Coding Assignment 5

Due Date: *Monday, December 8 (11.59PM)*

Submission: on Gradescope

Instructions:

The coding assignments may be completed in groups up to 4 students. If you do so, please make sure that you include everyone's full name, and that you *also select everyone's name when submitting the assignment on Gradescope*. This ensures that each group member will get a grade assigned and have access to the comments from the graders.

For this assignment, the following items need to be submitted:
(1) the **code** in R or *Python* (a Markdown file is ok; a Jupyter notebook is ok)
(2) a **pdf** file with your code and results with all necessary plots and comments.

Problem 1 (50 points)

In this problem we are going to work with the Mall Customers data set from Kaggle – you can download the data directly from Kaggle. This data set contains information about the Annual Income, Spending Score, Age, and Gender of 200 customers. Our **goal** is to *cluster* customers into distinct groups based on spending patterns and demographics.

Use one-hot-encoding to convert the categorical variable Age to numerical. Also, do not forget to scale (or standardize) the features, since the underlying Euclidean metric in the k-Means algorithm is sensitive to that.

(a) Use K-**Means Clustering** to assign each customer to a cluster:

i. Use the **Gap Statistic** and the **Silhouette Statistic** to determine the optimal number of clusters k_{gap} and k_{sil}. Fit the K-means algorithm to the data for both ks. *Note:* If you get the same k that is ok! You have more evidence that this is probably the appropriate number of clusters.

ii. Plot the Annual Income vs. Spending Score and color code the clusters for both ks . Comment on the separability of the clusters in each case.

iii. Interpret the clusters. For example, you can compute summary statistics for each cluster to understand the customers in each cluster (for each k), e.g. comment on the annual income level, spending score or age. What are your observations?

(b) Use **Hierarchical Clustering** to assign a cluster to each customer:

i. Plot the dendrogram for the data.

ii. Repeat questions (a - ii) and (b - iii) for the results of hierarchical clustering and comment on the differences and/or similarities with what you got in part (a).

Problem 2 (50 points)

In this problem we are going to create a simulated data set to illustrate the use of spectral clustering. In the literature, what we are going to simulate is known as the "Two Moons" data. Mathematically, each half moon is nothing by a half semi-circle of radius r that can be parametrized using polar coordinates as

$$x = rcos\theta, \;\; y = r\sin\theta$$

if it is the upper half, and

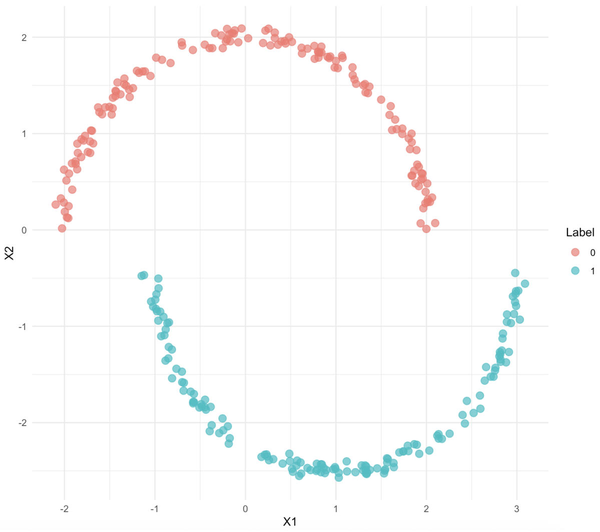
$$x' = dx - rcos\theta, \;\; y' = -r\sin\theta - dy$$

the lower half slightly shifted in each coordinate by dx and dy (so that the two half circles do not overlap. Here $\theta \in [0, \pi]$ and r is the radius.

Create the *two moons* synthetic data of size $n = 200$ as follows:

1. For $r = 2$, sample $\theta_i^1 \sim Uniform[0, \pi]$, and compute $x_i = rcos\theta_i^1$ and $y_i = r\sin\theta_i^1$. Add some noise to each coordinate, e.g. add a normal random variable with mean 0 and variance 0.05. **Assign the label '0' to this data**.
2. For $r = 2$, sample $\theta_i^2 \sim Uniform[0, \pi]$, and compute $x_i' = 1 - 2cos\theta_i^2$ and $y_i' = -2\sin\theta_i^2 - 0.5$. Add some noise to each coordinate, e.g. add a normal random variable with mean 0 and variance 0.05. **Assign the label '1' to this data.**
3. Combine everything into a data frame with 3 columns: the coordinates and the label. This is your data set!
4. Plot it!

If you use $n = 200$ and $Seed = 598$ (in R), you should get something similar to the following:



Note: Python has a function that creates this synthetic data automatically, but you will loose all the fun! *It is ok to use it, but make sure that you plot your synthetic data.*

Now, let's apply clustering techniques to this data set!

(a) Use **K-Means Clustering** to assign each point to a cluster (use $k = 2$). Plot the results by color-coding the data by cluster assignment. Comment on the results.

(b) Use **Spectral Clustering** to the same data set (use $k = 2$). Plot the results by color-coding the data by cluster assignment. Comment on the results and compare with part (a) above.
Note: Here, you are free to use the built-in ‘R‘ (`specc` in `kernlab`) or ‘Python‘ (`SpectralClustering` in `sklearn.cluster`) functions. In both cases, for you can use $k = 2$ and the Gaussian/RBF kernel.