K-Means clustering and Color Quantization

Daniel Amirtharaj
damirtha@buffalo.edu

1 Objective

To detect clusters within a given dataset using the k-means algorithm, as well as apply k-means for color quantization of images.

2 Analysis

2.1 k-means Clustering

Clustering is an unsupervised learning method which tries to partition data points based on their relative distribution in the feature space. The k-means clustering algorithm in particular, takes a parameter k, forms k different clusters in the dataset and classifies them according to the cluster center to which the data point is closest to. Euclidean distances are used here to compute distances. The cluster center is recalculated once a classification is made, and is the the mean of all points that have been assigned to the same cluster.
The following iterative steps are used to find k-clusters in the data,

1. Initialize μ_k cluster centers for each k (can be done randomly)
2. Repeat until convergence (when μ remains unchanged),
    1. Assign a cluster to each data point x_i in the dataset based on which cluster center μ_j is closest to the datapoint.
    2. Using the cluster assignments from the previous step, recompute μ_k for each cluster k.

3 Method

The method or procedure followed in solving each problem or task is laid out below. The following steps were followed to perform the k-means algorithm on a sample dataset,

1. Model parameters such as k and Mu were initialized as given in the project description.
2. k-means was applied on the sample data X following the algorithm as described in section 3.2.1, and scatter plots at certain snapshots of the algorithms iterations were captured and saved.

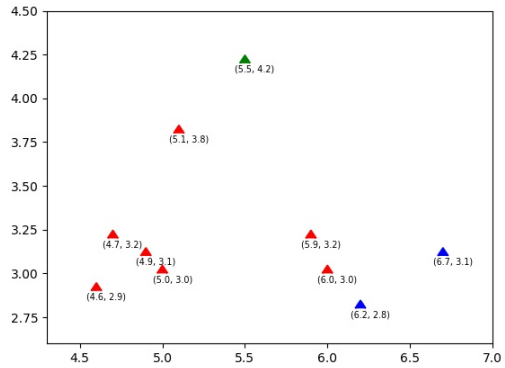
The following steps were followed to perform color quantization of images using k-means,

1. The image was converted into an array of data samples of dimension (d *by* f) where $d = m$ x n, and m x n is the shape of the input image.
2. Model parameters k were set as given and μ_k for each k were initialized randomly.
3. k-means was applied on the image array, following the algorithm as described in section 3.2.1.

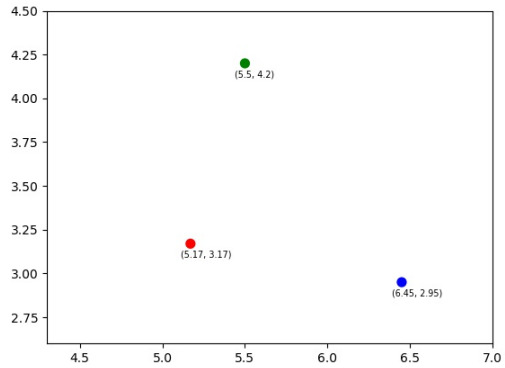
4. Once the algorithm converged, the resultant array was reshaped to that of the image with each data point assigned μ_k values if it belonged

4 Results

The following results were obtained at various steps of the program. Running k-means on the sample data set X for 2 iterations, gave the following outputs,



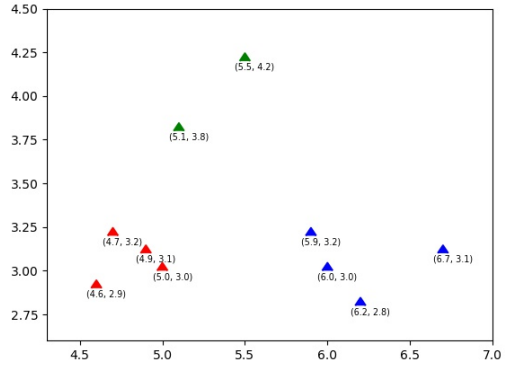
(a) Classification of data points based on initial μ.



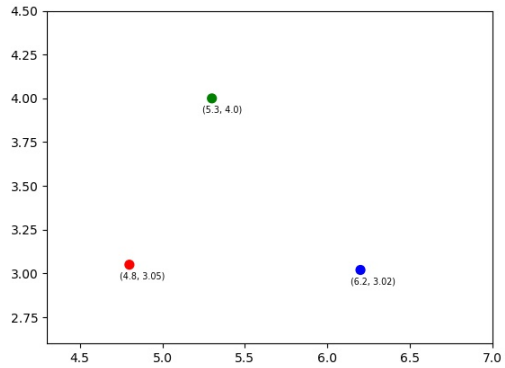
(b) μ computed in the first iteration.

Figure 1: Iteration 1 of k-means on given data sample.

Classification vector at first iteration : [1, 1, 3, 1, 2, 1, 1, 3, 1, 3]
μ computed at first iteration : μ_1 = [5.17 3.17], μ_2 = [5.5 4.2], μ_3 = [6.45 2.95]



(a) Classification of data points after 1st iteration and coomputation of μ.

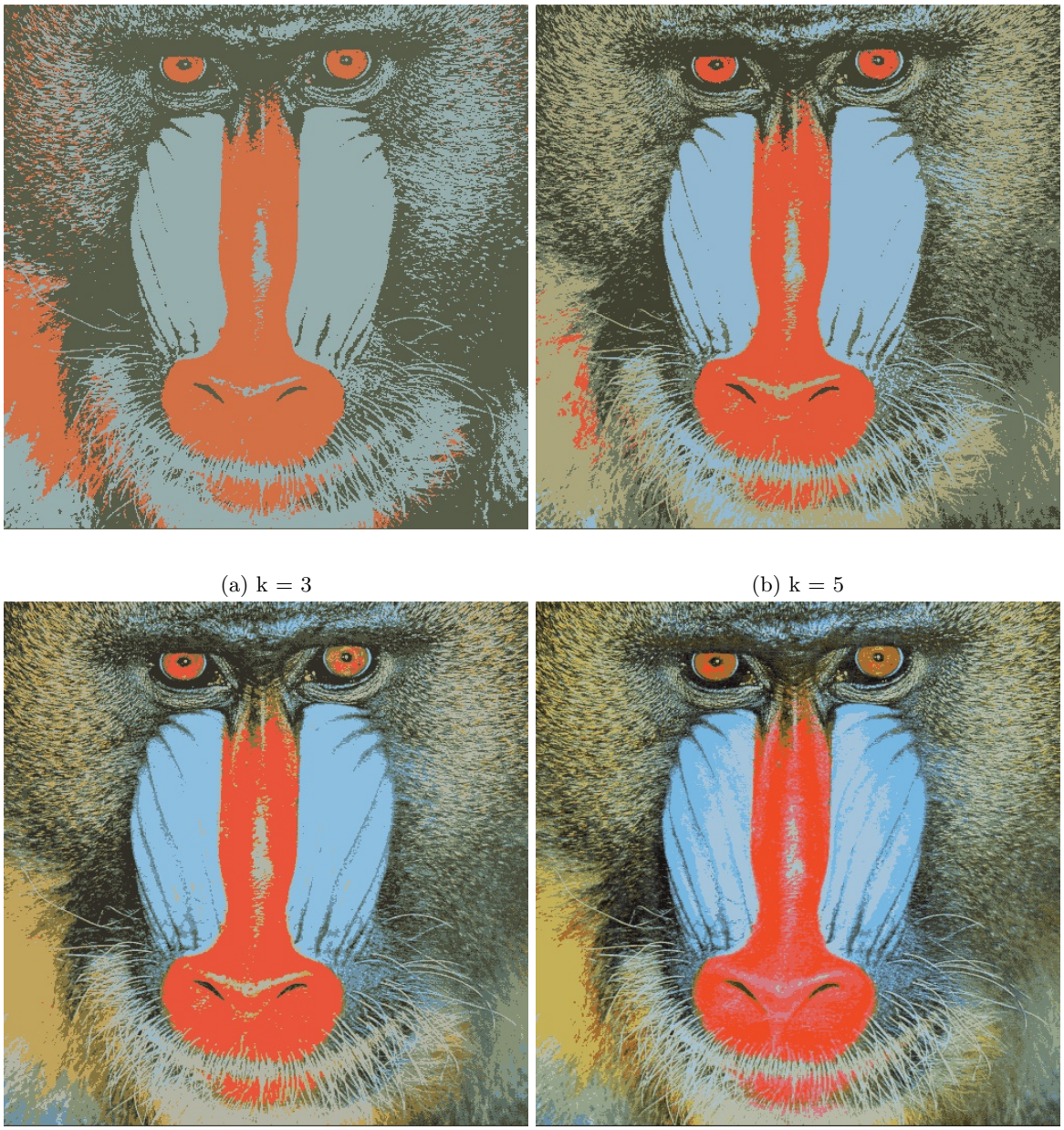


(b) μ computed in the second iteration.

Figure 2: Iteration 2 of k-means on given data sample.

Classification vector at second iteration : [3, 1, 3, 1, 2, 1, 1, 3, 2, 1]
μ computed at first iteration : μ_1 = [4.8 3.05], μ_2 = [5.3 4], μ_3 = [6.2 3.02]]

Applying k-means for color quantization of the baboon.jpg image yielded the following images for different values of k. It can be observed that these images have only k colors in them after the quantization.



(a) k = 3

(b) k = 5

(c) k = 10

(d) k = 20

Figure 3: k-means clustering for color quantization in image, for different values of k.