ISYE 6740 Summer 2021
Homework 1 (100 points + 2 bonus points)

1 Image compression using clustering [60 points]

In this programming assignment, you are going to apply clustering algorithms for image compression. Your task is implementing *K-means* for this purpose. **It is required you implementing the algorithms yourself rather than calling k-means from a package. However, it is ok to use standard packages such as file i/o, linear algebra, and visualization.**

Formatting instruction

Input

- `pixels`: the input image representation. Each row contains one data point (pixel). For image dataset, it contains 3 columns, each column corresponding to Red, Green, and Blue component. Each component has an integer value between 0 and 255.
- `k`: the number of desired clusters. Too high value of K may result in empty cluster error. Then, you need to reduce it.

Output

- `class`: cluster assignment of each data point in pixels. The assignment should be 1, 2, 3, etc. For $k = 5$, for example, each cell of class should be either 1, 2, 3, 4, or 5. The output should be a column vector with `size(pixels, 1)` elements.
- `centroid`: location of k centroids (or representatives) in your result. With images, each centroid corresponds to the representative color of each cluster. The output should be a matrix with K rows and 3 columns. The range of values should be [0, 255], possibly floating point numbers.

Hand-in

Both of your code and report will be evaluated. Upload them together as a zip file. In your report, answer to the following questions:

1. (20 points) Use k-means with squared-ℓ_2 norm as a metric, for `GeorgiaTech.bmp` and `football.bmp` and also choose a third picture of your own to work on. We recommend size of 320×240 or smaller. Run your k-means implementation with these pictures, with several different $k = 2, 4, 8, 16$. How long does it take to converge for each k (report the number of iterations, as well as actual running time)? Please write in your report, and also include the resulted compressed pictures for each k.
2. (20 points) Run your k-means implementation (with squared-ℓ_2 norm) with different initialization centroids. Please test two initialization strategies, compare the results (output image, running time, iterations) and report: (i) random initialization. Please try multiple time and report the best one (in terms of the image quality). (ii) poor initialization. Please design your own strategy, explain why it qualifies as a poor initialization, try multiple times, and report the results.

   How does this it affect your final result? (We usually randomize initial location of centroids in general.) Please also explain in the report how you initialize the centroid.

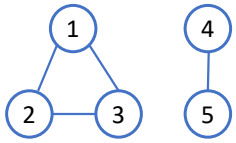
3. (20 points) Now try your k-means with the Manhattan distance (or ℓ_1 distance) and repeat the same steps in Part (1). Please note that the assignment of data point should be based on the Manhattan distance, and the cluster centroid (by minimizing the sum of deviance – as a result o fusing the Manhattan distance) will be taken as the "median" of each cluster. Comment on the difference of image compression results using the two methods.

Note

- You may see some error message about empty clusters when you use too large k. Your implementation should treat this exception as well. That is, do not terminate even if you have an empty cluster, but use smaller number of clusters in that case.
- We recommend you to test your code with several different pictures so that you can detect some problems that might happen occasionally.
- If we detect copy from any other student's code or from the web, you will not be eligible for any credit for the entire homework, not just for the programming part. Also, directly calling built-in functions or from other package functions is not allowed.

2 Spectral clustering and discover football colleague [40 points + 2 bonus]

First consider the following simple graph



1. (10 points) Write down the graph Laplacian matrix and find the eigenvectors associated with the zero eigenvalue. Explain how do you find out the number of disconnected clusters in graph and identify these disconnected clusters using these eigenvectors.

Now consider the football league example in the "spectral clustering" lecture. Use the data provided there (in demo, play_graph.txt, nodes.csv and edges.csv) for this question. Implement the spectral clustering algorithm yourself (you can borrow the idea from demo code, in particular, regarding importing data, read data etc.)

2. (10 points) For the graph Laplacian matrix and perform eigedecomposition on it. Plot the eigenvalues (ranked from the largest to the smallest), and based on the plot explain approximately how many clusters you believe there are and why.
3. (15 points) Now perform spectral clustering, using $k = 5$, $k = 7$, $k = 10$, and your choice of k (based on your answer in Part (2). Report the size of the largest cluster and the smallest cluster based on your result for each k. Report the results for $k = 10$ by listing the teams and which cluster they belong to (you can either include a table in your solution, or upload a file or spreadsheet that include your result).
4. (5 points) Now run the algorithm a few times for $k = 10$. You may notice the results are slightly different - please explain why. Also check which clusters that "Georgia Tech", "Georgia State", and "Georgia" (UGA) are in. Are they always in the same cluster or not - please explain your reasoning.
5. (Bonus, 2 points.) Please report what else you can discover from this data analysis and results. Try to be creative.