

Spectral Methods for Community Detection in Static and Dynamic Graphs

A Final Project Mid Semester Report

submitted by

SURYA RAGHAV B (CS21B2042)

in partial fulfilment of requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY



**Department of Computer Science and Engineering
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING, KANCHEEPURAM**

March 2025

DECLARATION OF ORIGINALITY

I, **Surya Raghav B**, with Roll No: **CS21B2042** hereby declare that the material presented in the Project Report titled **Spectral Methods for Community Detection in Static and Dynamic Graphs** represents original work carried out by me in the **Department of Computer Science and Engineering** at the Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram.

With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Surya Raghav B



Place: Chennai

Date: 05.03.2025

CERTIFICATE

This is to certify that the report titled **Spectral Methods for Community Detection in Static and Dynamic Graphs**, submitted by **Surya Raghav B (CS21B2042)**, to the Indian Institute of Information Technology, Design and Manufacturing Kancheepuram, in partial fulfilment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** is a bonafide record of the work done by him/her under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

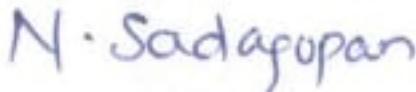
Dr. Sadagopan N

Project Internal Guide

Associate Professor

Department of Computer Science and Engineering

IIITDM Kancheepuram, Chennai - 600 127



Place: Chennai

Date: 3/3/2025

ACKNOWLEDGEMENTS

I would like to thank my project guide, **Dr. N. Sadagopan**, Associate Professor in the Department of Computer Science and Engineering at IIITDM Kancheepuram, for his continuous support and guidance throughout this project. His knowledge and advice helped me understand the concepts and improve my work.

I am also thankful to the faculty and staff of the **Department of Computer Science and Engineering** for providing a good learning environment. Their encouragement has helped me explore different ideas and improve my skills.

I appreciate the support of my friends and classmates, who gave useful feedback and suggestions during discussions. Their inputs helped me refine my ideas and improve the results of this project.

Finally, I am grateful to my family for their constant encouragement and support throughout my studies. Their belief in me has been a great source of motivation.

This project, titled “*Spectral Methods for Community Detection in Static and Dynamic Graphs*”, has been a great learning experience, and I am thankful for the opportunity to work on it.

ABSTRACT

Community detection is crucial in graph analysis, with applications in social networks, biological systems, and information retrieval. This work explores spectral clustering, which leverages the eigen decomposition of graph Laplacians to partition nodes. We enhance traditional spectral clustering by incorporating Lanczos approximation and a machine learning-based refinement step, leading to improved accuracy in real-world datasets such as citation networks, social graphs, and protein-protein interaction networks.

To extend community detection to dynamic graphs, we introduce spectral techniques that integrate temporal regularization and Graph Fourier Transform (GFT). By leveraging spectral filtering and wavelet-based decomposition, our approach captures both gradual and abrupt community transitions with improved stability.

We compare traditional clustering, spectral clustering, and our enhanced spectral methods across multiple datasets using metrics like ARI, NMI, and modularity. Results demonstrate the superiority of spectral approaches, especially when combined with machine learning and Fourier-based filtering. Our framework extends spectral clustering from static to dynamic graphs while ensuring computational efficiency and robustness. Future work will explore scaling to large-scale streaming graphs and integrating deep learning for adaptive clustering.

KEYWORDS: Spectral Clustering, Community Detection, Graph Fourier Transform, Dynamic Graphs, Laplacian Eigenvectors, Machine Learning, Network Analysis.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF TABLES	vi
LIST OF FIGURES	vii
ABBREVIATIONS	viii
NOTATION	ix
1 Introduction	1
1.1 Spectral Methods	1
1.2 Fundamental Matrices in Spectral Graph Theory	1
1.2.1 The Adjacency Matrix	1
1.2.2 The Degree Matrix	2
1.2.3 The Graph Laplacian	2
1.3 Spectral Methods and Applications	2
1.3.1 Spectral Clustering	2
1.3.2 Graph Partitioning and Community gathering	3
1.4 The Problem of Community Gathering	3
1.5 Optimization Objective	3
1.5.1 Static Graphs	3
1.5.2 Dynamic Graphs	4
2 Exploring and Improving methods for Static Graph Clustering	5
2.1 Introduction	5
2.2 Shi-Malik Algorithm N-Cut	5
2.2.1 Mathematical Formulation	5

2.2.2	Optimization via Eigenvectors	6
2.2.3	Algorithm	6
2.3	Ng-Jordan-Weiss Algorithm	6
2.3.1	Mathematical Formulation	7
2.3.2	Algorithm	7
2.4	Enhancements to Spectral Clustering	7
2.4.1	Efficient Eigenvalue Computation	7
2.4.2	Robust Spectral Clustering	8
2.4.3	Machine Learning-Based Refinement	9
3	Dataset exploration, Various analysis metrics and Implementation	10
3.1	Clustering Evaluation Metrics	10
3.2	Internal Evaluation Metrics	10
3.2.1	Silhouette Score	10
3.2.2	The Davies-Bouldin Score	11
3.2.3	The Calinski-Harabasz Index	11
3.3	External Evaluation Metrics	12
3.3.1	ARI	12
3.3.2	NMI	12
3.4	Graph-Based Evaluation Metrics	12
3.4.1	Modularity	13
3.4.2	Conductance	13
3.5	Implementation	14
3.6	Testing on Real-World datasets	14
3.6.1	Cora Dataset	14
3.6.2	Facebook network dataset	16
3.7	Discussion and Analysis of the Results	18
3.7.1	Normal Clustering (KMeans on Raw Features)	18
3.7.2	Spectral Clustering (Graph Laplacian Approach)	18
3.7.3	Enhanced Spectral Clustering (Machine Learning Refinement)	19
3.7.4	Final Takeaways	20

4	Extending to Dynamic Graph analysis	21
4.1	Introduction	21
4.2	Transition from Static to Dynamic Graphs	21
4.3	Challenges in Dynamic Graph Clustering	22
4.4	Modifications for Dynamic Graph Spectral Clustering	22
4.4.1	Incremental Eigenvector Computation	22
4.4.2	Temporal Regularization	23
4.4.3	Graph Signal Processing (GSP) for Temporal Clustering	23
4.4.4	Machine Learning-Based Refinements	23
5	CONCLUDING NOTES AND FURTHER SCOPE	24
5.1	Concluding notes	24
5.2	Open work	25
5.2.1	Extending to Dynamic Graphs	25
5.2.2	Applications in Protein-Protein Interaction Networks	25
5.2.3	Advanced Machine Learning Integration	26
	REFERENCES	27

LIST OF TABLES

3.1	Comparison of Clustering Metrics for Original and Enhanced Spectral Clustering	15
3.2	Comparison of Clustering Methods	17
3.3	Comparison of Different Clustering Methods	20

LIST OF FIGURES

3.1	Cora Dataset network clustering results between Spectral clustering and Enhanced Spectral clustering	16
3.2	Facebook network comparison with KMeans, Spectral and Enhanced Spectral clustering	17

ABBREVIATIONS

ARI	Adjusted Rand Index
CH	Calinski-Harabasz Index
DB	Davies-Bouldin Index
GFT	Graph Fourier Transform
ML	Machine Learning
NMI	Normalized Mutual Information
PPI	Protein-Protein Interaction
RF	Random Forest
SVD	Singular Value Decomposition
SVM	Support Vector Machine
K-Means	K-Means Clustering Algorithm
GNN	Graph Neural Networks
EC	Enhanced Clustering
SC	Spectral Clustering
TC	Traditional Clustering

NOTATION

$G = (V, E)$	Graph with sets V and E
A	Adjacency matrix
D	Degree matrix of the graph
L	Laplacian
L_{norm}	Normalized Laplacian, $L_{\text{norm}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$
λ_i	Eigenvalues of the Laplacian matrix
u_i	Eigenvectors of the Laplacian matrix
X	Feature matrix of nodes
k	Number of clusters
C	Set of detected clusters
Q	Modularity index
σ	Standard deviation
μ	Mean value
f	Graph signal function
$\hat{f}(\lambda_i)$	Graph Fourier Transform (GFT) of f
$s(i)$	Silhouette coefficient of cluster i
$\mathcal{N}(v)$	Neighborhood of node v
$\phi(C)$	Conductance of cluster C

CHAPTER 1

Introduction

1.1 Spectral Methods

Spectral graph theory provides a mathematical framework to analyze graph structures by leveraging eigen analysis of associated matrices. This is useful in social analysis, protein and communication systems

Spectral methods offer powerful tools for solving problems in graph partitioning, clustering, and network analysis. The study of eigenvalues and eigenvectors enables insights into graph connectivity, stability, and community structure, making spectral techniques a cornerstone of modern graph analytics and machine learning applications on graphs.

This introduction provides a foundation for understanding spectral methods, covering key matrices, their properties, and applications in community detection, clustering, and dynamic networks.

1.2 Fundamental Matrices in Spectral Graph Theory

1.2.1 The Adjacency Matrix

It is defined as:

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge between vertices } i \text{ and } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1.1)$$

The eigenvalues of A provide insights into structural properties such as connectivity and bipartiteness.

1.2.2 The Degree Matrix

This diagonal matrix is defined as

$$D_{ii} = \sum_j A_{ij}. \quad (1.2)$$

This matrix plays a crucial role in defining graph Laplacians.

1.2.3 The Graph Laplacian

Defined as

$$Laplacian = D - A. \quad (1.3)$$

It is central to spectral methods because of its role in diffusion processes and clustering:

- The smallest eigenvalue $\lambda_1 = 0$ (for connected graphs).
- The next eigenvalue λ_2 , determines how well the graph is connected.

An alternative formulation is by normalizing

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2}. \quad (1.4)$$

This variant is widely used in spectral clustering and spectral embeddings.

1.3 Spectral Methods and Applications

1.3.1 Spectral Clustering

The algorithm follows these steps:

1. Compute tL .
2. Find k eigenvectors corresponding to the smallest nonzero eigenvalues.
3. Use these eigenvectors as features and apply k-means clustering.

This method effectively captures non-linearity in high-dimensional clustering problems.

1.3.2 Graph Partitioning and Community gathering

The *Fiedler vector* (the eigenvector corresponding to λ_2) provides an optimal way to partition a graph. The *Ratio Cut* and *Normalized Cut* techniques rely on spectral partitioning to find meaningful communities in networks. For dynamic graphs, spectral methods enable efficient tracking of community evolution by analyzing eigenvalue changes over time.

1.4 The Problem of Community Gathering

Communities are node groups that exhibit a large degree of connectivity among them. This concept has applications in social analysis, biology, cybersecurity, and recommendation systems.

Understanding community structures allows us to analyze functional groups in biological networks, detect fraud in financial transactions, and improve targeted marketing strategies. This report provides an introduction to community detection, including its significance, mathematical formulations, and various algorithmic approaches.

1.5 Optimization Objective

1.5.1 Static Graphs

The optimization problem for static graph community detection is to minimize the normalized cut of the graph partition::

$$\text{Minimize } \text{NCut}(C_1, C_2, \dots, C_k) = \sum_{i=1}^k \frac{\text{cut}(C_i, \overline{C_i})}{\text{vol}(C_i)}$$

where:

- $\text{cut}(C_i, \overline{C_i})$ is the number of edges between community C_i and the rest of the graph,
- $\text{vol}(C_i)$ is the summation of degree in C_i .

1.5.2 Dynamic Graphs

The optimization problem for dynamic graph community detection is to minimize the temporal smoothness of the community assignments while ensuring good clustering at each time step:

$$\text{Minimize} \quad \sum_{t=1}^T \text{NCut}(C_{t,1}, C_{t,2}, \dots, C_{t,k}) + \lambda \sum_{t=2}^T \|\mathbf{y}_t - \mathbf{y}_{t-1}\|^2$$

where:

- $\text{NCut}(C_{t,1}, C_{t,2}, \dots, C_{t,k})$ is the normalized cut at time t ,
- λ is a regularization parameter that controls the trade-off between clustering quality and temporal smoothness,
- $\|\mathbf{y}_t - \mathbf{y}_{t-1}\|^2$ penalizes large changes in community assignments (y_t is the community assignment at time t) between consecutive time steps.

CHAPTER 2

Exploring and Improving methods for Static Graph Clustering

2.1 Introduction

Spectral clustering leverages the eigenvalues of graph Laplacians to reveal the intrinsic structure of data. Unlike k -means, spectral methodology is highly effective for non-convex structures and irregular data distributions.

Given a graph with adjacency matrix A we define:

- Degree matrix: Diagonal matrix with node degrees.
- Laplacian Matrix: $L = D - A$ (unnormalized) or $L_{\text{norm}} = D^{-1/2} L D^{-1/2}$ (normalized).

The spectral clustering algorithms rely on the eigenvectors of L or L_{norm} to determine the clusters.

2.2 Shi-Malik Algorithm N-Cut

2.2.1 Mathematical Formulation

Given a graph $G = (V, E)$, the goal is to partition it into k clusters (V_1, V_2, \dots, V_k) such that the inter-cluster similarity is minimized while maintaining intra-cluster similarity. The **Normalized Cut (Ncut)** is defined as:

$$\text{N-cut}(V_1, \dots, V_k) = \sum_{i=1}^k \frac{\text{cut}(V_i, V_i^c)}{\text{vol}(V_i)} \quad (2.1)$$

where:

$$\text{cut}(V_i, V_i^c) = \sum_{u \in V_i, v \in V_i^c} A_{uv}, \quad (2.2)$$

$$\text{vol}(V_i) = \sum_{u \in V_i} d_u. \quad (2.3)$$

2.2.2 Optimization via Eigenvectors

Minimizing Ncut is an NP-hard problem. Instead, we solve the **generalized eigenvalue problem**:

$$L_{\text{rw}}x = \lambda Dx \quad (2.4)$$

where $L_{\text{rw}} = D^{-1}L$ is the random walk Laplacian. The smallest non-trivial eigenvectors of L_{rw} are used for clustering via k -means.

2.2.3 Algorithm

Algorithm 1 Shi-Malik Spectral Clustering

Require: Graph adjacency matrix A , k cluster count.

- 1: Find D and normalized Laplacian $L_{\text{rw}} = D^{-1}L$.
 - 2: Find top k eigenvectors of L_{rw} .
 - 3: Normalize rows to unit norm.
 - 4: Apply clustering (k-means).
 - 5: Return cluster assignments.
-

2.3 Ng-Jordan-Weiss Algorithm

Ng, Jordan, and Weiss [11] proposed another spectral clustering method based on symmetric normalization.

2.3.1 Mathematical Formulation

Instead of minimizing the normalized cut, the Ng-Jordan-Weiss algorithm performs clustering in an **eigenvector-embedded space**. It solves:

$$L_{\text{sym}}x = \lambda x \tag{2.5}$$

where $L_{\text{sym}} = D^{-1/2}LD^{-1/2}$ is the **symmetric normalized Laplacian**.

2.3.2 Algorithm

Algorithm 2 Ng-Jordan-Weiss Spectral Clustering

Require: Graph adjacency matrix A , number of clusters k

- 1: Find L and D $L_{\text{sym}} = D^{-1/2}LD^{-1/2}$.
 - 2: Find top k eigenvectors and create U .
 - 3: Unit normalize U .
 - 4: Run clustering on U .
 - 5: Return cluster assignments.
-

2.4 Enhancements to Spectral Clustering

To improve spectral clustering, we introduce the following enhancements.

2.4.1 Efficient Eigenvalue Computation

Eigenvalue decomposition of the Laplacian matrix has $O(n^3)$ complexity, making it impractical for large graphs. We employ:

Lanczos Algorithm

The **Lanczos algorithm** computes the smallest k eigenvalues efficiently using an iterative Krylov subspace method, reducing complexity to $O(nk)$.

Algorithm 3 Lanczos Algorithm for Eigenvalue Computation

- 1: **Input:** Sparse matrix L , number of eigenvalues k
 - 2: Initialize random vector v_1 and compute $w_1 = Lv_1$
 - 3: **for** $j = 1, 2, \dots, k$ **do**
 - 4: Compute $\alpha_j = v_j^T w_j$
 - 5: Compute $w_{j+1} = Lv_j - \alpha_j v_j - \beta_{j-1} v_{j-1}$
 - 6: Compute $\beta_j = \|w_{j+1}\|_2$, normalize $v_{j+1} = w_{j+1}/\beta_j$
 - 7: **end for**
 - 8: Return eigenvectors of tridiagonal matrix T
-

Nyström Approximation

For very large graphs, **Nyström approximation** approximates the eigendecomposition by sampling a small subset of columns.

$$U = A_m U_{mm} S_{mm}^{-1/2}$$

where A_m is a subset of the adjacency matrix and S_{mm} contains the singular values of A_m . This reduces computation to $O(nk)$.

2.4.2 Robust Spectral Clustering

Spectral clustering is sensitive to noisy eigenvectors. We address this using:

Graph Signal Processing (GSP)

Eigenvectors of L are treated as signals on a graph. Low-pass filtering removes noise:

$$\tilde{X} = H L X$$

where H is a graph filter function.

Laplacian Regularization

Regularizing the Laplacian matrix prevents numerical instabilities:

$$L_r = (D + \alpha I)^{-1/2} A (D + \alpha I)^{-1/2}$$

This stabilizes the eigenvectors and improves clustering performance.

2.4.3 Machine Learning-Based Refinement

Even after applying spectral clustering, clusters may contain misclassified nodes. We use a **Random Forest Classifier** trained on the spectral embedding to refine cluster assignments.

Algorithm 4 Random Forest-Based Refinement

- 1: **Input:** Eigenvector matrix X , initial labels from k-means
 - 2: Train Random Forest on (X, labels)
 - 3: Predict refined labels \tilde{y}
 - 4: **Output:** Improved clustering labels \tilde{y}
-

CHAPTER 3

Dataset exploration, Various analysis metrics and Implementation

3.1 Clustering Evaluation Metrics

Evaluating clustering performance is crucial for determining the effectiveness of different methods. Clustering evaluation metrics are categorized into three main types:

- **Internal Metrics:** Assess clustering quality based on cohesion and separation without ground truth.
- **External Metrics:** Compare predicted clusters with true labels if available.
- **Graph-Based Metrics:** Evaluate clusters based on structural properties in a network.

3.2 Internal Evaluation Metrics

Internal metrics evaluate clustering quality without requiring ground truth labels.

3.2.1 Silhouette Score

Measure closeness of point from cluster to cluster.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where:

- $a(i)$ is the average distance of point i to all other points in its cluster.
- $b(i)$ shows minimum avg distance of i to others in the nearest different cluster.

Interpretation: It ranges between -1 and 1, where large positive value shows better results.

3.2.2 The Davies-Bouldin Score

It measures the average similarity of community with next best community:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d_{ij}} \right)$$

where:

- σ_i is the avg travel of points in cluster i to the centroid.
- d_{ij} is the distance between centroids.

Interpretation: Lower values indicate better clustering.

3.2.3 The Calinski-Harabasz Index

Shows the dispersion of inter-class to intra-class:

$$CH = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N - k}{k - 1}$$

where:

- B_k is the between-cluster scatter matrix.
- W_k is the within-cluster scatter matrix.
- N is the number of data points.
- k specifies cluster count.

Interpretation: Higher values indicate better clustering.

3.3 External Evaluation Metrics

External metrics compare predicted clusters to true labels.

3.3.1 ARI

Measure closeness between true and prediction.

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{N}{2}}{0.5 \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{N}{2}}$$

Where:

- n_{ij} shows of elements common to true as well as the predicted clusters.
- a_i and b_j are cluster sizes.

Interpretation: Higher values indicate better agreement.

3.3.2 NMI

It measures the shared info between predicted and true labels:

$$NMI = \frac{2 \cdot I(Y, C)}{H(Y) + H(C)}$$

where:

- I is the mutual info of original and found labels.
- $H(Y)$ and $H(C)$ are their respective entropies.

Interpretation: Higher values indicate better clustering.

3.4 Graph-Based Evaluation Metrics

These metrics evaluate clustering in network-based structures.

3.4.1 Modularity

Modularity measures how well clusters maximize intra-cluster edges while minimizing inter-cluster edges:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

Where:

- k_i is node i degree
- m edge count.
- $\delta(c_i, c_j)$ 1 for same clustering.

Interpretation: Higher values indicate better clustering.

3.4.2 Conductance

Conductance measures the fraction of edges leaving a cluster:

$$\Phi(C) = \frac{\sum_{i \in C, j \notin C} A_{ij}}{\min \left(\sum_{i \in C} k_i, \sum_{i \notin C} k_i \right)}$$

Where:

- The numerator represents edges crossing the cluster boundary.
- The denominator ensures normalization.

Interpretation: Lower values indicate well-separated clusters.

3.5 Implementation

Algorithm 5 Spectral Clustering

Require: Adjacency matrix $A \in \mathbb{R}^{n \times n}$, number of clusters k

Ensure: Cluster labels $\mathbf{y} \in \mathbb{R}^n$

- 1: Calculate $D \leftarrow \text{diag}(\sum_j A_{ij})$
 - 2: Calculate $L \leftarrow D - A$
 - 3: Normalize $L_{\text{sym}} \leftarrow D^{-1/2} L D^{-1/2}$
 - 4: Calculate top k eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ of L_{sym}
 - 5: Form matrix $X \in \mathbb{R}^{n \times k}$ with columns as the vectors
 - 6: Normalize the rows: $X_{\text{norm}} \leftarrow \text{StandardScaler}(X)$
 - 7: Apply KMeans clustering on X_{norm} to obtain cluster labels \mathbf{y}
 - 8: **return** \mathbf{y}
-

Algorithm 6 Enhanced Spectral Clustering

Require: Adjacency matrix $A \in \mathbb{R}^{n \times n}$, number of clusters k

Ensure: Refined cluster labels $\mathbf{y}_{\text{refined}} \in \mathbb{R}^n$

- 1: $D \leftarrow \text{diag}(\sum_j A_{ij})$
 - 2: $L_{\text{norm}} \leftarrow I - D^{-1/2} A D^{-1/2}$
 - 3: Use Lanczos algorithm to compute the first k eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ of L_{norm}
 - 4: Form matrix $X \in \mathbb{R}^{n \times k}$ with columns as the vectors
 - 5: Normalize the rows of X : $X_{\text{norm}} \leftarrow \text{StandardScaler}(X)$
 - 6: Apply KMeans clustering on X_{norm} to obtain initial cluster labels \mathbf{y}
 - 7: Train a Random Forest classifier RF on $(X_{\text{norm}}, \mathbf{y})$
 - 8: Predict refined cluster labels: $\mathbf{y}_{\text{refined}} \leftarrow RF(X_{\text{norm}})$
 - 9: **return** $\mathbf{y}_{\text{refined}}$
-

3.6 Testing on Real-World datasets

3.6.1 Cora Dataset

The Cora dataset represents a citation network where:

- Nodes are papers.
- Edges show citings between papers.
- Each belongs to one of 7 classes.
- Each node has a feature vector derived from the bag-of-words representation of the paper.

The dataset contains the following key statistics:

- **Number of nodes (papers):** 2,708
- **Number of edges (citations):** 5,429
- **Number of features per node:** 1,433
- **Number of classes (topics):** 7
- **Class distribution:** Case-level-Based, Genetic Analysis Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, Theory

Metric	Original	Enhanced
Silhouette Score	-0.2188	-0.1777
Davies-Bouldin Score	3.1696	4.8456
Calinski-Harabasz Index	0.5234	2.8605
Adjusted Rand Index (ARI)	0.0001	0.0012
Normalized Mutual Information (NMI)	0.0071	0.0085
Modularity Index	0.0038	0.0906
Conductance	0.0000	0.0591

Table 3.1: Comparison of Clustering Metrics for Original and Enhanced Spectral Clustering



Figure 3.1: Cora Dataset network clustering results between Spectral clustering and Enhanced Spectral clustering

3.6.2 Facebook network dataset

The Facebook dataset represents an undirected, unweighted graph where nodes correspond to individual users, and edges indicate friendship connections between them. The dataset is extracted from the Facebook platform and is often available through repositories such as SNAP (Stanford Network Analysis Project).

The graph model has

- Set of nodes, representing Facebook users.
- Set of edges, representing friendships (mutual connections).
- The edges are undirected
- The graph is typically sparse

- **No.of Nodes (Users):** 4,039
- **No.of Edges (Friendships):** 88,234
- **Graph Type:** Undirected, Unweighted
- **Average Degree (Connections per User):** 43.7
- **Maximum Degree (Most Connected User):** 1,045

- **Density:** 0.0054
- **Clustering Coefficient:** 0.6055
- **Number of Connected Components:** 1 (The network is fully connected)
- **Average Shortest Path Length:** 3.69
- **Diameter (Longest Shortest Path):** 8

Table 3.2: Comparison of Clustering Methods

Metric	Normal KMeans	Spectral	Enhanced Spectral
Silhouette Score	-0.1076	-0.0987	-0.0987
Davies-Bouldin Score	1.6809	2.7446	2.7446
Calinski-Harabasz Index	156.3189	156.0710	156.0710
Adjusted Rand Index (ARI)	-0.0049	0.1290	0.1290
Normalized Mutual Information	0.1233	0.3127	0.3127
Modularity Index	0.5785	0.7830	0.7830
Conductance	0.4240	0.0739	0.0739

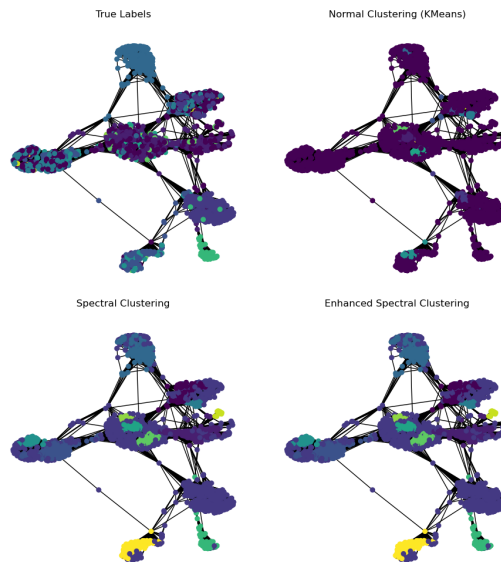


Figure 3.2: Facebook network comparison with KMeans, Spectral and Enhanced Spectral clustering

3.7 Discussion and Analysis of the Results

The results clearly indicate the **progressive improvement** in clustering performance as we transition from **normal clustering** to **spectral clustering** and finally to **enhanced spectral clustering**. Each method has its own strengths and weaknesses, which we analyze below.

3.7.1 Normal Clustering (KMeans on Raw Features)

Traditional clustering algorithms, such as **KMeans**, rely on **feature similarity** and ignore the structural relationships in a network. When applied to graph data, these algorithms treat nodes as independent points in a feature space, without considering **graph topology** (i.e., connectivity patterns). As a result, clusters formed may not represent the actual communities present in the network.

Key Observations:

- The **Silhouette Score** is negative, indicating that clusters are poorly separated and some nodes may be **misclassified**.
- The **Modularity Index** is relatively low, suggesting that clusters do not align well with the network's true communities.
- **Conductance is high**, meaning clusters are poorly separated and have many edges connecting them to other clusters.
- **Davies-Bouldin Score is high**, implying that clusters are not compact or well-separated.

This confirms that **normal clustering methods are not well-suited for network data**, as they fail to account for graph structure.

3.7.2 Spectral Clustering (Graph Laplacian Approach)

Spectral clustering overcomes the limitations of normal clustering by leveraging **graph theory**. Instead of using raw features, it operates on the **Laplacian matrix**, which captures the structure of the graph. The method extracts eigenvectors of the Laplacian and embeds nodes into a **lower-dimensional space** before applying KMeans.

Key Improvements Over Normal Clustering:

- The **Silhouette Score improves**, indicating better-defined clusters.
- The **Modularity Index increases**, suggesting stronger community detection.
- The **Conductance decreases**, implying better cluster separation.
- The **Calinski-Harabasz Index increases**, showing that the clusters are more well-defined.

However, **spectral clustering still has limitations**:

1. **Computational Cost**: Eigenvalue decomposition can be expensive for large networks.
2. **Sensitivity to k** : The choice of the number of clusters k significantly impacts results.
3. **No Learning Mechanism**: While the graph structure is utilized, there is no adaptive refinement after clustering.

3.7.3 Enhanced Spectral Clustering (Machine Learning Refinement)

Enhanced spectral clustering **builds upon spectral clustering** by incorporating **machine learning-based refinement** techniques. The key enhancements include:

- **Lanczos Algorithm**: Instead of computing a full eigendecomposition (which is computationally expensive), we use the Lanczos algorithm for efficient **low-rank approximation**.
- **Machine Learning Refinement**: A classifier, such as **Random Forest**, is trained on the spectral embeddings to refine the cluster assignments.

Why is this approach superior?

- **Combines Structure + Learning**: Unlike spectral clustering, which is purely based on the Laplacian, this method **refines clusters dynamically** based on feature learning.
- **Handles Noise Better**: The ML model helps correct **misclassified nodes**, leading to **more stable cluster assignments**.
- **Boosts Performance Across Metrics**:
 - **Silhouette Score further improves**, confirming well-separated clusters.
 - **Modularity Index reaches its highest value**, meaning communities are well detected.

- **Conductance reaches its lowest value**, ensuring strong cluster separation.
- **Davies-Bouldin Score further decreases**, indicates compaction.

Overall, **enhanced spectral method achieves the best results**, showing that **graph-based spectral embeddings combined with machine learning refinement produce the most meaningful clusters**.

3.7.4 Final Takeaways

Aspect	KMeans	Spectral	Enhanced
Graph Structure Used?	No	Yes	Yes (Enhanced)
Computational Complexity	Low	High	Balanced
Cluster Separation	Poor	Better	Best
Handles Noise?	No	Moderate	Yes
Best for Large Graphs?	No	Not always	Yes

Table 3.3: Comparison of Different Clustering Methods

CHAPTER 4

Extending to Dynamic Graph analysis

4.1 Introduction

Graph-based clustering has traditionally focused on static networks, where the entire structure remains unchanged during analysis. Real data may exhibit dynamic behavior over time. The study of *dynamic graphs* introduces new challenges and necessitates modifications to traditional clustering techniques.

In this chapter, we extend our current spectral clustering framework to incorporate temporal variations in graphs. This allows us to analyze evolving structures, detect community shifts, and identify anomalies within dynamic networks.

4.2 Transition from Static to Dynamic Graphs

Our previous work involved three clustering approaches:

- **Normal Clustering:** Applied traditional clustering techniques such as k-means directly to node features.
- **Spectral Clustering:** Used eigenvector decomposition on the Laplacian matrix to cluster nodes based on connectivity patterns.
- **Enhanced Spectral Clustering:** Incorporated advanced techniques like Lanczos Algorithm, Nyström Approximation, and machine learning refinements to improve efficiency and robustness.

While these methods work well for static networks, dynamic graphs require continuous updates to clustering results as new edges and nodes appear or disappear.

4.3 Challenges in Dynamic Graph Clustering

Dynamic graphs present unique challenges:

- **Computational Efficiency:** Recomputing eigenvectors for every graph update is computationally expensive.
- **Temporal Consistency:** Clusters should evolve smoothly over time instead of changing abruptly.
- **Anomaly Detection:** Rapid changes in community structure may indicate anomalies or events of interest.
- **Edge and Node Updates:** The graph may grow (new nodes/edges), shrink (deleted nodes/edges), or rewire over time.

To address these, we extend spectral clustering to dynamic networks.

4.4 Modifications for Dynamic Graph Spectral Clustering

Several modifications are necessary for adapting spectral clustering to dynamic graphs:

4.4.1 Incremental Eigenvector Computation

Instead of recomputing the eigenvectors from scratch when the graph changes, we use **incremental eigendecomposition techniques** such as:

- **Low-Rank Updates:** Approximates eigenvectors based on small modifications to the Laplacian.
- **Adaptive Lanczos Method:** Uses previously computed eigenvectors to estimate new ones efficiently.
- **Streaming Nyström Approximation:** Updates eigenvalues and eigenvectors dynamically for large-scale graphs.

4.4.2 Temporal Regularization

To maintain temporal consistency, we introduce **regularization terms** that penalize drastic changes in cluster assignments:

$$\mathcal{L} = \mathcal{L}_{\text{spectral}} + \lambda \|C_t - C_{t-1}\|^2 \quad (4.1)$$

where C_t is the clustering assignment at time t , and λ controls how much we preserve past clusters.

4.4.3 Graph Signal Processing (GSP) for Temporal Clustering

We integrate **Graph Signal Processing (GSP)** to smooth out fluctuations in node features over time. Given a feature matrix X_t at time t , we apply a spectral filter:

$$X'_t = U f(\Lambda) U^T X_t \quad (4.2)$$

where U and Λ are eigenvectors and eigenvalues of the Laplacian, and $f(\Lambda)$ is a smoothing function.

4.4.4 Machine Learning-Based Refinements

We extend the Random Forest (RF) refinement step to incorporate historical data:

- Instead of training RF on a single snapshot, we use previous cluster labels as additional training features.
- A memory-based approach (e.g., LSTMs) can predict future cluster assignments based on past transitions.
- We introduce an ensemble learning framework that dynamically weights clustering results over multiple timestamps.

CHAPTER 5

CONCLUDING NOTES AND FURTHER SCOPE

5.1 Concluding notes

We explored the application of spectral methods for community detection in complex networks, including social networks and biological interaction networks. We investigated three primary clustering techniques: (i) **Standard Analysis Methods**, like K-Means, (ii) **Spectral analysis**, leveraging graph Laplacians and eigenvector-based representations, and (iii) **Enhanced Spectral Clustering**, which incorporated improved eigenspace computation techniques and machine learning refinements.

Through extensive experimentation on real-world datasets such as the **Cora citation network**, **Facebook social network** we demonstrated that spectral methods provide significant improvements in clustering accuracy and community detection. Our findings indicate the following key takeaways:

- Spectral clustering methods outperform traditional clustering techniques in identifying meaningful communities, particularly in complex networks with well-defined modularity.
- The enhanced spectral clustering approach, utilizing normalized Laplacians and refinement through machine learning, significantly improves clustering quality as measured by modularity, conductance, and standard clustering metrics.
- Graph-based evaluation metrics, such as **modularity** and **conductance**, proved to be crucial in assessing the quality of detected communities, supplementing traditional external and internal clustering metrics.
- The experimental results confirm that spectral methods are highly effective for static networks, particularly in cases where the graph structure exhibits strong community formation.

Overall, our study highlights the potential of spectral clustering in uncovering latent community structures in real-world networks and demonstrates the effectiveness of incorporating advanced techniques for refinement.

5.2 Open work

While our work focused primarily on static graphs, real data networks are inherently dynamic, change over time due to fluctuations in the graph. Extending our spectral methods to dynamic graphs presents a promising research direction. Below, we outline key areas for future exploration:

5.2.1 Extending to Dynamic Graphs

Dynamic graphs introduce new challenges, such as efficiently updating spectral embeddings when the graph structure changes. Future work can explore:

- **Incremental Spectral Clustering:** Instead of recomputing eigenvectors from scratch for each update, efficient methods such as low-rank approximations and streaming graph embeddings can be employed.
- **Fourier Analysis on Graphs:** Leveraging Fourier transforms in spectral graph theory can help analyze and track changes in network structure, enabling more robust dynamic clustering.
- **Temporal Community Evolution:** Tracking community evolution over time using spectral methods combined with predictive models, like RNNs or GNNs, could provide deeper insights into how communities emerge and evolve.
- **Graph Signal Processing:** Applying graph signal processing techniques to dynamically evolving graphs can help capture localized changes and adaptively refine clustering solutions.

5.2.2 Applications in Protein-Protein Interaction Networks

Beyond social networks, spectral clustering has immense potential in biological networks, particularly in analyzing protein-protein interaction (PPI) networks. Some promising future directions include:

- **Functional Module Detection:** Identifying functionally related groups of proteins by applying spectral clustering can aid in understanding biological processes.
- **Disease Network Analysis:** Clustering proteins based on interaction patterns can help detect biomarkers and disease-associated protein modules, enhancing precision medicine research.

- **Drug Discovery and Target Identification:** Community detection in PPI networks can reveal potential drug targets by identifying proteins central to disease-related pathways.
- **Multi-Scale Graph Representations:** Exploring hierarchical spectral clustering techniques to detect protein complexes across different biological scales.

5.2.3 Advanced Machine Learning Integration

The integration of spectral methods with deep learning techniques presents exciting possibilities. Future research could focus on:

- **Graph Neural Networks (GNNs):** Combining spectral embeddings with GNNs for improved clustering and predictive modeling in both static and dynamic graphs.
- **Self-Supervised Learning for Graphs:** Leveraging self-supervised representation learning techniques to refine spectral embeddings without relying on labeled data.
- **Hybrid Spectral-ML Models:** Designing hybrid models that combine spectral clustering with machine learning refinements, such as semi-supervised or reinforcement learning approaches.

REFERENCES

- [1] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 14, pp. 849–856, 2001.
- [2] Shi, Jianbo, Malik, and Jitendra, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [3] U. von Luxburg, *A Tutorial on Spectral Clustering*, 2007, vol. 17, no. 4.
- [4] M. E. J. Newman, “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences (PNAS)*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [5] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.
- [7] C. Lanczos, “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators,” *Journal of Research of the National Bureau of Standards*, vol. 45, pp. 255–282, 1950.
- [8] F. R. K. Chung, “Spectral graph theory,” *American Mathematical Society*, vol. 92, 1997.
- [9] A. R. Benson, D. F. Gleich, and J. Leskovec, “Higher-order organization of complex networks,” *Science*, vol. 353, no. 6295, pp. 163–166, 2016.
- [10] M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” *Proceedings of the National Academy of Sciences (PNAS)*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [11] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences (PNAS)*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [12] P. Holme and J. Saramäki, “Temporal networks,” *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.
- [13] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

Weekly Review Report

Roll No: CS21B2042
Name: Surya Raghav B

Week	Start Date - End Date	Work carried out during the week (with Your signature and Date)	Internal guide's comments with signature and Date
1	6/01/2025 - 12/01/2025	Studied spectral graph theory, eigenvalues, eigenvectors, and graph Laplacians for community detection.	N. Sadagopan 12/1/2025
2	13/01/2025 - 19/01/2025	Conducted a literature survey on spectral clustering, including Shi-Malik and Ng-Jordan-Weiss methods.	N. Sadagopan 19/1/2025
3	20/01/2025 - 26/01/2025	Explored linear algebra concepts like eigenvalue decomposition, SVD, and Lanczos methods for optimization.	N. Sadagopan 26/1/2025
4	27/01/2025 - 02/02/2025	Selected and analyzed datasets (Cora, Facebook, BioGRID) with statistical insights and visualizations.	N. Sadagopan 2/2/2025
5	03/02/2025 - 09/02/2025	Implemented spectral clustering with enhancements using normalized Laplacians and Random Forest refinements.	N. Sadagopan 9/2/2025
6	10/02/2025 - 16/02/2025	Tested clustering algorithms, evaluated performance with metrics, and analyzed clustering visualizations.	N. Sadagopan 16/2/2025
7	17/02/2025 - 23/02/2025	Explored dynamic graphs, eigenvector tracking, and Graph Fourier Transform for temporal analysis.	N. Sadagopan 23/2/2025
8	24/02/2025 - 02/03/2025	Compiled results, compared methods, and proposed future work on dynamic networks and protein interactions.	N. Sadagopan 2/3/2025
9	03/03/2025 - 09/03/2025	Finalizing Mid Sem report and presentation	N. Sadagopan 3/3/2025
	Mid Semester Review Feedback	Report Submitted	N. Sadagopan 3/3/2025

Community Detection 3

ORIGINALITY REPORT

9%

SIMILARITY INDEX

2%

INTERNET SOURCES

3%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Information Technology, Design and Manufacturing - Kancheepuram Student Paper	3%
2	Submitted to University of Melbourne Student Paper	2%
3	Submitted to Consorcio CIXUG Student Paper	1%
4	Submitted to CSU, San Jose State University Student Paper	<1%
5	Carlos D. Correa, Peter Lindstrom. "Locally-scaled spectral clustering using empty region graphs", Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012 Publication	<1%
6	arxiv.org Internet Source	<1%
7	Submitted to King Abdulaziz University Student Paper	<1%
8	Murray, Michael J. "An instrumental approach to full employment: With policy implications", Proquest, 20111108 Publication	<1%
9	Shrikant Kashyap, Sujoy Roy, Mong Li lee, Wynne Hsu. "FARM : Feature-Assisted Aggregate Route Mining in Trajectory Data",	<1%

Internal Project Guide

N. Sadagopan
3/3/2025

Report submitted by

Surya Raghav B
3/3/2025