M156 – Lab assignment #5: K-means clustering

The objective of this lab is to get familiar with the K-means clustering technique and to apply it to a naive image segmentation/compression problem.

1 Theory

1. Recall the objective function which is minimized in the K-means algorithm and derive the updates for the cluster centroids $\boldsymbol{\mu}_k$ and the assignment variables r_{nk}.
2. The value of a given pixel in a gray-level image is usually coded on 8 bits. Give the number of bits required to code an entire color image with n rows and m columns.
3. The goal of image compression is to reduce the size of an image, while keeping the result as close to the original as possible. If an image is to be approximated by a segmentation coming from the K-means algorithm, what are the quantities that have to be stored in order to be able to reconstruct the segmented image?
4. Compute the number of bits required to store the compressed image, as function of K (using a minimal –integer– number of bits to store K).

2 Experimental Part

The data we are using is a $96 \times 144 \times 3$ color image of Royce Hall on campus. The image is stored in a `.mat` file `royce_hall_small.mat`.

1. Code the K-means algorithm in a separate function. The algorithm should take as input the color image (in a matrix form), and the number of desired classes. It should return the label of each pixel, and the values of the K centroids. The initialization of the clusters should be made using random pixel values. The algorithm should stop when the relative variation of the objective function between two iterations is below 1%.
2. Test the algorithm for different values of K. Coded as we presented it in class, it is rather slow and can take up to a few tens of seconds to converge. You can display the labels as images (use a colormap to distinguish easily between them). Another possibility is to create a segmented image in which each pixel has the value of the centroid it was assigned to by the algorithm.

   Show and comment on a few segmented images for a few relevant values of K between 1 and 50. Try to interpret the different clusters formed by the algorithm. What is a good range of values of K for segmentation? Recall that the goal of segmentation is to identify meaningful structures and regions of the image, not to get a good approximation of the original image.
3. Make a 3D scatterplot of the data using the three color channels. Add the clusters centroids for one of the values of K in the range identified in the previous question. Relate each centroid to a structure in the image by having a look to the color values (low value for all channels: dark object; high value for all the channels: bright object)
4. Add zero mean unit covariance Gaussian distributed noise to the image, such that the SNR is 10dB (see lab 1 for a definition of the SNR). What is the effect of the noise on the output of the algorithm?
5. For this question on (but not before), use the built in `kmeans` function of MATLAB. The implementation is much more efficient than the naive version you have here.

6. Run the K-means algorithm for all K values between 1 and 100. Show and comment on a few RMSE images for a few values of K (show them on the same scale for a meaningful comparison). The RMSE is defined as

$$RMSE = \frac{1}{\sqrt{3}}||\mathbf{x} - \hat{\mathbf{x}}||_2$$

where $\mathbf{x}$ is the original (vectorized) image, and $\hat{\mathbf{x}}$ is its approximation using the clustering algorithm.

7. Plot the mean value of the RMSE as a function of K. What is the behavior of the RMSE when K increases? What is the limit when $K \to N$, where N is the number of pixels (do not run the algorithm for this value of K! Stop at $K = 100$)? Explain then why K-means can be used for image compression. In this context, what is the role of parameter K?

8. Now, plot the mean RMSE against the compression ratio, which is defined as the ratio between the number of bits required to encode the compressed image, and the number of bits required to encode the original image. Explain the shape of the obtained curve. Say we want to get a compression ratio between 10 and 15 %. What value should we choose for K then?

9. Make the same scatterplot a in question 3, but with a larger value of K (say 50). Using clustering algorithms to compress images is sometimes referred to as *vector quantization*. In signal processing, quantization refers to the operation of discretizing a real valued function using only a few possible achieveable values (typically to allow storage of signals on a computer, which cannot handle continuous values), as shown in Figure 1. Explain why clustering amounts to doing something similar with vector valued functions.

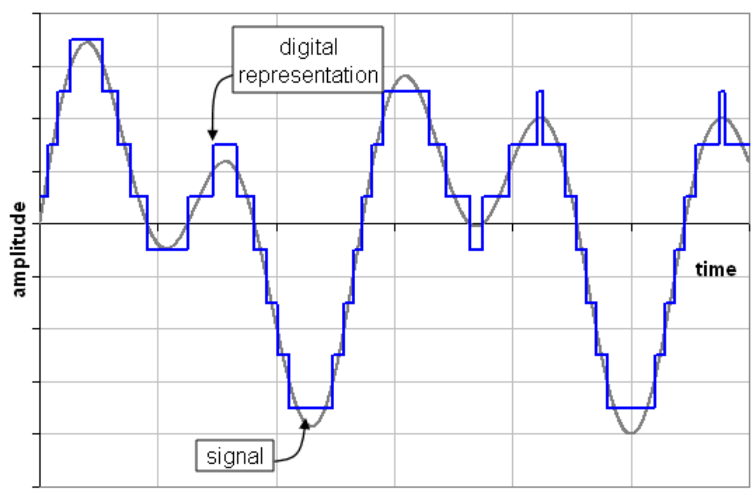

Figure 1: Quantization of an analog signal