TP6 : K-Means

Yan CHEN & Dajing GU

October 2020

In this TP, the algorithms of K-Means are analyzed and implemented in the program of Handwritten Digit Recognition. Code associated can be found in the GitHub repository: TP6 : K-Means .

1 Principle of K-Means

K-means is an unsupervised learning clustering algorithm. The main idea is to cluster samples of dataset around k points in space by distance closest to the center. Through an iterative method, gradually update the value of the cluster center until the clustering result does not change. The following are the specific steps of the algorithm.

1. Select k samples from the dataset as the center of the clusters
2. Calculate the distance, such as Euclidean distance between the k samples and the remaining samples of dataset.
3. Divide the remaining samples into k clusters according to the minimum distance.
4. Calculate the centre of each cluster by the average of each cluster
5. Calculate the distance between the k centers and the remaining samples of dataset and redivide the remaining samples into k clusters according to the minimum distance.
6. Repeat step 4 and step 5 until the clustering result does not change.

Advantages of the K-means
K means is simple and fast. We can usually obtain a better clustering result.

Problems with K-means

1. Under normal circumstances, K-means can get a good result. However, because the sample selected at the beginning is random, this may cause the optimization function to fall into a local minimum during the iteration process, and thus the clustering effect is not good. For example, in the iterative process, the centers of different clusters may be the same. In order to solve this problem, we can do K-means operation multiple times and select the result with minimum loss function.

2. The value k in the algorithm is given by the user. For the same data set, different values k have different classification effects. There is always an optimal k value. We can use Elbow method to solve the problem, refer to Figure 1.1. But, it exists the situation that there aren't Elbow, we can select the k value according to user, refer to Figure 1.2 [1].

3. When the data is non-spherical, K-means is no longer applicable. Fortunately, we can use density based clustering method. Such as DBSCAN (Density Based Spatial Clustering of Applications with Noise).

4. When the data set is very large, the convergence speed is slow. In fact, we can use Mini Batch K-means algorithm to accelerate the speed of convergence.

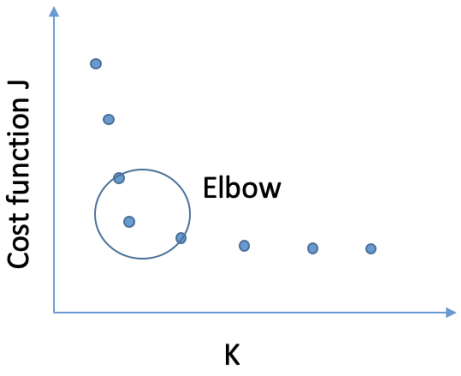

Figure 1.1: Elbow Method

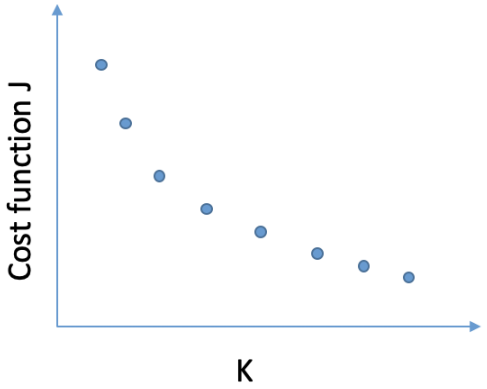

Figure 1.2: Non Elbow

2 Differences between K-Means and K-NN

The is a similar process of classifying a point in both K-Means and K-NN : the point that is found closest to a certain point.

However, K-Means and K-NN are completely different. K-Means is a clustering algorithm for unsupervised learning and has no sample output; while KNN is a classification algorithm for supervised learning and has a corresponding category output.

On the other hand, K-Means has an obvious training process to find the best centroids of k categories to determine the category of the sample, while there's no such a process in K-NN.

3 Inplementation of the algorithms

The dataset used in this project is different from that of TP4. The resolution of the dataset offered is 8×8, which is much smaller than 28×28 of the TP4, so the result of the classification is supposed to be worse than before.

In the code, TP6 K-Means, many of the parameters have been tested, and the best result is shown in this section.

It should be paid attention that the cluster classified by K-Means has no relation with the real label. Although **sklearn.metrics** has offered us some methods to mark the result of classification, we still create a function **getLabelMatched** to have clearer idea of the difference between the cluster classified and the true label.

In this function, the exhaustive method has been taken use. The best match of the true labels (*train_label*) and the clusters classified (*train_label_pred*) is obtained. Then we compare the *train_label_matched* and *train_label*, whose result ios shown in figure 3.1.

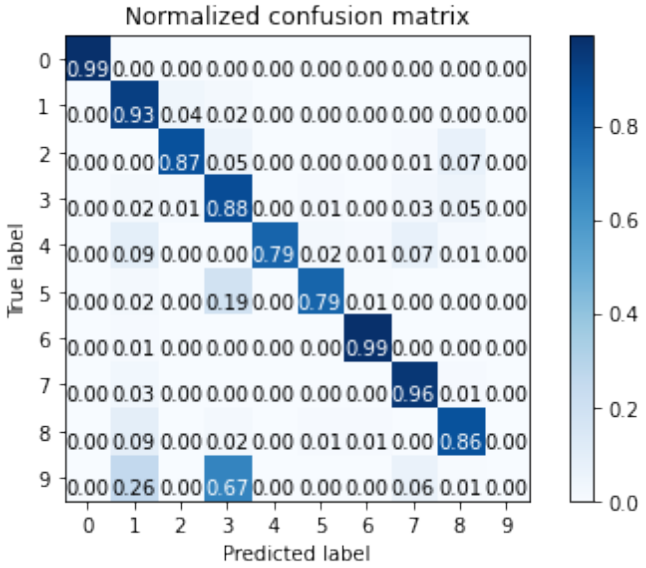

Figure 3.1: Confusion Matrix of the algorithms

The accuracy of the classification can reach 80.70%. In consideration of the few resolution of the dataset, the result is not so bad. The number 9 is often misclassified as 3. The partition of the clusters can be found in figure 3.2.

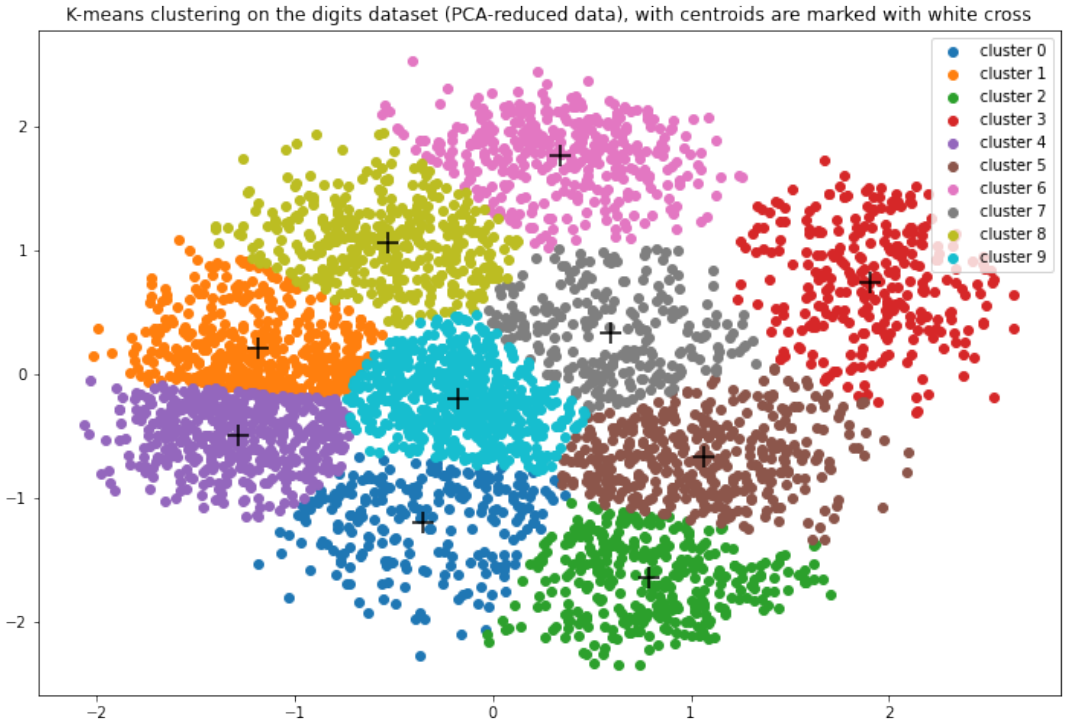

Figure 3.2: Classification of Clusters

References

[1] Andrew Ng. Machine learning. coursera. *Standford University,[Online]. Available: https://www. coursera. org/learn/machine-learning.[Accessed 15 February 2019]*, 2016.