Homework 1 Progamming Report

Liu Yujia GTID:903070716

October 2014

Programming: Image compression [30 pts]

1. Within the K-medoids framework, you have several choices for detailed implementation. Explain how you designed and implemented details of your K-medoids algorithm, including (but not limited to) how you chose representatives of each cluster, what distance measures you tried and chose one, or when you stopped iteration.

   Solution:

   Within the K-medoids framework, I choose the median of all data point in the cluster to represent the cluster. The distance measures is Euclidean distance. And I stop the iteration when the cluster assignment will not change anymore.

2. Attach a picture of your own. We recommend size of 320×240 or smaller.

   

3. Run your K-medoids implementation with the picture you chose above, with several different K. (e.g, small values like 2 or 3, large values like 16 or 32) What did you observe with different K? How long does it take to converge for each K?

   Solution:

   When K is 2 or 3, the compressed image usually looks comparatively resemble the original image in shape and but not in color. However, as the K increases(e.g more than 10), the compressed image looks more and more similar with the original image.

   For each K, I run the algorithm 5 times and take the average of the running times as its result. For $K = 2$, the running time is 10.00 seconds. For $K = 5$, the running time is 5.76 seconds. For $K = 10$, the running time is 4.21 seconds. For $K = 20$, the running time is 4.90 seconds.

4. Run your K-medoids implementation with different initial centroids/representatives. Does it affect final result? Do you see same or different result for each trial with different initial assignments? (We usually randomize initial location of centroids in general. To answer this question, an intentional poor assignment may be useful.)

   Solution:

Since the initial centroids are randomly chosen, we assume that the initial centroids are different for every times. Different result for each trial can be seen. When the initial centroids are badly chosen, the compressed image quality decreased a lot, especially when K is small.

5. Repeat question 2 and 3 with K-means. Do you see significant difference between K-medoids and K-means, in terms of output quality, robustness, or running time?

   Solution:

   We run K-means on the same image to compare these two algorithms.

   When running K-means, since the image is not large, it seems that there is no significant quality difference of the compressed image quality from that using the K-medoids algorithm and is also not easily affected by the initial assignment when K is relatively large. However, when K is small(e.g less than 5), it seems that K-medoids performs better than K-means basically and has stronger robustness.

   As for running time, similarly, I run the K-means algorithm for different K for 5 times and take the average. The result is followed. For $K = 2$, the running time is 0.76 seconds. For $K = 5$, the running time is 0.72 seconds. For $K = 10$, the running time is 0.95 seconds. For $K = 20$, the running time is 0.96 seconds. It is obvious that the K-means runs much faster than K-medoids.