ISYE 6740 Fall 2020
Homework 1

1 Clustering [25 points]

Given m data points x^i, $i = 1, \ldots, m$, K-means clustering algorithm groups them into k clusters by minimizing the distortion function over $\{r^{ij}, \mu^j\}$

$$J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} \|\mathrm{x}^i - \mu^j\|^2,$$

where $r^{ij} = 1$ if x^i belongs to the j-th cluster and $r^{ij} = 0$ otherwise.

1. (5 points) Prove (using mathematical arguments) that using the squared Euclidean distance $\|\mathrm{x}^i - \mu^j\|^2$ as the dissimilarity function and minimizing the distortion function, we will have

$$\mu^j = \frac{\sum_i r^{ij} \mathrm{x}^i}{\sum_i r^{ij}}.$$

   That is, μ^j is the center of j-th cluster.

2. (5 points) Discuss qualitatively how the algorithm will change if we replace the similarity function (the squared ℓ_2 distance here) by a general similar measure $d(x, y)$ (a legitimate measure that satisfies the requirements to be a similarity measure).

3. (5 points) Prove (using mathematical arguments) that K-means algorithm converges to a local optimum in finite steps.

4. (10 points) Calculate k-means by hands. Given 5 data points configuration in Figure 1. Assume $k = 2$ and use Manhattan distance (a.k.a. the ℓ_1 distance: given two 2-dimensional points (x_1, y_1) and (x_2, y_2), their distance is $|x_1 - x_2| + |y_1 - y_2|$). Assuming the initialization of centroid as shown, after one iteration of k-means algorithm, answer the following questions.

   (a) Show the cluster assignment;

   (b) Show the location of the new center;

   (c) Will it terminate in one step?

2 Image compression using clustering [40 points]

In this programming assignment, you are going to apply clustering algorithms for image compression. Your task is implementing the clustering parts with two algorithms: *K-means* and *K-medoids*. **It is required you implementing the algorithms yourself rather than calling from a package.**

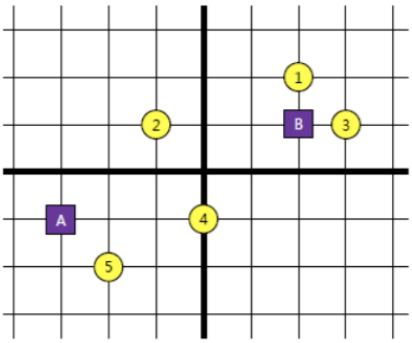

Figure 1: K-means.

K-medoids

In class, we learned that the basic K-means works in Euclidean space for computing distance between data points as well as for updating centroids by arithmetic mean. Sometimes, however, the dataset may work better with other distance measures. It is sometimes even impossible to compute arithmetic mean if a feature is categorical, e.g, gender or nationality of a person. With K-medoids, you choose a representative data point for each cluster instead of computing their average. Please note that K-medoid is different from generalized K-means: Generalized K-means still computes centre of a cluster is not necessarily one of the input data points (it is a point that minimizes the overall distance to all points in a cluster in a chosen distance metric).

Given m data points $\mathrm{x}^i (i = 1, \ldots, m)$, K-medoids clustering algorithm groups them into K clusters by minimizing the distortion function $J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} D(\mathrm{x}^i, \mu^j)$, where $D(\mathrm{x}, \mathrm{y})$ is a distance measure between two vectors x and y in same size (in case of K-means, $D(x, y) = \|\mathrm{x} - \mathrm{y}\|^2$), μ^j is the center of j-th cluster; and $r^{ij} = 1$ if x^n belongs to the k-th cluster and $r^{ij} = 0$ otherwise. In this exercise, we will use the following iterative procedure:

- Initialize the cluster center μ^j, $j = 1, ..., k$.
- Iterate until convergence:
  - Update the cluster assignments for every data point x^i: $r^{ij} = 1$ if $j = \arg\min_j D(\mathrm{x}^i, \mu^j)$, and $r^{ij} = 0$ otherwise.
  - Update the center for each cluster j: choosing another representative if necessary.

There can be many options to implement the procedure; for example, you can try many distance measures in addition to Euclidean distance, and also you can be creative for deciding a better representative of each cluster. We will not restrict these choices in this assignment. You are encouraged to try many distance measures as well as way of choosing representatives (e.g., ℓ_1 norm).

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. (10 points) 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.
2. (10 points) Attach a picture of your own. We recommend size of 320×240 or smaller. Run your k-medoids implementation with the picture you chose, as well as two pictures provided (beach.bmp and football.bmp), 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? Please write in your report.
3. (10 points) 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.) Please write in your report.
4. (10 points) 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? Please write in your report.

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 will grade using test pictures which are not provided. 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.

3 Political blogs dataset [35 points]

We will study a political blogs dataset first compiled for the paper Lada A. Adamic and Natalie Glance, "The political blogosphere and the 2004 US Election", in Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem (2005). The dataset nodes.txt contains a graph with $n = 1490$ vertices ("nodes")

corresponding to political blogs. Each vertex has a 0-1 label (in the 3rd column) corresponding to the political orientation of that blog. We will consider this as the true label and try to reconstruct the true label from the graph using the spectral clustering on the graph. The dataset edges.txt contains edges between the vertices. You may remove isolated nodes (nodes that are not connected any other nodes).

1. (5 points) Assume the number of clusters in the graph is k. Explain the meaning of k here intuitively.

2. (10 points) Use spectral clustering to find the $k = 2$, $k = 3$, and $k = 4$ clusters in the network of political blogs (each node is a blog, and their edges are defined in the file edges.txt). Then report the majority labels in each cluster, when $k = 2, 3, 4$, respectively. For example, if there are $k = 2$ clusters, and their labels are $\{0, 1, 1, 1\}$ and $\{0, 0, 1\}$ then the majority label for the first cluster is 1 and for the second cluster is 0. **It is required you implementing the algorithms yourself rather than calling from a package.**

3. (5 points) Now compare the majority label with the individual labels in each cluster, and report the *mismatch rate* for each cluster, when $k = 2, 3, 4$. For instance, in the example above, the mismatch rate for the first cluster is $1/4$ (only the first node differs from the majority) and the the second cluster is $1/3$.

4. (10 points) Now tune your k and find the number of clusters to achieve a reasonably small *mismatch rate*. Please explain how you tune k and what is the achieved mismatch rate.

5. (5 points) Please explain the finding and what can you learn from this data analysis (e.g., node within same community tend to share the same political view, or now? Did you find blogs that share the same political view more tightly connected than otherwise?)