MATH-BIOINF-STATS 547: Mathematics of Data
Due Date: 02/23/2023
Name:
Collaborators:

Problem Set 3: Clustering

For this problem set, please submit a `.pdf` document with a write-up of your results and observations. We encourage using Overleaf, but the MATLAB Live Editor or other word-processing software is acceptable. We have provided a LaTeX template to help get you started, which is available on Overleaf where you can make a copy of it.

Exercise 1 ($k-$Means)**.** Implement the $k-$Means Clustering Algorithm in any language of your choice according to the following pseduocode. Once implemented, apply your algorithm on the Fisher-Iris data set and visualize the results. Include your plots, and answer the below questions.

The $k-$Means Algorithm

We assume that we have n data points $(a_{ij})_{j=1}^{n} \in \mathbb{R}^m$, which we organize as columns in a matrix $A \in \mathbb{R}^{m\times n}$. Let $\Pi = (\pi_i)_{j=1}^{k}$ denote a partitioning of the vectors $a_1, a_2, ..., a_n$ into k clusters: $\pi_j = \{\nu \mid a_\nu \text{ belongs to cluster } j\}$. Let the mean of cluster j be

$$\mu_j = \frac{1}{n_j} \sum_{\nu\in\pi_j} a_\nu, \tag{1}$$

where n_j is the number of elements in π_j. The **centroid** is the data point c_j with minimum distance to the cluster mean

$$c_j = \min_{a_\nu | \nu\in\pi_j} \mu_j - a_\nu,$$

with respect to a valid metric. The $k-$Means Algorithm iteratively assigns data points to the cluster of the nearest centroid and then updates the centroid of each cluster based on equation 1. This is described with the pseudo-code:

Algorithm 1 k-Means

Require: Data matrix A with rows a_i, k the number of clusters
1: Randomly select k data points from A to start as cluster centroids $c_1, \ldots, c_k$
2: **do**
3:     **for** $a_i \in A$ **do** $\triangleright$ Assign data to clusters
4:         assign a_i to π_j, where c_j is the closest centroid to a_i
5:     **end for**
6:     **for** $j \in 1, \ldots, k$ **do** $\triangleright$ Update cluster centroids
7:         $\mu_j =$ the average of all $a_i \in \pi_j$
8:         $c_j = a_i$ such that a_i is closest to μ_j
9:     **end for**
10: **while** any c_j updated or a_i was assigned to a new cluster in the last iteration
11: **return** π_i and c_i

Lines 4 and 8 of the $k-$Means algorithm rely on the notion of closeness or distance between data points. This may be computed with standard Euclidean distance but often other metrics or kernel functions are used to measure the closeness of data for $k-$Means. Because $k-$Means may utilize a variety of distance metrics, it is a very popular, effective algorithm.

Data: Fisher's Iris data [1] is a classical data set used for introductory clustering and supervised learning problems. The data contain 4 measurements (length and width measurements on the sepals and petals) for three (i.e. $k = 3$) similar types of flowers. Two types of the flowers are linearly inseparable from one another while the third set of samples can be separated. This data set is built into MATLAB and may be loaded with the command "load fisheriris".

(a) What are the general properties of data that can be clustered well with $k-$Means?

(b) What are potential drawbacks of your implementation of $k-$Means, and how could the algorithm be improved to circumvent these issues?

(c) How can you evaluate the quality of the clusters of your algorithm?

(d) How do you deal with the random initialization of $k-$Means?

Solution:

Exercise 2 (Spectral Clustering)**.** Three variations of spectral clustering algorithms are described in "A tutorial on spectral clustering" (page 399) [2]. The first algorithm is what Luxburg refers to as "Unnormalized spectral clustering", the second is "Normalized spectral clustering according to Shi and Malik", and the third is "Normalized spectral clustering according to Ng *et al*". The version of algorithm 3 shown below is an adapted form of Andrew Ng's normalized spectral clustering algorithm described in "On spectral clustering: Analysis and an algorithm" [3] which may be helpful for your understanding.

Each of the three algorithms has been implemented in the MATLAB starter code. In this question, each of the three versions of spectral clustering are applied to the same data set in order to compare the quality and computational performance of each algorithm.

Data: The matlab variable `data_mat`: a 777×777 weighted adjacency matrix derived from Hi-C data. To ensure the matrix is connected, rows and columns where more than 10% of the entries were zeros were removed from the matrix. See figures 1 and 2 for examples of how the data and clusters are visualized for this question.

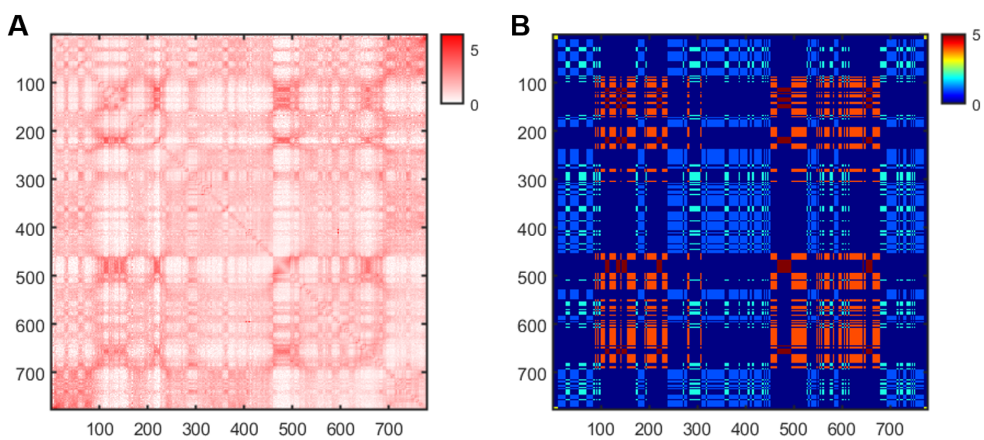

Figure 1: Weighted adjacency matrix and spectral clustering. (A) Log_2 transformed weighted adjacency matrix plotted using `imagesc`. (B) Ng clustering algorithm results with $k = 5$, plotted using `imagesc`.

(a) Set a k value in the starter code and run each section of code for that question.

(b) Plot the results of each clustering algorithm (see Figures 1 and 2 for examples).

(c) Which algorithm performed the best on this data? Which algorithm was the fastest?

(d) Repeat steps (a)-(c) with different values of k. What value of k do you think worked best and why?

Solution:

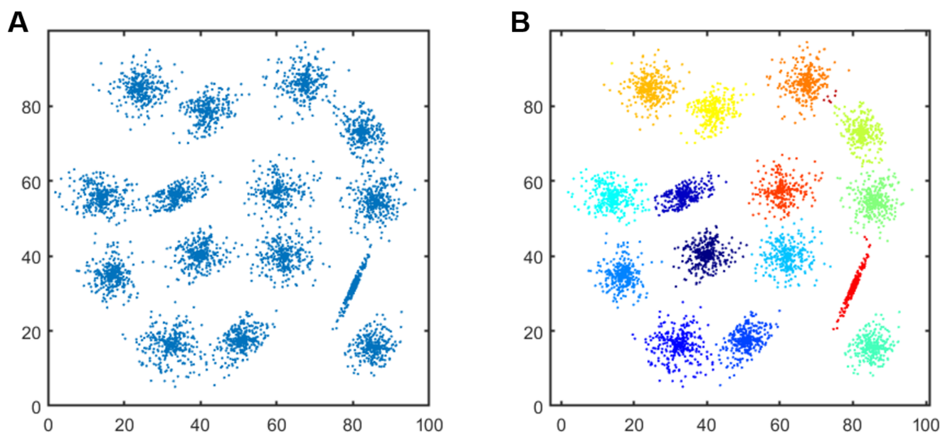

Figure 2: Data and spectral clustering. (A) 5,000 points plotted using `scatter`. (B) Ng clustering algorithm results with $k = 16$, plotted using `gscatter`.

Exercise 3 (Propose a Clustering Algorithm)**.** $k-$Means is the most fundamental clustering algorithm, and the significance of spectral clustering is motivated by our understanding of the singular values and spectra of the data, but there are an endless number of additional clustering algorithms. For an overview of, see the scipy clustering library overview page. Based on your understanding of data, the clustering algorithms we have seen, and any other resources, propose and implement a new clustering algorithm.

Data: Obtain a data set of personal interest to apply your proposed clustering algorithm. The data set should contain a sufficient number of samples ($n \geq 100$) and features per sample.

(a) Include a copy of the code for your implemented algorithm.

(b) Discuss the motivation of how you developed this algorithm, and how your algorithm compare to $K-$Means and spectral clustering?

(c) Discuss the data set you chose, why you chose it, and provide a reference (in the form of a citation or link) to your data. Apply your algorithm on the data discussed above.

(d) How do the output clusters of your algorithm compare to the other methods of clustering?

(e) How does the speed of your algorithm compare to the other methods of clustering?

(f) Interpret the output clusters in the context of the domain your data come from.

Solution:

References

[1] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[2] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[3] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.