Report 4: Graph Spectra

Group 150: Lorenzo Deflorian, Riccardo Fragale

November 30, 2025

1 Introduction

The goal of this homework is to implement the spectral graph clustering algorithm described in this paper `https://ai.stanford.edu/~ang/papers/nips01-spectral.pdf` and use it to analyze two given graphs. Spectral clustering is a powerful technique that leverages the eigenvalues and eigenvectors of the graph Laplacian matrix to identify natural clusters in the graph structure.

2 Implementation

The implementation is straightforward and can be found in the `miner/core/spectra/cluster_machine.py` module.

Given the graph represented by its adjacency matrix, the steps to follow are:

1. Remove loops from the graph by subtracting the diagonal of the graph from itself

2. Build the degree matrix D

3. Build the Laplacian matrix $L = D - A$

4. Compute the eigenvalues and eigenvectors of the Laplacian matrix L

5. Take the first k eigenvectors corresponding to the k largest eigenvalues to build the feature matrix X

6. Form the normalized feature matrix Y by normalizing each row to have unit length

7. Cluster the data using the k-means algorithm

The degree matrix D is a diagonal matrix where the element (i, i) is the degree of the i-th node, indicating how many edges are connected to that node.

The Laplacian matrix L is a symmetric matrix defined as $L = D - A$, where A is the adjacency matrix of the graph. It is used to compute the eigenvalues and eigenvectors of the graph structure. The idea behind the Laplacian matrix is to measure how far each node is from the other nodes. The eigenvectors of the Laplacian matrix capture the structure of the graph, with the first few eigenvectors encoding the most significant structural information about clusters.

Once we have the feature matrix X, we form the normalized feature matrix Y by normalizing each row to have unit length. Finally, the k-means algorithm is applied to cluster the data into k clusters based on the normalized feature vectors.

3   Experimental Results

One of the first questions we can ask is: what is the optimal number of clusters to use? To answer this question, we can check the eigenvalues of the Laplacian matrix and compare the gaps between them. The idea is that the larger the gap, the more distinct the clusters are.

The implementation of this analysis is straightforward and can be found in the `miner/core/spectra/gap_finder.py` module.

3.1   Example 1: Optimal Cluster Selection

Let's take a look at the eigenvalues of the Laplacian matrix for example 1.

Table 1 shows the gaps between consecutive eigenvalues of the Laplacian matrix. The largest gap is between λ_4 and λ_5 with a value of 0.1942, which suggests that $k = 4$ might be an appropriate number of clusters.

Table 1: Gaps between consecutive eigenvalues of the Laplacian matrix

Eigenvalue Pair	Gap Value
$\lambda_1 - \lambda_2$	0.0000
$\lambda_2 - \lambda_3$	0.0000
$\lambda_3 - \lambda_4$	0.0000
$\lambda_4 - \lambda_5$	0.1942
$\lambda_5 - \lambda_6$	0.0012
$\lambda_6 - \lambda_7$	0.0483
$\lambda_7 - \lambda_8$	0.0093
$\lambda_8 - \lambda_9$	0.0043
$\lambda_9 - \lambda_{10}$	0.0044
$\lambda_{10} - \lambda_{11}$	0.0590
$\lambda_{11} - \lambda_{12}$	0.0030
$\lambda_{12} - \lambda_{13}$	0.0173
$\lambda_{13} - \lambda_{14}$	0.0038
$\lambda_{14} - \lambda_{15}$	0.0275

Let's plot the clusters for $k = 4$ to visualize the results.

Graph Structure with Clusters (layout: spring)

Figure 1: Clusters for $k = 4$

3.1.1 Cluster Quality Evaluation

To validate our choice of $k = 4$, we compute the inter-cluster edges count, the intra-cluster edges count, the expansion ratio, and the conductance for different values of k to verify that we found the optimal number of clusters.

Table 2 shows the cluster quality metrics for different values of k. The best results are achieved when $k = 4$, which matches our earlier analysis based on eigenvalue gaps. For $k = 4$, we have zero inter-cluster edges and all edges are intra-cluster, resulting in perfect expansion ratio and conductance values of 0.00, indicating ideal cluster separation.

Table 2: Cluster quality metrics for different values of k

k	Inter-edges	Intra-edges	expansion ratio	conductance
2	0	923	0.00	0.00
3	0	923	0.00	0.00
4	0	923	0.00	0.00
5	12	911	0.71	0.41
6	35	888	0.71	0.32
7	126	797	0.71	0.50
8	163	760	0.93	0.48
9	191	732	1.61	0.62

3.2 Example 2: Optimal Cluster Selection

Now let's analyze example 2.

Table 3 shows the gaps between consecutive eigenvalues of the Laplacian matrix for example 2. The largest gap is between λ_2 and λ_3 with a value of 0.5451, which suggests that $k = 2$ might be an appropriate number of clusters.

Table 3: Gaps between consecutive eigenvalues of the Laplacian matrix for example 2

Eigenvalue Pair	Gap Value
$\lambda_1 - \lambda_2$	0.1656
$\lambda_2 - \lambda_3$	0.5451
$\lambda_3 - \lambda_4$	0.0187
$\lambda_4 - \lambda_5$	0.0173
$\lambda_5 - \lambda_6$	0.0042
$\lambda_6 - \lambda_7$	0.0098
$\lambda_7 - \lambda_8$	0.0028
$\lambda_8 - \lambda_9$	0.0094
$\lambda_9 - \lambda_{10}$	0.0136
$\lambda_{10} - \lambda_{11}$	0.0034
$\lambda_{11} - \lambda_{12}$	0.0106
$\lambda_{12} - \lambda_{13}$	0.0058
$\lambda_{13} - \lambda_{14}$	0.0078
$\lambda_{14} - \lambda_{15}$	0.0033

Plotting the clusters for $k = 2$ yields the following result:

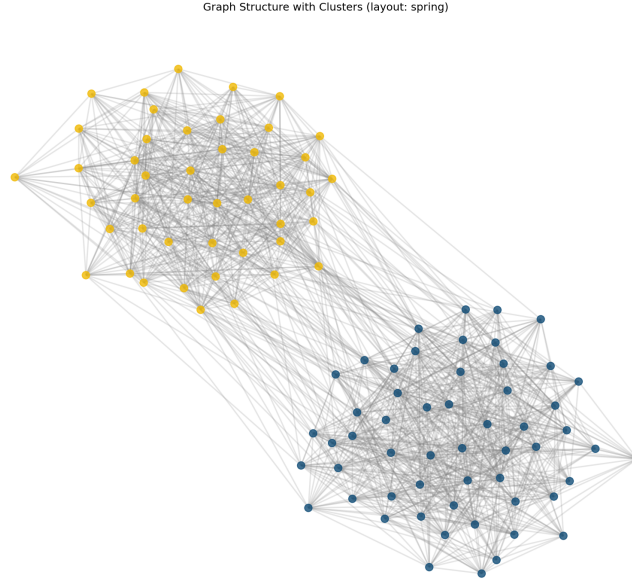Graph Structure with Clusters (layout: spring)

Figure 2: Clusters for $k = 2$

3.2.1 Cluster Quality Evaluation

Table 4 shows the cluster quality metrics for different values of k. The best results are achieved when $k = 2$, which matches our earlier analysis based on eigenvalue gaps. For $k = 2$, we have the lowest number of inter-cluster edges (113) and the highest number of intra-cluster edges (1096), with the best expansion ratio and conductance values.

Table 4: Cluster quality metrics for different values of k (example 2)

k	Inter-edges	Intra-edges	expansion ratio	conductance
2	113	1096	0.23	0.19
3	355	854	1.80	0.64
4	569	640	2.09	0.68
5	644	565	3.90	0.80
6	704	505	5.08	0.84
7	749	460	18.40	0.95
8	825	384	7.53	0.88
9	844	365	10.31	0.91

3.2.2 Fiedler Vector Visualization

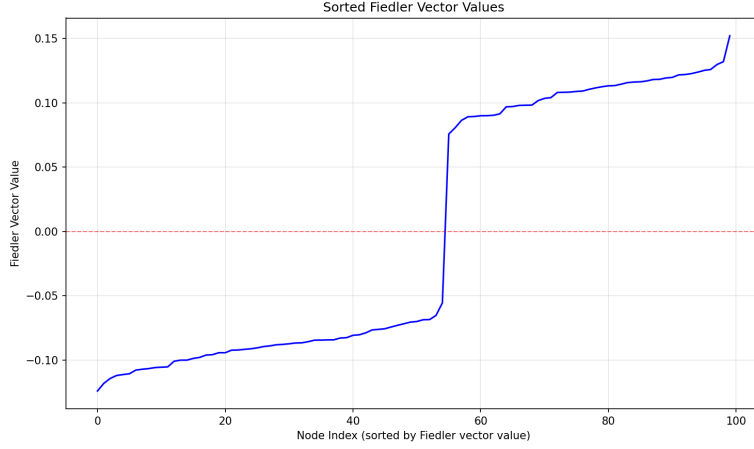To better understand the cluster separation, let's examine the Fiedler vector for example 2.

Figure 3: Fiedler vector for $k = 2$

From the Fiedler vector, we can see a clear separation between the two clusters. The steeper the slope, the more distinct the cluster boundaries are.

4 Conclusion

In conclusion, we have successfully implemented and applied the spectral graph clustering algorithm to analyze two different graphs. Through eigenvalue gap analysis, we determined the optimal number of clusters for each graph: $k = 4$ for example 1 and $k = 2$ for example 2.

The cluster quality metrics (inter-cluster edges, intra-cluster edges, expansion ratio, and conductance) confirmed our choices, with example 1 achieving perfect separation (zero inter-cluster edges) and example 2 achieving the best balance among all tested values of k. The visualization of the Fiedler vector further validated the cluster separation, demonstrating the effectiveness of spectral clustering in identifying natural graph communities.