Identifying Written Digits with K-Means

David Armstrong, *Team Member*, Ryan Croxford, *Team Member*

***Abstract*—The goal of this study was to create a process to identify the numerical value of handwritten digits, as well as identify outlier images within the set of images. The design we implemented creates 75 pictures of example digits (centroids) that a test set of images are cross-referenced with. Whichever example digit (centroid) is "closest" to the test image is assigned to that centroid.**

I. Introduction

In order to create our example digits, we used the k-means algorithm. Through this process we initialize centroids, assign each vector to the closest centroid, then update the centroid to equal the average of the vectors in that cluster. This process is then repeated for a predetermined number of iterations. We are given a training set of 1500 images of digits to develop our final centroids.

II. Methods

A. Initializing Centroids

We chose to use ten centroids in our initial set to account for each of the ten digits. To choose a fitting initial centroid we averaged each of the vectors that represented a single digit, repeating this ten times to include each digit. For example, each corresponding value of every vector representing a 3 in the training set is averaged to create a mix of every 3 vector.

B. Finalizing Centroids

In order to assign each vector to the closest centroid, we find the norm between a given vector and every centroid. The vector is then assigned to the centroid that has the smallest norm. In order to update the centroids, we average all vectors assigned to a centroid and designate that value as the updated centroid. When this process is repeated, some vectors may change which centroid they are assigned to.

After we finalized our ten centroids we took each cluster and divided it based on what the picture's true values were. For example, if there were 40 zeros that made their way into the nine cluster, we removed them and made an additional cluster out of that group of zeros. This process continued until we created our final product of 75 centroids.

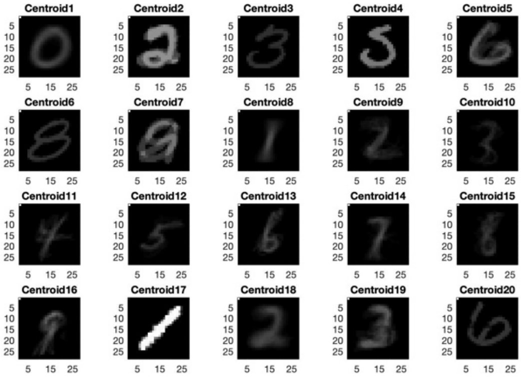

Fig. 1. Our first 20 out of 75 centroids formed with the k-means algorithm.

III. Results

A. Predictions

Using the final centroids, we predicted 156 of 200 test images correctly.

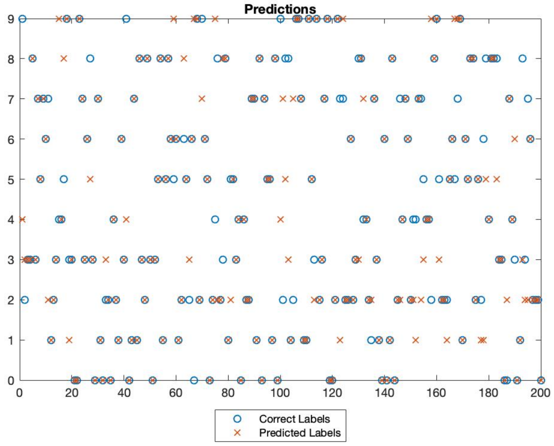

Fig. 1. Visual representation of our model's 78% accuracy.

The figure displays a fairly high accuracy for our k-means function; however, it is not perfect.

B. Cost

The cost of each iteration in the k-means algorithm can be calculated with (1) below. This is the sum of the distance between a vector and its centroid, squared.

$$\text{cost}(T) = \sum_{z \in T} \sum_{x \in C_z} \|x - z\|^2. \quad (1)$$

In this study, T is the number of clusters and x is the vector form of a training image that belongs to the particular cluster, C. For our study, the cost reduced between iterations in a negatively exponential manner.

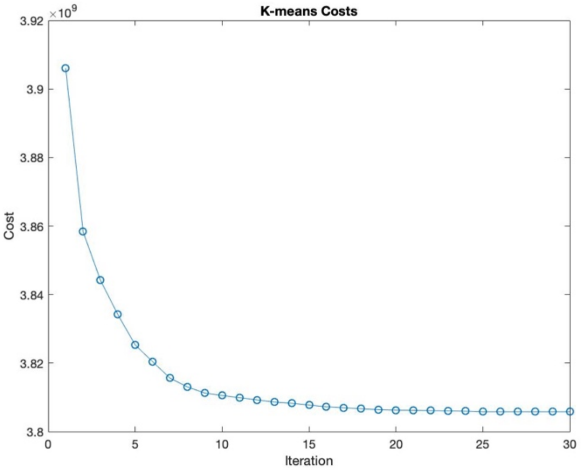

Fig. 2. The cost of the algorithm remains the same after the data stop switching centroids and the centroids stop changing values.

C. Outliers

When examining the test set vectors, we found that some of them had much higher pixel values across the board. The majority of vectors always had a zero in the first pixel position while the outliers had a 200 instead. Given this information, our process for flagging outliers was to flag the vector if the first pixel did not have a value greater than zero.

When assigning the test data to centroids, in order to counteract these outliers we subtracted each outlying vector by a vector of 200's. This scaled the outliers to be the same as the rest of the vectors. From there we were able to more accurately classify them into clusters.

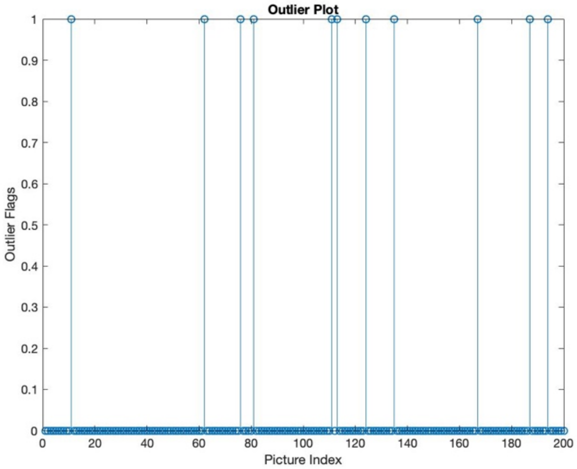

Fig. 3. This displays the 11 outliers that are hidden in the 200 test images.

IV. Conclusions

We conclude that our process for assigning vectors is successful for most cases but cannot perfectly predict a handwritten digit's intended value. We were unable to create clusters through k-means that could reliably sort the digits; however, for a practical use our function would be decently accurate.

V. References

https://cseweb.ucsd.edu/~dasgupta/291-geom/kmeans.pdf