Report 4: Graph Spectra

Thilini Yashodha, Hieu Vu - Group 41

November 29, 2025

1 Introduction

This assignment focuses on learning graph spectra clustering using the algorithm described in the paper "On Spectral Clustering: Analysis and an algorithm by Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. In this work, we construct the adjacency matrix, compute the normalised Laplacian, analyse eigenvalues to determine the number of clusters using the eigengap heuristic, run k-means on the spectral embedding, and visualise the clustering results.

2 Main problems and solutions

Building the adjacency matrix

- Edges were loaded from the datasets and constructed an undirected adjacency matrix
- $A_{ij} = 1$ if edge (i,j) exists

Computing the normalized Laplacian

- Computed using $L = I - D^{-1/2} A D^{-1/2}$
- Where
  - A is the adjacency matrix
  - D is the degree matrix

Eigen decomposition

- We computed all eigenvalues and eigenvectors of L.
- Eigenvalues were sorted in increasing order to apply the eigengap rule

Determining the number of clusters (K)

To identify the number of clusters K, we plotted the sorted eigenvalues of the normalized Laplacian. In this figure, the small eigenvalues appear first, followed by a sudden jump known as the eigengap. The number of eigenvalues close to zero before this jump indicates the number of communities in the graph. For example, example1 shows four small eigenvalues before the gap (K=4), while example2 shows two (K=2).

Normalize rows

Let X be the matrix of the first K eigenvectors. We renormalize each row of X to have unit length using,

$$Y_{ij} = X_{ij} / (\sum_j X_{ij}^{\ 2})^{1/2}$$

The matrix Y forms the low dimensional embedding used for clustering.

K-means clustering

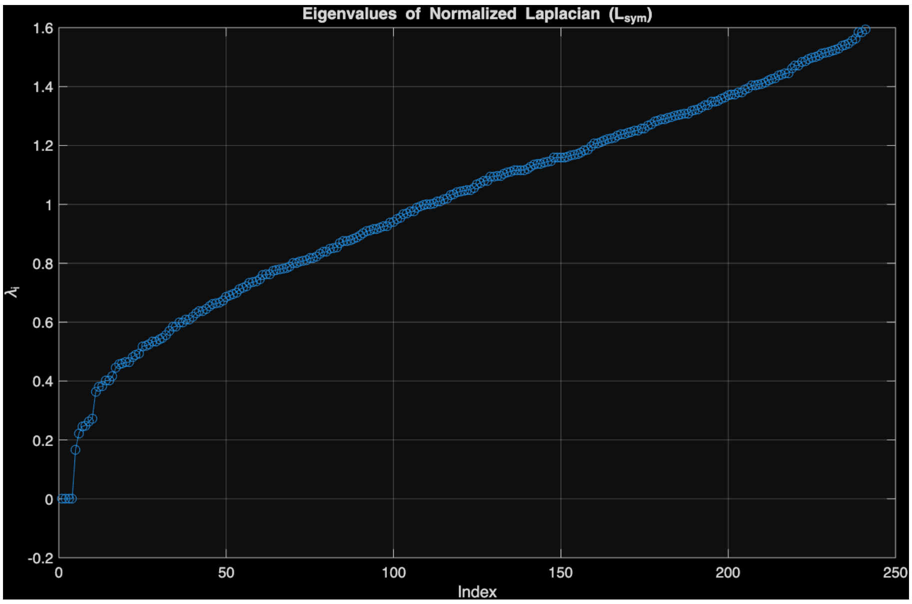
After computing the Y_{ij}, we apply k-means with K clusters to group nodes based on their positions in the low dimensional space. Because the eigenvectors separate communities geometrically, k-means can clearly assign each node to its corresponding cluster. The resulting labels are then used to visualize the final community structure.

3 Evaluation

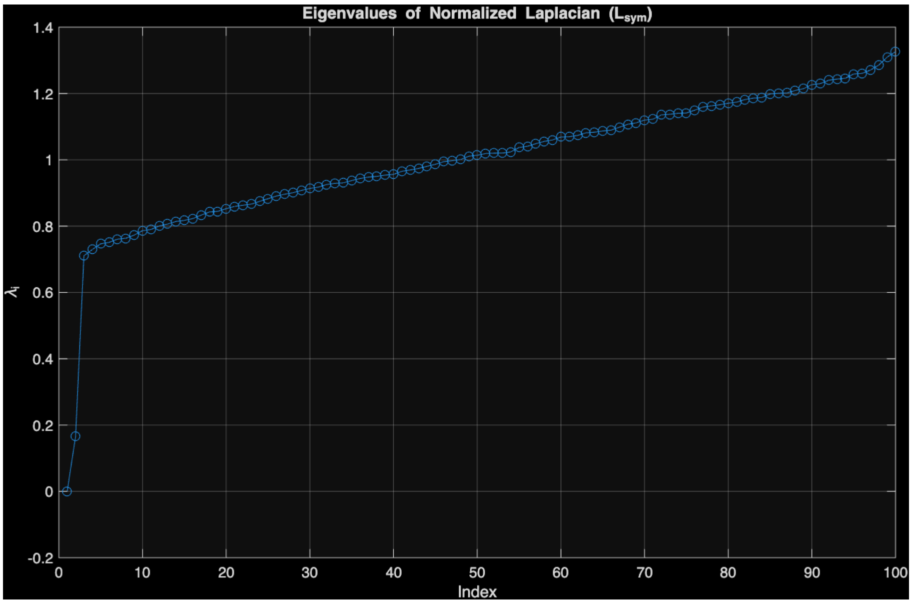
Sorted Eigenvalues

The paper assumes that the number of clusters K is provided as an input parameter. Since the paper does not specify how to choose K, in this assignment we follow the standard eigengap heuristic used in modern spectral clustering literature. We plot the eigenvalues of the normalized Laplacian in ascending order and identify the largest gap, the number of eigenvalues close to zero before this jump indicates the appropriate number of clusters. Selected K =4 for dataset 1 and K = 2 for dataset 2.

This figure plots the sorted eigenvalues of the normalized Laplacian L in ascending order. Each point on the graph corresponds to one eigenvalue λi, and the x axis represents the index after sorting. Small eigenvalues near zero appear at the left, while larger eigenvalues appear toward the right.



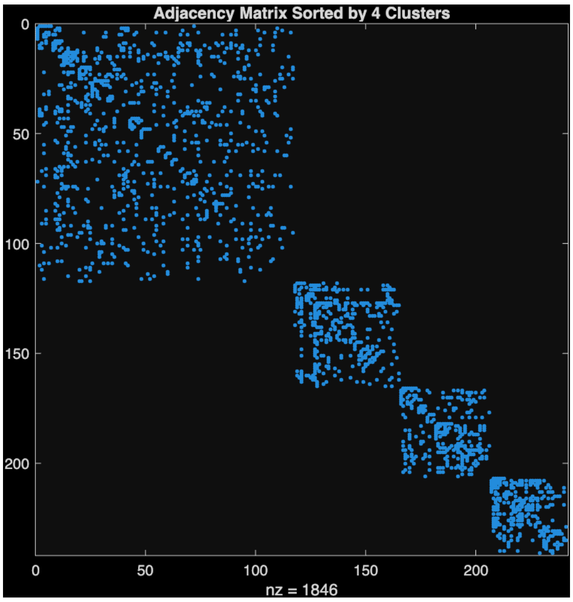
Dataset 1



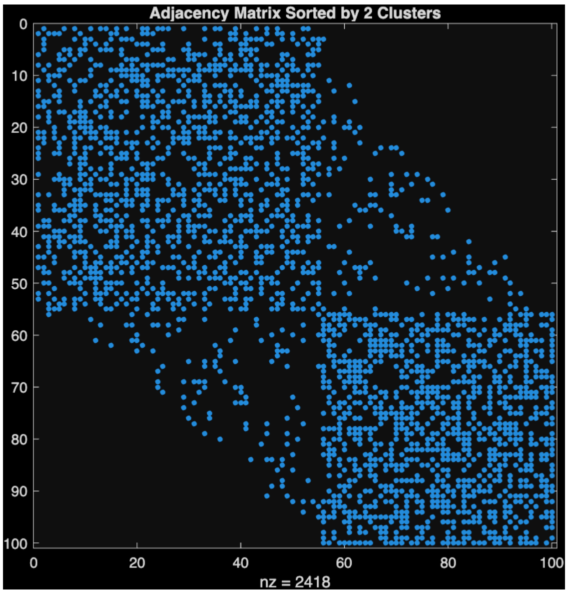
Dataset 2

Sparsity Pattern

The sparsity pattern visualises the structure of the adjacency matrix by marking only the non-zero entries. Each dot represents an edge between two nodes. Dense blocks indicate tightly connected groups (communities), while empty regions correspond to weak or missing connections. Plotting the sparsity pattern helps reveal the underlying community structure even before clustering.



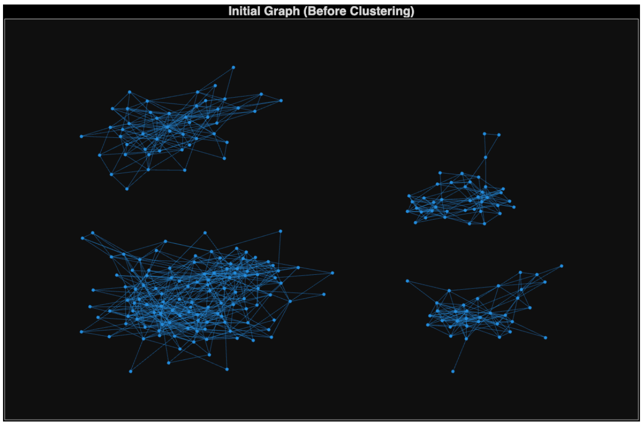
Dataset 1



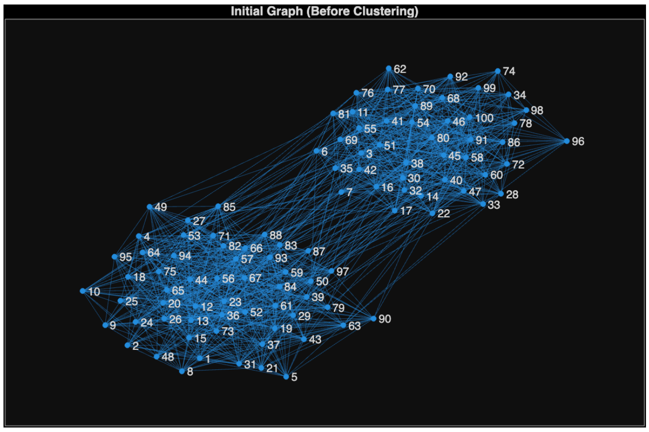
Dataset 2

Initial Graph

In order to identify the accuracy of the above graphs, we plot the raw network structure before applying spectral clustering. All nodes are displayed using a force directed layout, and no community information is used. The plot reveals the overall connectivity pattern of the graph but does not visually separate groups or clusters.



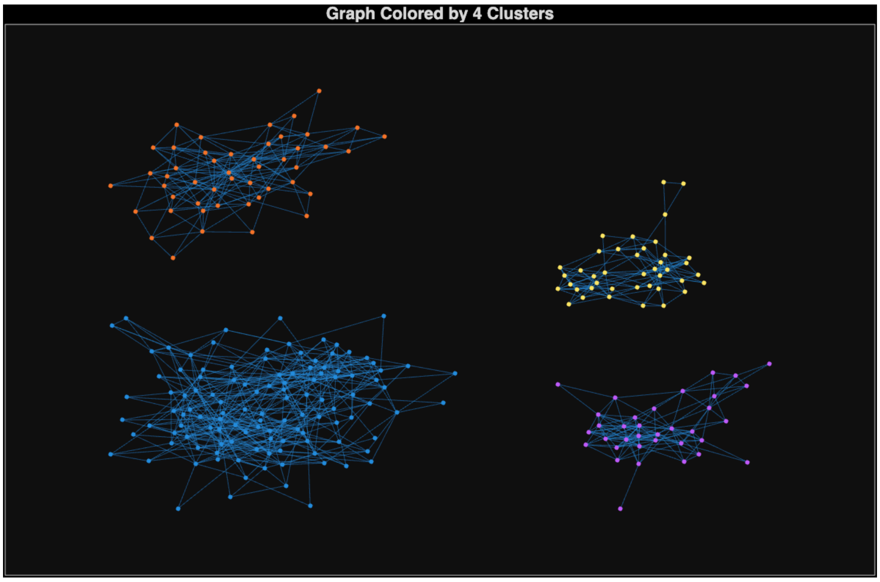
Dataset 1



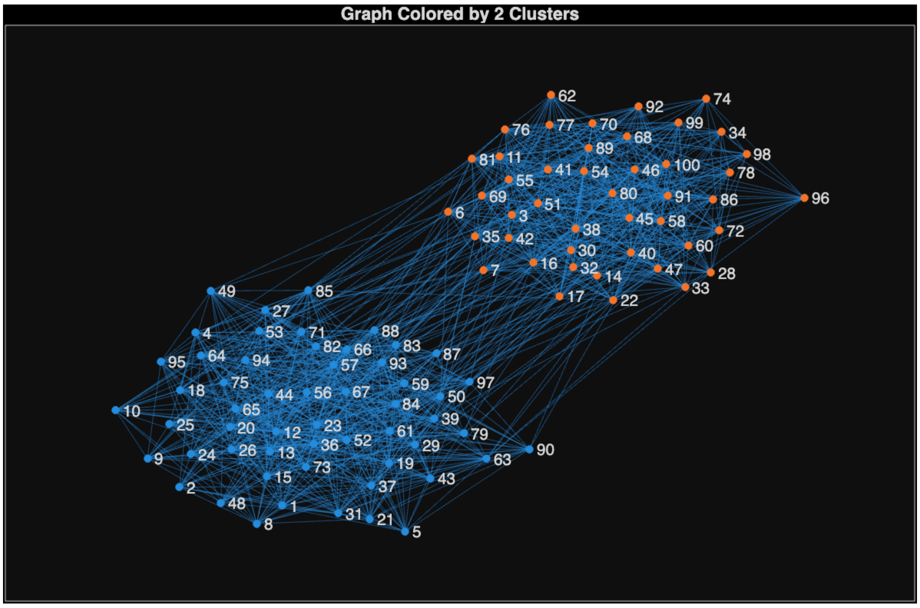
Dataset 2

Clustering

This figure visualises the same graph after applying spectral clustering. Nodes are colored according to the cluster assignments produced by the K-means step of the NJW algorithm. Distinct colour groups represent different communities, clearly showing how the algorithm partitions the network into meaningful clusters.



Dataset 1



Dataset 2

Conclusions

In this assignment, we implemented the spectral clustering algorithm and applied it to both a real world network and a synthetic graph. By constructing the normalized Laplacian, analyzing its eigenvalues, and using the eigengap heuristic, we identified the appropriate number of clusters for each dataset. The K means step successfully separated nodes into well defined communities, which was clearly visible in the cluster colored graph visualizations. Overall, the results demonstrate how spectral embedding reveals community structure that is not easily observable from the raw graph alone.