Please write a report for this project.

Problem 1: Graph Laplacian. There are two normalized versions of the graph Laplacian, a symmetric one and a non-symmetric one, given by

$$L_S = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{1}$$

$$L_N = D^{-1} L = I - D^{-1} W. \tag{2}$$

Prove the following results:

1. For every vector $f \in \mathbb{R}^n$ there holds

$$f^* L_S f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \Big(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \Big)^2. \tag{3}$$

2. λ is an eigenvalue of L_N with eigenvector u if and only if λ is an eigenvalue of L_S with eigenvector $w = D^{\frac{1}{2}} u$.

3. λ is an eigenvalue of L_N with eigenvector u if and only if λ and u solve the generalized eigenproblem $Lu = \lambda D u$.

4. 0 is an eigenvalue of L_N and the associated eigenvector is $\mathbf{1}$. 0 is an eigenvalue of L_S and the associated eigenvector is $D^{\frac{1}{2}} \mathbf{1}$.

Problem 2: k-means implementation. Implement your own k-means clustering with the following detailed specifications:

- The function has as input an $n \times d$ data matrix X and a number $1 < k < n$ of clusters.
- The function returns an $n \times 1$ vector C (with $1 \leq C[i] \leq k, i = 1, \cdots, n$) that contains the cluster number for each data point, and a $k \times d$ matrix M with the cluster centers as its rows.
- The flag `'distance'`, in the function argument, specifies the method used to measure the similarity of two measurements. Your algorithm should be able to accommodate two different choices of 'distance', the Euclidean distance and the cosine distance.
- Your implementation should initialize the centers by choosing k random data points, and should iteration until a proper stopping criterion of your choice is fulfilled. (Since we solve a non-convex optimization problem with potentially many local minima, you may need to find the best solution over multiple random initializations.)

Problem 3: How many clusters? For the dataset `kmeansgaussian.mat` in the zip file. There are 500 data points in $\mathbb{R}^4$. The challenge is that you do not know how many clusters are contained in this data set. Derive a strategy to determine the optimal number of clusters (that is, the optimal choice of k) for this data set with the code your developed in problem

2 using the Euclidean distance. What seems to be the best number of clusters in this case? Why?

Problem 4: Clustering the Fisher iris data set. In the 1920's, botanists collected measurements from three different species (setosa, versicolor, virginica) of the iris. They measured the sepal length and width, and the petal length and width of each flower. Thus per flower we have four measurements. The data set consists of measurements from 150 irises. The measurements became known as Fisher's iris data. In this problem we try to answer the question: Is it possible to define the specie of an iris based on these four measurements? Apply the code you developed in Problem 2 to cluster the data in `'iris.data.txt'` with both distances (Euclidean and cosine). Since the data are labeled so that we know which species each iris belongs to, we can verify if the data have been clustered correctly. Measure the performance of the clustering/classificiation by computing the number of misclustered data versus the total number of data, which gives a number between 0 (perfect) and 1 (all wrong). What is the best performance you can obtain?