ROB311 - Apprentissage pour la robotique

TP6 - KMeans : Digit Recognition

Mateus Lopes Ricci
Matheus Melo Monteverde

25 octobre 2020

1 Introduction

K-means clustering is an unsupervised learning algorithm that evaluates and groups data according to their characteristics. The algorithm aims at partitioning n observations into k clusters in which each observation belongs to the cluster with the closest mean (cluster centers or cluster centroid). To create the cluster and classify the instances, the algorithm makes a comparison between each value of each line by means of the distance. The euclidean distance is generally used to calculate how 'far' one instance is from another. The way to calculate this distance will depend on the number of attributes in the table provided. After calculating the distances the algorithm calculates centroids for each class. As the algorithm iterates, the value of each centroid is refined by the average of the values of each attribute of each instance that belongs to this centroid. An example of how a data set would be classified after using the K-Means algorithm can be seen in the figure below.

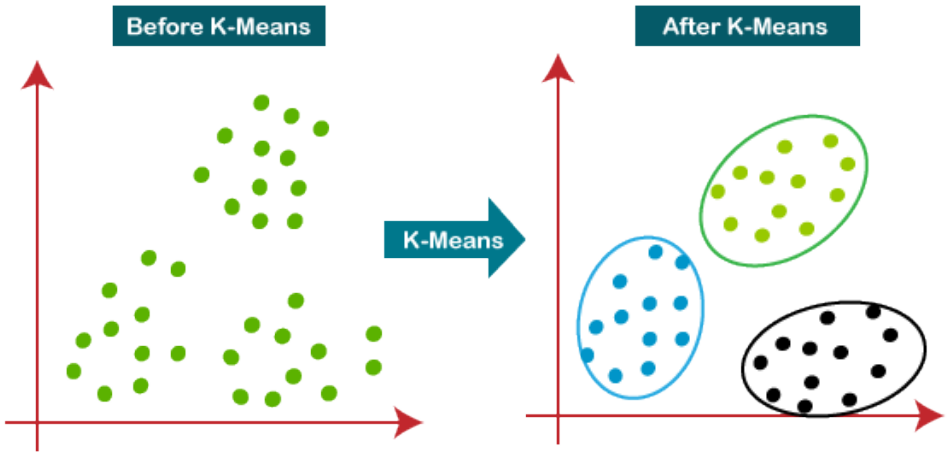
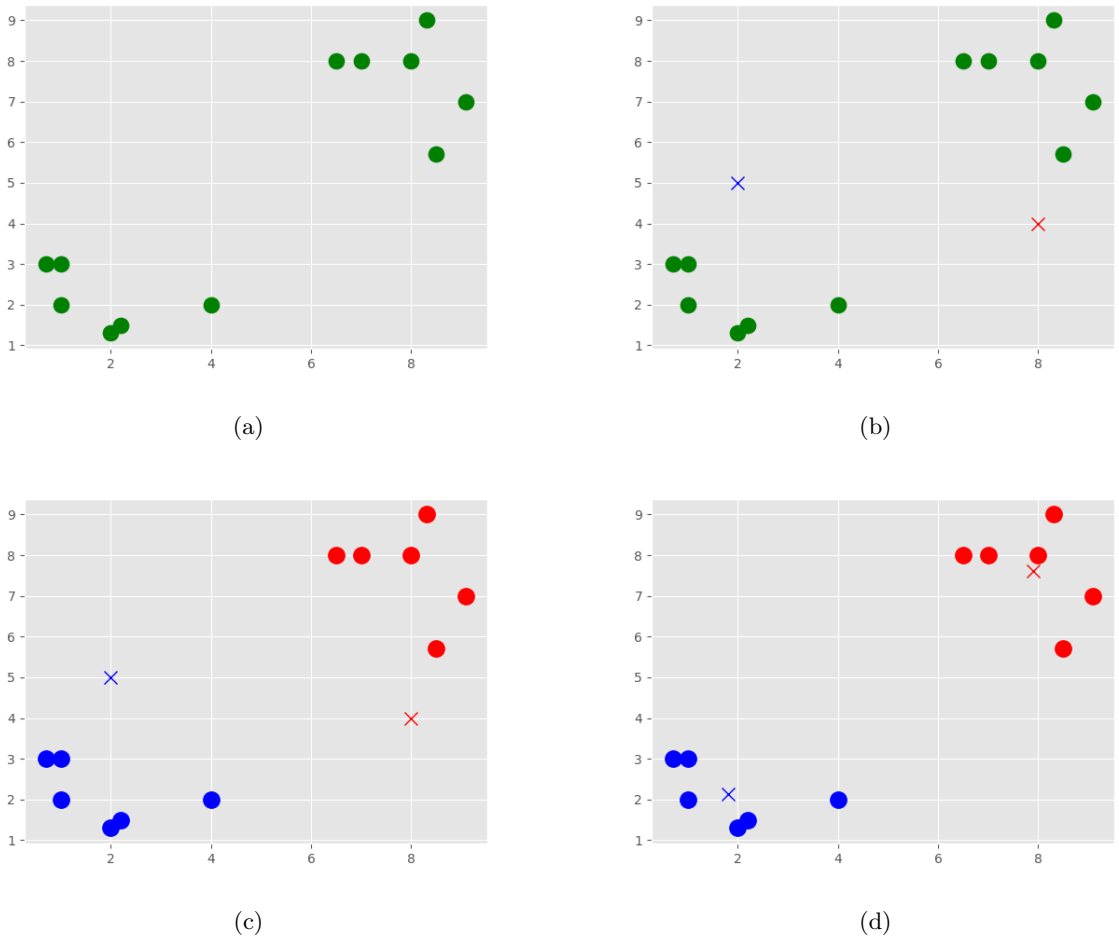


FIGURE 1 – K-Means algorithm

To simplify the understanding of the algorithm process a few steps can be followed.

1. **Provide values for the centroids :** In this step the k centroids should receive initial values. These values can be chosen randomly or based on the values of the first instances of the data set.
2. **Partition the instances in the classes :** The instances are classified according to their distance from the centroids of each class. The instance will belong to the class represented by the closest centroid. It is important to say that the algorithm ends if no instance changes classes, that is, if no instance is incorporated into a class other than the one it was before this step.
3. **Calculate the new centroids for each class :** In this step, the coordinate values of the centroids are recalculated. For each class that has more than one point, the new centroid value is calculated by averaging each attribute of all the points belonging to this class.
4. **Repeat until convergence :** the algorithm returns to step 2 iteratively repeating the refinement of the calculation of the centroid coordinates.

In the following pictures it is possible to see these steps of the K-Means algorithm for instances with two characteristics. In figure 2a the instances of the data set are scattered. Then, initial values are assigned to two centroids, representing two distinct classes, as shown in figure 2b. The centroids are represented by the x markers. In figure 2c it is possible to see that each of the instants is assigned to one of the two classes, through the calculation of the Euclidean distances. Finally, figure 2d illustrates the calculation of the new centroids position. After this step the algorithm continues its iterative process until the convergence.



(a) (b) (c) (d)

FIGURE 2 – K-Means algorithm steps

1.1 Data set

In this project it was desired to implement the K-Means in order to implement a digit recognition algorithm. The data set contains normalized bitmaps of handwritten digits from a preprinted form. Each instance of the data set is a 8x8 matrix resulting from the 32x32 bitmaps which were divided in 4x4 nonoverlapping blocks. The sum of each element in the 4x4 block represents its value on the 8x8 matrix. Therefore the value of each 8x8 matrix element varies from 0 to 16.

2 Principal Component Analysis

To facilitate the algorithm execution it was decided to attempt the Principal Component Analysis (PCA) together with the K-Means. It is the process of computing the principal components and using them to perform a change of basis on the data. In this project it was used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data variation as possible. This way it was possible to reduce the size of features without losing its main features.

3 Confusion Matrix

After creating the clusters with the training data set, utilizing the KMeans clustering, the testing instances were predicted. We represent the results obtained in a confusion matrix format, this way, it is possible to fin tendencies in the clusters created and compare to the label of the handwritten digits.

3.1 Using PCA

Unfortunately the results obtained with the use of the PCA were less accurate, despite varying the number of components. The corresponding confusion matrix with PCA is shown below.

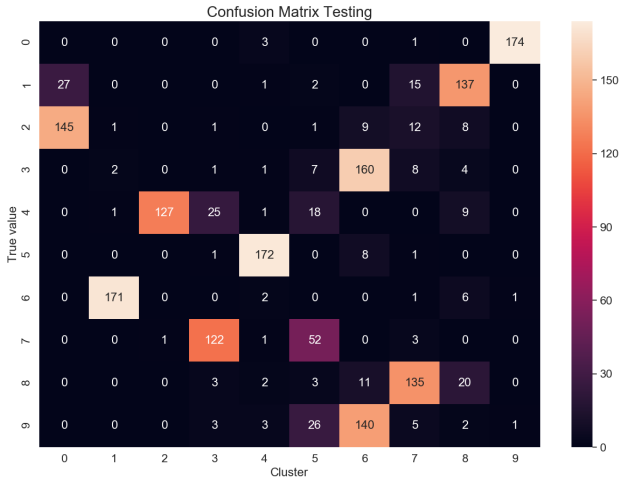


FIGURE 3 – Confusion Matrix for testing data set using PCA

3.2 Scaling the features

In order to attempt to increase the performance of the training we also tried scaling the input data.

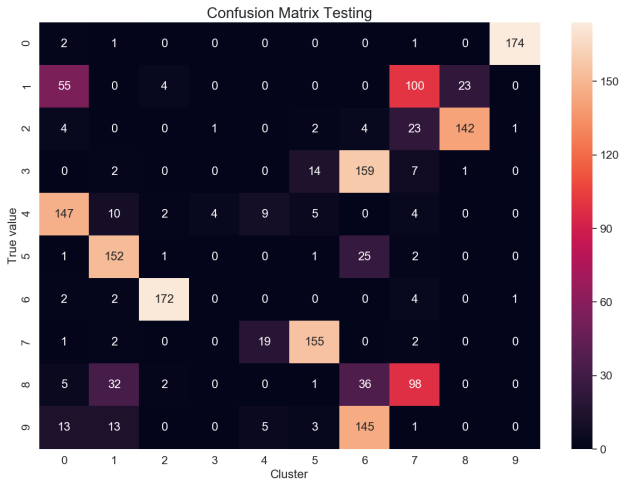


FIGURE 4 – Confusion Matrix for testing data set scaling the features

3.3 Best Results

After comparing the results of the clustering by different methods (such as combinations utilizing PCA, scaling the data and utilizing only the clustering), the best results that corresponds to utilizing only the clustering are shown in the confusion matrix below :

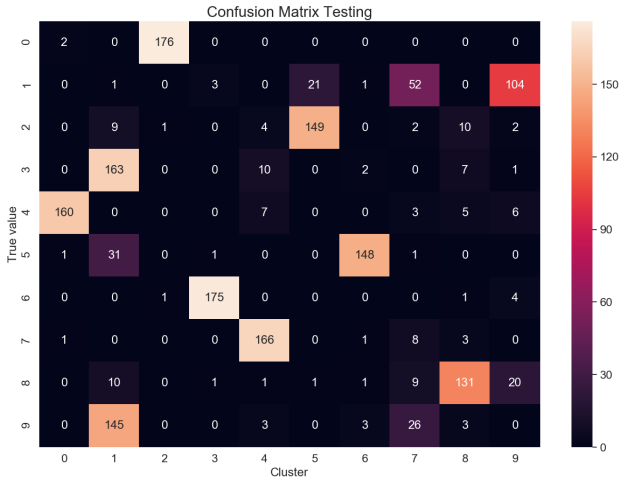


FIGURE 5 – Confusion Matrix for testing data set

4 Conclusion

Analyzing the results obtained by the K-Means algorithm, it is possible to conclude that it achieves considerably good results. This means that the method is reliable for clustering the instances of a certain data set, where the classes are unknown. In addition, within these classes it is possible to verify the tendencies and correlation among features, which are very important for the several applications of the algorithm.