The subset of traits heavily present in cluster 34 show a strong similarity of the trait types.

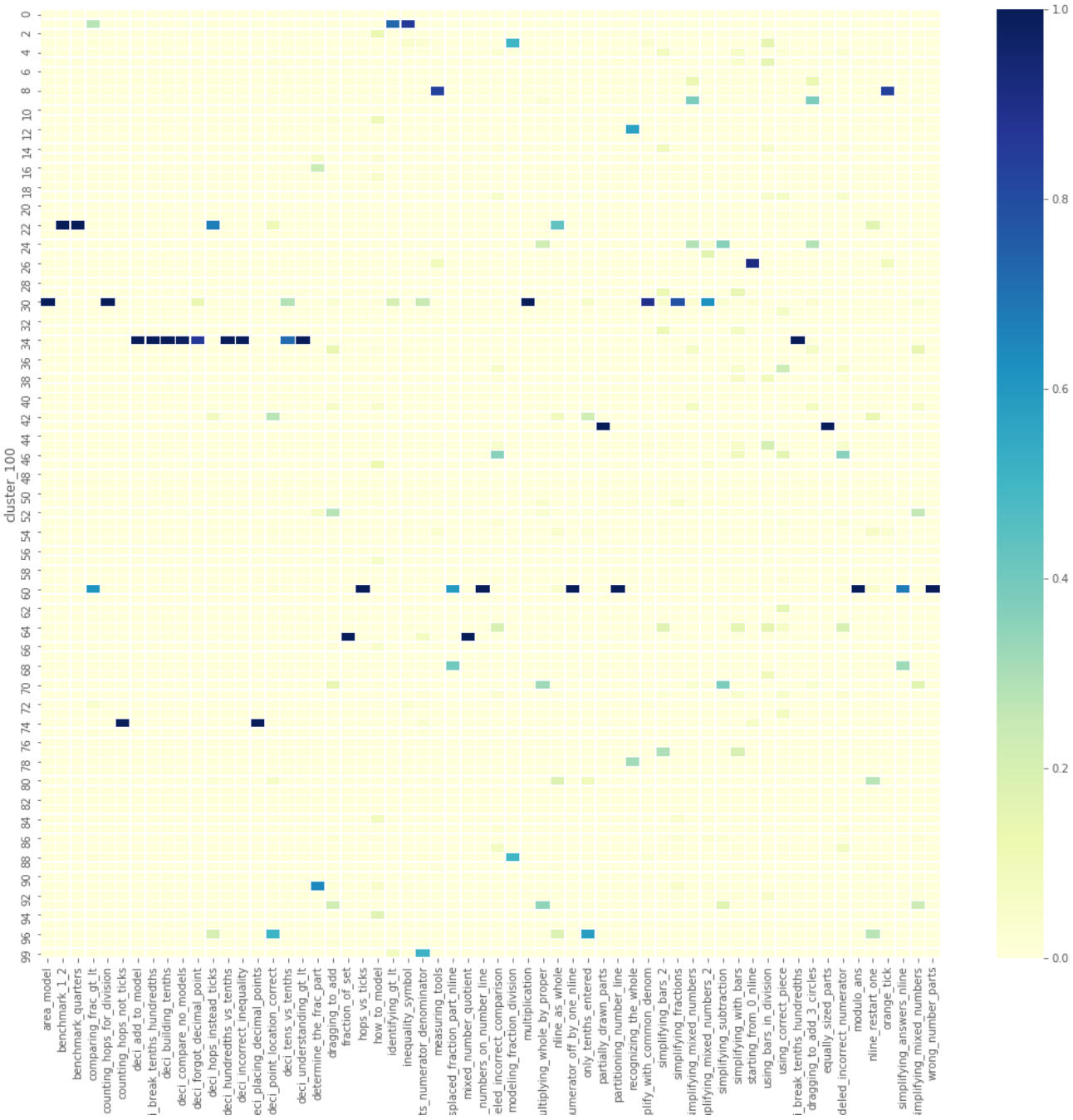


Figure 2: Heat map displaying the relative fraction of a trait per cluster

Woot Math was very interested in using the results to explore the data in the clusters with few or no traits mapped. They asked us to provide a list of samples associated with these clusters so they could look for new types of learning issues.

6.1.1 Drawbacks to this approach

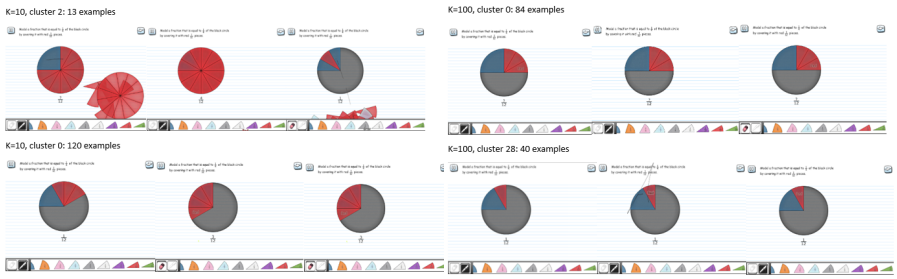
This clustering approach took all documents in the subset and clustered with k = 100. The high cluster count might force clustering along less pertinent features, reducing the importance of any one individual cluster. However this is not certain, and further evaluation on the distinguishing features and analysis on the documents themselves might reveal a deeper insight into the distribution of traits and documents across clusters.

6.2 Approach 2

The following visualizations were made for [qual_id: 'xSDXuo9OEH.bonus.OG_XxtbnEa'], using 495 incorrect problem attempts. Our script extracted 78 features.

As the feature extraction mainly captured modeling states, the clusters are particularly well-suited to identifying different modeling behaviors. Some new traits that we believe we have found include the following:

1. **Incomplete modeling**. The student started modeling the problem, but did not finish.
2. **Modeled correctly, but still missed the problem**. This is perhaps an error in the question representation.
3. **Unusual modeling**: Often outliers. These could indicate confusion about how to model, or perhaps boredom.
4. **Common modeling mistakes**: Majority of students misunderstanding how to complete a problem.



Low K: High heterogeneity *High K: High homgeneity*

Figure 3: Three example images pulled from a cluster for different K values.

6.2.1 Drawbacks to this approach

Due to our feature extraction method, too much importance is placed on model piece usage. Clusters are based mainly on the existence of a feature, rather than that feature's value. For example, adding a new model piece will have more of an impact on the feature vector and its subsequent clustering than a different answer response or an additional already-existing model piece will have. Because of this, clustering is either very homogeneous for large K, or noisy for small K.

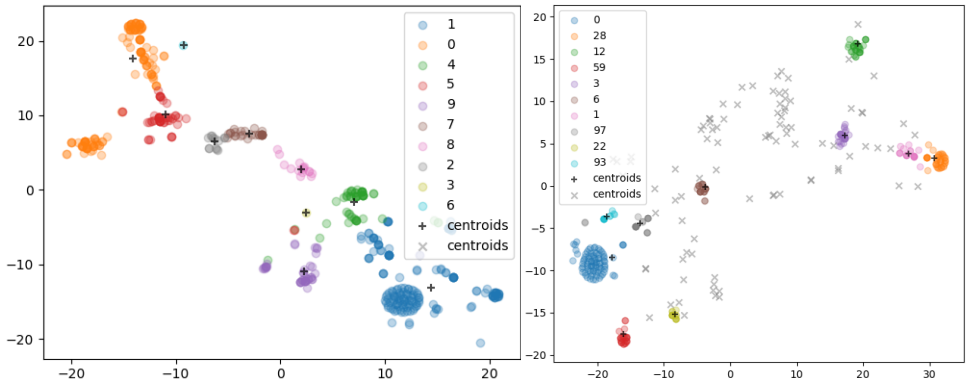
Another issue is the choice of K. The choice of K is sensitive, and depends a good deal on the specific problem. If the problem is simple or doesn't require modeling then the canvas states are limited. Fewer types of incorrect answers means fewer clusters can be found. On the other hand, some of the more complicated problems (e.g. those that require modeling) may have hundreds of unique incorrect answers. In this case, numeric clustering may be more beneficial, helping to identify similar but not-identical incorrect answers. This requires careful hand-tuning. Some clustering methods such as DBSCAN can automatically identify clusters, but these, too, require parameter tuning.

7 Cluster Evaluation

We need a method to determine the quality of the clusters we found. But determining how meaningful the clusters are is difficult without ground truths for comparison. To this end, we developed a script that automatically performs some

automatic exploratory data analysis (EDA) for the most populous clusters and downloads some screen shots of student canvases for each cluster. This lets us quickly examine the clusters, and use our intuition to infer why the students might be making these common mistakes.

We can also use visualization to see the clusters in relation to one another. These visualizations were produced by using PCA to reduce the dimensionality from ~75 features in the input vector to 40. Then T-SNE is used to reduce the dimensions to two, while attempting to preserve their spacial relationships as much as possible. You can see how similar the points in the clusters are to one another for different values of K in k-means.



Low K: K-means with K=10 *High K: K-means with K=100*

Figure 4: T-SNE visualization of clustering from 500 attempts on a specific fraction modeling problem

8 Future Work

The dataset was not designed for machine learning. As a result, we had to determine how to extract features to cluster on. But the clusters can only find data which is present in the dataset, it cannot infer any features. Therefore improved clustering is likely possible if data is collected with machine learning in mind. We have identified some possible features that may be useful in identifying student learning issues:

1. **Time series data/intermediate steps** in answering the question: student actions while answering a problem, rather than simply the final canvas state when the answer is submitted
2. **A canonical answer**: we often had to infer the answer from the question.
3. **A standardized feature set**: clustering across problems is difficult, because internal data representation my be completely different even though the concept is similar. For example, when adding two fractions, the data may look completely different in the dataset even though the concepts are very closely related. This makes clustering difficult.
4. **Hand-engineered features** to identify specific behavioral traits. For example:
   (a) **off-by-one errors** are common, but hard to identify without a specific feature highlighting them
   (b) **distance-from-answer**. If the answer is 5/6, then 4/6 is probably closer than 1/6.
5. **Attention**: we wonder if it is possible to identify when students have lost focus. Perhaps the final canvas states could show something that is so far from the correct answer that the student is clearly not engaged (outliers from clustering).

References

[1] R. Brent Milne, Sean A. Kelly, David C. Webb, *Effect of Adaptivity on Learning Outcomes in an Online Intervention for Rational Number Tutoring, "Woot Math," for Grades 3-6: A Multi-Site Randomized Controlled Trial.*, *wootmath.com/research* 2014.