Clustering Handwritten Digits and Detecting Outliers with the k-means Algorithm

Noah Waldman
Washington University in St. Louis
Introduction to Electrical Engineering – ESE 105
St. Louis, U. S.
n.g.waldman@wustl.edu

Ben Ko
Washington University in St. Louis
Introduction to Electrical Engineering – ESE 105
St. Louis, U. S.
ben.k@wustl.edu

***Abstract*— The k-means algorithm clusters data vectors based upon the user's choice of k initial centroids which act as representative vectors for partitioning the data vectors. Next, the algorithm finds the distance between a data vector and each centroid and assigns the data vector to the centroid it is closest to. After that, each centroid is updated by taking the average of all the data vectors in its cluster. This process is iterated until the cost (or the sum of all distances between centroids and their assigned data vectors) converges. The algorithm was fed 1000 greyscale images of handwritten digits from the MNIST data set. At this point, there are k centroids. From there, data vectors from an external test set were assigned to each centroid to predict their numerical meaning. It was found that the choice of initial centroids was accurate in predicting the image's meaning 75% of the time. Next, outliers were determined by finding the data vectors with the largest distance from their assigned centroids.**

I. Introduction

The k-means clustering algorithm extracts valuable information from data sets by clustering data into related groups. The algorithm assigns meaning to the data. Some examples of where the k-means clustering algorithm may be used include grouping residents of certain ZIP codes by a range of factors (such as age, socioeconomic background, and household size) to determine what type of people live where, creating search engines that recommend users webpages that they are likely to be interested in based on past searches, and clustering weather zones based upon patterns of weather in differing geographic locations [1]. This case study used the k-means algorithm to classify greyscale images of handwritten digits from the MNIST data set. The accuracy of the clusters created was tested by assigning previously unseen images to the clusters and finding the accuracy of the predictions made. From there, outliers were detected by finding the data vectors with the largest distance from their respectively assigned centroids. This case study provides a greater understanding of the k-means algorithm, the linear algebra behind it, and the notion of outliers.

II. Methods

The first task after completing the base k-means algorithm in MATLAB was to choose k initial centroids. The first choice of k decided upon was 10 because there are 10 digits. Having 10 centroids worked well; each centroid was well defined and they predicted the numerical meaning of data vectors in the test set with a 60% accuracy. With that baseline, more centroids were added one by one or in pairs to clarify underdefined centroids (e.g., many 9s had a similar slant to 7s, so the predictions for those digits were inaccurate). However, adding additional centroids for those digits counterintuitively resulted in the new post-algorithm centroids becoming a mix of the two digits resulting in even lower prediction accuracy. The next attempt to refine the k initial centroids was to add additional copies of well-defined centroids (e.g., 0s since they have a unique shape compared to other digits). This resulted in a much higher prediction accuracy. After tinkering with the choice of initial centroids for well-defined digits, it was found that choosing a second centroid with deviations from the first centroid led to the best predictions. E.g., in Fig.1, the 1 in centroid 3 is straight and the 1 in centroid 4 is slanted. With that in mind, the initial 16 centroids depicted in Fig. 1 were chosen.

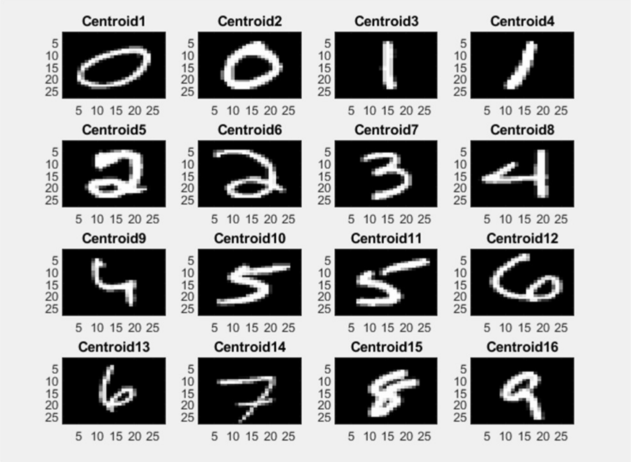

Fig. 1. The initial centroids selected for the clusters.

Prior to this point, the algorithm was run with an arbitrarily high number of iterations which caused it to take a long time. By looking at the cost of the k-means algorithm (the sum of the distances between each data vector and its assigned centroid) it was found that the cost of the algorithm converges (the change in cost was minimized and the centroids no longer changed) after 10 iterations as shown in Fig. 2, so 10 was chosen to be the number of iterations.

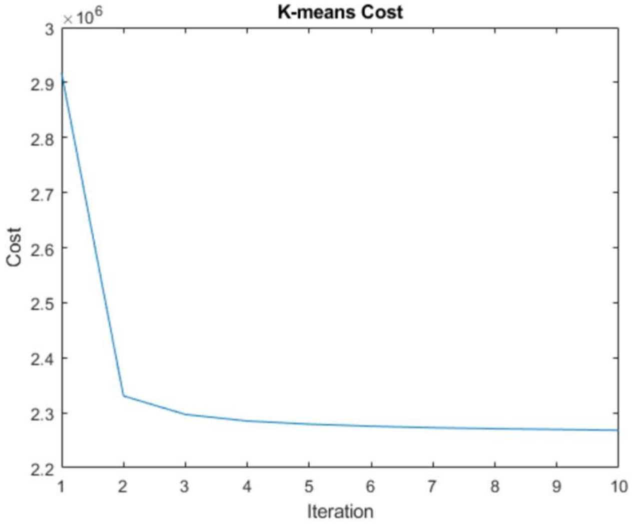

Fig. 2. The k-means cost after each iteration of algorithm.

After iterating the k-means algorithm 10 times with the initial centroids shown in Fig. 1, the final centroids were the centroids shown in Fig. 3.

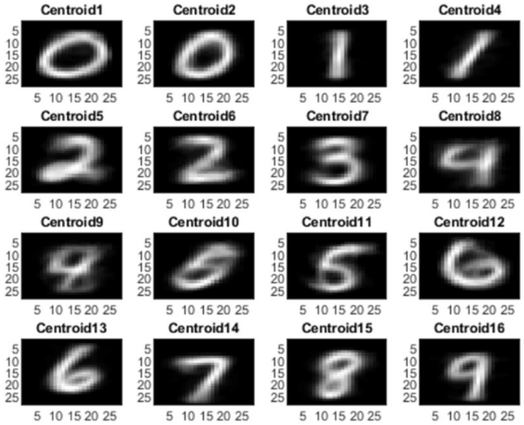

Fig. 3. The final centroids after created from the initial centroids after 10 iterations of the k-means algorithm.

To detect outliers, first, the distance between each data vector and its respective centroid was found. Then, the 10 vectors with the largest distance from their centroids were chosen as outliers. 95.4% of samples are within 2 standard deviations of the mean for a normal distribution, which implies that 4.6% of samples are outside of 2 standard deviations [2]. Although outliers are typically chosen to exist outside of 3 standard deviations, it was decided to choose outliers to exist outside of 2 standard deviations due to small sample size (if outliers were chosen to exist outside of 3 standard deviations there would be 0 outliers since 0.3% of 200 is less than 1). Since 4.6% of 200 is 10 (after taking the ceiling), it was decided there should be 10 outliers.

III. Results and Discussion

After assigning the test data vectors to the centroids shown in Fig. 3, the digit predictions marked by the orange xs in Fig. 4 were made (the blue circles indicate the actual digits). Of the 200 predictions made, 150 predictions matched the labels given, resulting in a 75% prediction accuracy. Predictions for 0, 1, and 2 were particularly accurate, but predictions for 5, 8, and 9 were inaccurate. The accurate predictions occurred due to those digits having unique shapes. The inaccurate predictions for 5 come from the 5 in centroid 10 having a faint outline of an 8 and the 5 in centroid 11 having a faint outline of a 6 as shown in Fig. 3. The lack of total predictions for 8 come from that same 5 in centroid 11 as shown in Fig. 3 resembling an 8, which caused many 8s to be assigned to centroid 11. Additionally, the 4 in centroid 9 of Fig. 3 resembles an 8, so many 8s were misassigned there. There appears to be a lack of predictions for 9 because the 4 in centroid 8 of Fig. 3 resembles a 9, so some 9s were misassigned to centroid 8. The other errors can be accounted for by outliers.

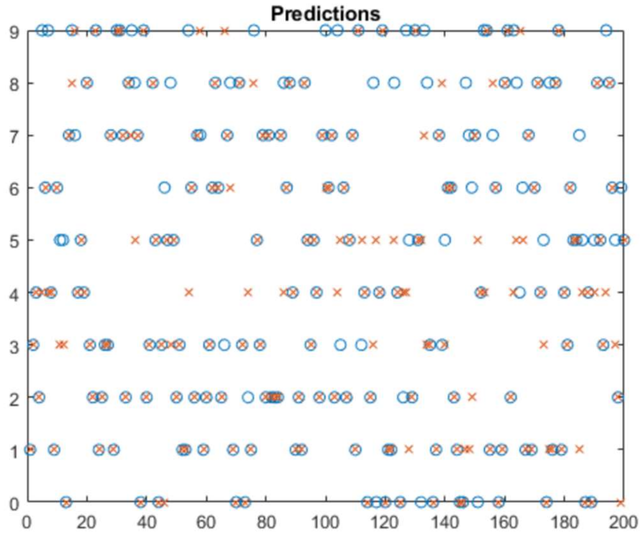

Fig. 4. The digit predictions made by assignment to closest centroid compared to the actual digits.

Fig. 5 displays the index values of the 10 outliers found. Of the outliers found, many of them appear to have blurred sections such as the 0 in Fig. 6.

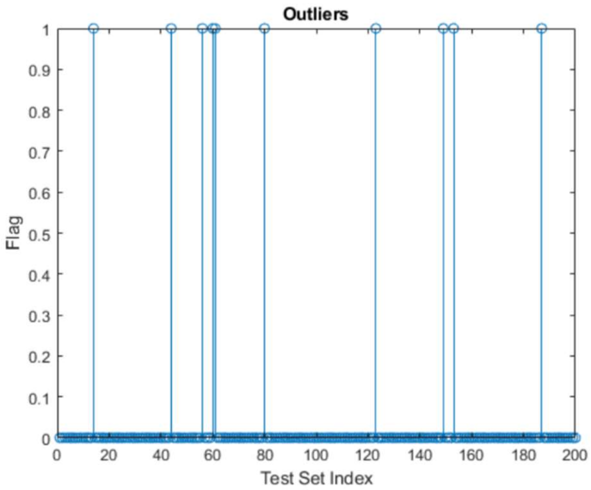

Fig. 5. The outliers predicted.

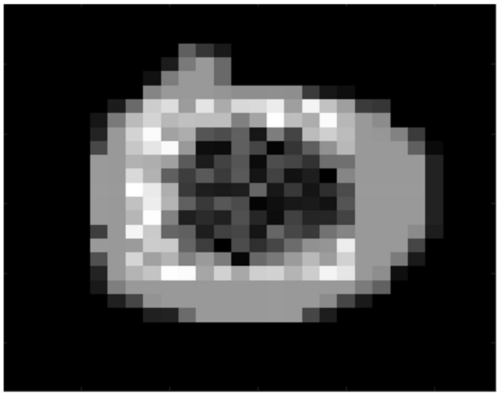

Fig. 6. The outlier that was farthest from its centroid.

A major limitation of the design used to detect outliers is that it requires the user to choose how many outliers they believe there should be, creating potential user error (e.g., a user can choose all 200 data vectors to be outliers). Additionally, the method assumes completely accurate assignment of data vectors to centroids (e.g., if a 0 were assigned to a centroid containing mostly 3s, it may be an outlier because it may be farther from the centroid than any of the 3s even if it was a normal 0). It was assumed that outliers would always be farther from the centroid than non-outliers.

IV. Conclusion

Although the k-means algorithm implemented in this case study was by no means perfect, it demonstrated the power of the algorithm to assign meaning to large data sets with high accuracy. The k-means algorithm successfully predicted the digits of 75% of the test data vectors after tuning the initial choice of centroids by adding additional centroids for well-defined digits. Outliers were detected by choosing the 10 data vectors with the largest distance from their respective centroids. Shortcomings not previously mentioned include using a small data set and the inability for people to concur on what a given digit image represents. Future related projects may consist of more complex image classification such as partitioning images of cats and dogs.

References

[1] S. Boyd and L. Vandenberghe, *Introduction to applied linear algebra: Vectors, matrices, and least squares*. Cambridge: Cambridge University Press, 2019.

[2] R. Sedgewick and K. Wayne, “Gaussian distribution,” *Princeton University,* 16-Apr-2010. [Online]. Available: https://introcs.cs.princeton.edu/java/11gaussian/. [Accessed: 15-Oct-2021].