|  |  |
| --- | --- |
|  | IMA7221 |

Report

**Topic: Unsupervised Learning**

Made by: Botirov Khojiakbar

**Introduction**

This work is based on applying K-means and Hierarchical Clustering using database of handwritten digits. Database presents train and test datasets, which are consisted of 3823 and 1797 samples respectively.

Train dataset represents matrix with 3823 x 65, where each line is digit of 8 x 8 pixels and last columns is label of the digit, same concept for test dataset.

Practical work was done in python using colab by google , for each clustering technique two steps were done: Training and Testing.

**K-means Clustering**

**Training**

K-means algorithm separates dataset into N groups of equal variance, where N is described by the mean of samples also known as cluster. K-means minimizes inertia, which can be recognized as a measure how internally coherent clusters are.

Algorithm starts from choosing random initialization of K cluster center from dataset, then assign each sample to the closest cluster center and this process is repeating until attainment of stability (convergence).

Global squared error “J” must be minimum, which is the sum of squared distances between sample and cluster center or mean.

**Dataset Loading and K-means implementation**

Train dataset was loaded from optdigits.tra file, then the dataset was divided into two ndarrays: pixels and labels. Using KMeans function from scikit-learn framework train dataset was clustered for the range of iterations 1 – 19, Figure 1 shows Global squared error for each iteration, min(J) is on 19th.

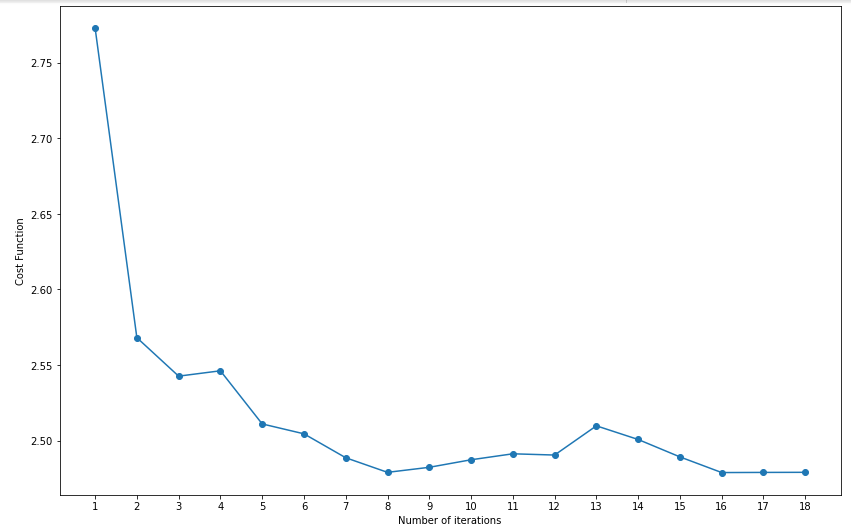


Figure 1.

Figure 2 and 3 represent histograms for real labels of train dataset and labels obtained after clustering. One can be concluded, that most frequent confusion is in digit 9, which was assigned to the wrong clusters.

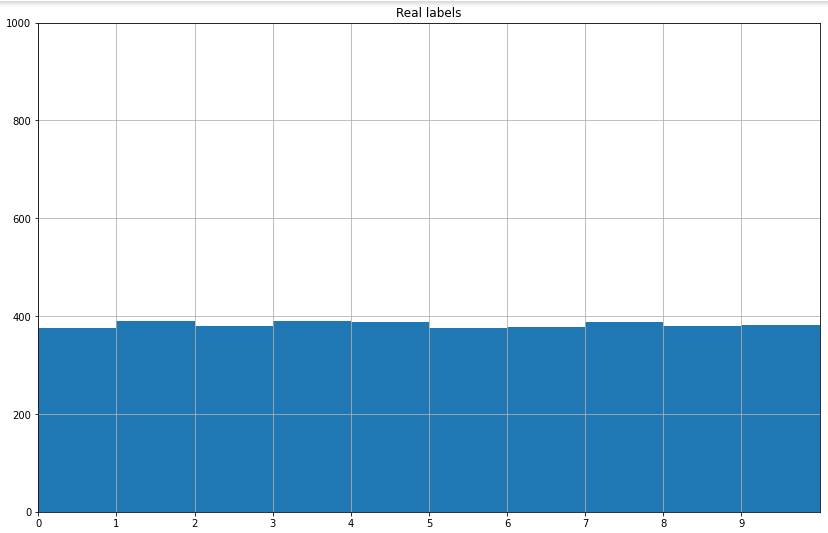


Figure 2

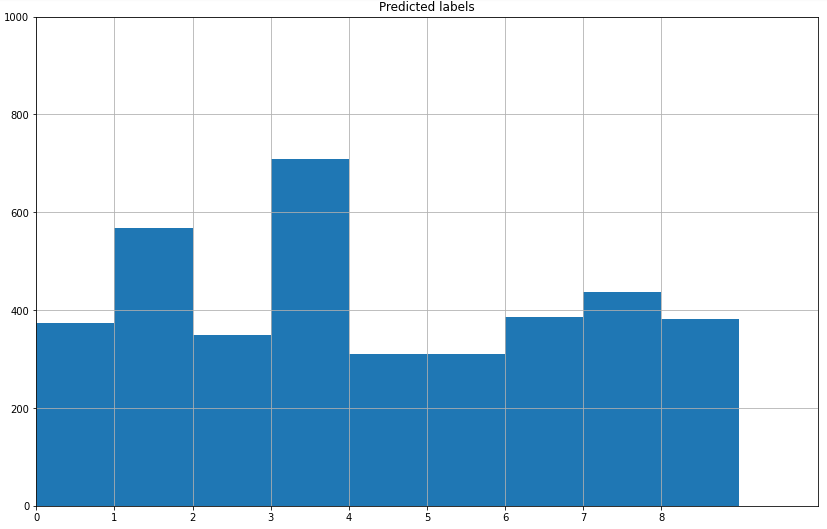


Figure 3

**Silhouette**

Silhouette analysis can be used to study the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1].

If silhouette coefficient close to 0, it refers that two clusters are very close to each other at boundary decision level and if coefficient is negative, it means that the samples were assigned to the wrong cluster.

This measure helps to determine proper number of clusters.

Figure 4 shows output of silhouette values for the range of number of clusters between 10 and 15. The highest value is for n\_clusters = 10.

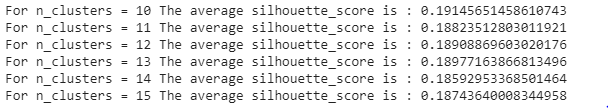


Figure 4

**Permutation**

Majority voting to give a label to each cluster is done in permutation(), where scipy.stats.mode() was used from scipy library.

The accuracy of K-means for train data is 80%.

**Testing**

Kmeans.predict() method was used to find the closest cluster and assigning digit to the closest cluster.

In confusion matrix each row of matrix the represents the instances in a predicted class while each column represents the instances in an actual class.

Confusion matrix is given by Figure 5 and the accuracy of K-means for test data is **79%**.

Global performance or the number of correctly clustered samples for K-means are **1321.**

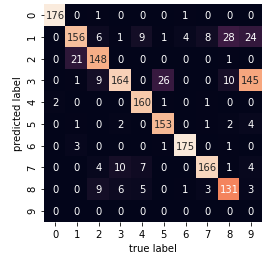
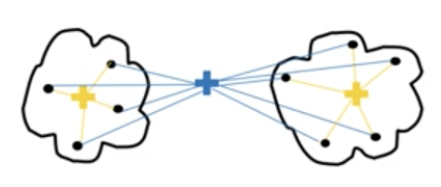


Figure 5

**Hierarchical clustering**

A simple algorithm for partitioning the dataset into groups called clusters. We used Agglomerative clustering – bottom up approach, starts with many small clusters and merge them together to create bigger clusters. At first point all data points a clusters, then merges two closest clusters until there is only one cluster. The distance for Ward Linkage is the sum of squared differences within all clusters.



Ward Linkage

Dendrograms are used to visualize the history of grouping and figure out the optimal number of clusters by putting threshold line, which is at the largest vertical distance that doesn’t intersect any of other clusters and the optimal number of clusters is equal to the number of vertical lines going thought the horizontal line. Figure 5 shows dendrogram for train dataset for number of clusters = 10.

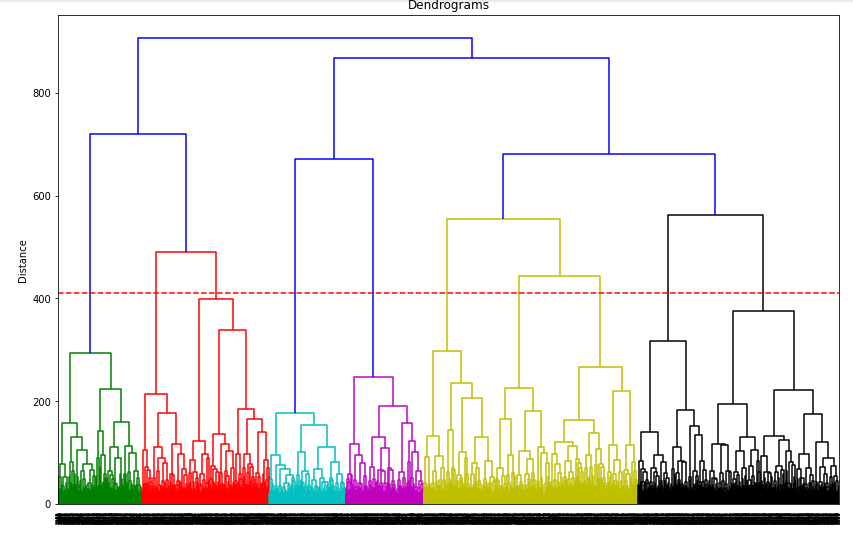


Figure 5

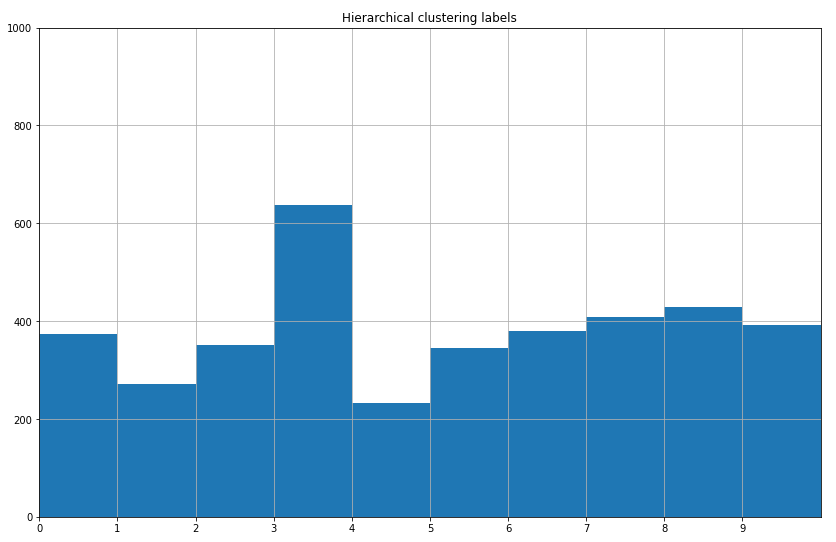


Figure 6. The histogram per cluster for Hierarchical clustering

Silhouette coefficient for Hierarchical clustering at number of clusters = 10 is 0.1745470931891432

The accuracy of Hierarchical clustering for test dataset is **86%**.

Global performance or the number of correctly clustered samples are **1549**.

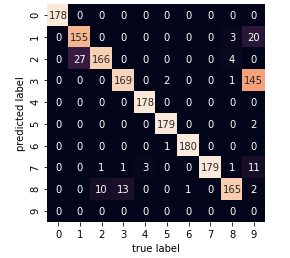


Figure 7. Confusion matrix of HC

In addition, silhouette coefficients for Hierarchical clustering are too close to each other, Figure 7 shows the output of silhouette values for HC at range of number of clusters between 10 and 15.

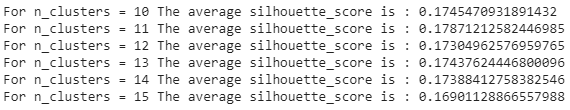


Figure 8

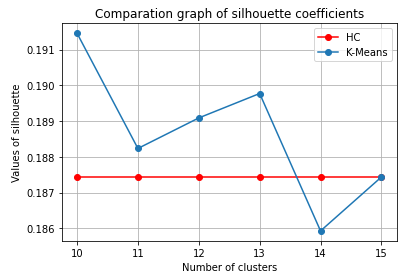


Figure 9. Comparison graph of silhouette values for both techniques.

**Conclude**

During this practical work, we better understood the working principle of two clustering techniques: K-means and Hierarchical Clustering. To realize this code, functions from the scikit-learn framework were widely used and the function permutation() was written to allocate the most frequent class in the each cluster. For given dataset one can be concluded, that digit 9 is the most confused and from confusion matrixes of both techniques it can be seen, that majority samples of predicted labels for digit “9” is allocated to digit “3”.

The accuracy of Hierarchical clustering is a little bit higher than for K-means, 86% against 79% for test dataset and Global performance better for HC than K-means. At the same time silhouette coefficient for K-means is higher than HC for n-clusters = 10, but for k-means the coefficient drops with increase of the number clusters.