Clustering

Giuseppe L'Erario

Introduction

In this report the focus is on unsupervised learning techniques for cluster analysis. The goal of clustering is to find structures into data when we do not have labeled dataset.

In particular we will apply *K-Means* algorithm and *Gaussian Mixture Model* on *digits* dataset.

1 K-Means

K-Means algorithm is based on the idea of the **centroids**, the average of similar points in a cluster. The weakness of K-Mean is that the number of k clusters must be specified a priori: it is clear that a bad choice of k leads to bad performances.

K-means algorithm can be summarized:

1. Pick k *centroids* randomly and far from each other as center of initial clusters;
2. Assign each point to the nearest centroid;
3. Move the *centroids* accordingly with the mean of the points assigned to it;
4. Iterate until the *centroids* do not move.

The similarity between the point depends on the distance we heave chosen. The commonly used distance used in clustering is the **square Euclidean distance**:

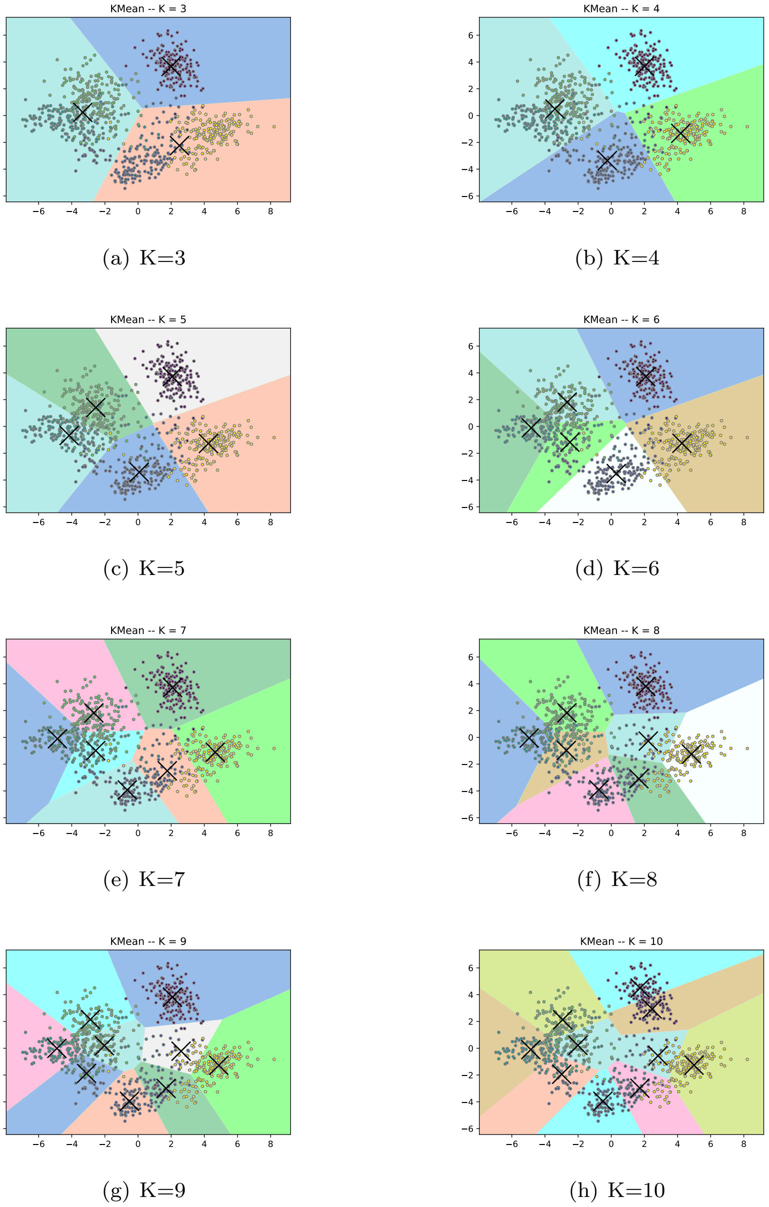
$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})^2 = \sum_i (x_i^{(i)} - x_i^{(j)})^2 = \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| \tag{1}$$

and the problem is the minimization of the **sum of squared errors (SSE)**:

$$\underset{S}{argmin} \sum_{i=1}^{K} \sum_{x_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\| \tag{2}$$

where $\boldsymbol{\mu}$ is the centroid vector for the cluster j.

We import the first five classes of the *digits* dataset, then preprocess data and apply PCA(2).



(a) K=3

(b) K=4

(c) K=5

(d) K=6

(e) K=7

(f) K=8

(g) K=9

(h) K=10

Figure 1: K-Mean clustering

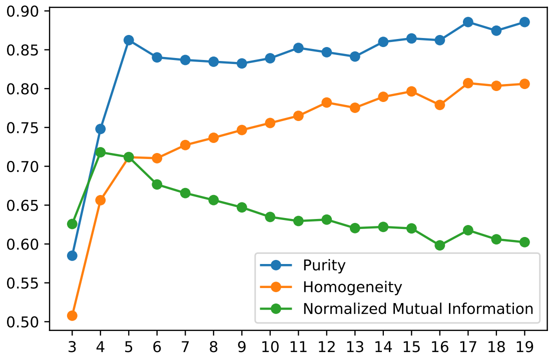


Figure 2: Performance on variyng of the number k of clusters

The results of *K-Means* clustering are shown in fig.1.

The performance are measured with three parameters:

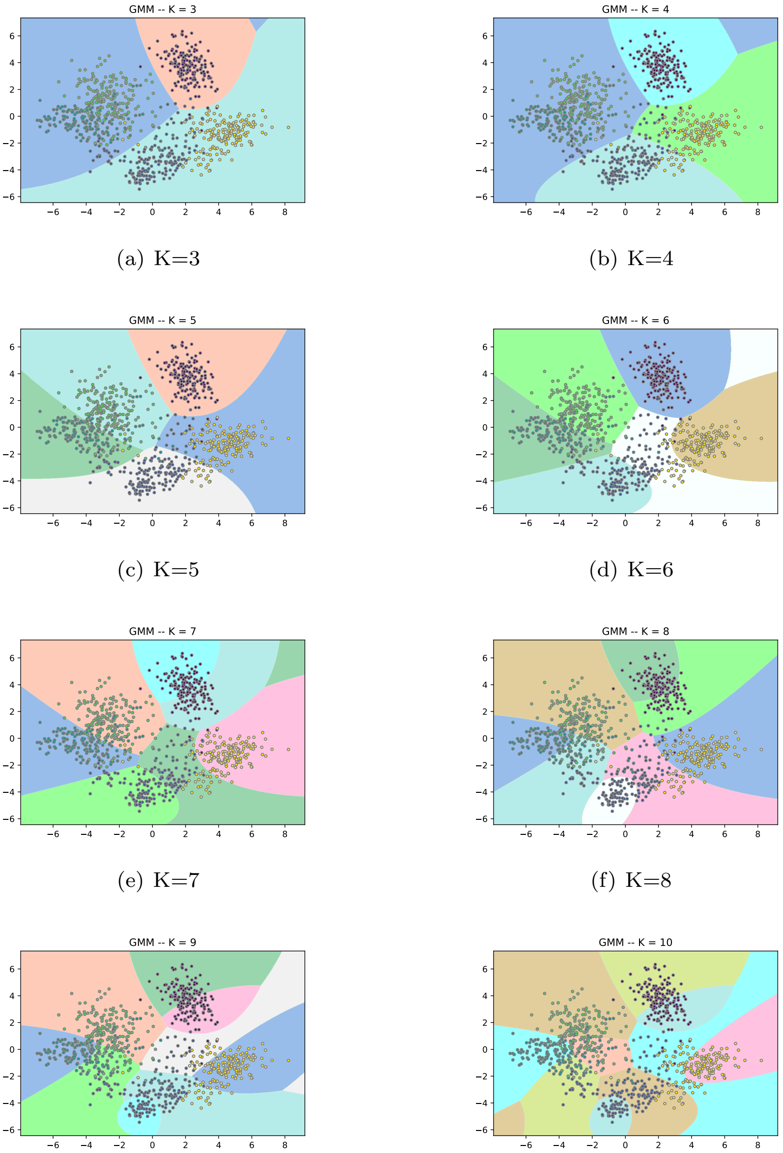
Purity : Each cluster is assigned to the most frequent class in the cluster itself. The accuracy is the rate between the sum of correct-assigned samples and the total samples. The issue of *purity* is that it does not give information about the trade off between the quality of the clustering and the number of the clusters. If we build N cluster for N samples we reach a purity equal to 1, trivially.

Homogeneity : If every cluster contains only data points belonging to a single class, then homogeneity is satisfied. A score equal to 1 means perfect homogeneous labeling.

Normalized Mutual Information : This function scales the *Mutual Information* score between 0 and 1. The *Mutual Information* measures the amount of information that we have about the data, knowing what the cluster is. Like the *purity*, the problem of the *Mutual Information* is that it reaches trivially the maximum value if we build N clusters for N samples. **NMI** fixes this problem.

In fig.2 can be seen that the *purity* reaches a peak when $k = 5$ and remains constant. The *homogeneity* improves with k, infact, the smaller are the clusters the more is the probability that in one cluster are present homogeneous samples. On the other side, *NMI* reach its maximum when $k = 5$ and then decreases, because penalizes the large cardinalities.

2 Gaussian Mixture Models



(a) K=3

(b) K=4

(c) K=5

(d) K=6

(e) K=7

(f) K=8

(g) K=9

(h) K=10

Figure 3: K-Mean clustering

A Gaussian mixture model is a probabilistic model that implies that all the data points are generated from a mixture of Gaussian distributions. These Gaussian distributions are fitted with *Expectation-Maximization* algorithm. This algorithm assumes random components and computes for each point the probability of being generated by each model. Then iteratively tunes the parameter to maximize the associated likelihood of the data.

The results of the GMM-based clustering is shown in fig.3.

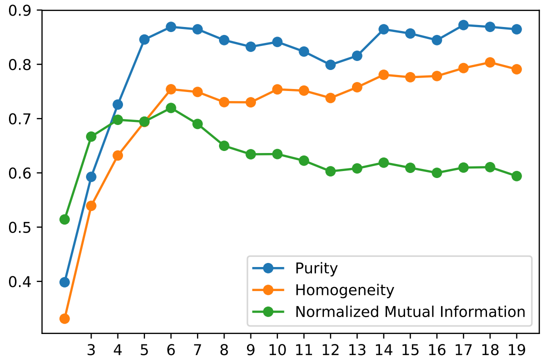


Figure 4: Performance on variyng of the number k of clusters

The behaviour of the GMM on variyng of k is quite similar to the K-means. *Purity* increases till $k = 5$ and remains quite constant. *Homogeneity* increases with k, while *NMI* decreases after a peak around $k = 5$. The only difference is that the *NMI* does not change drastically as with K-mean algorithm.