E0-270 Machine Learning - Assignment 2

Submitted by: Shingala Jaydeep Jaysukhbhai (SR: 21076)
Department of Computer Science and Automation, IISc, Bangalore.
jaydeeps@iisc.ac.in

1 Problem Statement

The task is to cluster the pixels of the image using the K-Means algorithm, with given number of clusters k = 2, 5, 10, 20, 50.

2 Introduction

we are given an RGB image of shape 512 × 512 × 3. Each pixel of the image is considered to be a data point. Now we want to cluster that image pixels into different clusters as given k = 2, 5, 10, 20, 50.

3 The K-Means Algorithm

Clustering is grouping of objects into subsets or clusters such that objects within one cluster are close to each other than those assigned to different clusters. Clustering of a set X of points into K partitions C = C1, C2, . . . , Ck is such that,

$$\bigcup_{j=1}^{k} C_j = X, \qquad C_i \cap C_j = \emptyset \quad \forall i \neq j, C_j \neq \emptyset \quad \forall j = 1, \ldots, K$$

K-Means is a clustering algorithm used to partition a given dataset into K clusters. The goal of K-Means is to minimize the sum of squared distances between each data point and the Cluster center of its assigned cluster. In other words, K-Means tries to find K cluster centers such that those center points can minimize the distances between each data point of the dataset and its assigned cluster center point.

Here's how the K-Means algorithm works:

1. Initialize K Cluster centers randomly from the data points.
2. For each data point of the dataset assign it to one of the clusters based on some distance calculation between datapoint and centers that we have.
3. Recalculate the centers based on the mean of the data points assigned to each cluster that is created.
4. Repeat steps 2 and 3 until convergence (i.e., until the Cluster centers no longer change more then ϵ).

For step 2, For each data point, we calculate the Euclidean distance between the data point and each Cluster center. Then data point is then assigned to the nearest Cluster center having minimum ecludian distance out of all cluster centers.

The Euclidean distance between two points (p1, p2, p3) and (q1, q2, q3) is calculated as:

$$\text{distance} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}$$

The K-Means algorithm is based on the concept of minimizing the sum of squared distances between each data point and the Cluster center of its assigned cluster. This is known as the within-cluster sum

of squares (WCSS), and it is given by the following formula:

The Optimization problem:

$$\min_{C} WCSS = \min_{C} \sum_i \sum_j ||x_j - c_i||^2$$

Here, i represents the cluster index, j represents the data point index within the i-th cluster, x is the j-th data point in cluster i, and c is the Cluster center of cluster i.

To minimize the WCSS. To do this, the algorithm alternates between two steps:

Assignment step: Each data point is assigned to the nearest center of cluster based on the minimum Euclidean distance.

Update step: The Cluster centers are updated to the mean of the data points assigned to them.

This process is repeated until convergence, which occurs when the assignments no longer change.

The assignment step can be written mathematically as:

$$S(i) = \{x_j : ||x_j - c_i||^2 \leq ||x_j - c_k||^2 \forall k \neq i\}$$

Here, S(i) is the set of data points assigned to cluster i. This equation says that each data point is assigned to the cluster whose Cluster center is closest to it.

The update step can be written mathematically as:

$$c_i = \frac{1}{|S(i)|} \sum_{j \in S(i)} x_j$$

Here, |S(i)| is the number of data points assigned to cluster i. This equation says that the Cluster center of cluster i is updated to the mean of the data points assigned to it.

The K-Means algorithm is guaranteed to converge, but it may converge to a local minimum rather than the global minimum. To mitigate this, the algorithm is often run multiple times with different initial Cluster centers, and the solution with the lowest WCSS is chosen.

4 Methedology

Here, we will be using K-Means clustering algorithm from scratch to cluster image pixels into different desired clusters.

1. Initialize the algorithm by selecting the number of clusters (k) and randomly initializing the Cluster center of each cluster.
2. Assign each data point to its nearest Cluster center, creating k clusters.
3. Recalculate the Cluster centers of each cluster by taking the mean of all data points assigned to that cluster.
4. Repeat steps 2-3 until the old cluster center and new cluster centers are more then ϵ apart or until a maximum number of iterations is reached.
5. Output the final clusters and their Cluster centers. For convergence we can consider ϵ

$$\delta = ||\mathbf{C}_{t+1} - \mathbf{C}_t|| \tag{1}$$

where:

- δ is the change in distance between the old and new cluster center assignments
- $|| \cdot ||$ represents the Euclidean norm of the vector
- $\mathbf{C}t$ represents the cluster center assignments at iteration t
- $\mathbf{C}t + 1$ represents the cluster center assignments at iteration $t + 1$ after the means of the assigned data points are calculated.

5 Things to be considered while using KMeans

1. Initialisation of the starting Cluster centers is crucial and important. if we do not initialise properly then we might not get good desired clusteres after convergence.
2. This clustering is Hard clustering, which means we are assigning definitely any point to any cluster and that also to only one cluster.
3. Though here the number of wanted clusters were given but, deciding on how many clusters to use is also crucial.

6 Results

The original image that was given for K-Means clustering is as given below.

Figure 1: Original image

7 Mean Squared Error

MSE is a typical indicator of image quality that is frequently used to assess how well image processing or compression methods perform. Lower numbers denote higher quality, and it measures the average amount of error between the original and processed images. Here MSE calculated by squared difference between each pixel of original image and clustered image.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I_{original}(i) - I_{clustered}(i))^2$$

Figure 2: Clustered image for K = 2

Figure 3: Clustered image for K = 5

Figure 4: Clustered image for K = 10

Figure 5: Clustered image for K = 20

Figure 6: Clustered image for K = 50

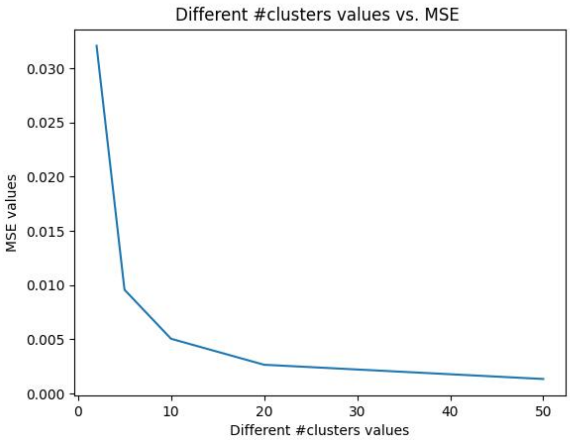

Figure 7: Plot of Different number of clusters (k) Vs. MSE between original and K clustred image

The MSE error values with respect to different Number of clusters while running the implemented code is given below in table:

Table 1: Table of MSE values for K Number of clusters

K	MSE Values
2	0.03206917602224996
5	0.009577312447572142
10	0.005056930487915403
20	0.0026609903711270195
50	0.0013490113538570952

8 Conclusion

KMeans clustering algorithm can be used for Image compression and size reduction if we encode an image with some pixel values by finding approximate representations for each pixel, that is their cluster centers. The convergence in KMeans is crucial, and we also need to be careful when choosing initial cluster centers. Experimentally we observed that after the number of clusters as k = 10, we almost represented the image correctly.