Report 4: Graph Spectra

Group 150: Lorenzo Deflorian, Riccardo Fragale

December 1, 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 $L = D^{-1/2}AD^{-1/2}$
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 $L = D^{-1/2}AD^{-1/2}$, where A is the adjacency matrix of the graph and D is the degree matrix. This is the normalized Laplacian matrix. 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.

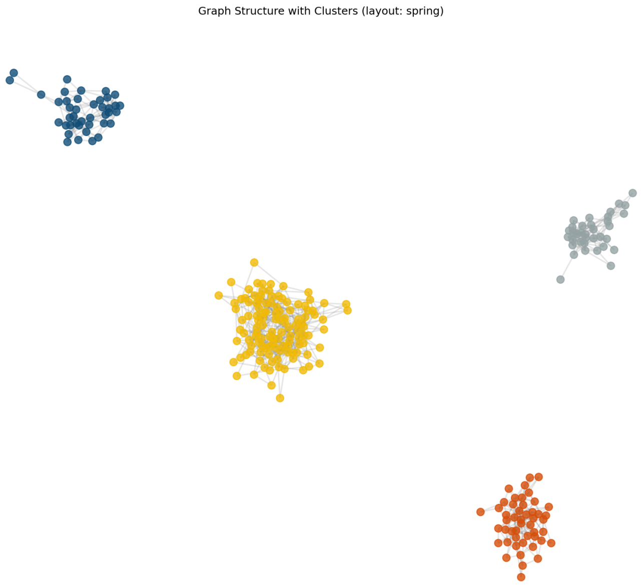

Figure 1: Clusters for $k = 4$

3.1.1 Cluster Quality Evaluation

To validate our choice of $k = 4$, we evaluate the graphs on the following metrics:

- **Inter-edges:** The number of edges connecting nodes from different clusters. Lower values indicate better cluster separation.
- **Intra-edges:** The number of edges connecting nodes within the same cluster. Higher values indicate stronger internal connectivity within clusters.
- **Expansion ratio:** The ratio of inter-cluster edges to intra-cluster edges. Lower values indicate better cluster quality, with 0.00 representing perfect separation.
- **Conductance:** The fraction of total edge volume that points outside the cluster. Lower values indicate better cluster quality, with 0.00 representing perfect separation.
- **Modularity:** A measure of the quality of the clustering that compares the number of edges within clusters to the expected number in a random graph. Higher values (closer to 1) indicate better community structure.

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. The modularity score also confirms this choice, with $k = 4$ achieving the highest modularity value of 0.6700.

Table 2: Cluster quality metrics for different values of k

k	Inter-edges	Intra-edges	expansion ratio	conductance	modularity
2	0	923	0.00	0.00	0.4999
3	0	923	0.00	0.00	0.6207
4	0	923	0.00	0.00	0.6700
5	12	911	0.71	0.41	0.6654
6	35	888	0.71	0.32	0.5363
7	126	797	0.71	0.50	0.6658
8	163	760	0.93	0.48	0.5806
9	191	732	1.61	0.62	0.6029

Since we have more than two clusters, let's plot the spectral embedding for $k = 4$.

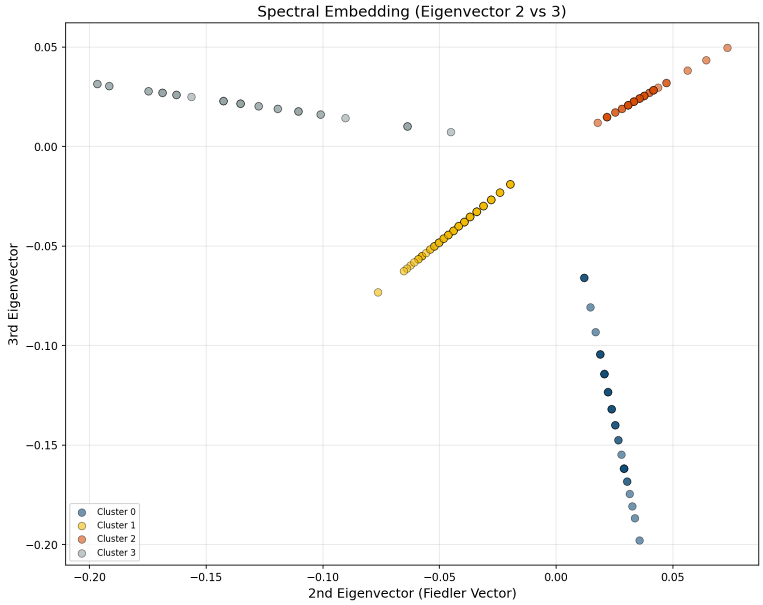

Figure 2: Spectral embedding for $k = 4$

3.1.2 Spectral Embedding Interpretation

The spectral embedding visualization in Figure 2 shows how the nodes are distributed in the reduced space defined by the 2nd and 3rd eigenvectors.

One interesting observation is that the points form distinct lines (or rays) radiating from a central point, rather than round clusters. This happens because nodes in the same community have similar eigenvector values, but due to degree variations within each cluster, they scale differently, creating these linear structures. This visualization demonstrates why Step 6 of our implementation (normalizing rows to unit length) is important: when we normalize these points, the rays collapse into tight, dense points on the unit circle, making them easier for K-means to separate.

The plot confirms that $k = 4$ is the correct choice. The four clusters are moving in clearly different directions with very little overlap between them. The Grey cluster moves along the negative X-axis, the Blue cluster moves down the negative Y-axis, and the Orange and Yellow clusters move in positive directions at distinct angles. This clear separation in the reduced eigenvector space validates that the spectral clustering algorithm has successfully identified four distinct communities in the graph.

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:

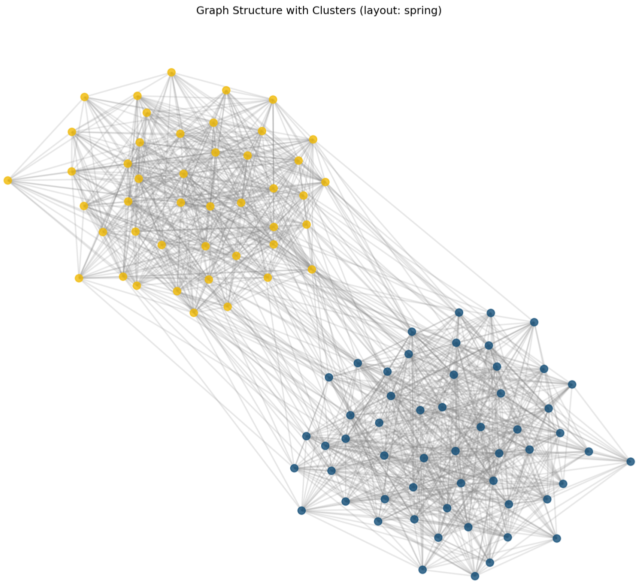

Figure 3: Clusters for $k = 2$

3.2.1 Cluster Quality Evaluation

The graphs are evaluated on the following metrics:

- **Inter-edges:** The number of edges connecting nodes from different clusters. Lower values indicate better cluster separation.
- **Intra-edges:** The number of edges connecting nodes within the same cluster. Higher values indicate stronger internal connectivity within clusters.
- **Expansion ratio:** The ratio of inter-cluster edges to intra-cluster edges. Lower values indicate better cluster quality, with 0.00 representing perfect separation.
- **Conductance:** The fraction of total edge volume that points outside the cluster. Lower values indicate better cluster quality, with 0.00 representing perfect separation.
- **Modularity:** A measure of the quality of the clustering that compares the number of edges within clusters to the expected number in a random graph. Higher values (closer to 1) indicate better community structure.

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. The modularity score also confirms this choice, with $k = 2$ achieving the highest modularity value of 0.4028.

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

k	Inter-edges	Intra-edges	expansion ratio	conductance	modularity
2	113	1096	0.23	0.19	0.4028
3	355	854	1.80	0.64	0.3516
4	569	640	2.09	0.68	0.2732
5	644	565	3.90	0.80	0.2515
6	704	505	5.08	0.84	0.2301
7	749	460	18.40	0.95	0.1815
8	825	384	7.53	0.88	0.1940
9	844	365	10.31	0.91	0.1661

3.2.2 Fiedler Vector Visualization

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

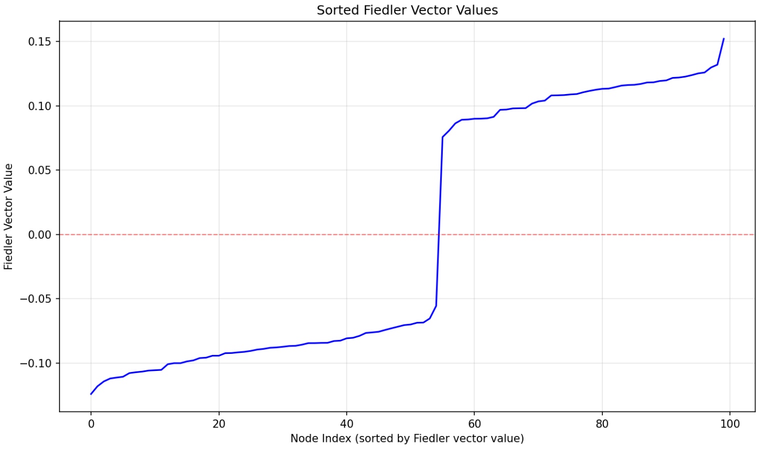

Figure 4: Fiedler vector for $k = 2$

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, conductance, and modularity) confirmed our choices, with example 1 achieving perfect separation (zero inter-cluster edges) and the highest modularity score of 0.6700, and example 2 achieving the best balance among all tested values of k with the highest modularity score of 0.4028. The visualization of the Fiedler vector and spectral embedding further validated the cluster separation, demonstrating the effectiveness of spectral clustering in identifying natural graph communities.