TSKS12 Modern Channel Coding, Inference and Learning

Laboration 1:

Implementation and Application of K-means Clustering Algorithm

I. Introduction

In this lab, you will implement K-means clustering algorithm and apply it to compress a colored image. In order to gain an intuition on how the K-means clustering algorithm works, in the first part of this exercise, you are given a toy two dimensional data set with 7 observations or data points and you are supposed to group them into 2 clusters. The second part of this lab involves application of the K-means clustering algorithm in image compression to reduce the number of colors that occur in a colored image to only those that are most common in that image.

II. Background on K-means Clustering Algorithm

The K-means clustering algorithm is a heuristic algorithm of partitioning N observations or data points in an I dimensional space into K cohesive clusters. It is basically a method to cluster similar data examples together. Each cluster is parametrized by a vector $\boldsymbol{m}^{(k)}$ which is called the mean or the cluster centroid. The data points are usually denoted as $\{\boldsymbol{x}^{(n)}\}$ where n runs from 1 to the number of data points N. The vector $\boldsymbol{x}^{(n)}$ is I-dimensional and it is assumed that the $\boldsymbol{x}^{(n)}$ lives in a real space, i.e., $\boldsymbol{x}^{(n)} \in \mathbb{R}^I$. Furthermore, we use the following metric to define the distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i (x_i - y_i)^2. \tag{1}$$

The K-means algorithm is an iterative procedure that starts by making a guess about the initial cluster centroids or means and then refines this guess by repeatedly assigning the observations to their nearest mean and then recomputing the means based on the assignments. It consists of the following steps: 1)

Initialize the K means to random values, 2) Assign each observation or data point to its closest centroid based on a certain distance metric, and 3) Recomputing or updating the mean of each cluster using the data points assigned to it. One then has to repeat the assignment and the update steps until the assignments do not change. At this point, the K-means algorithm is said to converge.

Note that the converged solution may not be ideal and depends on the initial guess about the means. Therefore, in practice, the K-means algorithm is usually run with different random initializations and different solutions are obtained. You should choose the one that gives the lowest distortion or the cost function.

III. SET UP

The following hardware and software is needed for this lab:

- A computer
- MATLAB/ OCTAVE installed

With the theoretical background and the set up in place, you need to submit the codes and the saved output in a report for the following two tasks:

IV. TASK-I

Using the K-means clustering algorithm and the distance metric in (1) discussed above, you are supposed to group the following 7 data points into two clusters: $\boldsymbol{x}^{(1)} = (1, 1)$, $\boldsymbol{x}^{(2)} = (1.5, 2)$, $\boldsymbol{x}^{(3)} = (3, 4)$, $\boldsymbol{x}^{(4)} = (5, 7)$, $\boldsymbol{x}^{(5)} = (3.5, 5)$, $\boldsymbol{x}^{(6)} = (4.5, 5)$, and $\boldsymbol{x}^{(7)} = (3.5, 4.5)$. Suppose that you assigned $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(4)}$ as initial cluster centroids for K-means clustering. How many iterations are needed to converge? On a 7×7 space with all the seven points, display the two clusters and the new means after each iteration until convergence.

V. TASK-II

In this task, the objective is to apply K-means clustering algorithm for image compression. You are supposed to pick up a $256 \times 256 \times 3$ colored image and reduce the number of colors in it to K colors. Basically, use K-means algorithm to select the K colors that will be used to represent the compressed image. More precisely, treat every pixel in the image as a data point and use the K-means algorithm to find the K colors that best cluster the pixel in the 3-dimensional RGB space. Once, the cluster centroids

on the image have been computed, you should then use the K colors to replace the pixels in the original image.

Note that in a 24-bit color representation of any image, each pixel is represented as three 8 bit unsigned integers ranging from 0 to 255 that specify the red, green, and blue intensity values. Any colored image that you pick contains thousands of colors and you will reduce it to L_0. The photo can thus be stored in a more efficient way. You only need to store the RGB values of the K selected colors and for each pixel in the image you now only need to store the index of the color at that location.

These are the steps that you need to follow for this exercise:

1) Read a colored image of dimension $256 \times 256 \times 3$.
2) Initialize the means: Select K colors randomly from the image and use them as initial means.
3) Find the closest cluster centroid or mean: Navigate through each pixel in the image and compute its distance from the K means. The pixel gets assigned to its closest cluster centroid or mean.
4) Update means: Once each pixel is assigned to its closest centroid, for each centroid, the algorithm should recompute the means of the points that were assigned to it.
5) Repeat steps 2 and 3 until convergence: you basically need to check the distance between the initial and the updated means and see if the difference is less than a certain threshold, then you can terminate.
6) Once you have the final cluster centroids, you can now replace each pixel in the image with the mean that it is closes to.

So, to summarize you basically have 256×256 data points of three dimensions each and you need to cluster them into $K = 8, 16, 64$ groups (try these three different values of K). Find the compression ratio in each case and display both the original and the image compressed using K-means. Try out different colored images and display both the original and compressed images. You can as well play around with the distance metric in (1) and use the following distance metric and see how that affects the compressed image:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i |(x_i - y_i)|. \tag{2}$$

In the report, you should submit the codes along with comments related to the implementation of the K-means algorithm and all your MATLAB/OCTAVE generated results in a pdf file with proper description of the steps followed.