K Means Clustering Analysis of MNIST Digit Dataset

Tomoki Koike (koike@purdue.edu)

Fall 2019

Every Boilermaker Engineering Code – Intermediate Level Programming

Contents

[Objective 2](#_Toc27247567)

[Theory & Procedure 2](#_Toc27247568)

[Results 4](#_Toc27247569)

[Appendix 9](#_Toc27247570)

[I. Figures 9](#_Toc27247571)

[II. References 13](#_Toc27247572)

# Objective

The main objectives of this project were the following:

1. To acquire knowledge of K-means clustering, which is a fundamental method of data analysis and a method that can be applied to machine learning.
2. Apply the knowledge obtained throughout the course of python programming and develop an original algorithm of K-means clustering for an introduction of real data analysis.
3. To experimentally test your algorithm with the provided MNIST digit dataset and analyze the retrieved results. Then implement this to further perfect your algorithm.

# Theory & Procedure

The primary focus of this project was K-means clustering. Clustering is a type of machine learning approach called unsupervised learning. This method manages problems where another class is unnecessary to learn from and is able to find groups that share similitude within the certain group. Clustering is the task of dividing the data depending on the resemblance in the sample; that is, grouping the similar points together and separating the dissimilar points into other groups. K-means clustering is one of those many clustering methods where the data is looped through n observations and for each time the observations are partitioned into k clusters based on points’ distance from the nearest mean. The distance between the points and the mean value is calculated using the squared Euclidean distance:

Equation 1: Square Euclidean Distance Formula

As aforementioned, the clusters are then created contingent to the distance from the mean/centroid value for the k-th partition. The number of observations depend on whether the clustered data for the n-th and (n+1)-th trial become identical or until the iterations reaches the limit value which is set beforehand. The mean/centroid for the first observation will either be preset or will be randomly generated depending on the user inputs of the program; and for each proceeding observations, the mean/centroid value will be upgraded by calculating the average distances from one point to all the other points within the new cluster (Liu, 2019).

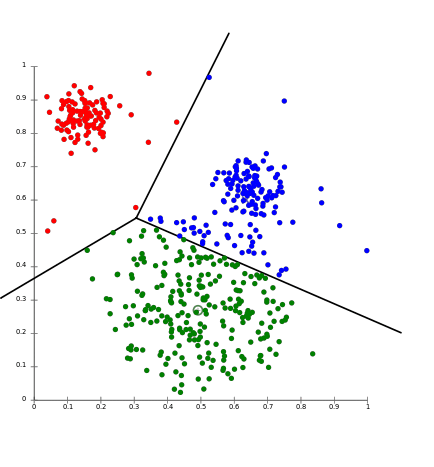


Figure 1: Example of K-means Clustering Plot (from Amazon AWS)

Once, the repetitive clustering has terminated and all clustering outputs are properly obtained, the silhouette coefficient is then computed to determine the best k-value for the given dataset. The optimal k-value is determined by the measure of how much the computed silhouette coefficient is close to the value of one. The silhouette coefficient is a value with a range of [-1, 1]; and, it is ideal to have a value of 1 for the k-means clustering. In other words, if the value is close to -1, this indicates that the k value selected for the clustering is far from optimal.

In the following paragraph the breakdown of the K-means clustering algorithm will be explained. Primarily the K-means clustering algorithm will be a class function. Within this class function there will be three essential functions:

1. readData

This function will read the MNIST digit dataset and convert it to a pandas dataframe. This function will accept the raw data file as an argument.

1. cluster

The second function will handle the clustering of the data. There will be four arguments. The first one is the dataframe parsed from the previous function “readData.” The second argument, “iterCount” will accept the number of iterations, which is the number of n observations to conduct the clustering. The third argument will be the user input for the k-value for the clustering. The last input is the list of ID numbers (one of the columns in the MNIST digit dataset) for the points selected to be initial centroid of the first observation to start with. This argument can be an empty <list> where in such case the initial centroid will be randomly generated by the program. This function will return a <list<list>> of the ID numbers for all clusters.

1. calculateSC

This function will compute the silhouette coefficient from the output of the “cluster” function. It will only take the output from the “cluster” function as an argument and will return the silhouette coefficient as the output.

# Results

There were 2 parts to the analysis.

1. Part 1: Using the subset of the digits [2, 4, 6, 7] within the MNIST digit dataset
2. Part 2: Using the subset of the digits [6, 7] within the MNIST digit dataset

The first task for the analysis was to create a visual image from the given dataset. The following scatter plots illustrate the visual image for the digits 2, 4, 6, and 7 respectively.

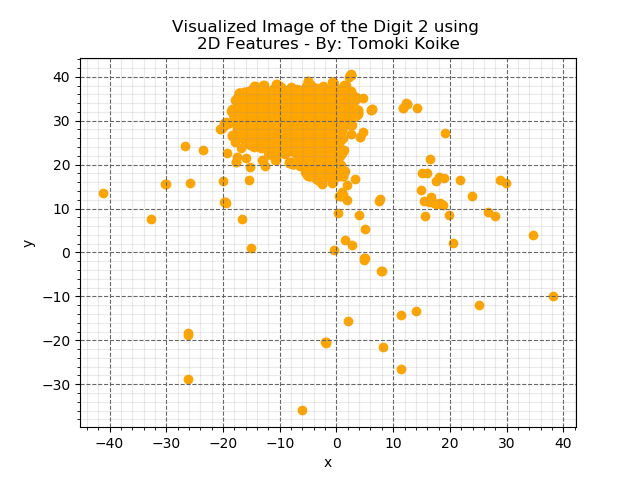


Figure 2: Digit 2 Visual Image Scatter Plot

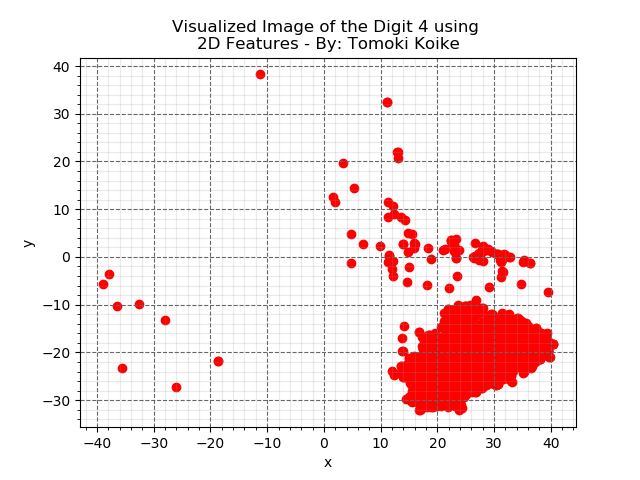


Figure 3: Digit 4 Visual Image Scatter Plot

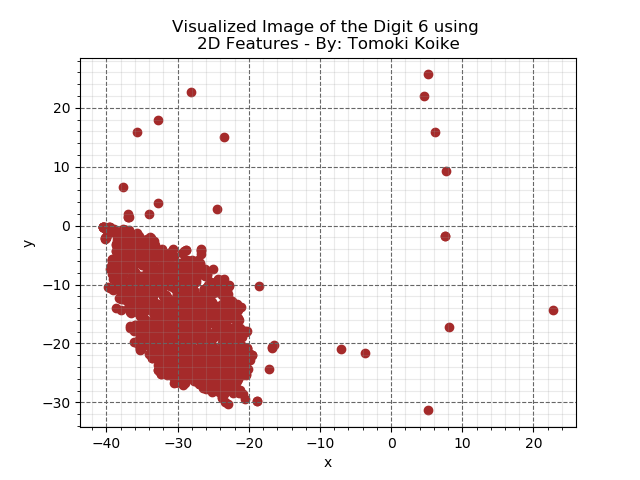


Figure 4: Digit 6 Visual Image Scatter Plot

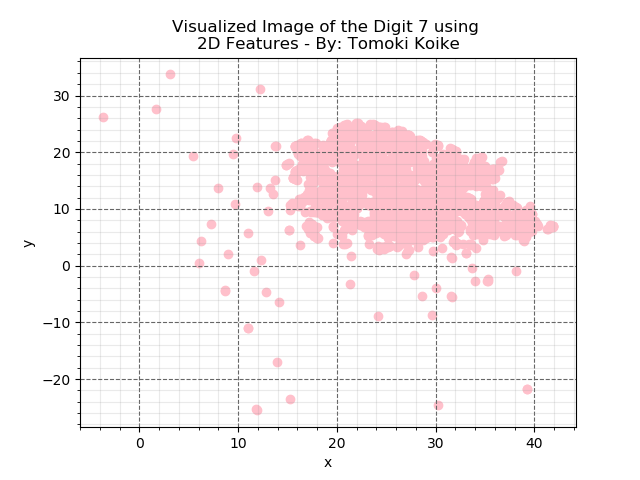


Figure 5: Digit 7 Visual Image Scatter Plot

From the scatter plots, we are able to tell that each digit has presumably 2 or 4 clusters. The digits 4 and 6 clearly have 2 clusters; however, 2 and 7 rather scatter more randomly that it looks like it has 2 or perhaps 4 clusters.

Next, we have clustered that data for k-values 2, 4, 8, and 16, and for each k-value the clustering was repeated 5 different times. From this procedure, 5 different silhouette coefficients were obtained, and the average silhouette coefficient was calculated for those 5 experiments. To note, this was then repeated for part 1 and 2. The results for part 1 and 2 appears as the following.

Part 1:

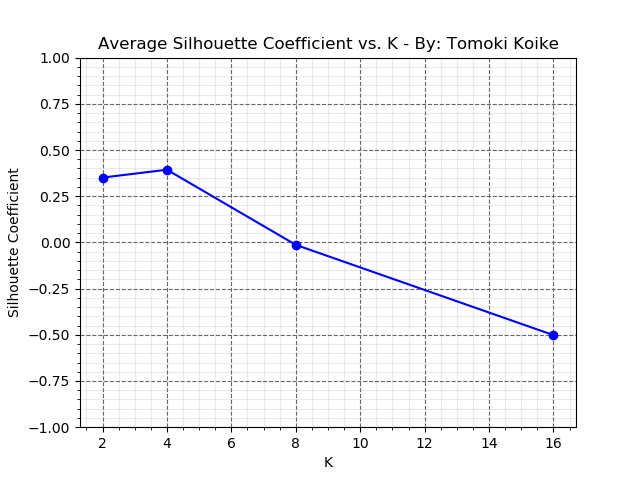


Figure : Part 1 Results for K-values vs Average Silhouette Coefficients

Part 2:

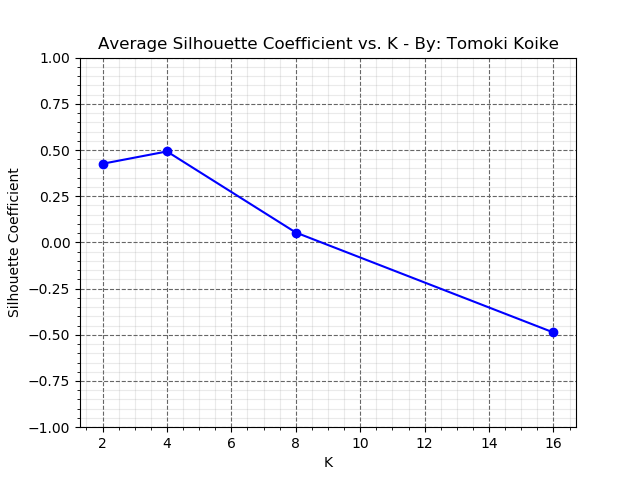


Figure : Part 2 Results for K-values vs Average Silhouette Coefficients

The plots show that the most optimal k-values for this analysis is 4 because at k = 4 the silhouette coefficient peaks for both part 1 and part 2. This agrees with what we have said before by examining the visual image for each digit. Thus, we will conclude this analysis successful by obtaining the optimal k-value of 4 for the given MNIST digit dataset.

# Appendix

## Figures

A screenshot of a video game

Description automatically generated

Figure : Digit 0 Visual Image Scatter Plot

A close up of a map

Description automatically generated

Figure : Digit 1 Visual Image Scatter Plot

A screenshot of a cell phone

Description automatically generated

Figure : Digit 3 Visual Image Scatter Plot

A screenshot of a cell phone

Description automatically generated

Figure : Digit 5 Visual Image Scatter Plot

A screenshot of a cell phone

Description automatically generated

Figure : Digit 8 Visual Image Scatter Plot

A close up of a map

Description automatically generated

Figure : Digit 9 Visual Image Scatter Plot

## References

Liu, Xiaojin. (2019). *Every Boildermaker Engineering Codes Intermediate Level Python Programming Fall   
2019: Lecture 11 – K-means Clustering* [PowerPoint Slides]. Retrieved from: https://mycourses‌.purdue.edu/webapps/blackboard/content/listContent.jsp?course\_id=\_449400\_1&content\_id=\_13942410\_1