SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria
INGEGNERIA INFORMATICA
(Intelligenza Artificiale)

Corso: Machine Learning

Docente: Prof. LUCA IOCCHI

K-MEANS CLUSTERING ALGORITHM

REDJAN SHABANI
(1013173)

1- DATA CLUSTERING

Suppose we have a dataset:

$$D = \left\{x^{(1)}, \dots, x^{(d)}\right\} \quad with\ x^{(i)} \in \mathbb{R}^n$$

and, let's suppose that there is some reason to believe that this data are grouped in subset according to a property. An example of clustering problem is the segmentation of an image by considering pixels position and color. An RGB image is a set of pixels, where each pixel have an associated color defined by the vector$[\rho\ \gamma\ \beta]^T$, where ρ represents the level of the red, γ representes the level of the green and β the level of blue. In RGB images, the components of the color vector, associated to each pixel, are integer values in the interval$[0; 255]$. Digital images are defined by three matrices R, G, B. We can represent each pixel with coordinates (i, j), as a 5-dimensional vector:

$$\boldsymbol{p}_{i,j} = \begin{bmatrix} x \\ y \\ \rho \\ \gamma \\ \beta \end{bmatrix}$$

Let's now suppose that we want to distinguish the objects captured by the image. In general objects are projected to the images as region with soft color variation. From this presupposition we can try to apply the inverse process; determine regions in the image by considering color variations.

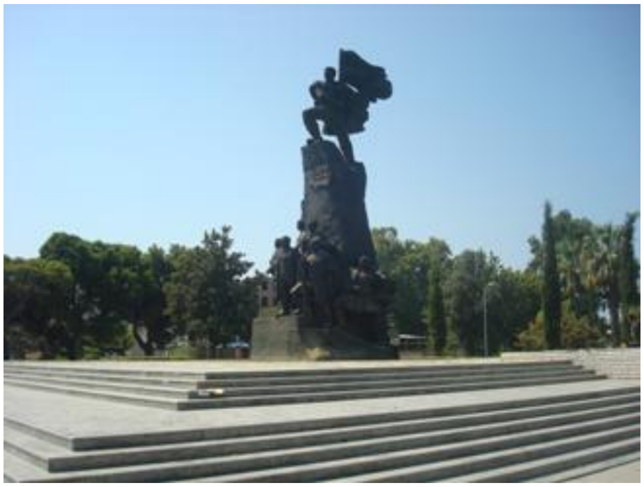

Figure 1-Original Image (Flag's Square - Valona)

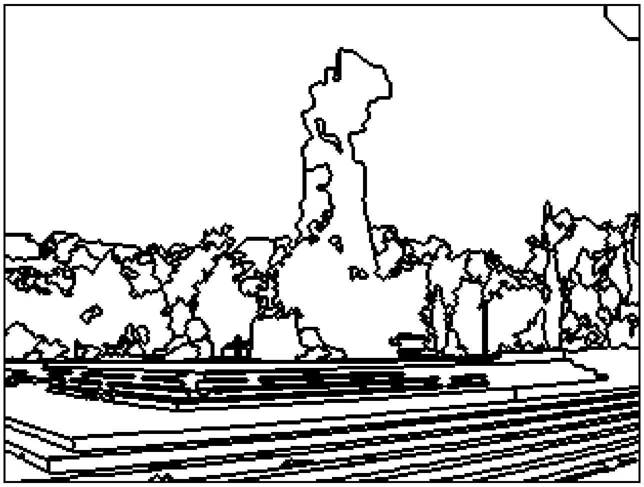

Figure 2-Clustered image obtained with Mean Shift Algorithm

It is possible define a certain numbers of color interval and classify the image by this color producing some images that visualizes only the associated interval of color (fig.1). The problem with this approach is that different objects may have the same color, so we have to consider even the spatial distribution of the homogeneous regions, for example by considering connected regions as the projection of a particular object. In formal terms the clustering is the process of grouping data, by using spatial distribution information of the dataset.

2- K-MEANS ALGORITHM

K-Means algorithm starts from an assumption that data are classified in predetermined classes (K classes). For each data point in the dataset

$$D = \left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\right\}$$

we introduce an

$$r_k^{(n)} = \begin{cases} 1 & iff\ \boldsymbol{x}^{(n)} \in C_k \\ 0 & otherwise \end{cases}$$

where $k \in \{1,2,\dots,K\}$ is the counter index for the classes and C_k is the k^{th} class. We can define an objective function to be minimized, as a measure of sum of squared distances of data points from the assigned class's centroids, sometimes called *distortion measure*, given by:

$$J = \sum_{n=1}^{N}\left(\sum_{k=1}^{K} r_k^{(n)} \left\|\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k\right\|^2\right)$$

where μ_k is the centre of mass of data point in C_k:

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}(C_k) = \frac{1}{|C_k|}\sum_{x \in C_k} \boldsymbol{x}$$

The parameters $\boldsymbol{\mu}_k$ have to be intended as the centres of masses of all entities of this class, not as the centre of mass only our observed data, classified members of C_k.

The minimization of J is done by repeating iteratively two phases:

1- changing $r_k^{(n)}$ and maintaining fixed all $\boldsymbol{\mu}_k$

2- changing $\boldsymbol{\mu}_k^{(k)}$ and maintaining fixed $r_k^{(n)}$

The values for $r_k^{(n)}$ may be chosen considering the nearest centroid. This result gives us a simple solution for choosing the $r_k^{(n)}$ parameters, more precisely:

$$r_k^{(n)} = \begin{cases} 1 & iff\ k = argmin_j \left\{ \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_j \right\|^2 \right\} \\ 0 & otherwise \end{cases}$$

Now consider the dependence of J with respect to $\boldsymbol{\mu}_k$:

$$\frac{\partial J}{\partial \mu_i} = \frac{\partial}{\partial \mu_i}\left[\sum_{n=1}^{N}\left(\sum_{k=1}^{K} r_k^{(n)} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k \right\|^2\right)\right] = \frac{\partial}{\partial \mu_i}\left[\sum_{n=1}^{N} r_i^{(n)} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i \right\|^2\right] = \sum_{n=1}^{N} \frac{\partial}{\partial \mu_i}\left[r_i^{(n)} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i \right\|^2\right]$$
$$= 2\sum_{n=1}^{N} r_i^{(n)} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i \right\|$$

Imposing zero value for the derivative:

$$\sum_{n=1}^{N} r_i^{(n)} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i \right\| = 0$$

$$\Rightarrow \sum_{n=1}^{N} r_i^{(n)} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i \right\| = 0$$

$$\Rightarrow \sum_{n=1}^{N} r_i^{(n)} \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i \sum_{n=1}^{N} r_i^{(n)} = 0$$

$$\boldsymbol{\mu}_i = \frac{\sum_{n=1}^{N} r_i^{(n)} \boldsymbol{x}^{(n)}}{\sum_{n=1}^{N} r_i^{(n)}}$$

The kmeansClusterin.m contains a MatLab implementation of K-Means algorithm. You can find even two test scripts, one applied to a distribution of data obtained from four gaussians and another test is applied for clustering an image in gray scale.

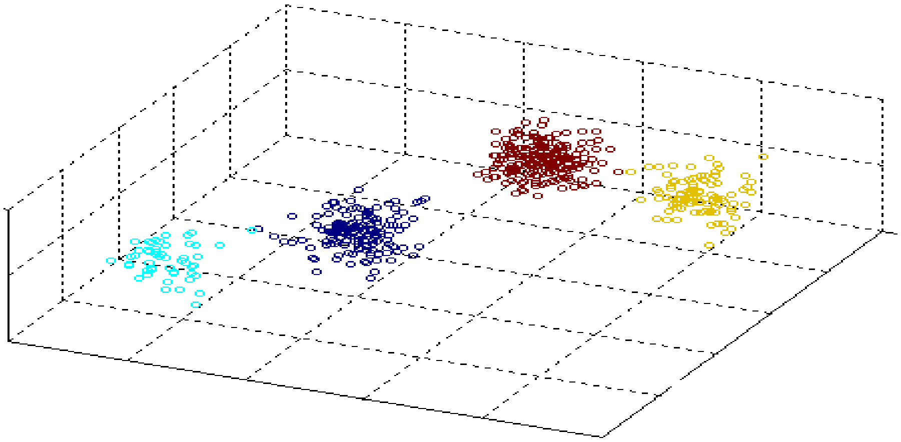

Figure 3-Clusters detected by K-Means(K=4) over a dataset of 3-D points obtained from four gaussians.

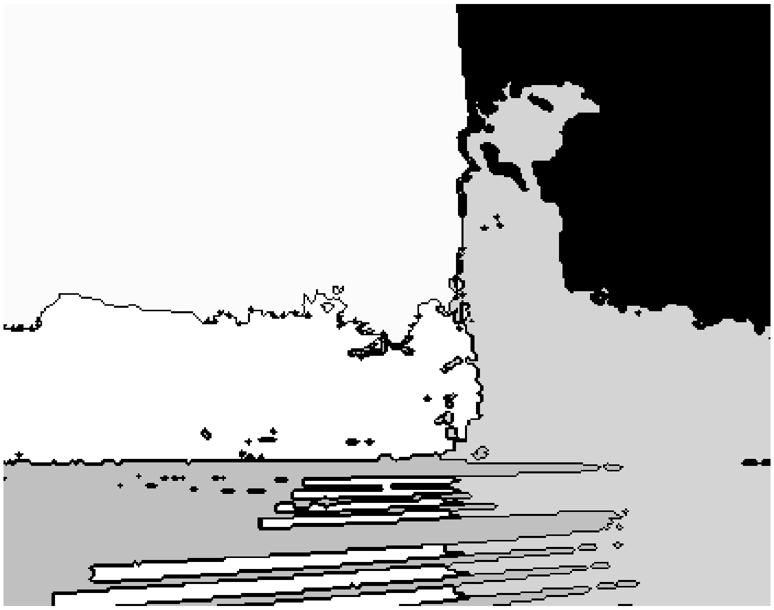

Figure 4-Segmetation with K-Means(K=5) of the image(245x326) in fig.1

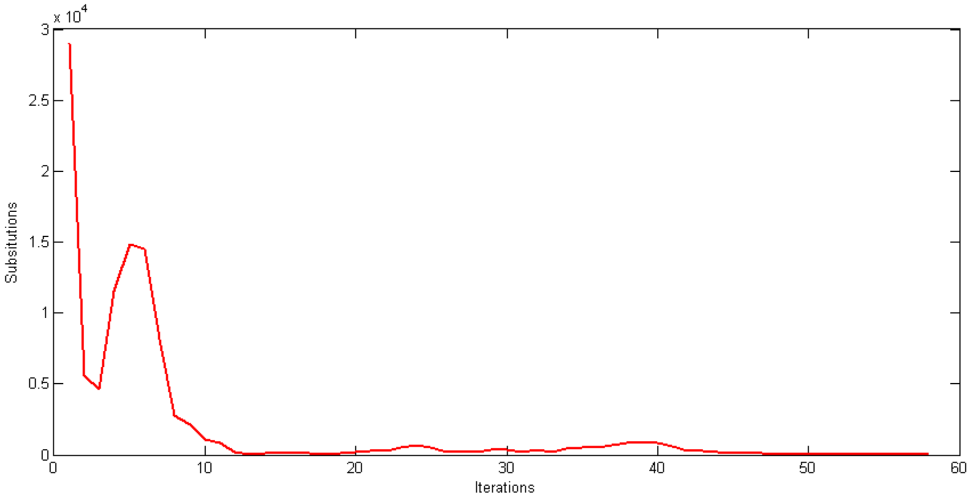

Figure 5-Number of substitutions over each iteration for the segmentation of the image in fig.1.