DASC 521-Homework 08: Spectral Clustering

Ender Erkaya

December 2021

1 Introduction

In this homework, we demonstrate spectral clustering algorithm for $D = 2$ dimensional data set. First, we create a connectivity matrix. Then, using it we create a graph Laplacian matrix whose diagonals represent masses of their corresponding nodes and each $L_{ij} = \{-1\}$ value represents an outward connection from node i to j. After symmetric normalization, we create Z matrix using eigenvectors corresponding $R = 5$ smallest eigenvalues. Then, k-means algorithm is run through Z matrix and we visualize memberships and centroids on data X.

2 Visualized Raw Data

The raw unlabeled data is given below as figure 1:

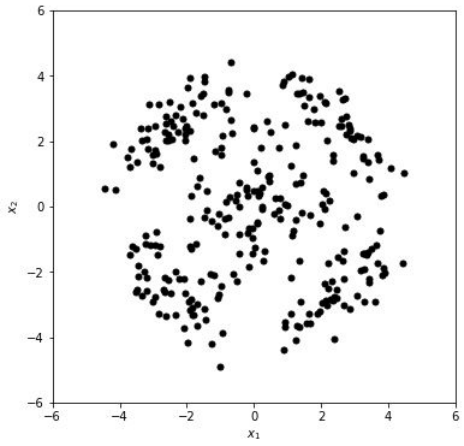

Figure 1: Unlabeled Data

3 Connectivity Matrix

First, a square pairwise distance matrix is calculated using "squareform" and "pdist" functions of python. Then, pairwise distance matrix is binarized to create connectivity matrix as follows:

$$B(D < \delta) = 1$$
$$B(D \geq \delta) = 0$$
$$B(diag(B)) = 0$$

The created matrices are visualized as follows in figures 2, 3:

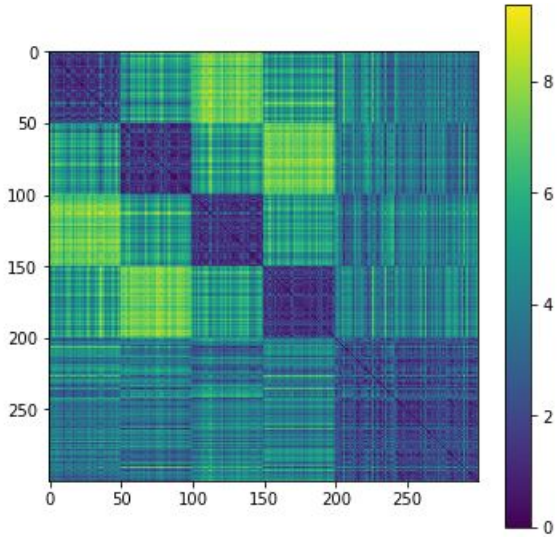

Figure 2: Distance Matrix

The connectivity between observations is visualized in figure 4:

4 Graph Laplacian and Z Matrix

Constructing a graph Laplacian matrix L as

$$L = D - B$$

and using symmetric normalization yields:

$$L_s = D^{\frac{-1}{2}} L D^{\frac{-1}{2}} = I - D^{\frac{-1}{2}} B D^{\frac{-1}{2}}$$

We construct Z matrix using eigenvectors of L_s corresponding R smallest eigenvalues. Hence, eigenvector decomposition is applied. After eigenvector composition, we need to find R smallest eigenvalue index. To achieve this, sorting and taking first indices can be applied. To avoid sorting complexity, a recursive minimum is applied for $R = 5$ steps.

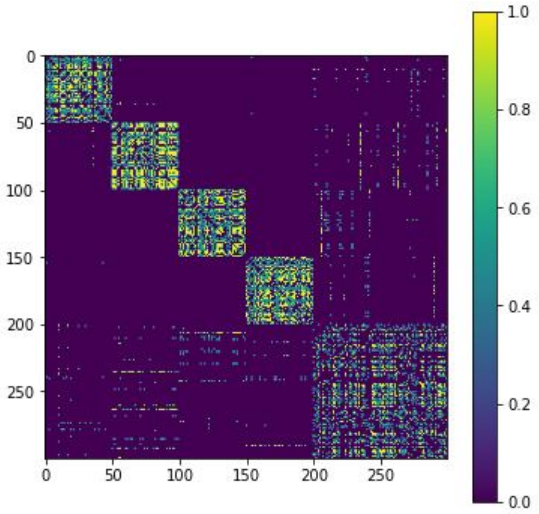

Figure 3: Connectivity Matrix

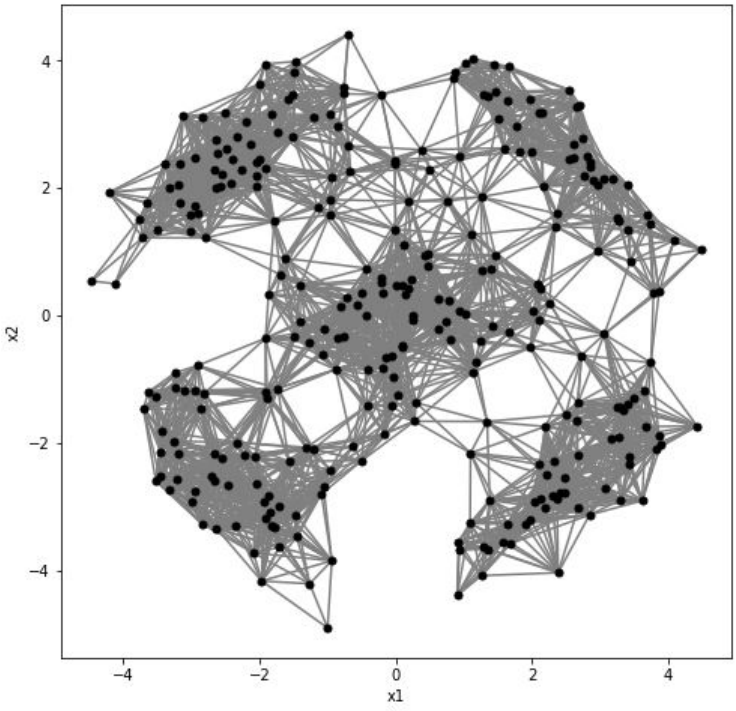

Figure 4: Connectivity Visual

5 K-means Algorithm on Z

After obtaining Z matrix, we use it as a new feature space and run k-means algorithm on Z. As initialization of centroids, we used rows $[[28, 142, 203, 270, 276], :]$ of Z. Alternatively, updating centroids and memberships, we converge a solution of $K = 5$ clusters. Memberships and centroids in each step is visualized as below:

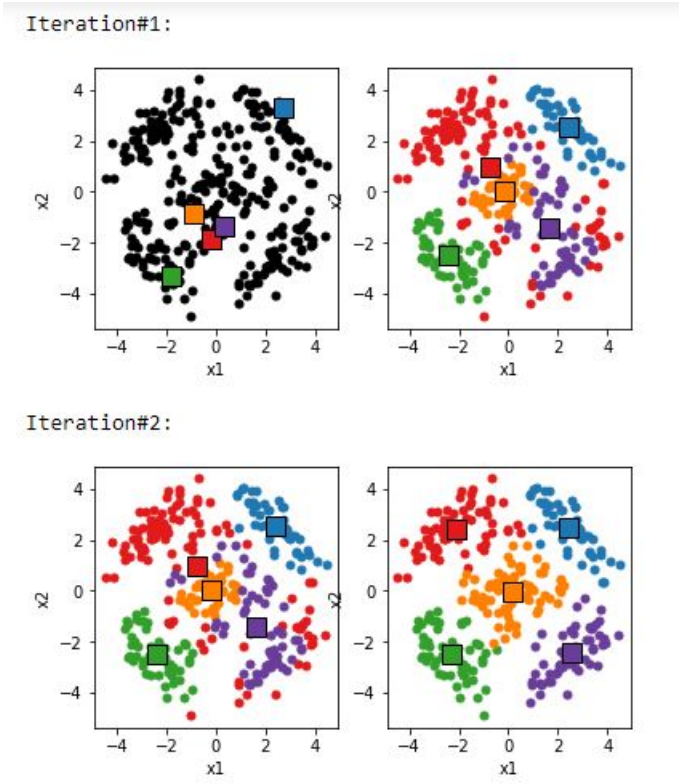

Figure 5: Iterations 1-2

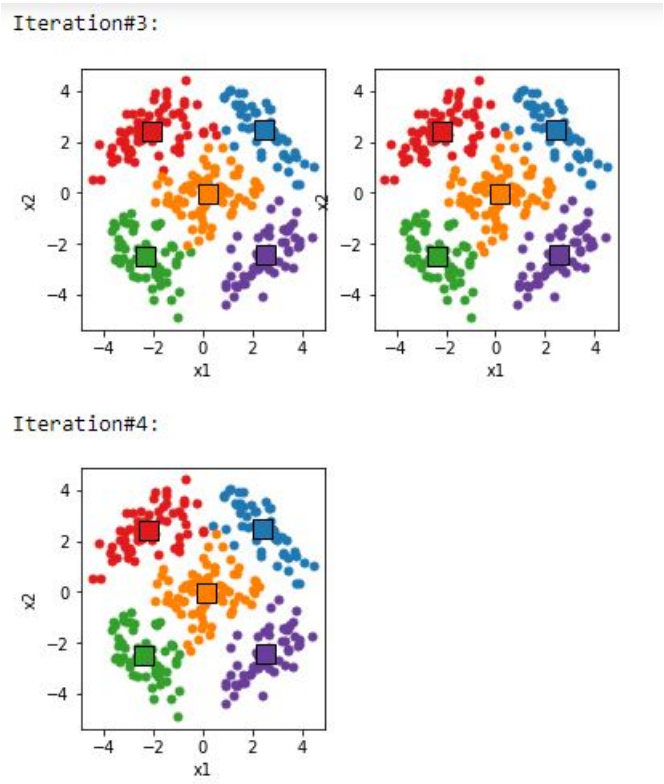

Figure 6: Iterations 3-4

6 Final Clusters Visualized

After convergence, the final clusters and centroids are visualized as below:

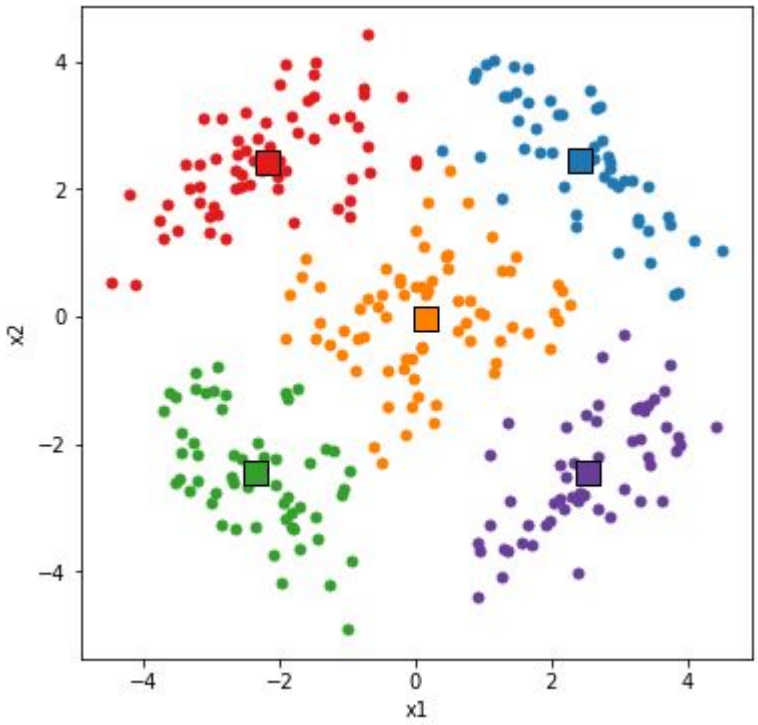

Figure 7: Final Clusters

7 Conclusion

Spectral clustering algorithm is implemented via kmeans algorithm using a created feature matrix on Laplacian matrix. The algorithm's performance depends on learning parameter δ. As δ is reduced, the algorithm learns more local connections, skipping the bigger picture, ie overfitting. If δ is increased much, the algorithm capacity discriminating local connections reduces. Hence, the connectivity threshold δ should be carefully chosen. If it is too low, the normalization results error due to zero weight on diagonal, ie singular data points. While implementation, it is observed that convergence clusters of the algorithm is highly dependent on the selection of initial choice. If we would choice other rows, the final clusters become different for this data set. This dependence of initialization is a property of k-means algorithm. It may be further developed to use another clustering algorithm instead of k-means algorithm.