Analysis and Actions on Graph Data

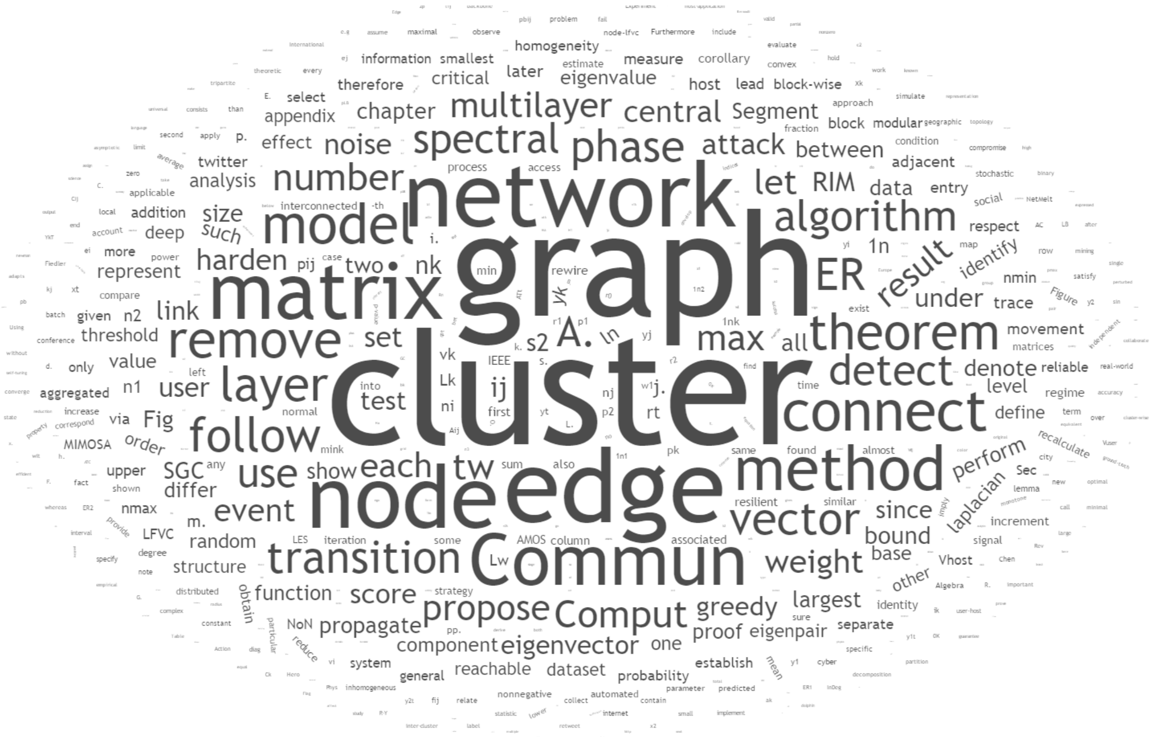
by

Pin-Yu Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in The University of Michigan
2016

Doctoral Committee:

Professor Alfred O. Hero III, Chair
Assistant Professor Danai Koutra
Assistant Professor Daniel M. Romero
Associate Professor Vijay Gautam Subramanian


problem
node-lfvc
Furthermore
include
homogeneity
information
smallest
measure
corollary
critical
later
eigenvalue
therefore
host
lead
block-wise
select
multilayer
chapter
central
Segment
appendix
block
modular
spectral
phase
attack
effect
noise
between
twitter
number
adjacent
let
RIM
analysis
network
data
entry
model
algorithm
size
deep
such
graph
respect
more
harden
ER
identify
represent
matrix
1n
result
nk
nmin
link
under
trace
n2
remove
A.
theorem
movement
threshold
max
s2
all
value
layer
detect
denote
user
ij
n1
aggregated
Fig
test
cluster
connect
level
define
follow
perform
SGC
each
tw
use
method
upper
show
differ
node
edge
event
vector
since
laplacian
nmax
LFVC
m.
random
bound
base
transition
Commun
weight
Vhost
structure
other
function
score
largest
propose
identity
Comput
greedy
propagate
separate
proof
eigenpair
component
eigenvector
one
system
reachable
dataset
establish
general
probability

For my dad, my mom, my brother, Pin-Yu (Bao), and my families in Taiwan

ACKNOWLEDGEMENTS

I am very fortunate to have Professor Alfred Hero as my advisor. During my doctoral study he has opened a new research door in graph data analytics for me, and has been continuously providing his full support, both in research and career development. I am also very fortunate to have Professor Vijay Subramanian, Professor Daniel Romero, and Professor Danai Koutra as my dissertation committee members. Their efforts and comments have made my dissertation more solid and profound.

I would like to thank members in the Hero group for their friendship and research discussion, especially to Sijia Liu for exciting research collaboration, to Yu-Hui Chen for providing his dissertation template, and to Michele Feldkamp in the EECS office for her administrative assistance. I also would like to thank Pai-Shun Ting and Chun-Chen Tu for their friendship and company in weekly grocery shopping. You have made Ann Arbor an even more unforgettable place. Outside of University of Michigan, I would like to thank Doctor Sutanay Choudhury at Pacific Northwest National Laboratory for giving me an internship opportunity, thank Siheng Chen at Carnegie Mellon University for his friendship and remarkable insight in graph signal processing, thank Professor Shin-Ming Cheng at National Taiwan University of Science and Technology for his friendship and productive research collaboration.

Lastly, I woule like to thank my parents, my brother Pin-Jung Chen, my girl friend Pin-Yu Chen, my best friends Chun-Yu Yang and Domi Liu, and other families and friends in Taiwan, for being part of my life. Your encouragement and company have fulfilled my life and have given me the momentum for becoming a better man.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Analysis and Actions on Graph Data

by

Pin-Yu Chen

Chair: Alfred O. Hero III

Graphs are commonly used for representing relations between entities and handling data processing in various research fields, especially in social, cyber and physical networks. Many data mining and inference tasks can be interpreted as certain actions on the associated graphs, including graph spectral decompositions, and insertions and removals of nodes or edges. For instance, the task of graph clustering is to group similar nodes on a graph, and it can be solved by graph spectral decompositions. The task of cyber attack is to find effective node or edge removals that lead to maximal disruption in network connectivity.

In this dissertation, we focus on the following topics in graph data analytics:

1. Fundamental limits of spectral algorithms for graph clustering in single-layer and multilayer graphs.

2. Efficient algorithms for actions on graphs, including graph spectral decompositions and insertions and removals of nodes or edges.

3. Applications to deep community detection, event propagation in online social networks, and topological network resilience for cyber security.

For 1, we established fundamental principles governing the performance of graph clustering for both spectral clustering and spectral modularity methods, which play an important role in unsupervised learning and data science. The framework is then extended to multilayer graphs entailing heterogeneous connectivity information.

For 2, we developed efficient algorithms for large-scale graph data analytics with theoretical guarantees, and proposed theory-driven methods for automatic model order selection in graph clustering.

For 3, we proposed a disruptive method for discovering deep communities in graphs, developed a novel method for analyzing event propagation on Twitter, and devised effective graph-theoretic approaches against explicit and lateral attacks in cyber systems.

CHAPTER I

Introduction

1.1 Motivation

Many real-world data are often represented as graphs, ranging from relations in social networks (e.g., friendship), links in cyber networks (e.g., the World Wide Web), connections in physical systems (e.g., computer networks, power systems), biological interactions and chemical reactions, to information flows in networks (e.g., transportation systems, routing). Therefore, graphs are common themes in various research fields, and many data mining and inference tasks, such as clustering, merging, pruning, visualization, summarization, sparsification, extraction, sampling, etc., can be interpreted as certain actions on the associated graphs. Specifically, actions on graphs include but are not limited to graph spectral decompositions, insertions of nodes or edges, removals of nodes or edges, and edge weight modification. In this dissertation, we are interested in understanding the principles for graph data analysis and processing through actions on graphs. In particular, we focus on two types of actions on graphs, namely *graph spectral decompositions* and *insertions and removals of nodes or edges*, for graph data analytics in graph clustering (also known as community detection) and cyber security. We also show some applications to discovering deep communities in graphs, analyzing event propagation on Twitter, and enhancing network resilience to cyber attacks.

1.2 Highlights of the Dissertation

This dissertation addresses three topics in graph data analytics:

1. Fundamental limits of spectral algorithms for graph clustering, including performance analysis of spectral clustering and spectral modularity methods in single-layer and multilayer weighted graphs. (Chapters III, IV, and VI)

2. Efficient algorithms for actions on graphs, including incremental eigenpair computation of graph Laplacian matrices, automated model order selection methods for graph clustering in single-layer and multilayer graphs, and greedy approaches for insertions and deletions of nodes or edges with theoretic guarantees. (Chapters II, V, VI, VII, and X)

3. Applications to community detection, event propagation in online social networks, and cyber security, including deep community extraction, event propagation on Twitter, and topological network resilience to explicit and implicit attacks. (Chapters VII, VIII, IX, and X)

1.3 Matrix Representations for Graphs

One common methodology of graph data analytics is to represent the data of interest as a graph for inference and processing, where a node represents an entity (e.g., a pixel in an image or a user in a social network), and an edge represents similarity (e.g., a distance metric between two multivariate data samples) or actual relation (e.g., friendship) between nodes. Therefore, graphs are useful representations that characterize explicit relations for relational data (e.g., friendship between users in a social network), or implicit dependencies for attributional data (e.g., correlations between multivariate data samples in a dataset).

Mathematically, a graph consisting of n nodes and m edges is denoted by $G =$

$(\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes with cardinality $|\mathcal{V}| = n$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges with cardinality $|\mathcal{E}| = m$, and $\mathbf{W}$ is the $n \times n$ nonnegative matrix of edge weights. Throughout this dissertation we assume there is at most one edge between any ordered node pairs, and all edge weights are positive. The connectivity structure of G is characterized by an $n \times n$ adjacency matrix $\mathbf{A}$, where its entry $[\mathbf{A}]_{ij} = 1$ if there is a directed edge connecting from node i to node j, and $[\mathbf{A}]_{ij} = 0$ otherwise. If G is undirected, then an edge $(i, j) \in \mathcal{E}$ means that $[\mathbf{A}]_{ij} = [\mathbf{A}]_{ji} = 1$. The entry of the weight matrix $[\mathbf{W}]_{ij} > 0$ if $[\mathbf{A}]_{ij} = 1$, and $[\mathbf{W}]_{ij} = 0$ otherwise. Therefore, for undirected graphs $\mathbf{W}$ and $\mathbf{A}$ are symmetric matrices, and for unweighted graphs $\mathbf{W} = \mathbf{A}$.

Throughout this dissertation bold uppercase letters (e.g., $\mathbf{X}$ or $\mathbf{X}_k$) denote matrices and $[\mathbf{X}]_{ij}$ denotes the entry of the i-th row and the j-th column of $\mathbf{X}$, bold lowercase letters (e.g., $\mathbf{x}$ or $\mathbf{x}_i$) denote column vectors, $(\cdot)^T$ denotes matrix or vector transpose, italic letters (e.g., x or x_i) denote scalars, and calligraphic uppercase letters (e.g., $\mathcal{X}$ or $\mathcal{X}_i$) denote sets. The $n \times 1$ vector of ones (zeros) is denoted by $\mathbf{1}_n$ ($\mathbf{0}_n$). The matrix $\mathbf{I}$ denotes an identity matrix and the matrix $\mathbf{O}$ denotes the matrix of zeros. The notation $\lambda_k(\mathbf{X})$ denotes the k-th smallest eigenvalue (in absolute value) of a square matrix $\mathbf{X}$, and its associated eigenvector is called the k-th smallest eigenvector. The notation $\sigma_k(\mathbf{X})$ denotes the k-th largest singular value of a rectangular matrix $\mathbf{X}$, and its associated left (right) singular vector is called the k-th largest left (right) singular vector.

1.3.1 Block model for graph data

Here we introduce a block model for graph data, where each block either characterizes the connectivity structure within one cluster or between two clusters. The block model not only allows us to generate synthetic graphs with ground-truth cluster assignment for performance evaluation, but also provides parametric network models for graph data analysis. Without loss of generality, we assume there are K clusters in the graph, and cluster k has n_k nodes and m_k edges such that $\sum_{k=1}^{K} n_k = n$ and $\sum_{k=1}^{K} m_k = m$. For analysis purposes, we reorder the nodes in the graph such that the adjacency matrix $\mathbf{A}$ of the entire graph has block-wise connectivity structure, which is represented as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{C}_{12} & \mathbf{C}_{13} & \cdots & \mathbf{C}_{1K} \\ \mathbf{C}_{21} & \mathbf{A}_2 & \mathbf{C}_{23} & \cdots & \mathbf{C}_{2K} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{K1} & \mathbf{C}_{K2} & \cdots & \cdots & \mathbf{A}_K \end{bmatrix}. \tag{1.1}$$

We call the matrix representation in (1.1) a *block model* for graph data. The block matrix $\mathbf{A}_k$ is the adjacency matrix of within-cluster edges of cluster k, and the block matrix $\mathbf{C}_{ij}$ is the adjacency matrix of between-cluster edges between clusters i and j. For undirected graphs, $\mathbf{A}$ is symmetric, $\mathbf{A}_k = \mathbf{A}_k^T$ and $\mathbf{C}_{ij} = \mathbf{C}_{ji}^T$. Similar block models can be defined for the weight matrix $\mathbf{W}$.

A popular block model for undirected graph data is the stochastic block model (SBM) [74]. SBM assumes each entry in $\mathbf{A}$ is random and mutually independent, and the entry in $\mathbf{A}_k$ ($\mathbf{C}_{ij}$, $i \neq j$) is a Bernoulli(p_k) (Bernoulli(p_{ij})) random variable. Other statistical network models can be found in the recent survey paper [65].

1.3.2 A signal plus noise perspective

Throughout this dissertation, the established phase transition analysis on the eigendecomposition of different matrices representing a graph is based on a *signal plus noise block model*, where the within-cluster connections are viewed as signal and the between-cluster connections are viewed as noise. For the purpose of illustration, Fig. 1.1 shows the connectivity structure of a graph generated from a block model

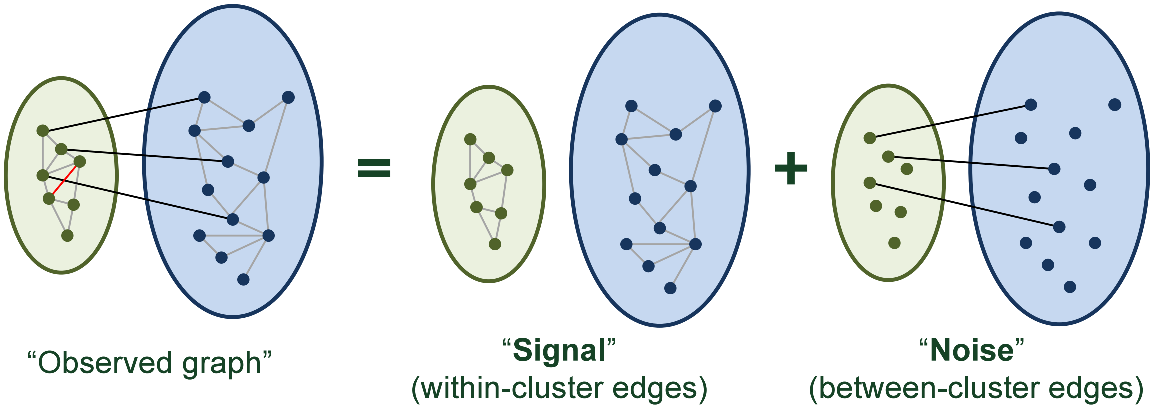


Figure 1.1: An illustration of the connectivity structure of a graph generated from a block model with $K = 2$ clusters.

with $K = 2$ clusters. Fixing the signal and varying the noise, we are interested in the behavior of eigendecomposition of different matrices for graph data analytics.

1.3.3 Graph Laplacian matrices and their properties

Graph Laplacian matrices are widely used for graph data analysis due to their special matrix properties and their close relation to graph-cut based metrics. For undirected weighted graphs, the (unnormalized) graph Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{S} - \mathbf{W}, \tag{1.2}$$

where $\mathbf{S} = \text{diag}(s_1, s_2, \ldots, s_n)$ is the diagonal matrix of nodal strengths (i.e., weighted degrees) and $s_i = \sum_{j=1}^{n} [\mathbf{W}]_{ij}$ is the strength of node i. In particular, for undirected unweighted graphs, $\mathbf{S} = \mathbf{D}$, where $\mathbf{D} = \text{diag}(d_1, d_2, \ldots, d_n)$ is the diagonal matrix of degrees, and $d_i = \sum_{j=1}^{n} [\mathbf{A}]_{ij}$ is the degree of node i.

The graph Laplacian matrix $\mathbf{L}$ has the following properties:

1. $\mathbf{1}_n$ is in the null space of $\mathbf{L}$, i.e., $\mathbf{L}\mathbf{1}_n = \mathbf{0}_n$.

2. $\mathbf{L}$ is positive semidefinite (PSD) and $\lambda_1(\mathbf{L}) = 0$, i.e., $0 = \lambda_1(\mathbf{L}) \leq \cdots \leq \lambda_n(\mathbf{L})$.

3. The number of connected components in G is the number of zero eigenvalues

of $\mathbf{L}$.

4. For any non-complete undirected unweighted graph G, the second smallest eigenvalue of $\mathbf{L}$, $\lambda_2(\mathbf{L})$, also known as the algebraic connectivity of G, is a lower bound on node and edge connectivity [55].

One popular variant of the graph Laplacian matrix is the normalized graph Laplacian matrix, which is defined as

$$\mathbf{L}_{\mathcal{N}} = \mathbf{S}^{-\frac{1}{2}}\mathbf{L}\mathbf{S}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{S}^{-\frac{1}{2}}\mathbf{W}\mathbf{S}^{-\frac{1}{2}}, \tag{1.3}$$

where $\mathbf{S}$ is assumed to be invertible and $\mathbf{S}^{-\frac{1}{2}} = \operatorname{diag}(\frac{1}{\sqrt{s_1}}, \frac{1}{\sqrt{s_2}}, \ldots, \frac{1}{\sqrt{s_n}})$. Interested readers can refer to [38, 98] and the references therein for more details.

It is worth mentioning that over the past two decades, the graph Laplacian matrix and its variants have been widely adopted for solving various research tasks, including graph partitioning [128], data clustering [97], community detection [25, 166], consensus in networks [114], dimensionality reduction [12], entity disambiguation [178], graph signal processing [145], centrality measures for graph connectivity [24], interconnected physical systems [133], network vulnerability assessment [27], image segmentation [144], among others.

1.3.4 Spectral clustering

In many graph clustering tasks, *spectral clustering* methods [97, 108, 166, 176] are used for clustering nodes in the graph by inspecting the eigenstructure of $\mathbf{L}$. To partition the nodes in the graph into K ($K \geq 2$) clusters, spectral clustering [97] uses the K eigenvectors associated with the K smallest eigenvalues of $\mathbf{L}$. Each node can be viewed as a K-dimensional vector in the subspace spanned by these eigenvectors. K-means clustering [72] is then implemented on the K-dimensional vectors to group the nodes into K clusters. Vector normalization of the obtained

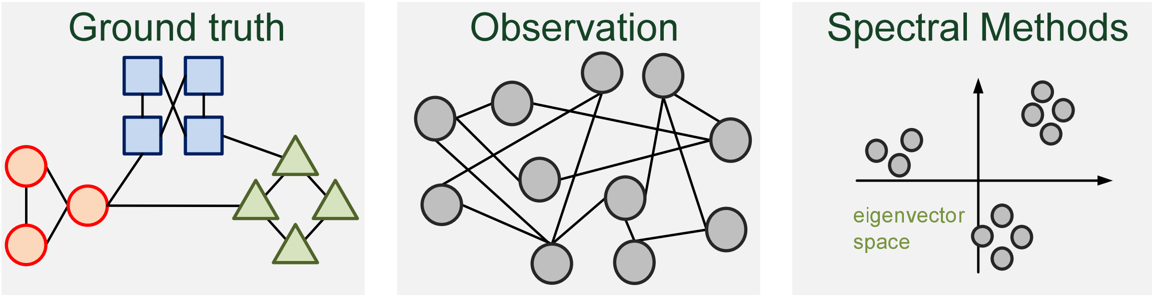


Figure 1.2: An illustration of graph spectral decomposition methods for graph clustering. Graph spectral decomposition methods transform the observed graph into a representation in a low-dimensional vector space to reveal the ground-truth clusters.

K-dimensional vectors or degree normalization of the adjacency matrix can be used to stabilize K-means clustering [97, 108, 176]. Fig. 1.2 illustrates the methodology of graph spectral decomposition methods for graph clustering, which includes spectral clustering. Graph spectral decomposition methods transform the observed graph into a representation in a low-dimensional vector space to reveal the ground-truth clusters.

The success of spectral clustering can be explained by the fact that acquiring K smallest eigenvectors of $\mathbf{L}$ is equivalent to solving a relaxed graph cut minimization problem, which partitions a graph into K clusters by minimizing various objective functions including min cut, ratio cut or normalized cut [97].

1.4 Overview of Graph Clustering Methods

Broadly speaking, actions on graphs can be viewed as various means for optimizing an objective function associated with a data analysis task. For graph clustering, the fundamental task is to partition the nodes in a graph into groups based on the graph connectivity structure. Different graph clustering methods lead to different edge removal strategies that uncover the cluster structures, and the associated cost often relates to the total edge weight of the removed edges, which is known as the min-cut score. In particular, one principal method for graph clustering is through graph spectral decompositions.

Throughout this dissertation we will use the terms “graph clustering” and “community detection” interchangeably, as they both aim at clustering nodes in a graph but may target different objective functions based on the data features, e.g., a similarity graph of an image, or a friendship graph of a social network.

1.4.1 Graph clustering methods and analysis for single-layer graphs

Graph clustering has been an active research field across various disciplines, including machine learning, physics, data mining, graph signal processing, computer science, bio-informatics, network analysis, and data science. Here we categorize the related work based on the methodologies for graph clustering. Interested readers can refer to the [56] and the references therein for more details.

- **Spectral algorithms.** In principle, spectral algorithms utilize the eigenstructure of matrices associated with the graph data for clustering. For instance, spectral clustering [97, 108, 166, 176] uses the graph Laplacian matrix, the spectral modularity method [105, 106] uses the modularity matrix, and the spectral redemption method [87, 139] uses the nonbacktracking matrix.

- **Inference methods.** Inference methods are built upon generative network models such as the stochastic block model (SBM) [74] and its variants [4]. The task is to infer the cluster assignment for each node based on the graph connectivity structure. In [80], the authors infer clusters based on a maximum likelihood method. Other methods such as belief propagation and message passing are proposed in [47, 73, 179].

- **Hierarchical methods.** Hierarchical methods can often relate to clustering based on dendrograms. Popular methods for creating dendrograms include removing edges of high betweenness measure [63], greedy modularity maximization approaches [18, 104], label propagation [135], and node removals based on centrality measures [165].

- **Random walk based methods.** Random walk based methods specify a transition matrix of random walks on a graph and use its stationary distribution of random walks for clustering. Typical methods can be found in [125, 160].

- **Performance limits for graph clustering.** Recently there is growing interest in understanding the universal limits of graph clustering, as well as performance limits of specific graph clustering methods. Most of the works assume the graph data is a realization of a generative network model, such as the SBM [181]. For instance, the performance limit of the spectral modularity method is studied in [102], the performance limit of eigenspectrum-based approaches is studied in [122], and the information theoretic limit is studied in [1]. A summary of inferential limits under the SBM can be found in [2].

1.4.2 Model order selection

A long-standing challenge for graph clustering is the selection of cluster counts K for partitioning. For spectral clustering, it is equivalent to selecting the number K of smallest eigenvectors of $\mathbf{L}$ that best fits the data, which we call it as the model order selection problem. Most existing model selection algorithms specify an upper bound $K_{\max}$ on the number K of clusters and then select K based on optimizing some objective function, e.g., the goodness of fit of the k-cluster model for $k = 2, \ldots, K_{\max}$. In [108], the objective is to minimize the sum of cluster-wise Euclidean distances between each data point and the centroid obtained from K-means clustering. In [124], the objective is to maximize the gap between the K-th largest and the $(K+1)$-th largest eigenvalue. In [176], the authors propose to minimize an objective function that is associated with the cost of aligning the eigenvectors with a canonical coordinate system. A model based method for determining the number of clusters is proposed in [126]. In [105], the authors propose to iteratively divide a cluster based on the leading eigenvector of the modularity matrix until no significant improvement in the modularity measure can be achieved. The Louvain method in [18] uses a greedy algorithm for modularity maximization. In [87, 139], the authors propose to use the eigenvectors of the nonbacktracking matrix for graph clustering, where the number of clusters is determined by the number of real eigenvalues with magnitude larger than the square root of the largest eigenvalue.

1.4.3 Graph clustering methods and analysis for multilayer graphs

Graph clustering on multilayer graphs aims to find a consensus cluster assignment on each node in the common node set shared by different layers. Layer aggregation has been a principal method for processing and mining multilayer graphs [20, 46, 153, 154, 156, 168], as it transforms a multilayer graph into a single aggregated graph, facilitating application of data analysis techniques designed for single-layer graphs. Extending from the stochastic block model (SBM) for graph clustering in single-layer graphs [74], multilayer SBM has been proposed for graph clustering on multilayer graphs [10, 71, 121, 149, 156, 159]. Under the assumption of two equally-sized clusters, the authors in [156] show that if each layer is an independent realization of a common SBM, the inferential limit for cluster detectability decays with $O(L^{-\frac{1}{2}})$, where L is the number of layers. In [149], a layer selection method based on a multilayer SBM is proposed to improve the performance of graph clustering by identifying a subset of layers of a common SBM. However, multilayer SBM assumes homogeneous connectivity structure for within-cluster and between-cluster edges in each layer, and it assumes layer-wise independence.

In addition to inference approaches based on multilayer SBM, other methods have been proposed for graph clustering on multilayer graphs, including information-theoretic approaches [77, 117], k-nearest neighbor method [67], nonnegative matrix factorization [110], flow-based approach [45], linked matrix factorization [155], random walk [89], tensor decomposition [43], subspace methods [51, 52], and greedy multilayer modularity maximization [101]. More details on multilayer graph models can be found in the recent surveys on graph clustering on multilayer graphs [81, 84].

It is worth mentioning that the aforementioned work requires the knowledge of the number of clusters (model order) for graph clustering, especially for matrix decomposition-based methods [43, 51, 52, 110, 155] and multilayer SBM [10, 71, 121, 149, 156, 159]. However, in many practical cases the model order is not known in advance. Although many model order selection methods have been proposed for automated graph clustering without prespecifying the model order for single-layer graphs [18, 31, 87, 176], little has been developed for automated model order selection for graph clustering on multilayer graphs. Moreover, many layer aggregation methods assign uniform weights over layers such that the aggregated graph is insensitive to the quality of clusters in each layer [153, 154, 156].

1.5 Overview of Node Centrality Measures

Node and edge centralities are quantitative measures that are used to evaluate the level of importance and/or influence of a node or an edge in the network. Centrality measures can be classified into two categories, *global* and *local* measures. Global centrality measures require complete topological information for their computation, whereas local centrality measures only require partial topological information from neighboring nodes. For instance, acquiring shortest path information between every node pair is a global method required for the betweenness centrality measure, and acquiring degree information of every node is a local method. Some commonly used centrality measures are:

- **Betweenness** [58]: betweenness measures the fraction of shortest paths passing through a node relative to the total number of shortest paths in the network. Specifically, betweenness is a global measure defined as betweenness$(i) = \sum_{k \neq i} \sum_{j \neq i, j > k} \frac{\phi_{kj}(i)}{\phi_{kj}}$, where ϕ_{kj} is the total number of shortest paths from k to j and $\phi_{kj}(i)$ is the number of such shortest paths passing through i. A similar notion is used to define the edge betweenness centrality [63].

- **Closeness** [140]: closeness is a global measure of geodesic distance of a node to all other nodes. A node is said to have high closeness if the sum of its shortest path distances to other nodes is small. Let $\rho(i,j)$ denote the shortest path distance between node i and node j in a connected graph. Then we define closeness$(i) = 1/\sum_{j \in \mathcal{V}, j \neq i} \rho(i,j)$.

- **Eigenvector centrality** (eigen centrality) [107]: eigenvector centrality is the i-th entry of the eigenvector associated with the largest eigenvalue of the adjacency matrix $\mathbf{A}$. It is defined as eigen$(i) = \lambda_{\max}^{-1} \sum_{j \in \mathcal{V}} \mathbf{W}_{ji} \boldsymbol{\xi}_j$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{A}$ and $\boldsymbol{\xi}$ is the left eigenvector associated with $\lambda_{\max}$. It is a global measure since eigenvalue decomposition of $\mathbf{A}$ requires global knowledge of the graph topology.

- **Degree** (d_i): degree is the simplest local node centrality measure which accounts for the number of neighboring nodes.

- **Ego centrality** [54]: consider the (d_i+1)-by-(d_i+1) local adjacency matrix of node i, denoted by $\mathbf{A}(i)$, and let $\mathbf{R}$ be a matrix of ones. Ego centrality can be viewed as a local version of betweenness that computes the shortest paths between its neighboring nodes. Since $[\mathbf{A}^2(i)]_{kj}$ is the number of two-hop walks between k and j, and $\left[\mathbf{A}^2(i) \circ \left(\mathbf{E} - \mathbf{A}(i)\right)\right]_{kj}$ is the total number of two-hop shortest paths between k and j for all $k \neq j$, where $\circ$ denotes entrywise matrix product. Ego centrality is defined as ego$(i) = \sum_k \sum_{j>k} 1/\left[\mathbf{A}^2(i) \circ \left(\mathbf{E} - \mathbf{A}(i)\right)\right]_{kj}$.

Chapters	Graph spectral decompositions	Insertions/Removals of nodes/edges
I		
II	✓	
III	✓	
IV	✓	
V	✓	
VI	✓	
VII	✓	✓
VIII	✓	✓
IX	✓	✓
X	✓	✓
XI		

Table 1.1: Dissertation outline with respect to two actions on graphs

1.6 Dissertation Outline and Contributions

The dissertation outline with respect to two actions on graphs, graph spectral decompositions and insertions and removals of nodes or edges, is listed in Table 1.1. The contributions of each chapter are summarized as follows.

- Chapter II proposes an efficient incremental eigenpair computation method for graph Laplacian matrices. The proposed method adapts a novel matrix transform that enables fast incremental eigenpair computation of increasing eigenpair orders. In particular, the proposed method significantly outperforms the batch computation method in terms of computation time. This incremental eigenpair computation method can be readily applied to iterative graph clustering of increasing orders in the following chapters.

- Chapter III establishes phase transition analysis of the spectral modularity method under the stochastic block model (SBM) with $K = 2$ clusters. The phase transition results are shown to be universal in the sense that the critical threshold affecting the performance of spectral modularity method is independent of the clusters sizes as long as they grow with comparable rates.

- Chapter IV establishes phase transition analysis of spectral graph clustering

(SGC) under the random interconnection model (RIM). The analysis identifies the role of inter-cluster edge connection probabilities in the success of SGC. Specifically, we show that under the RIM, a graph can be separated into two regimes: a regime where SGC is successful, and a regime where SGC is unsuccessful. It is called a phase transition since the regimes are separated by a critical value of the inter-cluster inter-cluster edge connection probability. The phase transition analysis is then extended to undirected weighted graphs generated by the RIM.

- Chapter V develops an automated model order selection (AMOS) algorithm for determining the number of clusters in single-layer weighted graphs based on the phase transition analysis established in Chapter IV. AMOS works by iterative spectral graph clustering of increasing model orders, and finds the minimal model order such that the identified clusters meet the phase transition criterion for clustering reliability. AMOS also provides statistical clustering reliability guarantees for graph data inference. Experimental results on several real-life datasets show that AMOS can produce consistent clusters when compared with the meta information, e.g., ground-truth clusters or geographical separations.

- Chapter VI establishes phase transition analysis of spectral graph clustering in multi-layer graphs via convex layer aggregation. Based on the phase transition analysis, a multilayer iterative model order selection algorithm (MIMOSA) is proposed for model selection and layer weight adaption for graph clustering in multilayer graphs with statistical clustering reliability. The success of MIMOSA can be explained by a multilayer signal plus noise model since its layer weight adaption process is sensitive to the noise level of each layer.

- Chapter VII introduces a new centrality measure called local Fielder vector centrality (LFVC). Stemming from algebraic connectivity minimization via node/edge removals, LFVC evaluates the importance of a node or an edge for graph connectivity. In particular, we show that greedy node removals based on LFVC have bounded performance guarantee relative to the optimal batch removals. Based on LFVC removals, a deep community detection algorithm is proposed to extract important communities from the graph.

- Chapter VIII proposes a method for identifying influential links for event propagation on Twitter, also applicable to other social network applications, where the influence of a link is evaluated in terms of the effect of its removal on event propagation. The proposed method incorporates the network of networks structure embedded in Twitter follower networks, and can effectively identify influential links in several actual event propagation traces.

- Chapter IX proposes an edge rewiring algorithm for enhancing network resilience to centrality attacks. The proposed method works by swapping edges in the graph for improved algebraic connectivity without introducing additional edges, and it can be implemented in a distributed fashion. Experimental results on real-like network show that the proposed method can effectively enhance the network resilience by only rewiring a few edges in the graph.

- Chapter X develops a graph-theoretic framework for cyber resilience against lateral movement attacks. By modeling the interactions among user, hosts, and applications in an enterprise network as a tripartite heterogeneous graph and mapping feasible preventative actions to operations on the associated graph matrices, we propose greedy algorithms with performance guarantees to enhance enterprise cyber resiliency. Experimental results show that the proposed methods can greatly contain simulated lateral movement attacks in actual enterprise networks and actual lateral movement attacks extracted from real-world traces.

1.7 List of Relevant Publications

1.7.1 Journal publications

- P.-Y. Chen and A. O. Hero, “Deep Community Detection,” *IEEE Transactions on Signal Processing*, vol. 63, no. 21, pp. 5706-5719, Nov. 2015 [24]

- P.-Y. Chen and A. O. Hero, “Phase Transitions in Spectral Community Detection,” *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4339-4347, Aug. 2015 [25]

- P.-Y. Chen and A. O. Hero, “Universal Phase Transition in Community Detectability under a Stochastic Block Model,” *Physical Review E*, vol. 91, no.3, pp. 032804, Mar. 2015 [29]

- P.-Y. Chen and A. O. Hero, “Assessing and Safeguarding Network Resilience to Nodal Attacks,” *IEEE Communications Magazine*, vol. 52, no. 11, pp. 138-143, Nov. 2014 [27]

1.7.2 Conference publications

- P.-Y. Chen and A. O. Hero, “Multilayer Spectral Graph Clustering via Convex Layer Aggregation,” *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2016

- P.-Y. Chen, B. Zhang, M. Hasan and A. O. Hero, “Incremental Method for Spectral Clustering of Increasing Orders,” *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) Workshop on Mining and Learning with Graphs (MLG)*, Aug. 2016 [36]

- P.-Y. Chen, S. Choudhury, and A. O. Hero, “Multi-Centrality Graph Spectral Decompositions and Their Application to Cyber Intrusion Detection,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4553-4557, Mar. 2016 [34]

- P.-Y. Chen and A. O. Hero, "Phase Transitions in Spectral Community Detection of Large Noisy Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3402-3406, Apr. 2015 [30]

- P.-Y. Chen and A. O. Hero, "Local Fiedler Vector Centrality for Detection of Deep and Overlapping Communities in Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1120-1124, May 2014 [28]

- P.-Y. Chen and A. O. Hero, "Node Removal Vulnerability of the Largest Component of a Network," *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 587-590. Dec. 2013 [26]

1.7.3 Submitted papers

- P.-Y. Chen and A. O. Hero, "Multilayer Spectral Graph Clustering via Convex Layer Aggregation: Theory and Algorithm," 2016

- P.-Y. Chen, C.-C. Tu, P.-S. Ting, Y.-Y. Luo, D. Koutra, and A. O. Hero, "Identifying Influential Links for Event Propagation on Twitter: A Network of Networks Approach," 2016 [35]

- P.-Y. Chen, Thibaut Gensollen, and A. O. Hero, "AMOS: An Automated Model Order Selection Algorithm for Spectral Graph Clustering," 2016

- P.-Y. Chen, S. Sutanay, L. Rodriguez, A. O. Hero, and I. Ray, "Enterprise Cyber Resiliency Against Lateral Movement: A Graph Theoretic Approach," 2016

- P.-Y. Chen and A. O. Hero, "Phase Transitions and a Model Order Selection Criterion for Spectral Graph Clustering," 2016 [31]

CHAPTER II

Incremental Method for Spectral Graph Clustering of Increasing Orders

As introduced in Chapter I, the smallest eigenvalues and the associated eigenvectors (i.e., the smallest eigenpairs) of a graph Laplacian matrix have been widely used for spectral clustering and community detection. However, the majority of applications require computation of a large number $K_{\max}$ of eigenpairs, where $K_{\max}$ is an upper bound on the number K of clusters, called the order of the clustering algorithms. In this chapter, we propose an incremental method for constructing the eigenspectrum of the graph Laplacian matrix. This method leverages the eigenstructure of graph Laplacian matrix to obtain the $(k+1)$-th eigenpairs of the Laplacian matrix given a collection of all the k smallest eigenpairs. Our proposed method adapts the Laplacian matrix such that the batch eigenvalue decomposition problem transforms into an efficient sequential leading eigenpair computation problem.

Generally, the number of clusters K is selected to be much smaller than n (the number of data points), making full eigen decomposition (such as QR decomposition) unnecessary. An efficient alternative is to use methods that are based on power iteration, such as Arnoldi method or Lanczos method, which computes the leading eigenpairs through repeated matrix vector multiplication. ARPACK [94] library is a popular parallel implementation of different variants of Arnoldi and Lanczos method, which is used by many commercial software including Matlab.

However, in most situations the best value of K is unknown and a heuristic is used by clustering algorithms to determine the number of clusters, e.g., fixing a maximum number of clusters $K_{\max}$ and detecting a large gap in the values of the $K_{\max}$ largest sorted eigenvalues or normalized cut score [108, 124]. Alternatively, this value of K can be determined based on domain knowledge [11]. For example, a user may require that the largest cluster size be no more than 10% of the total number of nodes or that the total inter-cluster edge weight be no greater than a certain amount. In these cases, the desired choice of K cannot be determined *a priori*. Over-estimation of the upper bound $K_{\max}$ on the number of clusters is expensive as the cost of finding K eigenpairs using the power iteration method grows rapidly with K. On the other hand, choosing an insufficiently large value for $K_{\max}$ runs the risk of severe bias. Setting $K_{\max}$ to the number of data points n is generally computationally infeasible, even for a moderate-sized graph. Therefore, an incremental eigenpair computation method that effectively computes the K-th smallest eigenpair of graph Laplacian matrix by utilizing the previously computed $K-1$ smallest eigenpairs is needed. Such an iterative method obviates the need to set an upper bound $K_{\max}$ on K, and its efficiency can be explained by the adaptivity to increments in K.

In this chapter, we propose an efficient method for incremental computation of smallest eigenpairs by exploiting the eigenspace structure of graph Laplacian matrices. For each increment, given the previously computed smallest eigenpairs, we show that computing the next smallest eigenpair is equivalent to computing a leading eigenpair of a particular matrix, which transforms potentially tedious numerical computation (such as the tridiagonalization step in Lanczos algorithm [90]) to simple matrix power iterations of known computational efficiency [90]. Our experimental results show that for a given K, the proposed incremental computation provides a significant reduction in computation time compared to a batch computation method which computes the K smallest eigenpairs in a single batch. Also, as K increases, the gap between the incremental approach and the batch approach widens, providing an order of magnitude speed-up.

It is worth noting that the proposed method aims to incrementally compute the smallest eigenpair of a given graph Laplacian matrix. There are several works that are named as incremental eigenvalue decomposition methods [50, 79, 111, 112, 136]. However, these works focus on updating the eigenstructure of graph Laplacian matrix of dynamic graphs when nodes (data samples) or edges are inserted or deleted into the graph.

2.1 Incremental Eigenpair Computation for Graph Laplacian Matrices

2.1.1 Notation for eigenpairs

The i-th smallest eigenvalue and its associated unit-norm eigenvector of $\mathbf{L}$ are denoted by $\lambda_i(\mathbf{L})$ and $\mathbf{v}_i(\mathbf{L})$, respectively. That is, the eigenpair $(\lambda_i, \mathbf{v}_i)$ of $\mathbf{L}$ has the relation $\mathbf{L}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, and $\lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}) \leq \ldots \leq \lambda_n(\mathbf{L})$. The eigenvectors have unit Euclidean norm and they are orthogonal to each other such that $\mathbf{v}_i^T \mathbf{v}_j = 1$ if $i = j$ and $\mathbf{v}_i^T \mathbf{v}_j = 0$ if $i \neq j$. The eigenvalues of $\mathbf{L}$ are said to be distinct if $\lambda_1(\mathbf{L}) < \lambda_2(\mathbf{L}) < \ldots < \lambda_n(\mathbf{L})$. Similar notation is used for $\mathbf{L}_{\mathcal{N}}$.

2.1.2 Theoretical foundations of the proposed method

The following lemmas and corollaries provide the cornerstone for establishing the incremental computation method. The main idea is that we utilize the eigenspace structure of graph Laplacian matrix to inflate specific eigenpairs via a particular perturbation matrix, without affecting other eigenpairs. The proposed method can be viewed as a specialized Hotelling's deflation method [118] designed for graph Lapla-

Type / Graph Laplacian	Unnormalized	Normalized
Connected Graphs	Lemma 2.1, Theorem 2.6	Corollary 2.2, Corollary 2.8
Disconnected Graphs	Lemma 2.3, Theorem 2.7	Corollary 2.4, Corollary 2.9

Table 2.1: Utility of the established lemmas, corollaries, and theorems in Chapter II.

cian matrices by exploiting their spectral properties and associated graph characteristics. It works for both connected, and disconnected graphs using both normalized and unnormalized graph Laplacian matrices. For illustration purposes, in Table 2.1 we group the established lemmas, corollaries, and theorems under different graph types and different graph Laplacian matrices. The proofs of the established lemmas, theorems and corollaries are given in Appendix A.

Lemma 2.1. *Assume that G is a connected graph and $\mathbf{L}$ is the graph Laplacian with s_i denoting the sum of the entries in the i-th row of the weight matrix $\mathbf{W}$. Let $s = \sum_{i=1}^{n} s_i$ and define $\widetilde{\mathbf{L}} = \mathbf{L} + \frac{s}{n}\mathbf{1}_n\mathbf{1}_n^T$. Then the eigenpairs of $\widetilde{\mathbf{L}}$ satisfy $(\lambda_i(\widetilde{\mathbf{L}}), \mathbf{v}_i(\widetilde{\mathbf{L}})) = (\lambda_{i+1}(\mathbf{L}), \mathbf{v}_{i+1}(\mathbf{L}))$ for $1 \leq i \leq n-1$ and $(\lambda_n(\widetilde{\mathbf{L}}), \mathbf{v}_n(\widetilde{\mathbf{L}})) = (s, \frac{\mathbf{1}_n}{\sqrt{n}})$.*

Corollary 2.2. *For a normalized graph Laplacian matrix $\mathbf{L}_{\mathcal{N}}$, assume G is a connected graph and let $\widetilde{\mathbf{L}}_{\mathcal{N}} = \mathbf{L}_{\mathcal{N}} + \frac{2}{s}\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n\mathbf{1}_n^T\mathbf{S}^{\frac{1}{2}}$. Then $(\lambda_i(\widetilde{\mathbf{L}}_{\mathcal{N}}), \mathbf{v}_i(\widetilde{\mathbf{L}}_{\mathcal{N}})) = (\lambda_{i+1}(\mathbf{L}_{\mathcal{N}}), \mathbf{v}_{i+1}(\mathbf{L}_{\mathcal{N}}))$ for $1 \leq i \leq n-1$ and $(\lambda_n(\widetilde{\mathbf{L}}_{\mathcal{N}}), \mathbf{v}_n(\widetilde{\mathbf{L}}_{\mathcal{N}})) = (2, \frac{\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n}{\sqrt{s}})$.*

Lemma 2.3. *Assume that G is a disconnected graph with $\delta \geq 2$ connected components, and let $s = \sum_{i=1}^{n} s_i$, let $\mathbf{V} = [\mathbf{v}_1(\mathbf{L}), \mathbf{v}_2(\mathbf{L}), \ldots, \mathbf{v}_\delta(\mathbf{L})]$, and let $\widetilde{\mathbf{L}} = \mathbf{L} + s\mathbf{V}\mathbf{V}^T$. Then $(\lambda_i(\widetilde{\mathbf{L}}), \mathbf{v}_i(\widetilde{\mathbf{L}})) = (\lambda_{i+\delta}(\mathbf{L}), \mathbf{v}_{i+\delta}(\mathbf{L}))$ for $1 \leq i \leq n-\delta$, $\lambda_i(\widetilde{\mathbf{L}}) = s$ for $n-\delta+1 \leq i \leq n$, and $[\mathbf{v}_{n-\delta+1}(\widetilde{\mathbf{L}}), \mathbf{v}_{n-\delta+2}, (\widetilde{\mathbf{L}}),$
$\ldots, \mathbf{v}_n(\widetilde{\mathbf{L}})] = \mathbf{V}$.*

Corollary 2.4. *For a normalized graph Laplacian matrix $\mathbf{L}_{\mathcal{N}}$, assume G is a disconnected graph with $\delta \geq 2$ connected components. Let $\mathbf{V}_\delta = [\mathbf{v}_1(\mathbf{L}_{\mathcal{N}}), \mathbf{v}_2(\mathbf{L}_{\mathcal{N}}), \ldots, \mathbf{v}_\delta(\mathbf{L}_{\mathcal{N}})]$, and let $\widetilde{\mathbf{L}}_{\mathcal{N}} = \mathbf{L}_{\mathcal{N}} + 2\mathbf{V}_\delta\mathbf{V}_\delta^T$. Then $(\lambda_i(\widetilde{\mathbf{L}}_{\mathcal{N}}), \mathbf{v}_i(\widetilde{\mathbf{L}}_{\mathcal{N}})) = (\lambda_{i+\delta}(\mathbf{L}_{\mathcal{N}}), \mathbf{v}_{i+\delta}(\mathbf{L}_{\mathcal{N}}))$ for $1 \leq i \leq n-\delta$, $\lambda_i(\widetilde{\mathbf{L}}_{\mathcal{N}}) = 2$ for $n-\delta+1 \leq i \leq n$, and $[\mathbf{v}_{n-\delta+1}(\widetilde{\mathbf{L}}_{\mathcal{N}}), \mathbf{v}_{n-\delta+2}, (\widetilde{\mathbf{L}}_{\mathcal{N}}), \ldots, \mathbf{v}_n(\widetilde{\mathbf{L}}_{\mathcal{N}})] = \mathbf{V}_\delta$.*

Remark 2.5. The columns of any matrix $\mathbf{V}' = \mathbf{V}\mathbf{R}$ with an orthonormal transformation matrix $\mathbf{R}$ (i.e., $\mathbf{R}^T\mathbf{R} = \mathbf{I}$) are also the largest δ eigenvectors of $\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{L}}_{\mathcal{N}}$ in Lemma 2.3 and Corollary 2.4. Without loss of generality we consider the case $\mathbf{R} = \mathbf{I}$.

2.1.3 Incremental eigenpair computation method

Given the K smallest eigenpairs of a graph Laplacian matrix, we prove that computing the $(K+1)$-th smallest eigenpair is equivalent to computing the leading eigenpair (the eigenpair with the largest eigenvalue in absolute value) of a certain perturbed matrix. The advantage of this transformation is that the leading eigenpair can be efficiently computed via matrix power iteration methods [90, 94].

Let $\mathbf{V}_K = [\mathbf{v}_1(\mathbf{L}), \mathbf{v}_2(\mathbf{L}), \ldots, \mathbf{v}_K(\mathbf{L})]$ be the matrix with columns being the K smallest eigenvectors of $\mathbf{L}$ and let $\mathbf{\Lambda}_K = \text{diag}(s-\lambda_1(\mathbf{L}), s-\lambda_2(\mathbf{L}), \ldots, s-\lambda_K(\mathbf{L}))$ be the diagonal matrix with $\{s - \lambda_i(\mathbf{L})\}_{i=1}^K$ being its main diagonal. The following theorems show that given the K smallest eigenpairs of $\mathbf{L}$, the $(K+1)$-th smallest eigenpair of $\mathbf{L}$ is the leading eigenvector of the original graph Laplacian matrix perturbed by a certain matrix.

Theorem 2.6. (connected graphs) *Given $\mathbf{V}_K$ and $\mathbf{\Lambda}_K$, assume that G is a connected graph. Then the eigenpair $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L}))$ is a leading eigenpair of the matrix $\widetilde{\mathbf{L}} = \mathbf{L} + \mathbf{V}_K\mathbf{\Lambda}_K\mathbf{V}_K^T + \frac{s}{n}\mathbf{1}_n\mathbf{1}_n^T - s\mathbf{I}$. In particular, if $\mathbf{L}$ has distinct eigenvalues, then $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L})) = (\lambda_1(\widetilde{\mathbf{L}}) + s, \mathbf{v}_1(\widetilde{\mathbf{L}}))$.*

The next theorem describes an incremental eigenpair computation method when the graph G is a disconnected graph with δ connected components. The columns of the matrix $\mathbf{V}_\delta$ are the δ smallest eigenvectors of $\mathbf{L}$. Note that $\mathbf{V}_\delta$ has a canonical representation where the nonzero entries of each column are a constant and their indices indicate the nodes in each connected component [26, 97], and the columns of $\mathbf{V}_\delta$ are the δ smallest eigenvectors of $\mathbf{L}$ with eigenvalue 0 [26]. Since the δ smallest eigenpairs with the canonical representation are trivial by identifying the connected components in the graph, we only focus on computing the $(K+1)$-th smallest eigenpair given K smallest eigenpairs, where $K \geq \delta$. The columns of the matrix $\mathbf{V}_{K,\delta} = [\mathbf{v}_{\delta+1}(\mathbf{L}), \mathbf{v}_{\delta+2}(\mathbf{L}), \ldots, \mathbf{v}_K(\mathbf{L})]$ are the $(\delta+1)$-th to the K-th smallest eigenvectors of $\mathbf{L}$ and the matrix $\mathbf{\Lambda}_{K,\delta} = \text{diag}(s-\lambda_{\delta+1}(\mathbf{L}), s-\lambda_{\delta+2}(\mathbf{L}), \ldots, s-\lambda_K(\mathbf{L}))$ is the diagonal matrix with $\{s-\lambda_i(\mathbf{L})\}_{i=\delta+1}^{K}$ being its main diagonal. If $K=\delta$, $\mathbf{V}_{K,\delta}$ and $\mathbf{\Lambda}_{K,\delta}$ are defined as the matrix with all entries being zero, i.e., $\mathbf{O}$.

Theorem 2.7. (disconnected graphs) *Assume that G is a disconnected graph with $\delta \geq 2$ connected components, given $\mathbf{V}_{K,\delta}$, $\mathbf{\Lambda}_{K,\delta}$ and $K \geq \delta$, the eigenpair $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L}))$ is a leading eigenpair of the matrix $\widetilde{\mathbf{L}} = \mathbf{L}+\mathbf{V}_{K,\delta}\mathbf{\Lambda}_{K,\delta}\mathbf{V}_{K,\delta}^T+s\mathbf{V}_\delta\mathbf{V}_\delta^T-s\mathbf{I}$. In particular, if $\mathbf{L}$ has distinct nonzero eigenvalues, then $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L})) = (\lambda_1(\widetilde{\mathbf{L}})+s, \mathbf{v}_1(\widetilde{\mathbf{L}}))$.*

Following the same methodology for proving Theorem 2.6 and using Corollary 2.2, for normalized graph Laplacian matrices, let $\mathbf{V}_K = [\mathbf{v}_1(\mathbf{L}_\mathcal{N}), \mathbf{v}_2(\mathbf{L}_\mathcal{N}), \ldots, \mathbf{v}_K(\mathbf{L}_\mathcal{N})]$ be the matrix with columns being the K smallest eigenvectors of $\mathbf{L}_\mathcal{N}$ and let $\mathbf{\Lambda}_K = \text{diag}(2-\lambda_1(\mathbf{L}_\mathcal{N}), 2-\lambda_2(\mathbf{L}_\mathcal{N}), \ldots, 2-\lambda_K(\mathbf{L}_\mathcal{N}))$. The following corollary provides the basis for incremental eigenpair computation for normalized graph Laplacian matrix of connected graphs.

Corollary 2.8. *For the normalized graph Laplacian matrix $\mathbf{L}_\mathcal{N}$ of a connected graph G, given $\mathbf{V}_K$ and $\mathbf{\Lambda}_K$, the eigenpair $(\lambda_{K+1}(\mathbf{L}_\mathcal{N}), \mathbf{v}_{K+1}(\mathbf{L}_\mathcal{N}))$ is a leading eigenpair of the matrix $\widetilde{\mathbf{L}}_\mathcal{N} = \mathbf{L}_\mathcal{N}+\mathbf{V}_K\mathbf{\Lambda}_K\mathbf{V}_K^T+\frac{2}{s}\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n\mathbf{1}_n^T\mathbf{S}^{\frac{1}{2}}-2\mathbf{I}$. In particular, if $\mathbf{L}_\mathcal{N}$ has distinct eigenvalues, then $(\lambda_{K+1}(\mathbf{L}_\mathcal{N}), \mathbf{v}_{K+1}(\mathbf{L}_\mathcal{N})) = (\lambda_1(\widetilde{\mathbf{L}}_\mathcal{N})+2, \mathbf{v}_1(\widetilde{\mathbf{L}}_\mathcal{N}))$.*

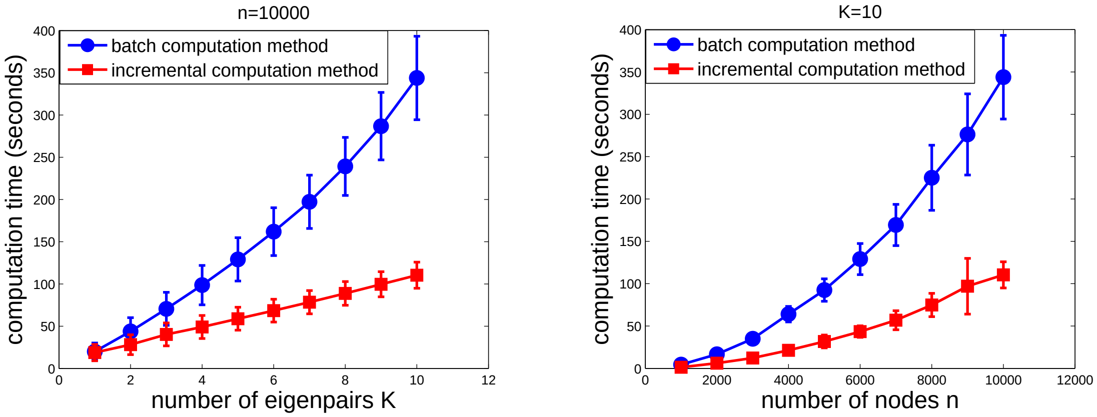
For disconnected graphs with δ connected components, let $\mathbf{V}_{K,\delta} = [\mathbf{v}_{\delta+1}(\mathbf{L}_\mathcal{N}), \mathbf{v}_{\delta+2}(\mathbf{L}_\mathcal{N}), \ldots, \mathbf{v}_K(\mathbf{L}_\mathcal{N})]$ with columns being the $(\delta+1)$-th to the K-th smallest eigenvectors of $\mathbf{L}_\mathcal{N}$ and let $\mathbf{\Lambda}_{K,\delta} = \text{diag}(2-\lambda_{\delta+1}(\mathbf{L}_\mathcal{N}), 2-\lambda_{\delta+2}(\mathbf{L}_\mathcal{N}), \ldots, 2-\lambda_K(\mathbf{L}_\mathcal{N}))$. Based on Corollary 2.4, the following corollary provides an incremental eigenpair computation method for normalized graph Laplacian matrix of disconnected graphs.

Corollary 2.9. *For the normalized graph Laplacian matrix $\mathbf{L}_{\mathcal{N}}$ of a disconnected graph G with $\delta \geq 2$ connected components, given $\mathbf{V}_{K,\delta}$, $\mathbf{\Lambda}_{K,\delta}$ and $K \geq \delta$, the eigenpair $(\lambda_{K+1}(\mathbf{L}_{\mathcal{N}}), \mathbf{v}_{K+1}(\mathbf{L}_{\mathcal{N}}))$ is a leading eigenpair of the matrix $\widetilde{\mathbf{L}}_{\mathcal{N}} = \mathbf{L}_{\mathcal{N}} + \mathbf{V}_{K,\delta}\mathbf{\Lambda}_{K,\delta}\mathbf{V}_{K,\delta}^T + \frac{2}{s}\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n\mathbf{1}_n^T\mathbf{S}^{\frac{1}{2}} - 2\mathbf{I}$. In particular, if $\mathbf{L}_{\mathcal{N}}$ has distinct eigenvalues, then $(\lambda_{K+1}(\mathbf{L}_{\mathcal{N}}), \mathbf{v}_{K+1}(\mathbf{L}_{\mathcal{N}})) = (\lambda_1(\widetilde{\mathbf{L}}_{\mathcal{N}}) + 2, \mathbf{v}_1(\widetilde{\mathbf{L}}_{\mathcal{N}}))$.*

2.1.4 Computational complexity analysis

Here we analyze the computational complexity of the proposed incremental eigenpair computation method and compare it with the batch computation method. The proposed incremental method utilizes the existing K smallest eigenpairs to compute the $(K+1)$-th smallest eigenpair as described in Sec. 2.1.3, whereas the batch computation method recomputes all eigenpairs for each value of K. Both methods can be easily implemented via well-developed numerical computation packages such as ARPACK [94].

Following the analysis in [88], the average relative error of the leading eigenvalue from Lanczos algorithm [90] has an upper bound of the order $O\left(\frac{(\ln n)^2}{t^2}\right)$, where n is the number of nodes in the graph and t is the number of iterations for Lanczos algorithm. Therefore, when one sequentially computes from $k = 2$ to $k = K$ smallest eigenpairs, for the proposed incremental computation method the upper bound on the average relative error of K smallest eigenpairs is $O\left(\frac{K(\ln n)^2}{t^2}\right)$ since in each increment computing the corresponding eigenpair can be transformed to a leading eigenpair computation process as described Sec. 2.1.3. On the other hand, for the batch computation method, the upper bound on the average relative error of K smallest eigenpairs is $O\left(\frac{K^2(\ln n)^2}{t^2}\right)$ since for the k-th increment ($k \leq K$) it needs to compute all k smallest eigenpairs from scratch. These results also imply that to reach the same average relative error ϵ for sequential computation of K smallest eigenpairs, the incremental method requires $\Omega\left(\sqrt{\frac{K}{\epsilon}}\ln n\right)$ iterations, whereas the batch method



(a) Computation time with $n = 10000$ and different number of eigenpairs K.

(b) Computation time with $K = 10$ and different number of nodes n

Figure 2.1: Sequential eigenpair computation time on Erdos-Renyi random graphs with edge connection probability $p = 0.1$. The markers are averaged computation time of 50 trials and the error bar represents standard deviation.

requires $\Omega\left(\frac{K \ln n}{\sqrt{\epsilon}}\right)$ iterations.

2.2 Experimental Results

We compare the computation time between the proposed incremental method and the batch method by perform experiments on synthetic connected graphs generated by Erdos-Renyi random graphs. We implement the proposed incremental eigenpair computation method using Matlab R2015a's "eigs" function, which is based on ARPACK package [94]. Note that, this function takes a parameter K and returns K leading eigenpairs of the given matrix. Following Theorem 2.6, the incremental method works by sequentially perturbing the graph Laplacian matrix $\mathbf{L}$ with a particular matrix and computing the leading eigenpair of the perturbed matrix $\widetilde{\mathbf{L}}$ by calling $eigs(\widetilde{\mathbf{L}}, 1)$. For the batch computation method, we use $eigs(\mathbf{L}, K)$ to compute the desired K eigenpairs from scratch as K increases. The Matlab implementation of both the batch mode and the proposed incremental method are available for download from https://sites.google.com/site/pinyuchenpage/codes.

To illustrate the advantage of the proposed incremental eigenpair computation method, we compare the computation time between the proposed incremental method and the batch method both for varying K and varying graph size. The Erdos-Renyi random graphs that we build are used for this comparison. Fig. 2.1 (a) shows the computation time of incremental and batch computation methods for sequentially computing from $K = 2$ to $K = 10$ smallest eigenpairs. Fig. 2.1 (b) shows the computation time of both methods with respect to different graph size n. It is observed that the difference in computation time grows polynomially as n increases, which suggests that the proposed incremental method is more efficient than the batch computation method, especially for large graphs.

CHAPTER III

Phase Transitions in Spectral Modularity Method under a Stochastic Block Model

In this chapter, we study the performance of the spectral modularity method [106] under the stochastic block model (SBM) of two communities. We prove the existence of an asymptotic phase transition threshold on community detectability for the spectral modularity method. The phase transition on community detectability occurs as the inter-community edge connection probability p grows. This phase transition separates a sub-critical regime of small p, where modularity-based community detection successfully identifies the communities, from a super-critical regime of large p where successful community detection is impossible. We show that, as the community sizes become large, the asymptotic phase transition threshold p^* is equal to $\sqrt{p_1 p_2}$, where p_i $(i = 1, 2)$ is the within-community edge connection probability. Thus the phase transition threshold is universal in the sense that it does not depend on the ratio of community sizes. The universal phase transition phenomenon is validated by simulations for moderately sized communities. Using the derived expression for the phase transition threshold we propose an empirical method for estimating this threshold from real-world data.

It has been observed in the literature that community detectability (i.e., the fraction of correctly identified nodes) degrades rapidly as the number of inter-community edges increases beyond a certain critical value [17, 47, 48, 87, 102, 131, 132, 138, 180]. This chapter establishes a mathematical expression for the critical phase transition threshold in modularity-based community detection under a stochastic block model. This phase transition threshold governs the community modularity measure of the graph as a function of the respective edge connection probabilities p_1 and p_2 within community 1 and community 2. Defining p as the edge connection probability between the two communities the critical phase transition threshold on p takes on the simple asymptotic form $p^* = \sqrt{p_1 p_2}$, in the limit as the two community sizes converge (at comparable rate) to infinity. Remarkably, p^* does not depend on the community sizes, and in this sense it is a universal threshold.

Newman [106] proposed a measure called modularity that evaluates the number of excessive edges of a graph compared with the corresponding degree-equivalent random graph. More specifically, define the modularity matrix as $\mathbf{B} = \mathbf{A} - b\mathbf{d}\mathbf{d}^T$, where $\mathbf{d}$ is the degree vector of the graph and $b = \frac{1}{2m}$ is the reciprocal of the total number of edges in the graph. The last term $b\mathbf{d}\mathbf{d}^T$ can be viewed as the expected adjacency matrix of the degree-equivalent random graph. Newman proposed to compute the largest eigenvector of $\mathbf{B}$ and perform K-means clustering [72] or take the sign function on this vector to cluster the nodes into two communities. Since the n-dimensional vector of all ones, $\mathbf{1}_n$, is always in the null space of $\mathbf{B}$, i.e, $\mathbf{B}\mathbf{1}_n = \mathbf{0}_n$, where $\mathbf{0}_n$ is the n-dimensional vector of all zeros, the (unnormalized) modularity is the largest eigenvalue of $\mathbf{B}$ and has the representation

$$\lambda_{\max}(\mathbf{B}) = \max_{\mathbf{x}^T\mathbf{x}=1,\ \mathbf{x}^T\mathbf{1}_n=0} \mathbf{x}^T\mathbf{B}\mathbf{x}. \tag{3.1}$$

3.1 Phase Transition Analysis

Consider a stochastic block model [74] consisting of two community structures parameterized by edge connection probability p_i within community i ($i = 1, 2$) and edge connection probability p between the two communities. Let n_i denote the size of community i such that $n_1+n_2=n$. Recall from (1.1) that the overall $n\times n$ adjacency matrix of the entire graph can be represented as

$$\mathbf{A}=\begin{bmatrix}\mathbf{A}_1 & \mathbf{C}\\ \mathbf{C}^T & \mathbf{A}_2\end{bmatrix}, \tag{3.2}$$

where $\mathbf{A}_i$ is the n_i-by-n_i adjacency matrix of an Erdos-Renyi random graph with edge connection probability p_i and $\mathbf{C}$ is the n_1-by-n_2 adjacency matrix of the inter-community edges where each entry in $\mathbf{C}$ is a Bernoulli(p) random variable.

Using the network model in (3.2), let $\mathbf{d}=\mathbf{A}\mathbf{1}_n=[\mathbf{d}_1^T\ \mathbf{d}_2^T]^T$ denote the degree vector of the graph with $\mathbf{d}_1\in\mathbb{R}^{n_1}$ and $\mathbf{d}_2\in\mathbb{R}^{n_2}$. Then $b=(\mathbf{1}_n^T\mathbf{A}\mathbf{1}_n)^{-1}=(\mathbf{d}_1^T\mathbf{1}_{n_1}+\mathbf{d}_2^T\mathbf{1}_{n_2})^{-1}$. Let $\widetilde{\mathbf{d}}_i=\mathbf{A}_i\mathbf{1}_{n_i}$ denote the degree vector of community i. Since $\mathbf{A}\mathbf{1}_n=\mathbf{d}$, with (3.2) the degree vectors $\mathbf{d}_1$, $\mathbf{d}_2$, $\widetilde{\mathbf{d}}_1$, and $\widetilde{\mathbf{d}}_2$ satisfy the following equations:

$$\mathbf{d}_1=\widetilde{\mathbf{d}}_1+\mathbf{C}\mathbf{1}_{n_2} \text{ and } \mathbf{d}_2=\widetilde{\mathbf{d}}_2+\mathbf{C}^T\mathbf{1}_{n_1}. \tag{3.3}$$

Let $b_i=(\widetilde{\mathbf{d}}_i^T\mathbf{1}_{n_i})^{-1}$. The modularity matrix of community i is denoted by $\mathbf{B}_i=\mathbf{A}_i-b_i\widetilde{\mathbf{d}}_i\widetilde{\mathbf{d}}_i^T$. Using these notations, the modularity matrix of the entire graph can be represented as

$$\mathbf{B}=\begin{bmatrix}\mathbf{B}_1+b_1\widetilde{\mathbf{d}}_1\widetilde{\mathbf{d}}_1^T-b\mathbf{d}_1\mathbf{d}_1^T & \mathbf{C}-b\mathbf{d}_1\mathbf{d}_2^T\\ \mathbf{C}^T-b\mathbf{d}_2\mathbf{d}_1^T & \mathbf{B}_2+b_2\widetilde{\mathbf{d}}_2\widetilde{\mathbf{d}}_2^T-b\mathbf{d}_2\mathbf{d}_2^T\end{bmatrix}. \tag{3.4}$$

Let $\mathbf{y}=[\mathbf{y}_1^T\ \mathbf{y}_2^T]^T$ denote the largest eigenvector of $\mathbf{B}$, where $\mathbf{y}_1\in\mathbb{R}^{n_1}$ and $\mathbf{y}_2\in\mathbb{R}^{n_2}$. Following the definition of modularity in (3.1) and (3.4), $\mathbf{y}=\arg\max_{\mathbf{x}}\Gamma(\mathbf{x})$, where

$$\begin{aligned}\Gamma(\mathbf{x}) = {} & \mathbf{x}_1^T\mathbf{B}_1\mathbf{x}_1 + \mathbf{x}_2^T\mathbf{B}_2\mathbf{x}_2 + b_1(\widetilde{\mathbf{d}}_1^T\mathbf{x}_1)^2 + b_2(\widetilde{\mathbf{d}}_2^T\mathbf{x}_2)^2 \\ & - b(\mathbf{d}_1^T\mathbf{x}_1)^2 - b(\mathbf{d}_2^T\mathbf{x}_2)^2 + 2\mathbf{x}_1^T\mathbf{C}\mathbf{x}_2 - 2b(\mathbf{d}_1^T\mathbf{x}_1)(\mathbf{d}_2^T\mathbf{x}_2) \\ & - \mu(\mathbf{x}_1^T\mathbf{x}_1 + \mathbf{x}_2^T\mathbf{x}_2 - 1) - \nu(\mathbf{x}_1^T\mathbf{1}_{n_1} + \mathbf{x}_2^T\mathbf{1}_{n_2}), \end{aligned} \tag{3.5}$$

and $\mathbf{x} = [\mathbf{x}_1^T\ \mathbf{x}_2^T]^T$, $\mathbf{x}_1 \in \mathbb{R}^{n_1}$, and $\mathbf{x}_2 \in \mathbb{R}^{n_2}$. μ and ν are Lagrange multipliers of the constraints $\mathbf{x}^T\mathbf{x} = 1$ and $\mathbf{x}^T\mathbf{1}_n = 0$ in (3.1), respectively.

Differentiating (3.5) with respect to $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively, and substituting $\mathbf{y}$ to the equations, we obtain

$$2\mathbf{B}_1\mathbf{y}_1 + 2b_1(\widetilde{\mathbf{d}}_1^T\mathbf{y}_1)\widetilde{\mathbf{d}}_1 - 2b(\mathbf{d}_1^T\mathbf{y}_1)\mathbf{d}_1 - 2b(\mathbf{d}_2^T\mathbf{y}_2)\mathbf{d}_1 + 2\mathbf{C}\mathbf{y}_2 - 2\mu\mathbf{y}_1 - \nu\mathbf{1}_{n_1} = \mathbf{0}_{n_1}; \tag{3.6}$$

$$2\mathbf{B}_2\mathbf{y}_2 + 2b_2(\widetilde{\mathbf{d}}_2^T\mathbf{y}_2)\widetilde{\mathbf{d}}_2 - 2b(\mathbf{d}_2^T\mathbf{y}_2)\mathbf{d}_2 - 2b(\mathbf{d}_1^T\mathbf{y}_1)\mathbf{d}_2 + 2\mathbf{C}^T\mathbf{y}_1 - 2\mu\mathbf{y}_2 - \nu\mathbf{1}_{n_2} = \mathbf{0}_{n_2}. \tag{3.7}$$

Left multiplying (3.6) by $\mathbf{1}_{n_1}^T$ and left multiplying (3.7) by $\mathbf{1}_{n_2}^T$ and recalling that $\mathbf{B}_i\mathbf{1}_{n_i} = \mathbf{0}_{n_i}$ and $b_i = (\widetilde{\mathbf{d}}_i^T\mathbf{1}_{n_i})^{-1}$, we have

$$2(\widetilde{\mathbf{d}}_1^T\mathbf{y}_1) - 2b(\mathbf{d}_1^T\mathbf{y}_1)(\mathbf{d}_1^T\mathbf{1}_{n_1}) - 2b(\mathbf{d}_2^T\mathbf{y}_2)(\mathbf{d}_1^T\mathbf{1}_{n_1}) + 2\mathbf{1}_{n_1}^T\mathbf{C}\mathbf{y}_2 - 2\mu\mathbf{y}_1^T\mathbf{1}_{n_1} - \nu n_1 = 0; \tag{3.8}$$

$$2(\widetilde{\mathbf{d}}_2^T\mathbf{y}_2) - 2b(\mathbf{d}_2^T\mathbf{y}_2)(\mathbf{d}_2^T\mathbf{1}_{n_2}) - 2b(\mathbf{d}_1^T\mathbf{y}_1)(\mathbf{d}_2^T\mathbf{1}_{n_2}) + 2\mathbf{1}_{n_2}^T\mathbf{C}^T\mathbf{y}_1 - 2\mu\mathbf{y}_2^T\mathbf{1}_{n_2} - \nu n_2 = 0. \tag{3.9}$$

Summing (3.8) and (3.9) and using (3.3) gives $\nu = 0$. Left multiplying (3.6) by $\mathbf{y}_1^T$ and left multiplying (3.7) by $\mathbf{y}_2^T$, substituting $\nu = 0$ and summing the equations, with (3.4) we have $\mu = \lambda_{\max}(\mathbf{B})$.

Let $\overline{\mathbf{C}} = p\mathbf{1}_{n_1}\mathbf{1}_{n_2}^T$, a matrix whose elements are the means of entries in $\mathbf{C}$. Let

$\sigma_i(\mathbf{M})$ denote the i-th largest singular value of a rectangular matrix $\mathbf{M}$ and write $\mathbf{C} = \overline{\mathbf{C}} + \mathbf{\Delta}$, where $\mathbf{\Delta} = \mathbf{C} - \overline{\mathbf{C}}$. Latala's theorem [91] implies that the expected value of $\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_1 n_2}}\right)$ converges to 0 as n_1 and n_2 approach infinity, denoted $\mathbb{E}\left[\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_1 n_2}}\right)\right] \to 0$ as $n_1, n_2 \to \infty$. The proof is given in Appendix C.10 with the condition that $\overline{W} = 1$. Furthermore, by Talagrand's concentration theorem [150],

$$\sigma_1\left(\frac{\mathbf{C}}{\sqrt{n_1 n_2}}\right) \xrightarrow{\text{a.s.}} p \text{ and } \sigma_i\left(\frac{\mathbf{C}}{\sqrt{n_1 n_2}}\right) \xrightarrow{\text{a.s.}} 0,\ \forall i \geq 2 \tag{3.10}$$

when $n_1, n_2 \to \infty$, where $\xrightarrow{\text{a.s.}}$ means almost sure convergence. The proof is given in Appendix C.10 with the condition that $\overline{W} = 1$. Note that the convergence rate is maximal when $n_1 = n_2$ because $n_1 + n_2 \geq 2\sqrt{n_1 n_2}$ and the equality holds if $n_1 = n_2$.

Throughout this chapter we further assume $\frac{n_1}{n_2} \to c > 0$ as $n_1, n_2 \to \infty$. This means the community sizes grow with comparable rates. As proved in [13], the singular vectors of $\mathbf{C}$ and $\overline{\mathbf{C}}$ are close to each other in the sense that the square of inner product of their left/right singular vectors converges to 1 almost surely when $\sqrt{n_1 n_2} p \to \infty$. Consequently, the concentration results in (3.10) and [13] imply that

$$\frac{\mathbf{C}\mathbf{1}_{n_2}}{n_2} \xrightarrow{\text{a.s.}} p\mathbf{1}_{n_1} \text{ and } \frac{\mathbf{C}^T\mathbf{1}_{n_1}}{n_1} \xrightarrow{\text{a.s.}} p\mathbf{1}_{n_2}. \tag{3.11}$$

Furthermore, since under the stochastic block model setting each entry of the adjacency matrix $\mathbf{A}_i$ in (3.2) is a Bernoulli(p_i) random variable, following the same concentration arguments in (3.10) and (3.11) we have

$$\frac{\mathbf{A}_1\mathbf{1}_{n_1}}{n_1} \xrightarrow{\text{a.s.}} p_1\mathbf{1}_{n_1} \text{ and } \frac{\mathbf{A}_2\mathbf{1}_{n_2}}{n_2} \xrightarrow{\text{a.s.}} p_2\mathbf{1}_{n_2}. \tag{3.12}$$

By the fact that $\widetilde{\mathbf{d}}_i = \mathbf{A}_i\mathbf{1}_{n_i}$, (3.12) implies that

$$\frac{\widetilde{\mathbf{d}}_1}{n_1} \xrightarrow{\text{a.s.}} p_1\mathbf{1}_{n_1} \text{ and } \frac{\widetilde{\mathbf{d}}_2}{n_2} \xrightarrow{\text{a.s.}} p_2\mathbf{1}_{n_2}. \tag{3.13}$$

Applying (3.11), (3.12) and (3.13) to (3.3) and recalling that $\frac{n_1}{n_2} \to c > 0$, we have

$$\frac{\mathbf{d}_1}{n_1} \xrightarrow{\text{a.s.}} \left(p_1 + \frac{p}{c}\right) \mathbf{1}_{n_1} \text{ and } \frac{\mathbf{d}_2}{n_2} \xrightarrow{\text{a.s.}} \left(p_2 + cp\right) \mathbf{1}_{n_2}. \tag{3.14}$$

Therefore the reciprocal of the total degree in the graph b has the relation

$$n_1 n_2 b = \frac{n_1 n_2}{\mathbf{d}_1^T \mathbf{1}_{n_1} + \mathbf{d}_2^T \mathbf{1}_{n_2}} \xrightarrow{\text{a.s.}} \frac{1}{cp_1 + 2p + \frac{p_2}{c}}. \tag{3.15}$$

Substituting these limits into (3.8) and (3.9) and recalling that $\nu = 0$ and $\mathbf{y}_1^T \mathbf{1}_{n_1} = -\mathbf{y}_2^T \mathbf{1}_{n_2}$, we have

$$\mathbf{y}_1^T \mathbf{1}_{n_1} \left(\frac{\mu}{n} - \frac{p_1 p_2 - p^2}{cp_1 + 2p + \frac{p_2}{c}} \right) \xrightarrow{\text{a.s.}} 0; \tag{3.16}$$

$$\mathbf{y}_2^T \mathbf{1}_{n_2} \left(\frac{\mu}{n} - \frac{p_1 p_2 - p^2}{cp_1 + 2p + \frac{p_2}{c}} \right) \xrightarrow{\text{a.s.}} 0. \tag{3.17}$$

Since $\mu = \lambda_{\max}(\mathbf{B})$, for each inter-community edge connection probability p, one of the two cases below has to be satisfied:

$$\text{Sub-critical regime: } \frac{\lambda_{\max}(\mathbf{B})}{n} \xrightarrow{\text{a.s.}} \frac{p_1 p_2 - p^2}{cp_1 + 2p + \frac{p_2}{c}} \tag{3.18}$$

$$\text{Super-critical regime: } \mathbf{y}_1^T \mathbf{1}_{n_1} \xrightarrow{\text{a.s.}} 0 \text{ and } \mathbf{y}_2^T \mathbf{1}_{n_2} \xrightarrow{\text{a.s.}} 0 \tag{3.19}$$

In the sub-critical regime, observe that $\frac{\lambda_{\max}(\mathbf{B})}{n}$ converges to $\frac{p_1 p_2 - p^2}{cp_1 + 2p + \frac{p_2}{c}}$ almost surely such that the corresponding asymptotic largest eigenvector $\mathbf{y}$ of $\mathbf{B}$ remains the same (unique up to its sign) for different p. Left multiplying (3.6) by $\mathbf{y}_1^T$ and left multiplying (3.7) by $\mathbf{y}_2^T$, summing these two equations, and using the limiting expressions (3.4), (3.11), (3.12), (3.13), (3.14), (3.15), and (3.18), in the sub-critical regime, we have

$$\frac{\mathbf{y}_1^T \mathbf{B}_1 \mathbf{y}_1}{n} + \frac{\mathbf{y}_2^T \mathbf{B}_2 \mathbf{y}_2}{n} + f(p) \xrightarrow{\text{a.s.}} 0, \tag{3.20}$$

where $f(p) = \frac{p_1 p_2 - p^2}{cp_1 + 2p + \frac{p_2}{c}} \left[\frac{\left(\sqrt{c} + \frac{1}{\sqrt{c}}\right)^2 \left(\mathbf{y}_1^T \mathbf{1}_{n_1}\right)^2}{n} - 1 \right]$. Since $f(p)$ is a Laurent polynomial of p with finite powers, and (3.20) has to be satisfied over all values of p in the sub-critical regime,

$$\frac{\mathbf{y}_1^T \mathbf{B}_1 \mathbf{y}_1}{n} + \frac{\mathbf{y}_2^T \mathbf{B}_2 \mathbf{y}_2}{n} \xrightarrow{\text{a.s.}} 0 \text{ and } f(p) \xrightarrow{\text{a.s.}} 0. \tag{3.21}$$

Furthermore, we can show that, in the sub-critical regime, $\mathbf{y}_1$ and $\mathbf{y}_2$ converge almost surely to constant vectors with opposite signs,

$$\sqrt{\frac{nn_1}{n_2}} \mathbf{y}_1 \xrightarrow{\text{a.s.}} \pm \mathbf{1}_{n_1} \text{ and } \sqrt{\frac{nn_2}{n_1}} \mathbf{y}_2 \xrightarrow{\text{a.s.}} \mp \mathbf{1}_{n_2}. \tag{3.22}$$

The proof is given in Appendix B.1. Therefore, in the sub-critical regime the two communities can be almost perfectly detected. On the other hand, in the super-critical regime the spectral modularity method fails to detect the two communities since by (3.19) $\mathbf{y}_1$ and $\mathbf{y}_2$ must have both positive and negative entries.

Next we derive the asymptotic universal phase transition threshold p^* for transition from the sub-critical regime to the super-critical regime that occurs as p increases. Note that in the super-critical regime, since $\mathbf{y}_1^T \mathbf{1}_{n_1} \xrightarrow{\text{a.s.}} 0$ and $\mathbf{y}_2^T \mathbf{1}_{n_2} \xrightarrow{\text{a.s.}} 0$, using (3.1), (3.4), (3.11), (3.12), (3.13) and (3.14) we have

$$
\begin{aligned}
\frac{\lambda_{\max}(\mathbf{B})}{n} &= \frac{1}{n}\Big[\mathbf{y}_1^T\mathbf{B}_1\mathbf{y}_1 + \mathbf{y}_2^T\mathbf{B}_2\mathbf{y}_2 + b_1(\widetilde{\mathbf{d}}_1^T\mathbf{y}_1)^2 + b_2(\widetilde{\mathbf{d}}_2^T\mathbf{y}_2)^2 - b(\mathbf{d}_1^T\mathbf{y}_1)^2 - b(\mathbf{d}_2^T\mathbf{y}_2)^2 \\
&\quad + 2\mathbf{y}_1^T\mathbf{C}\mathbf{y}_2 - 2b(\mathbf{d}_1^T\mathbf{y}_1)(\mathbf{d}_2^T\mathbf{y}_2)\Big] \\
&\xrightarrow{\text{a.s.}} \frac{1}{n}\Big\{\mathbf{y}_1^T(p_1\mathbf{1}_{n_1}\mathbf{1}_{n_1}^T - p_1\mathbf{1}_{n_1}\mathbf{1}_{n_1}^T)\mathbf{y}_1 + \mathbf{y}_2^T(p_2\mathbf{1}_{n_2}\mathbf{1}_{n_2}^T - p_2\mathbf{1}_{n_2}\mathbf{1}_{n_2}^T)\mathbf{y}_2 \\
&\quad + b_1(n_1p_1\mathbf{y}_1^T\mathbf{1}_{n_1})^2 + b_2(n_2p_2\mathbf{y}_2^T\mathbf{1}_{n_2})^2 - b\Big[(n_1p_1 + n_2p)\,\mathbf{y}_1^T\mathbf{1}_{n_1}\Big]^2 \\
&\quad - b\Big[(n_2p_2 + n_1p)\,\mathbf{y}_2^T\mathbf{1}_{n_2}\Big]^2 + 2p(\mathbf{y}_1^T\mathbf{1}_{n_1})(\mathbf{y}_2^T\mathbf{1}_{n_2}) \\
&\quad - 2b\Big[(n_1p_1 + n_2p)\,\mathbf{y}_1^T\mathbf{1}_{n_1}\Big]\Big[(n_2p_2 + n_1p)\,\mathbf{y}_2^T\mathbf{1}_{n_2}\Big]\Big\} \\
&= 0. \qquad (3.23)
\end{aligned}
$$

Consequently, by (3.18) and (3.23), the phase transition occurs at $p = p^*$ almost surely when $\frac{p_1p_2 - p^{*2}}{cp_1 + 2p^* + \frac{p_2}{c}} = 0$. This implies an asymptotic universal phase transition threshold on community detectability:

$$
p^* \xrightarrow{\text{a.s.}} \sqrt{p_1p_2} \qquad (3.24)
$$

as $n_1, n_2 \to \infty$ and $\frac{n_1}{n_2} \to c > 0$. Note that the limit (3.24) does not depend on the community sizes. In this sense, the phase transitions are universal as they only depend on the within-community connection probabilities p_1 and p_2.

Moreover, the same phase transition results hold for a more general setting where $p_i = \Omega(\frac{1}{n^\epsilon})$ and $p = \Omega(\frac{1}{n^\epsilon})$ for any $\epsilon \in [0,1)$ by following the same derivation procedures. As a comparison, the phase transition threshold under the sparse network setting, where $p_i = \Omega(\frac{1}{n})$ and $p = \Omega(\frac{1}{n})$ [47, 48, 87, 102, 131, 180], is different from the threshold established in this chapter where $p_i = \Omega(\frac{1}{n^\epsilon})$ and $p = \Omega(\frac{1}{n^\epsilon})$ for any $\epsilon \in [0,1)$. Also note that when $p_i = \Omega(\frac{1}{n^\epsilon})$ and $p = \Omega(\frac{1}{n^\epsilon})$ for any $\epsilon \in [0,1)$, the community detectability undergoes an abrupt transition at the threshold whereas the

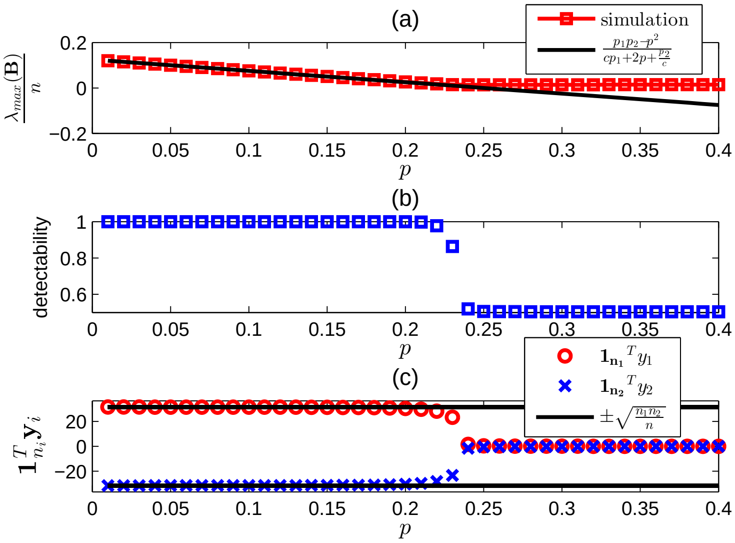


Figure 3.1: Validation of theoretical critical phase transition threshold (3.24) for two communities generated by a stochastic block model. The curves represent averages over 100 realizations of the model. Here $n_1 = n_2 = 2000$ and $p_1 = p_2 = 0.25$ so that the predicted critical phase transition is $p^* = 0.25$. (a) When $p < p^*$, $\frac{\lambda_{\max}(\mathbf{B})}{n}$ converges to $\frac{p_1p_2-p^2}{cp_1+2p+\frac{p_2}{c}}$ as predicted in (3.18). When $p > p^*$, $\frac{\lambda_{\max}(\mathbf{B})}{n}$ converges to 0 as predicted by (3.23). (b) Fraction of nodes that are correctly identified using the spectral modularity method. Community detectability undergoes a phase transition from perfect detectability to low detectability at $p = p^*$. (c) The spectral modularity method fails to detect the communities when $p > p^*$ since the components of the largest eigenvector of $\mathbf{B}$, $\mathbf{y}_1$ and $\mathbf{y}_2$, undergo transitions at $p = p^*$ as predicted by (3.19) and (3.22).

transition is smoother for sparse networks.

3.2 Numerical Experiments

3.2.1 Validation of phase transition analysis

We validate the asymptotic phase transition phenomenon predicted by our theory, and in particular the critical phase transition threshold (3.24), showing that the asymptotic theory provides remarkably accurate predictions for the case of finite small community sizes. Fig. 3.1 (a) shows that $\frac{\lambda_{\max}(\mathbf{B})}{n}$ converges to $\frac{p_1p_2-p^2}{cp_1+2p+\frac{p_2}{c}}$ when $p < p^*$ and $\frac{\lambda_{\max}(\mathbf{B})}{n}$ converges to 0 when $p > p^*$, as predicted by (3.16) and (3.23).

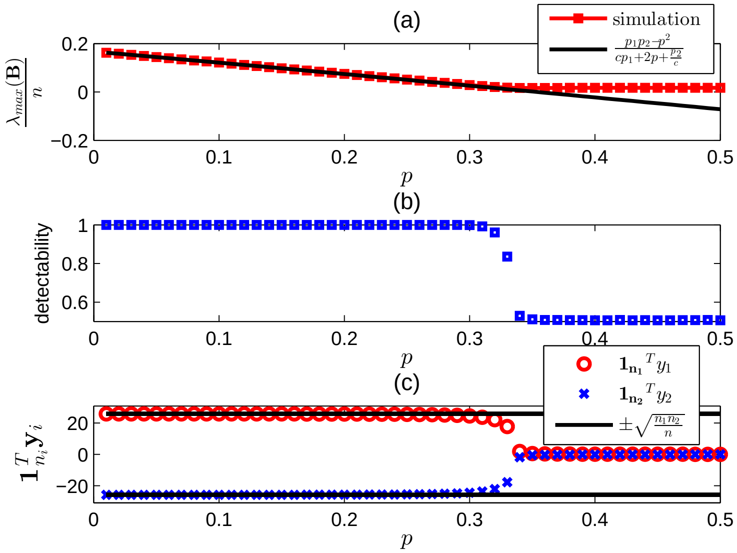


Figure 3.2: Validation of theoretical critical phase transition threshold (3.24) for two communities generated by a stochastic block model. The curves represent averages over 100 realizations of the model. Here $n_1 = 1000$, $n_2 = 2000$, $p_1 = 0.5$, and $p_2 = 0.25$ so that the predicted critical phase transition is $p^* = 0.3536$. Similar phase transition phenomenon can be observed for this network setting.

Fig. 3.1 (b) shows the phase transition from perfect detectability to low detectability at the critical value $p = p^*$. The numerical phase transition thresholds are accurately predicted by (3.24). Fig. 3.1 (c) further validates the predictions in (3.19) and (3.22) that $\mathbf{y}_1$ and $\mathbf{y}_2$ converge almost surely to constant vectors with opposite signs in the sub-critical regime of $p < p^*$ and $\mathbf{y}_1^T\mathbf{1}_{n_1}$ and $\mathbf{y}_2^T\mathbf{1}_{n_2}$ converge to 0 almost surely in the super-critical regime of $p > p^*$. Similarly in Fig. 3.2, the results are shown for a different stochastic block model where the sizes of the two communities are not the same. These results validate that the asymptotic phase transition threshold p^* in (3.24) is a universal phenomenon that does not depend on the community sizes. We have observed (see Appendix B.2) that the asymptotic phase transition expression in (3.24) is accurate even in cases of relatively small community sizes, e.g. down to sizes as small as 100.

3.2.2 Empirical estimator of the phase transition threshold

Using the derived expression of the phase transition threshold in (3.24), we propose an empirical method for estimating the threshold in order to evaluate the reliability of community detection on real-world data *a posteriori*. Let $\widehat{n}_i$ and $\widehat{m}_i$ denote the size and the number of edges of the identified community i, and let $\widehat{m}_{12}$ denote the number of identified external edges between communities. Define the empirical estimators

$$\widehat{p} = \frac{\widehat{m}_{12}}{\widehat{n}_1 \widehat{n}_2}; \tag{3.25}$$

$$\widehat{p}_i = \frac{\widehat{m}_i}{\widehat{n}_i^2}; \tag{3.26}$$

$$\widehat{p^*} = \sqrt{\widehat{p}_1 \widehat{p}_2}. \tag{3.27}$$

We apply these estimators to the political blog data in [3], where this dataset contains 1222 blogs, labeled as either conservative or liberal, and an edge corresponds to a hyperlink reference between blogs. The detectability using the spectral modularity method is 0.9419 (the labels are predicted by taking the sign function on the leading eigenvector of the modularity matrix). The corresponding empirical estimates are $\widehat{p} = 0.0042$, $\widehat{p}_1 = 0.0244$, $\widehat{p}_2 = 0.0179$, and $\widehat{p^*} = 0.0209$. The high detectability of the spectral modularity method is consistent with the fact that the empirical estimate $\widehat{p}$ is below the empirical phase transition threshold $\widehat{p^*}$.

CHAPTER IV

Phase Transitions in Spectral Graph Clustering under the Random Interconnection Model

Recall from Chapter I that spectral clustering [97, 108, 176] is a principal method for graph clustering, which we call is as spectral graph clustering (SGC). It works by transforming the graph adjacency matrix into a graph Laplacian matrix [98], computing its eigendecomposition, and performing K-means clustering [72] on the eigenvectors to partition the nodes into clusters. In recent years, researchers have established phase transitions of the accuracy of clustering nodes in a graph under a diverse set of network models [2, 5, 25, 29, 48, 70, 102]. A widely used network model is the stochastic block model (SBM) [74], where the edge connections within and between clusters are independent Bernoulli random variables. Under the SBM, a phase transition on the cluster interconnectivity probability separates clustering accuracy into two regimes: a regime where correct graph clustering is possible, and a regime where correct graph clustering is impossible. The critical values that separate these two regimes are called phase transition thresholds. A summary of phase transition analysis under the SBM can be found in [2].

In this chapter we analyze the performance of spectral clustering on undirected unweighted graphs generated by a random interconnection model (RIM), where each cluster can have arbitrary internal connectivity structure and inter-cluster edges are assumed to be random. The RIM does not impose any distributional assumptions on the within-cluster connectivity structure, but assumes the between-cluster edges are generated by a SBM. Under the RIM, we establish a breakdown condition on the ability to identify correct clusters using SGC. Furthermore, when all of the cluster interconnection probabilities are identical, a model we call the homogeneous RIM, this breakdown condition specifies a critical phase transition threshold $p^* \in [0,1]$ on the inter-cluster connection probability p. When this interconnection probability is below the critical phase transition threshold, spectral clustering can perfectly detect the clusters. On the other hand, when the interconnection probability is above the critical phase transition threshold, spectral clustering fails to identify the clusters. This breakdown condition and phase transition analysis apply to weighted graphs as well, where the critical phase transition threshold depends not only on the interconnection probability but also on the weights of the interconnection edges. In Sec. 4.2, Theorems 4.1, 4.2 and 4.8 apply to unweighted undirected graphs while Theorems 4.9 extends these theorems to weighted undirected graphs.

4.1 Random Interconnection Model (RIM) and Spectral Clustering

4.1.1 Random interconnection model (RIM)

Assume there are K clusters in the graph and denote the size of cluster k by n_k. The size of the largest and smallest cluster is denoted by $n_{\max}$ and $n_{\min}$, respectively. Using the block model notations for graphs in Sec. 1.3,. let $\mathbf{A}_k$ denote the $n_k \times n_k$ adjacency matrix representing the internal edge connections in cluster k and let $\mathbf{C}_{ij}$ $(i,j \in \{1,2,\ldots,K\})$ be an $n_i \times n_j$ matrix representing the adjacency matrix of inter-cluster edge connections between the cluster pair (i,j). The proposed random interconnection model (RIM) assumes that: (1) the adjacency matrix $\mathbf{A}_k$ is associated with a connected graph of n_k nodes but is otherwise arbitrary; (2) the $K(K-1)/2$ matrices $\{\mathbf{C}_{ij}\}_{i>j}$ are random mutually independent, and each $\mathbf{C}_{ij}$ has i.i.d. Bernoulli distributed entries with Bernoulli parameter $p_{ij} \in [0,1]$. We call this model a *homogeneous* RIM when all random interconnections have equal probability, i.e., $p_{ij} = p$ for all $i \neq j$. Otherwise, the model is called an inhomogeneous RIM. In the next section, Theorems 4.1 and 4.8 apply to general RIM while Theorems 4.2 and 4.9 are restricted to the homogeneous RIM.

The stochastic block model (SBM) [74] is a special case of the RIM in the sense that the RIM does not impose any distributional constraints on $\mathbf{A}_k$. In contrast, under the SBM $\mathbf{A}_k$ is a Erdos-Renyi random graph with some edge connection probability $p_k \in [0,1]$.

4.1.2 Mathematical formulation for spectral graph clustering

For analysis purposes, throughout this chapter we will focus on the case where the observed graph is connected. If the graph is not connected, the connected components can be easily found and the proposed algorithm can be applied to each connected component separately. Since the smallest eigenvalue of $\mathbf{L}$ is always 0 and the associated eigenvector is $\frac{\mathbf{1}_n}{\sqrt{n}}$, only the higher order eigenvectors will affect the clustering results. By the Courant-Fischer theorem [78], the $K-1$ eigenvectors associated with the $K-1$ smallest nonzero eigenvalues of $\mathbf{L}$, represented by the columns of the eigenvector matrix $\mathbf{Y} \in \mathbb{R}^{n \times (K-1)}$, are the solution of the minimization problem

$$\begin{aligned} &S_{2:K}(\mathbf{L}) = \min_{\mathbf{X} \in \mathbb{R}^{n \times (K-1)}} \operatorname{trace}(\mathbf{X}^T \mathbf{L} \mathbf{X}), \\ &\text{subjec to } \mathbf{X}^T \mathbf{X} = \mathbf{I}_{K-1},\ \mathbf{X}^T \mathbf{1}_n = \mathbf{0}_{K-1}, \end{aligned} \tag{4.1}$$

where the optimal value $S_{2:K}(\mathbf{L}) = \operatorname{trace}(\mathbf{Y}^T \mathbf{L} \mathbf{Y})$ of (4.1) is the partial sum of the second to the K-th smallest eigenvalues of $\mathbf{L}$, and $\mathbf{I}_{K-1}$ is the $(K-1) \times (K-1)$

identity matrix. The constraints in (4.1) impose orthonormality and centrality on the eigenvectors.

4.2 Breakdown Condition and Phase Transition Analysis

In this section we establish a mathematical condition (Theorem 4.1) under which SGC fails to accurately identify clusters under the RIM. Furthermore, under the homogeneous RIM assumption of identical interconnection probability $p_{ij} = p$ governing the entries of the matrices $\{\mathbf{C}_{ij}\}$ in (1), the condition leads to (Theorem 4.2) a critical phase transition threshold p^* where, if $p < p^*$ spectral clustering correctly identifies the communities with probability one, while if $p > p^*$ spectral clustering fails. The phase transition analysis developed in this section will be used to establish an automated model order selection algorithm for spectral graph clustering in Chapter V. Proofs of the established theorems and corollaries in this section are given in Appendix C. In the sequel, there are a number of limit theorems stated about the behavior of random matrices and vectors whose dimensions go to infinity as the sizes n_k of the clusters goes to infinity while their relative sizes n_k/n_ℓ are held constant. For simplicity and convenience, the limit theorems are often stated in terms of the the finite, but arbitrarily large, dimensions n_k, $k = 1, \ldots, K$.

Based on the RIM (1.1), Theorem 4.1 establishes a general breakdown condition under which spectral clustering fails to correctly identify the clusters.

Theorem 4.1 (breakdown condition for SGC)**.**

Let $\widetilde{\mathbf{A}}$ be the $(K-1) \times (K-1)$ matrix with (i,j)-th element

$$[\widetilde{\mathbf{A}}]_{ij} = \begin{cases} (n_i + n_K)\, p_{iK} + \sum_{z=1, z \neq i}^{K-1} n_z p_{iz}, & \text{if } i = j, \\ n_i \cdot \big(p_{iK} - p_{ij}\big) & \text{if } i \neq j. \end{cases}$$

The following holds almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. If $\lambda_i\left(\frac{\widetilde{\mathbf{A}}}{n}\right) \neq \lambda_j\left(\frac{\mathbf{L}}{n}\right)$

for all $i = 1, 2, \ldots, K-1$ and $j = 2, 3, \ldots, K$, then spectral clustering cannot be successful.

Proof. The proof is given in Appendix C.1. □

Since the eigenvalues of $\widetilde{\mathbf{A}}$ depend only on the RIM parameters p_{ij} and n_k whereas the eigenvalues of $\mathbf{L}$ depend not only on these parameters but also on the internal adjacency matrices $\mathbf{A}_k$, Theorem 4.1 specifies how the graph connectivity structure affects the success of SGC.

For the special case of homogeneous RIM, where $p_{ij} = p$, for all $i \neq j$, Theorem 4.2 establishes the existence of a phase transition in the accuracy of SGC as the interconnection probability p increases. A similar phase transition likely exists for the inhomogeneous RIM (i.e., p_{ij}'s are not identical), but an inhomogeneous extension of Theorem 4.2 is an open problem. Nonetheless, Theorem 4.8 shows that the homogeneous RIM phase transition threshold p^* in Theorem 4.2 can be used to bound clustering accuracy when the RIM is inhomogeneous.

Theorem 4.2 (phase transition in unweighted graphs)**.**

Let $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T$ be the cluster partitioned eigenvector matrix associated with the graph Laplacian matrix $\mathbf{L}$ obtained by solving (4.1), where $\mathbf{Y}_k \in \mathbb{R}^{n_k \times (K-1)}$ with its rows indexing the nodes in cluster k. Let $c^* = \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{S_{2:K}(\mathbf{L}_k)}{n} \right\}$. Under the homogeneous RIM in (1.1) with constant interconnection probability $p_{ij} = p$, there exists a critical value p^* such that the following holds almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$:

$$\text{(a)} \begin{cases} \text{If } p \leq p^*, \ \frac{S_{2:K}(\mathbf{L})}{n} = (K-1)p; \\ \text{If } p > p^*, \ c^* + (K-1)\left(1 - \frac{n_{\max}}{n}\right)p \leq \frac{S_{2:K}(\mathbf{L})}{n} \leq c^* + (K-1)\left(1 - \frac{n_{\min}}{n}\right)p. \end{cases}$$

In particular, if $p > p^*$ and $c = 1$, $\frac{S_{2:K}(\mathbf{L})}{n} = c^* + \frac{(K-1)^2}{K}p$.

Furthermore,

$$\text{(b)} \begin{cases} \text{If } p < p^*, \ \mathbf{Y}_k = \mathbf{1}_{n_k} \mathbf{1}_{K-1}^T \mathbf{V}_k = \left[v_1^k \mathbf{1}_{n_k}, v_2^k \mathbf{1}_{n_k}, \ldots, v_{K-1}^k \mathbf{1}_{n_k} \right], \ \forall \ k \in \{1, 2, \ldots, K\}; \\ \text{If } p > p^*, \ \mathbf{Y}_k^T \mathbf{1}_{n_k} = \mathbf{0}_{K-1}, \ \forall \ k \in \{1, 2, \ldots, K\}; \\ \text{If } p = p^*, \ \forall \ k \in \{1, 2, \ldots, K\}, \ \mathbf{Y}_k = \mathbf{1}_{n_k} \mathbf{1}_{K-1}^T \mathbf{V}_k \text{ or } \mathbf{Y}_k^T \mathbf{1}_{n_k} = \mathbf{0}_{K-1}, \end{cases}$$

where $\mathbf{V}_k = \operatorname{diag}(v_1^k, v_2^k, \ldots, v_{K-1}^k) \in \mathbb{R}^{(K-1) \times (K-1)}$ *is a diagonal matrix.*

Finally, p^* *satisfies:*

(c) $p_{LB} \leq p^* \leq p_{UB}$*, where*

$$\begin{cases} p_{LB} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1) n_{\max}}; \\ p_{UB} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1) n_{\min}}. \end{cases}$$

In particular, $p_{LB} = p_{UB}$ *when* $c = 1$.

Proof. The proof is given in Appendix C.2. □

Theorem 4.2 (a) establishes a phase transition of the normalized partial eigenvalue sum $\frac{S_{2:K}(\mathbf{L})}{n}$ at some critical value p^*, called the critical phase transition threshold. When $p \leq p^*$ the quantity $\frac{S_{2:K}(\mathbf{L})}{n}$ is exactly $(K-1)p$. When $p > p^*$ the slope in p of $\frac{S_{2:K}(\mathbf{L})}{n}$ changes and the intercept $c^* = \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{S_{2:K}(\mathbf{L}_k)}{n} \right\}$ depends on the cluster having the smallest partial eigenvalue sum. When all clusters have the same size (i.e., $n_{\max} = n_{\min} = \frac{n}{K}$) so that $c = 1$, $\frac{S_{2:K}(\mathbf{L})}{n}$ undergoes a slope change from $K-1$ to $\frac{(K-1)^2}{K}$.

Theorem 4.2 (b) establishes that $p > p^*$ makes the entries of the matrix $\mathbf{Y}_k$ incoherent, making it impossible for SGC to separate the clusters. On the other hand, $p < p^*$ makes $\mathbf{Y}_k$ coherent, and hence the row vectors in the eigenvector matrix $\mathbf{Y}$ possess cluster-wise separability. This is stated as follows.

Corollary 4.3 (separability of the row vectors in the eigenvector matrix $\mathbf{Y}$ when $p < p^*$)**.**

Under the same assumptions as in Theorem 4.2, when $p < p^*$*, the following properties of* $\mathbf{Y}$ *hold almost surely as* $n_k \to \infty$ *and* $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

(a) *The columns of* $\mathbf{Y}_k$ *are constant vectors.*

(b) *Each column of* $\mathbf{Y}$ *has at least two nonzero cluster-wise constant components,* *and these constants have alternating signs such that their weighted sum equals* 0 *(i.e.,* $\sum_k n_k v_j^k = 0,\ \forall\ j \in \{1, 2, \ldots, K-1\}$*).*

(c) *No two columns of* $\mathbf{Y}$ *have the same sign on the cluster-wise nonzero components.*

Proof. The proof is given in Appendix C.3. □

These properties imply that for $p < p^*$ the rows in $\mathbf{Y}_k$ corresponding to different nodes are identical, while the row vectors in $\mathbf{Y}_k$ and $\mathbf{Y}_\ell$, $k \neq \ell$, corresponding to different clusters are distinct. Hence, K-means clustering on these row vectors can group the nodes into correct clusters. Note that when $p > p^*$, from Theorem 4.2 (b) the row vectors of $\mathbf{Y}_k$ corresponding to the same cluster sum to a zero vector. This means that the entries of each column in $\mathbf{Y}_k$ have alternating signs and hence K-means clustering on the rows of $\mathbf{Y}$ yields incorrect clusters.

Furthermore, as a demonstration of the breakdown condition in Theorem 4.1, observe that when $p_{ij} = p$, Theorem 4.1 implies that the matrix $\frac{\widetilde{\mathbf{A}}}{n}$ reduces to a diagonal matrix $p\mathbf{I}_{K-1}$, and from (C.15) we know that $\lambda_j\left(\frac{\mathbf{L}}{n}\right) = p$ for $j = 2, 3, \ldots, K$ almost surely when $p < p^*$. Therefore, under the homogeneous RIM spectral clustering can only be successful when p is below the critical threshold value p^*.

Theorem 4.2 (c) provides upper and lower bounds on the critical threshold value p^* for the phase transition to occur when $p_{ij} = p$. These bounds are determined by the cluster having the smallest partial eigenvalue sum $S_{2:K}(\mathbf{L}_k)$, the number of clusters K, and the size of the largest and smallest cluster ($n_{\max}$ and $n_{\min}$). When all cluster sizes are identical (i.e., $c = 1$), these bounds become tight. Based on Theorem 4.2 (c), the following corollary specifies the properties of p^* and the connection to algebraic connectivity of each cluster.

Corollary 4.4 (properties of p^* and its connection to algebraic connectivity)**.** *Under the same assumptions as in Theorem 4.2, the following statements hold almost surely as* $n_k \to \infty$ *and* $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

(a) *If* $\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k) = \Omega(n_{\max})$*, then* $p^* > 0$*.*

(b) *If* $\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k) = o(n_{\min})$*, then* $p^* = 0$*.*

(c) $\frac{\min_{k \in \{1,2,\ldots,K\}} \lambda_2(\mathbf{L}_k)}{n_{\max}} \leq p^* \leq \frac{\min_{k \in \{1,2,\ldots,K\}} \lambda_K(\mathbf{L}_k)}{n_{\min}}$.

Proof. The proof is given in Appendix C.4. □

The following corollary specifies the bounds on the critical value p^* for some special types of clusters. These results provide theoretical justification of the intuition that strongly connected clusters, e.g., complete graphs, have high critical threshold value, and weakly connected clusters, e.g., star graphs, have low critical threshold value.

Corollary 4.5 (bounds on the critical value p^* for special type of cluster)**.** *Under the same assumptions as in Theorem 4.2, the following statements hold almost surely as* $n_k \to \infty$ *and* $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

(a) *If each cluster is a complete graph, then* $c \leq p^* \leq 1$*.*

(b) *If each cluster is a star graph and* $K < n_{\min}$*, then* $p^* = 0$*.*

Proof. The proof is given in Appendix C.5. □

Furthermore, if the internal connectivity of each cluster (i.e., $\mathbf{A}_k$) is a Erdos-Renyi random graph with edge connection probability p_k (i.e., the SBM), under the same assumptions as in Theorem 4.2 we can show that almost surely,

$$c \cdot \min_{k \in \{1,2,\ldots,K\}} p_k \leq p^* \leq \frac{1}{c} \cdot \min_{k \in \{1,2,\ldots,K\}} p_k. \tag{4.2}$$

The proof of (4.2) is given in Appendix C.6.

The next corollary summarizes the results from Theorem 4.2 for the case of $K = 2$ to elucidate the phase transition phenomenon. Note that it follows from Corollary 4.6 (b) that below the phase transition ($p < p^*$) the rows in $\mathbf{Y}$ corresponding to different clusters are constant vectors with entries of opposite signs, and thus K-means clustering is capable of yielding correct clusters. On the other hand, above the phase transition ($p > p^*$) the entries corresponding to each cluster have alternating signs, and thus K-means clustering fails.

Corollary 4.6 (special case of Theorem 4.2 when $K = 2$)**.**
When $K = 2$, let $\mathbf{Y} = [\mathbf{y}_1^T \ \mathbf{y}_2^T]^T$ and let $c^* = \frac{\lambda_2(\mathbf{L}_1)+\lambda_2(\mathbf{L}_2)-\left|\lambda_2(\mathbf{L}_1)-\lambda_2(\mathbf{L}_2)\right|}{2n}$. Then there exists a critical value p^* such that the following holds almost surely as $n_1, n_2 \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$.

$$\text{(a)} \begin{cases} \text{If } p \leq p^*, \ \frac{\lambda_2(\mathbf{L})}{n} = p; \\ \text{If } p > p^*, \ c^* + \frac{c}{1+c}p \leq \frac{\lambda_2(\mathbf{L})}{n} \leq c^* + \frac{1}{1+c}p. \end{cases}$$

In particular, if $p > p^*$ and $c = 1$, $\frac{S_{2:K}(\mathbf{L})}{n} = c^* + \frac{p}{2}$.

$$\text{(b)} \begin{cases} \text{If } p < p^*, \ \sqrt{\frac{nn_1}{n_2}}\mathbf{y}_1 = \pm\mathbf{1}_{n_1} \text{ and } \sqrt{\frac{nn_2}{n_1}}\mathbf{y}_2 = \mp\mathbf{1}_{n_2}; \\ \text{If } p > p^*, \ \mathbf{y}_1^T\mathbf{1}_{n_1} = 0 \text{ and } \mathbf{y}_2^T\mathbf{1}_{n_2} = 0; \end{cases}$$

(c) *$p_{LB} \leq p^* \leq p_{UB}$, where*

$$\begin{cases} p_{LB} = \frac{\lambda_2(\mathbf{L}_1)+\lambda_2(\mathbf{L}_2)-\left|\lambda_2(\mathbf{L}_1)-\lambda_2(\mathbf{L}_2)\right|}{n+|n_1-n_2|}; \\ p_{UB} = \frac{\lambda_2(\mathbf{L}_1)+\lambda_2(\mathbf{L}_2)-\left|\lambda_2(\mathbf{L}_1)-\lambda_2(\mathbf{L}_2)\right|}{n-|n_1-n_2|}. \end{cases}$$

When $c = 1$, $p_{LB} = p_{UB} = \frac{\lambda_2(\mathbf{L}_1)+\lambda_2(\mathbf{L}_2)-\left|\lambda_2(\mathbf{L}_1)-\lambda_2(\mathbf{L}_2)\right|}{n}$.

Proof. The proof is given in Appendix C.7. □

The above phase transition analysis can also be applied to the inhomogeneous RIM for which the p_{ij}'s are not constant. Let $p_{\min} = \min_{i \neq j} p_{ij}$ and $p_{\max} = \max_{i \neq j} p_{ij}$. The corollary below shows that under the inhomogeneous RIM when $p_{\max}$ is below p^*, which is the critical threshold value specified by Theorem 4.2 for the homogeneous RIM, the smallest $K - 1$ nonzero eigenvalues of the graph Laplacian matrix $\frac{\mathbf{L}}{n}$ lie within the internal $[p_{\min}, p_{\max}]$ with probability one.

Corollary 4.7 (bounds on the smallest $K - 1$ nonzero eigenvalues of $\mathbf{L}$ under the inhomogeneous RIM)**.**

Under the RIM with interconnection probabilities $\{p_{ij}\}$, let $p_{\min} = \min_{i \neq j} p_{ij}$, $p_{\max} = \max_{i \neq j} p_{ij}$, and let p^* be the critical threshold value of the homogeneous RIM specified by Theorem 4.2. If $p_{\max} < p^*$, the following statement holds almost surely as $n_k \to \infty$

and $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

$$p_{\min} \leq \lambda_j \left(\frac{\mathbf{L}}{n} \right) \leq p_{\max}, \ \forall \ j = 2, 3, \ldots, K. \tag{4.3}$$

Proof. The proof is given in Appendix C.8. □

In particular, Corollary 4.7 implies that the algebraic connectivity of the inhomogeneous RIM $\lambda_2(\frac{\mathbf{L}}{n})$ is between $p_{\min}$ and $p_{\max}$ almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$.

For graphs following the inhomogeneous RIM, Theorem 4.8 below establishes that accurate clustering is possible if it can be determined that $p_{\max} < p^*$. As defined in Theorem 4.2, let $\mathbf{Y} \in \mathbb{R}^{n \times (K-1)}$ be the eigenvector matrix of $\mathbf{L}$ under the inhomogeneous RIM, and let $\widetilde{\mathbf{Y}} \in \mathbb{R}^{n \times (K-1)}$ be the eigenvector matrix of the graph Laplacian $\widetilde{\mathbf{L}}$ of another random graph, independent of $\mathbf{L}$, generated by a homogeneous RIM with cluster interconnectivity parameter p. We can specify the distance between the subspaces spanned by the columns of $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ by inspecting their principal angles [97]. Since $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ both have orthonormal columns, the vector $\mathbf{v}$ of $K-1$ principal angles between their column spaces is $\mathbf{v} = [\cos^{-1} \sigma_1(\mathbf{Y}^T \widetilde{\mathbf{Y}}), \ldots, \cos^{-1} \sigma_{K-1}(\mathbf{Y}^T \widetilde{\mathbf{Y}})]^T$, where $\sigma_k(\mathbf{M})$ is the k-th largest singular value of real rectangular matrix $\mathbf{M}$. Let $\mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}) = \operatorname{diag}(\mathbf{v})$, and let $\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})$ be defined entrywise. When $p < p^*$, the following theorem provides an upper bound on the Frobenius norm of $\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})$, which is denoted by $\| \sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}) \|_F$.

Theorem 4.8 (distance between column spaces spanned by $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$)**.**

Under the RIM with interconnection probabilities $\{p_{ij}\}$*, let* p^* *be the critical threshold value for the homogeneous RIM specified by Theorem 4.2, and define* $\delta_{p,n} = \min\{p, |\lambda_{K+1}(\frac{\mathbf{L}}{n}) - p|\}$*. For a fixed* p*, if* $p < p^*$ *and* $\delta_{p,n} \to \delta_p > 0$ *as* $n_k \to \infty$*,* *the following statement holds almost surely as* $n_k \to \infty$ *and* $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

$$\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F \leq \frac{\|\mathbf{L} - \widetilde{\mathbf{L}}\|_F}{n\delta_p}. \tag{4.4}$$

Furthermore, let $p_{\max} = \max_{i \neq j} p_{ij}$*. If* $p_{\max} < p^*$*,*

$$\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F \leq \min_{p \leq p_{\max}} \frac{\|\mathbf{L} - \widetilde{\mathbf{L}}\|_F}{n\delta_p}. \tag{4.5}$$

Proof. The proof is given in Appendix C.9. □

As established in Corollary 4.3, under the homogeneous RIM when $p < p^*$ the row vectors of the eigenvector matrix $\widetilde{\mathbf{Y}}$ are perfectly cluster-wise separable as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Under the inhomogeneous RIM, thus it establishes that cluster separability can still be expected provided that $\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F$ is small and $p < p^*$. As a result, we can bound the clustering accuracy under the inhomogeneous RIM by inspecting the upper bound (4.4) on $\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}\|_F$. Note that if $p_{\max} < p^*$, we can obtain a tighter upper bound on (4.4).

Next we extend Theorem 4.2 to undirected weighted random graphs obeying the homogeneous RIM. The edges within each cluster are assumed to have nonnegative weights and the weights of inter-cluster edges are assumed to be independently drawn from a common nonnegative bounded distribution. Let $\mathbf{W}$ denote the $n \times n$ symmetric nonnegative weight matrix of the entire graph. Then the corresponding graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{S} - \mathbf{W}$, where $\mathbf{S} = \mathrm{diag}(\mathbf{W}\mathbf{1}_n)$ is the diagonal matrix of nodal strengths of the weighted graph. Similarly, the symmetric graph Laplacian matrix $\mathbf{L}_k$ of each cluster can be defined. The following theorem establishes a phase transition phenomenon for such weighted graphs. The critical value depends not only on the inter-cluster edge connection probability but also on the mean of inter-cluster edge weights.

Theorem 4.9 (phase transition in weighted graphs).

Under the same assumptions as in Theorem 4.2, assume the weight matrix $\mathbf{W}$ *is symmetric, nonnegative, and bounded, and the weights of the upper triangular part of* $\mathbf{W}$ *are independently drawn from a common nonnegative bounded distribution with mean* $\overline{W}$. *Let* $t = p \cdot \overline{W}$ *and let* $c^* = \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{S_{2:K}(\mathbf{L}_k)}{n} \right\}$. *Then there exists a critical value* t^* *such that the following holds almost surely as* $n_k \to \infty$ *and* $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

$$\text{(a)} \begin{cases} \text{If } t \leq t^*, \ \frac{S_{2:K}(\mathbf{L})}{n} = (K-1)t; \\ \text{If } t > t^*, \ c^* + (K-1)\left(1 - \frac{n_{\max}}{n}\right)t \leq \frac{S_{2:K}(\mathbf{L})}{n} \leq c^* + (K-1)\left(1 - \frac{n_{\min}}{n}\right)t. \end{cases}$$

In particular, if $t > t^*$ *and* $c = 1$, $\frac{S_{2:K}(\mathbf{L})}{n} = c^* + \frac{(K-1)^2}{K} t$.

$$\text{(b)} \begin{cases} \text{If } t < t^*, \ \mathbf{Y}_k = \mathbf{1}_{n_k} \mathbf{1}_{K-1}^T \mathbf{V}_k = \left[v_1^k \mathbf{1}_{n_k}, v_2^k \mathbf{1}_{n_k}, \ldots, v_{K-1}^k \mathbf{1}_{n_k} \right], \ \forall \ k \in \{1,2,\ldots,K\}; \\ \text{If } t > t^*, \ \mathbf{Y}_k^T \mathbf{1}_{n_k} = \mathbf{0}_{K-1} \ \forall \ k \in \{1,2,\ldots,K\}; \\ \text{If } t = t^*, \ \forall \ k \in \{1,2,\ldots,K\}, \ \mathbf{Y}_k = \mathbf{1}_{n_k} \mathbf{1}_{K-1}^T \mathbf{V}_k \text{ or } \mathbf{Y}_k^T \mathbf{1}_{n_k} = \mathbf{0}_{K-1}, \end{cases}$$

where $\mathbf{V}_k = \operatorname{diag}(v_1^k, v_2^k, \ldots, v_{K-1}^k) \in \mathbb{R}^{(K-1)\times(K-1)}$ *is a diagonal matrix.*

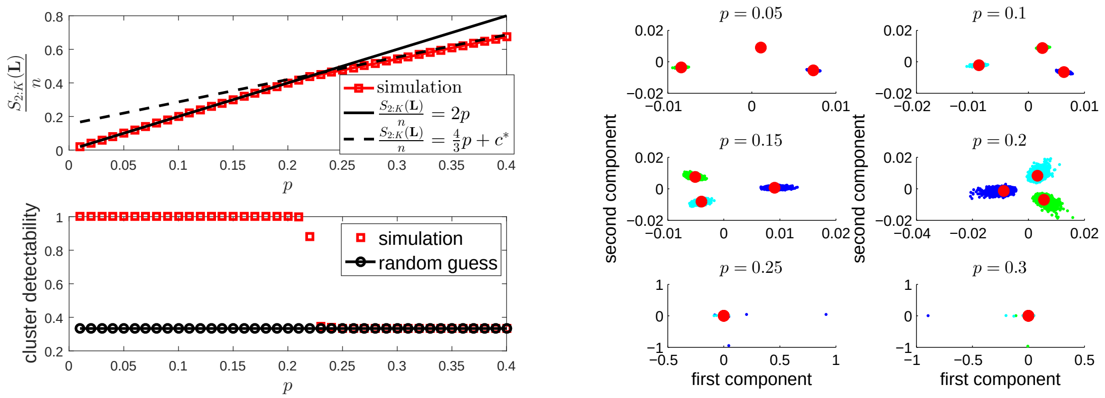
(c) $t_{LB} \leq t^* \leq t_{UB}$, *where*

$$\begin{cases} t_{LB} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1) n_{\max}}; \\ t_{UB} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1) n_{\min}}. \end{cases}$$

In particular, $t_{LB} = t_{UB}$ *when* $c = 1$.

Proof. The proof is given in Appendix C.10. □

Theorem 4.9 reduces to the case of unweighted graphs in Theorem 4.2 when $\overline{W} = 1$. Theorem 4.1 and Theorem 4.8 can be extended to weighted graphs under the inhomogeneous RIM, where the edge weights between clusters i and j, $i \neq j$, are independently drawn from a common nonnegative distribution with mean $\overline{W}_{ij}$ and bounded fourth moment.



(a) Phase transition in normalized partial sum of eigenvalues $\frac{S_{2:K}(\mathbf{L})}{n}$ and cluster detectability.

(b) Row vectors in $\mathbf{Y}$ with respect to different p. Colors and red solid circles represent clusters and cluster-wise centroids.

Figure 4.1: Phase transition of clusters generated by Erdos-Renyi random graphs. $K = 3$, $n_1 = n_2 = n_3 = 8000$, and $p_1 = p_2 = p_3 = 0.25$. The empirical critical phase transition threshold value predicted by Theorem 4.2 is $p^* = 0.2301$.

4.3 Numerical Experiments: Validation of Phase Transitions in Simulated Networks

We simulate graphs generated by the homogeneous RIM to validate the phase transition analysis. Fig. 4.1 (a) shows the phase transition in partial eigenvalue sum $S_{2:K}(\mathbf{L})$ and cluster detectability (i.e., the fraction of correctly identified nodes) for clusters generated by Erdos-Renyi random graphs with varying inter-cluster edge connection probability p. Random guessing leads to cluster detectability $\frac{1}{K}$. The simulation results verify Theorem 4.2 that the simulated graphs transition from almost perfect detectability to low detectability and undergo a change of slope in $S_{2:K}(\mathbf{L})$ when p exceeds the critical value p^*. In addition, the separability of the row vectors of $\mathbf{Y}$ in Corollary 4.3 is demonstrated in Fig. 4.1 (b). Similar phase transitions can be found for clusters generated by the Watts-Strogatz small world network model [163] in Fig. 4.2. Fig. 4.3 shows phase transition of weighted graphs where the inter-cluster edge weights are independently drawn from a common exponential distribution with mean $\overline{W}$, which verifies the results in Theorem 4.9. The effect of different cluster

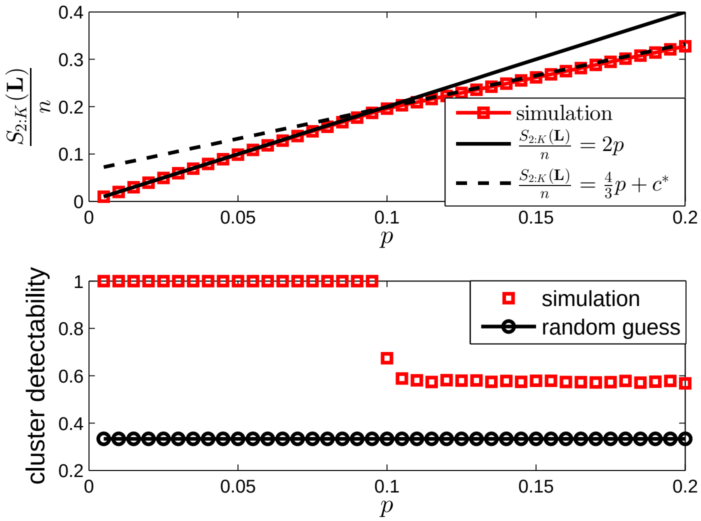


Figure 4.2: Phase transition of clusters generated by the Watts-Strogatz small world network model. $K = 3$, $n_1 = n_2 = n_3 = 1000$, average number of neighbors = 200, and rewire probability for each cluster is 0.4, 0.4, and 0.6. The empirical critical threshold value predicted by Theorem 4.2 is $p^* = 0.0985$.

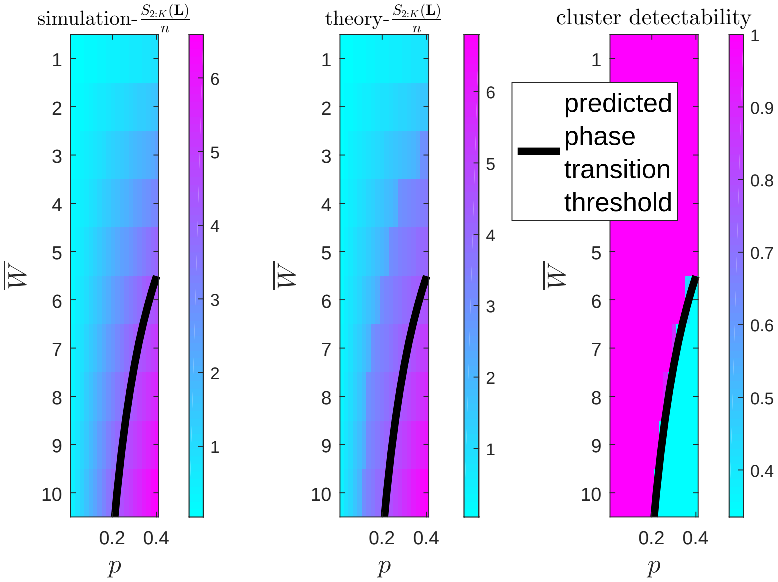


Figure 4.3: Phase transition of clusters generated by Erdos-Renyi random graphs with exponentially distributed edge weight with mean 10. $K = 3$, $n_1 = n_2 = n_3 = 8000$, and $p_1 = p_2 = p_3 = 0.25$. The predicted phase transition threshold curve from Theorem 4.9 is $p \cdot \overline{W} = \frac{K \min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1)n}$.

sizes and sensitivity to the inhomogeneous RIM are discussed in Appendix C.11.

CHAPTER V

AMOS: An Automated Model Order Selection Criterion for Spectral Graph Clustering

One of the longstanding open problems in unsupervised classification is the so-called model order selection problem: automated selection of the correct number of classes or clusters. In the context of spectral graph clustering (SGC), this is equivalent to the problem of finding the number of connected components or communities in an undirected graph. In this chapter we propose a solution to the SGC model selection problem under a homogeneous random interconnection model (RIM) using a novel selection criterion that falls out of an asymptotic phase transition analysis established in Chapter IV, which we call automated model order selection (AMOS).

AMOS works by sequentially increasing the model order while running multi-stage tests for testing for RIM structure. Specifically, for a given model order and an estimated cluster membership map obtained from SGC, we first test for local RIM structure for a single cluster pair using a binomial test of homogeneity. This is repeated for all cluster pairs and, if they pass the RIM test, we proceed to the second stage of testing, otherwise we increase the model order and start again. The second stage consists of testing whether the RIM is globally homogeneous or inhomogeneous. This is where the phase transition results are used - if any of the estimated inter-cluster connection probabilities exceed the critical phase transition threshold the model order is increased. In this manner, the outputs from AMOS are the SGC clustering results of minimal model order that are deemed reliable.

Comparing to other automated graph clustering methods, experiments on real-world network datasets show that the AMOS algorithm indeed outputs clusters that are more consistent with the ground truth meta information. For example, when applied to network data with longitude and latitude meta information, such as the Internet backbone map across North American and Europe, and the Minnesota road map, the clusters identified by the AMOS algorithm are more consistent with known geographic separations.

5.1 Automated Model Order Selection (AMOS) Algorithm for Spectral Graph Clustering

Based on the phase transition analysis in Sec. 4.2, we propose an automated model order selection (AMOS) algorithm for selecting the number of clusters in spectral graph clustering (SGC). This algorithm produces p-values of hypothesis tests for testing the RIM and phase transition. In particular, under the homogeneous RIM, we can estimate the critical phase transition threshold for each put/ative cluster found and use this estimate to construct a test of reliability of the cluster. The statistical tests in the AMOS algorithm are implemented in two phases. The first phase is to test the RIM assumption based on the interconnectivity pattern of each cluster (Sec. 5.1.2), and the second phase is to test the homogeneity and variation of the interconnectivity parameter p_{ij} for every cluster pair i and j in addition to making comparisons to the critical phase transition threshold (Sec. 5.1.3). The flow diagram of the proposed algorithm is displayed in Fig. 5.1, and the algorithm is summarized in Algorithm 5.2. The AMOS codes can be downloaded from https://github.com/tgensol/AMOS. Next we explain the functionality of each block in the diagram.

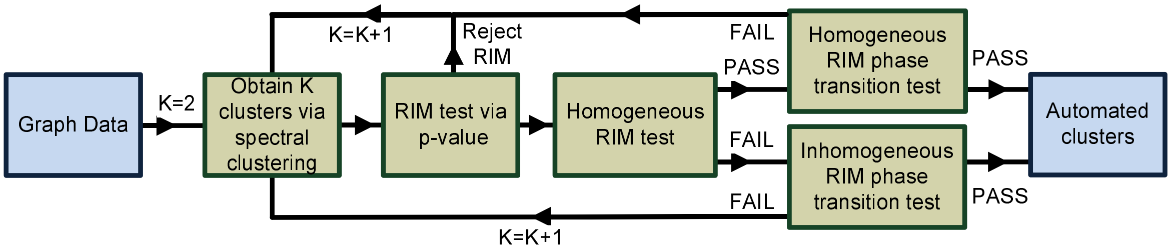

Graph Data
K=2
Obtain K clusters via spectral clustering
RIM test via p-value
Homogeneous RIM test
Homogeneous RIM phase transition test
Inhomogeneous RIM phase transition test
Automated clusters
K=K+1
Reject RIM
FAIL
PASS
FAIL
FAIL
PASS
PASS
K=K+1

Figure 5.1: Flow diagram of the proposed automated model order selection (AMOS) scheme in spectral graph cluster (SGC).

5.1.1 Input graph data and spectral clustering

The input graph data is a matrix that can be a symmetric adjacency matrix $\mathbf{A}$, a degree-normalized symmetric adjacency matrix $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, a symmetric weight matrix $\mathbf{W}$, or a normalized symmetric weight matrix $\mathbf{S}^{-\frac{1}{2}}\mathbf{W}\mathbf{S}^{-\frac{1}{2}}$, where $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_n)$ and $\mathbf{S} = \mathrm{diag}(\mathbf{W}\mathbf{1}_n)$ are assumed invertible. Spectral clustering is then implemented on the input data to produce K clusters $\{\widehat{G}_k\}_{k=1}^{K}$, where $\widehat{G}_k$ is the k-th identified cluster with number of nodes $\widehat{n}_k$ and number of edges $\widehat{m}_k$. Initially K is set to 2. The AMOS algorithm works by iteratively increasing K and performing spectral clustering on the data until the output clusters meet a level of significance criterion specified by the RIM test and phase transition estimator. In particular, the incremental eigenpair computation method developed in Chapter II can be readily applied to AMOS.

5.1.2 RIM test via p-value for local homogeneity testing

Given clusters $\{\widehat{G}_k\}_{k=1}^{K}$ obtained from spectral clustering with model order K, let $\widehat{\mathbf{C}}_{ij}$ be the $\widehat{n}_i \times \widehat{n}_j$ interconnection matrix of edges connecting clusters i and j. Our goal is to compute a p-value to test the hypothesis that the matrix $\mathbf{A}$ in (1.1) satisfies the RIM. More specifically, we are testing the null hypothesis that $\widehat{\mathbf{C}}_{ij}$ *is a realization of a random matrix with i.i.d. Bernoulli entries* (RIM) and the alternative hypothesis that $\widehat{\mathbf{C}}_{ij}$ *is a realization of a random matrix with independent Bernoulli entries* (not RIM), for all $i \neq j$, $i > j$. Since the RIM homogeneity model for the interconnection

Algorithm 5.1 p-value computation of V-test for the RIM test

```
**Input:** An $n_i \times n_j$ interconnection matrix $\widehat{\mathbf{C}}_{ij}$
**Output:** p-value$(i, j)$
$\mathbf{x} = \widehat{\mathbf{C}}_{ij}\mathbf{1}_{n_j}$ (# of nonzero entries of each row in $\widehat{\mathbf{C}}_{ij}$)
$\mathbf{y} = n_j\mathbf{1}_{n_i} - \mathbf{x}$ (# of zero entries of each row in $\widehat{\mathbf{C}}_{ij}$)
$X = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{1}_{n_i}$ and $Y = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{1}_{n_i}$.
$N = n_i n_j(n_j - 1)$ and $V = \left(\sqrt{X} + \sqrt{Y}\right)^2$.
Compute test statistic $Z = \frac{V-N}{\sqrt{2N}}$
Compute p-value$(i, j)$= $2 \cdot \min\{\Phi(Z), 1 - \Phi(Z)\}$
```

matrices $\mathbf{C}_{ij}$ will only be valid when the clusters have been correctly identified, this RIM test can be used to test the quality of a graph clustering algorithm.

To compute a p-value for the RIM we use the V-test [129] for homogeneity testing of the row sums or column sums of $\widehat{\mathbf{C}}_{ij}$. Specifically, given s independent binomial random variables, the V-test tests that they are all identically distributed. For concreteness, here we apply the V-test to the row sums. Given a candidate set of clusters, the V-test is applied independently to each of the $\binom{K}{2}$ interconnection matrices $\widehat{\mathbf{C}}_{ij}$.

For any interconnection matrix $\widehat{\mathbf{C}}_{ij}$ the test statistic Z of the V-test converges to a standard normal distribution as $n_i, n_j \to \infty$, and the p-value for the hypothesis that the row sums of $\widehat{\mathbf{C}}_{ij}$ are i.i.d. is p-value$(i, j) = 2 \cdot \min\{\Phi(Z), 1 - \Phi(Z)\}$, where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard normal distribution. The proposed V-test procedure is summarized in Algorithm 5.1. The RIM test on $\widehat{\mathbf{C}}_{ij}$ rejects the null hypothesis if p-value$(i, j) \leq \eta$, where η is the desired single comparison significance level. Since the $\mathbf{C}_{ij}$'s are independent, the p-value threshold parameter η can be easily translated into a multiple comparisons significance level for detecting homogeneity of all $\mathbf{C}_{ij}$'s. It can also be translated into a threshold for testing the homogeneity of at least one of these matrices using family-wise error rate Bonferroni corrections or false discovery rate analysis [14, 147].

Algorithm 5.2 Automated model order selection (AMOS) algorithm for spectral graph clustering (SGC)

```
**Input:** a connected undirected weighted graph, p-value significance level $\eta$, RIM confidence interval parameters $\alpha$, $\alpha'$
**Output:** number of clusters $K$ and identified clusters $\{\widehat{G}_k\}_{k=1}^K$
Initialization: $K = 2$. Flag $= 1$.
while Flag$= 1$ do
    Obtain $K$ clusters $\{\widehat{G}_k\}_{k=1}^K$ via spectral clustering $(*)$
    # Local homogeneity testing #
    for $i = 1$ to $K$ do
        for $j = i + 1$ to $K$ do
            Calculate p-value$(i, j)$ from Algorithm 5.1.
            if p-value$(i, j) \leq \eta$ then Reject RIM
                Go back to $(*)$ with $K = K + 1$.
            end if
        end for
    end for
    Estimate $\widehat{p}$, $\widehat{\overline{W}}$, $\{\widehat{p}_{ij}\}$, $\{\widehat{\overline{W}}_{ij}\}$, and $\widehat{t}_{\mathrm{LB}}$ specified in Sec. 5.1.3.
    # Homogeneous RIM test #
    if $\widehat{p}$ lies within the confidence interval in (5.1) then
        # Homogeneous RIM phase transition test #
        if $\widehat{p} \cdot \widehat{\overline{W}} < \widehat{t}_{\mathrm{LB}}$ then Flag$= 0$.
        else Go back to $(*)$ with $K = K + 1$.
        end if
    else if $\widehat{p}$ does not lie within the confidence interval in (5.1) then
        # Inhomogeneous RIM phase transition test #
        if $\prod_{i=1}^K \prod_{j=i+1}^K F_{ij}\left(\frac{\widehat{t}_{\mathrm{LB}}}{\widehat{\overline{W}}_{ij}}, \widehat{p}_{ij}\right) \geq 1 - \alpha'$ then Flag$= 0$.
        else Go back to $(*)$ with $K = K + 1$.
        end if
    end if
end while
Output $K$ clusters $\{\widehat{G}_k\}_{k=1}^K$.
```

5.1.3 Phase transition tests

Once the identified clusters $\{\widehat{G}_k\}_{k=1}^K$ pass the RIM test in Sec. 5.1.2, one can empirically determine the reliability of the clustering results using the phase transition analysis in CH. IV. AMOS first tests the assumption of homogeneous RIM, and performs the *homogeneous RIM phase transition test* by comparing the empirical estimate $\widehat{p}$ of the interconnectivity parameter p with the empirical estimate $\widehat{p}_{\mathrm{LB}}$ of the lower bound p_{LB} on p^* based on Theorem 4.2. If the test on the assumption of homogeneous RIM fails, AMOS then performs the *inhomogeneous RIM phase transition test* by comparing the empirical estimate $\widehat{p}_{\max}$ of $p_{\max}$ with $\widehat{p}_{\text{LB}}$ based on Theorem 4.8.

In a nutshell, after the identified clusters $\{\widehat{G}_k\}_{k=1}^K$ pass the RIM test in Sec. 5.1.2, the AMOS algorithm (Fig. 5.1) runs a serial process of statistical tests, including homogeneous RIM test, homogeneous and inhomogeneous RIM phase transition tests. Each of these is considered separately in what follows.

- **Homogeneous RIM test**:

The homogeneous RIM test is summarized as follows. Given clusters $\{\widehat{G}_k\}_{k=1}^K$, we estimate the interconnectivity parameters $\{\widehat{p}_{ij}\}$ by $\widehat{p}_{ij} = \frac{\widehat{m}_{ij}}{\widehat{n}_i \widehat{n}_j}$, where $\widehat{m}_{ij}$ is the number of inter-cluster edges between clusters i and j, and $\widehat{p}_{ij}$ is the maximum likelihood estimator (MLE) of p_{ij}. Under the homogeneous RIM, the estimate of the parameter p is $\widehat{p} = \frac{2(m-\sum_{k=1}^K \widehat{m}_k)}{n^2-\sum_{k=1}^K \widehat{n}_k^2}$, where $\widehat{m}_k$ is the number of within-cluster edges of cluster k and m is the total number of edges in the graph. A generalized log-likelihood ratio test (GLRT) is used to test the validity of the homogeneous RIM. By the Wilk's theorem [167], an asymptotic $100(1-\alpha)\%$ confidence interval for p in an assumed homogeneous RIM is

$$\begin{aligned}
&\Bigg\{ p : \xi_{\binom{K}{2}-1,1-\frac{\alpha}{2}} \leq 2\sum_{i=1}^{K}\sum_{j=i+1}^{K} \mathbb{I}_{\{\widehat{p}_{ij}\in(0,1)\}} \left[\widehat{m}_{ij}\ln\widehat{p}_{ij} + (\widehat{n}_i\widehat{n}_j - \widehat{m}_{ij})\ln(1-\widehat{p}_{ij})\right] \\
&-2\left(m - \sum_{k=1}^{K} \widehat{m}_k\right)\ln p - \left[n^2 - \sum_{k=1}^{K} \widehat{n}_k^2 - 2\left(m-\sum_{k=1}^{K}\widehat{m}_k\right)\right]\ln(1-p) \leq \xi_{\binom{K}{2}-1,\frac{\alpha}{2}} \Bigg\},
\end{aligned} \tag{5.1}$$

where $\xi_{q,\alpha}$ is the upper α-th quantile of the central chi-square distribution with degree of freedom q. The derivation of the confidence interval in (5.1) is given in Appendix D.1.

The identified clusters pass the homogeneous RIM test if $\widehat{p}$ is within the confidence interval specified in (5.1). Intuitively, if $\widehat{p}_{ij}$ are close to $\widehat{p}$, then the interconnectivity structure of the identified clusters are regarded homogeneous. On the other hand, if there is a large variation in $\{\widehat{p}_{ij}\}$, the homogeneity RIM test fails.

- **Homogeneous RIM phase transition test**:

By Theorem 4.2, if the identified clusters follow the homogeneous RIM (i.e., pass the homogeneous RIM test), then they are deemed reliable if $\widehat{p} < \widehat{p}_{\mathrm{LB}}$, an estimate of the lower bound on the critical phase transition threshold value, which is

$$\widehat{p}_{\mathrm{LB}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\widehat{\mathbf{L}}_k)}{(K-1)\widehat{n}_{\max}}. \tag{5.2}$$

- **Inhomogeneous RIM phase transition test**:

If the identified clusters fail the homogeneous RIM test, we then use the maximum of MLEs of p_{ij}'s, denoted by $\widehat{p}_{\max} = \max_{i>j} \widehat{p}_{ij}$, as a test statistic for testing the null hypothesis H_0: $\widehat{p}_{\max} < p^*$ against the alternative hypothesis H_1: $\widehat{p}_{\max} \geq p^*$. The test accepts H_0 if $\widehat{p}_{\max} < p^*$ and hence by Theorem 4.8 the identified clusters are deemed reliable. Using the Anscombe transformation on the $\widehat{p}_{ij}$'s for variance stabilization [8], let $A_{ij}(x) = \sin^{-1}\sqrt{\frac{x+\frac{c'}{\widehat{n}_i\widehat{n}_j}}{1+\frac{2c'}{\widehat{n}_i\widehat{n}_j}}}$, where $c' = \frac{3}{8}$. By the central limit theorem, $\sqrt{4\widehat{n}_i\widehat{n}_j+2} \cdot \left(A_{ij}(\widehat{p}_{ij}) - A_{ij}(p_{ij})\right) \xrightarrow{d} N(0,1)$ for all $p_{ij} \in (0,1)$ as $\widehat{n}_i, \widehat{n}_j \to \infty$, where $\xrightarrow{d}$ denotes convergence in distribution and $N(0,1)$ denotes the standard normal distribution [8]. Therefore, under the null hypothesis that $\max_{i>j} p_{ij} < p^*$, from [23, Theorem 2.1] an asymptotic $100(1-\alpha')\%$ confidence interval for $\widehat{p}_{\max}$ is $[0,\psi]$, where $\psi(\alpha', \{\widehat{p}_{ij}\})$ is a function of the precision parameter $\alpha' \in [0,1]$ and $\{\widehat{p}_{ij}\}$, which satisfies $\prod_{i=1}^{K}\prod_{j=i+1}^{K} \Phi\left(\sqrt{4\widehat{n}_i\widehat{n}_j+2} \cdot \left(A_{ij}(\psi) - A_{ij}(\widehat{p}_{ij})\right)\right) = 1-\alpha'$, and $\Phi(\cdot)$ is the cdf of the standard normal distribution. As a result, if $\psi < p^*$, then $\widehat{p}_{\max} < p^*$ with probability at least $1-\alpha'$. Note that verifying $\psi < p^*$ is equivalent to checking the condition

$$\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij}(p^*, \widehat{p}_{ij}) \geq 1 - \alpha', \tag{5.3}$$

where $F_{ij}(p^*, \widehat{p}_{ij}) = \Phi\left(\sqrt{4\widehat{n}_i\widehat{n}_j + 2} \cdot \left(A_{ij}(p^*) - A_{ij}(\widehat{p}_{ij})\right)\right) \cdot \mathbb{I}_{\{\widehat{p}_{ij} \in (0,1)\}} + \mathbb{I}_{\{\widehat{p}_{ij} < p^*\}} \mathbb{I}_{\{\widehat{p}_{ij} \in \{0,1\}\}}$. For implementation of the inhomogeneous RIM phase transition test, we replace $F_{ij}(p^*, \widehat{p}_{ij})$ in (5.3) with $F_{ij}(\widehat{p}_{\mathrm{LB}}, \widehat{p}_{ij})$, and check whether $\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij}(\widehat{p}_{\mathrm{LB}}, \widehat{p}_{ij}) \geq 1 - \alpha'$ or not. Since $p_{\mathrm{LB}} \leq p^*$, by the monotonicity of $\Phi(\cdot)$ and $\sin^{-1}(\cdot)$, $\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij}(p_{\mathrm{LB}}, \widehat{p}_{ij}) \geq 1 - \alpha'$ implies $\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij}(p^*, \widehat{p}_{ij}) \geq 1 - \alpha'$.

These phase transition tests can be extended to weighted graphs by defining the RIM parameter $t_{ij} = p_{ij} \cdot \overline{W}_{ij}$ for weighted graphs, and using the empirical estimators $\widehat{t}_{ij} = \widehat{p}_{ij} \cdot \widehat{\overline{W}}_{ij}$ and $\widehat{t}_{\mathrm{LB}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\widehat{\mathbf{L}}_k)}{(K-1)\widehat{n}_{\max}}$ in the AMOS algorithm, where $\widehat{\overline{W}}_{ij}$ is the average weight of the inter-cluster edges between clusters i and j. The details are given in Appendix D.2.

5.1.4 Computational complexity analysis

Let n and m be the number of nodes and edges in the graph, respectively. Fixing a model order K (i.e., the number of clusters) in the AMOS iteration as displayed in Fig. 5.1, the computational complexity of AMOS consists of three parts.

1. Based on the incremental eigenpair computation method in CH. II, acquiring an additional smallest eigenvector for spectral graph clustering takes $O(m+n)$ iterations via power iteration approach, since the number of nonzero entries in the graph Laplacian matrix $\mathbf{L}$ is $m+n$.

2. The estimation of the RIM parameters $\{p_{ij}\}$ and $\{\overline{W}_{ij}\}$ takes $O(m)$ operations since they only depend on the number of edges and edge weights. The estimation of t_{LB} takes $O(K(m+n) \cdot K) = O(K^2(m+n))$ iterations for computing the least partial eigenvalue sum among K clusters.

Dataset	Node	Edge	Ground truth meta information
IEEE reliability test system (RTS) [68]	73 power stations	108 power lines	3 interconnected power subsystems
Hibernia Internet backbone map [85]	55 cities	162 connections	city names and geographic locations
Cogent Internet backbone map [85]	197 cities	243 connections	city names and geographic locations
Minnesota road map [64]	2640 intersections	3302 roads	geographic locations

Table 5.1: Summary of real-world single-layer graph datasets.

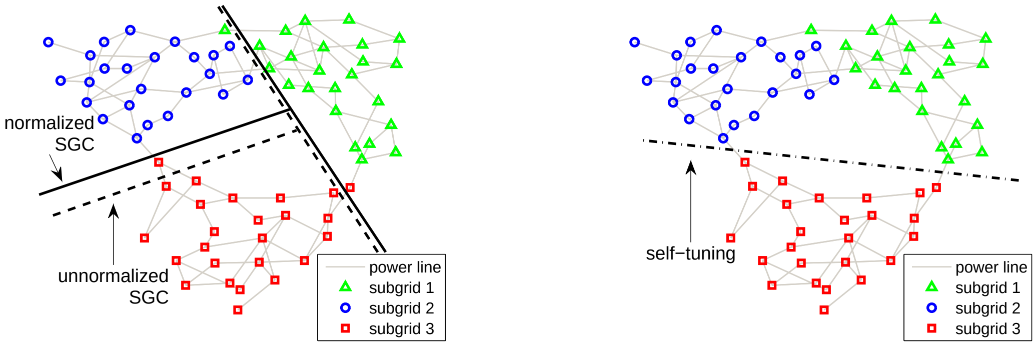
3. K-means clustering takes $O(nK^2)$ operations [174] for clustering n data points of dimension $K-1$ into K groups.

As a result, if the AMOS algorithm outputs K clusters, then the iterative process leads to total computational complexity of $O(K^3(m+n))$ operations.

5.2 Experiments: Automated model order selection (AMOS) on real-world network data

We implement the proposed AMOS algorithm (Algorithm 5.2) on several real-world network datasets with $\alpha = \alpha' = 0.05$, $\eta = 10^{-5}$ and compare the results with the self-tuning spectral clustering method proposed in [176] with $K_{\max} = \lceil n/4 \rceil$. Comparisons to the nonbacktracking matrix method [87, 139] and the Louvain method [18] can be found in the supplementary file. Note that no information beyond network topology is used for clustering. The meta information provided by these datasets are used *ex post facto* to validate the clustering results. The details of these datasets are summarized in Table 5.1.

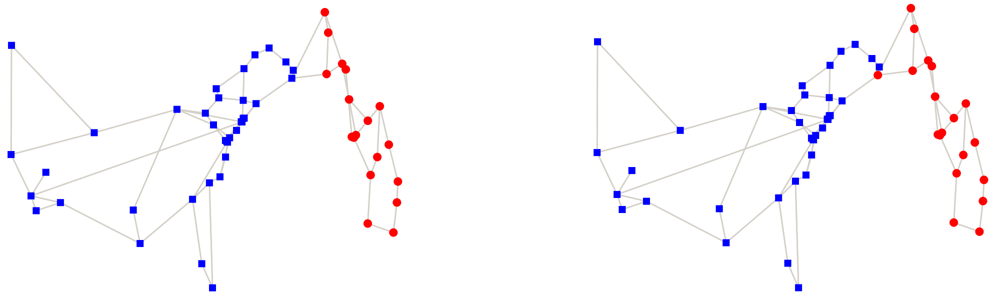
Fig. 5.2 shows the clustering results of IEEE reliability test system for power system. Marker shapes represent different power subsystems. It is observed that AMOS correctly selects the number of true clusters (subsystems), and unnormalized SGC (taking adjacency matrix as the input data) misidentifies 3 nodes while normalized



(a) Proposed AMOS algorithm. The number of clusters is 3.

(b) Self-tuning spectral clustering [176]. The number of clusters is 2.

Figure 5.2: IEEE reliability test system [68]. Normalized (unnormalized) spectral graph clustering (SGC) misidentifies 2 (3) nodes, whereas self-tuning spectral clustering fails to identify the third cluster.



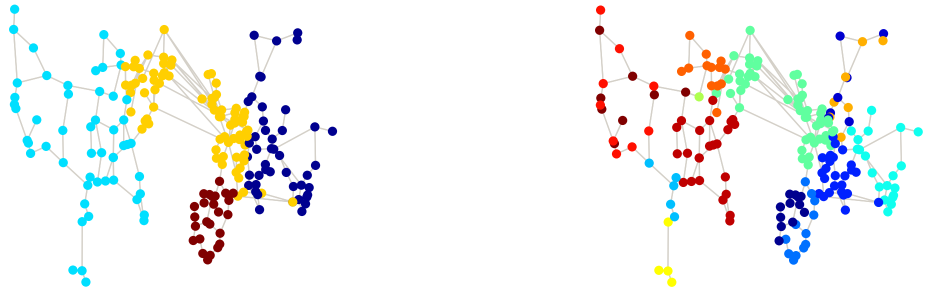
(a) Proposed AMOS algorithm. The number of clusters is 2.

(b) Self-tuning spectral clustering [176]. The number of clusters is 2.

Figure 5.3: The Hibernia Internet backbone map across Europe and North America [85]. Cities of different continents are perfectly clustered via automated SGC, whereas one city in North America is clustered with the cities in Europe via self-tuning spectral clustering. Automated clusters found by AMOS, including city names, can be found in Fig. D.3.

SGC (taking degree-normalized adjacency matrix as the input data) only misidentifies 2 nodes. Self-tuning spectral clustering fails to identify the third cluster.

We implement AMOS with normalized SGC for the rest of datasets, and the different colors represent different automated clusters. Fig. 5.3 shows the automated



(a) Proposed AMOS algorithm. The number of clusters is 4.

(b) Self-tuning spectral clustering [176]. The number of clusters is 14.

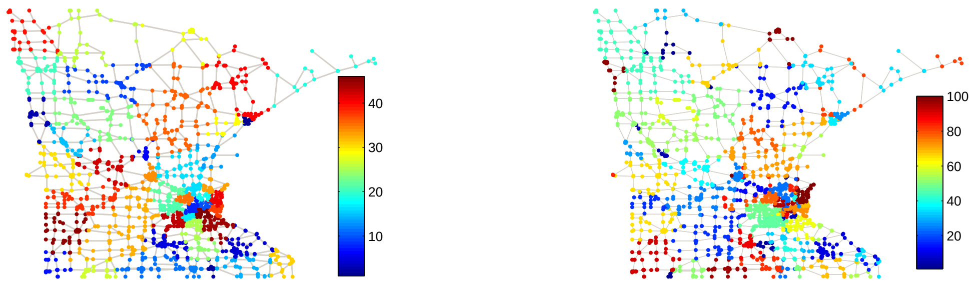
Figure 5.4: The Cogent Internet backbone map across Europe and North America [85]. Clusters from automated SGC are consistent with the geographic locations, whereas clusters from self-tuning spectral clustering are inconsistent with the geographic locations. Automated clusters found by AMOS, including city names, can be found in Fig. D.4.

clusters of the Hibernia Internet backbone map. AMOS outputs two clusters that perfectly separates the cities in North America and Europe, whereas one city in North America is clustered with the cities in Europe via self-tuning spectral clustering.

Fig. 5.4 shows the automated clusters of the Cogent Internet backbone map across North America and Europe. The clusters yielded by AMOS are consistent with the geographic locations except that North Eastern America and West Europe are identified as one cluster due to many transoceanic connections, wheres the clusters yielded by self-tuning spectral clustering are inconsistent with the geographic locations.

In Fig. 5.5, the clusters of the Minnesota road map via AMOS are shown to be aligned with the geographic separations, whereas some clusters identified via self-tuning clustering are inconsistent with the geographic separations and several clusters have small sizes[1]. In addition, when compared with the nonbacktracking matrix method [87, 139] and the Louvain method [18] (see Appendix D.3), the output clusters from the proposed AMOS algorithm are shown to be more consistent with the ground

[1] For the Minnesota road map we set $K_{\max} = 100$ for self-tuning spectral clustering to speed up the computation.



(a) Proposed AMOS algorithm. The number of clusters is 46.

(b) Self-tuning spectral clustering [176]. The number of clusters is 100.

Figure 5.5: Minnesota road map [64]. Clusters from automated SGC are aligned with the geographic separations, whereas some clusters from self-tuning spectral clustering are inconsistent with the geographic separations and self-tuning spectral clustering identifies several small clusters.

truth meta information.

5.2.1 External and internal clustering metrics

We use the following external and internal clustering metrics to evaluate the performance of different automated graph clustering methods. External metrics can be computed only when ground-truth cluster labels are known, whereas internal metrics can be computed in the absence of ground-truth cluster labels. In particular, we denote the K clusters identified by a multilayer graph clustering algorithm by $\{\mathcal{C}_k\}_{k=1}^{K}$, and denote the K' ground truth clusters by $\{\mathcal{C}'_k\}_{k=1}^{K'}$.

- **External clustering metrics**

1. normalized mutual information (NMI) [175]: NMI is defined as

$$\mathrm{NMI}(\{\mathcal{C}_k\}_{k=1}^{K}, \{\mathcal{C}'_k\}_{k=1}^{K'}) = \frac{2 \cdot I(\{\mathcal{C}_k\}, \{\mathcal{C}'_k\})}{|H(\{\mathcal{C}_k\}) + H(\{\mathcal{C}'_k\})|}, \tag{5.4}$$

where I is the mutual information between $\{\mathcal{C}_k\}_{k=1}^{K}$ and $\{\mathcal{C}'_k\}_{k=1}^{K'}$, and H is the entropy of clusters. Larger NMI means better clustering performance.

2. Rand index (RI) [175]: RI is defined as

$$\mathrm{RI}(\{\mathcal{C}_k\}_{k=1}^{K}, \{\mathcal{C}'_k\}_{k=1}^{K'}) = \frac{TP+TN}{TP+TN+FP+FN}, \tag{5.5}$$

where TP, TN, FP and FN represent true positive, true negative, false positive, and false negative decisions, respectively. Larger RI means better clustering performance.

3. F-measure [175]: F-measure is the harmonic mean of the precision and recall values for each cluster, which is defined as

$$\text{F-measure}(\{\mathcal{C}_k\}_{k=1}^{K}, \{\mathcal{C}'_k\}_{k=1}^{K'}) = \frac{1}{K}\sum_{k=1}^{K} \text{F-measure}_k, \tag{5.6}$$

where $\text{F-measure}_k = \frac{2 \cdot PREC_k \cdot RECALL_k}{PREC_k + RECALL_k}$, and $PREC_k$ and $RECALL_k$ are the precision and recall values for cluster $\mathcal{C}_k$. Larger F-measure means better clustering performance.

- **Internal clustering metrics**

1. conductance [144]: conductance is defined as

$$\mathrm{conductance}(\{\mathcal{C}_k\}_{k=1}^{K}) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{conductance}_k, \tag{5.7}$$

where $\mathrm{conductance}_k = \frac{W_k^{out}}{2 \cdot W_k^{in} + W_k^{out}}$, and W_k^{in} and W_k^{out} are the sum of within-cluster and between-cluster edge weights of cluster $\mathcal{C}_k$, respectively. Lower conductance means better clustering performance.

2. normalized cut (NC) [144]: NC is defined as

$$\mathrm{NC}(\{\mathcal{C}_k\}_{k=1}^{K}) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{NC}_k, \tag{5.8}$$

Dataset	Method	K	True K	NMI	RI	F	C	NC
RTS	AMOS	**3**	3	**0.8958**	**0.9642**	**0.9448**	0.0461	0.0682
	Louvain	6		0.7406	0.8387	0.6733	0.1439	0.1687
	NB	**3**		0.7535	0.8752	0.8121	0.0695	0.1000
	ST	2		0.7382	0.7808	0.7474	**0.0208**	**0.0407**
Hibernia	AMOS	**2**	2	**1.0000**	**1.0000**	**1.0000**	0.0296	0.0573
	Louvain	6		0.2713	0.5118	0.3256	0.2216	0.2630
	NB	**2**		0.7333	0.8949	0.9019	0.0273	0.0530
	ST	**2**		0.8787	0.9636	0.9667	**0.0283**	**0.0500**
Cogent	AMOS	4	2	**0.4242**	**0.6277**	0.5303	**0.0356**	**0.0487**
	Louvain	11		0.2451	0.5424	0.2584	0.1864	0.2044
	NB	3		0.2632	0.5444	**0.5765**	0.0732	0.1089
	ST	14		0.3435	0.5492	0.2868	0.1481	0.1640
Minnesota	AMOS	46	-	-	-	-	**0.0739**	**0.0756**
	Louvain	33					0.2899	0.2987
	NB	35					0.1399	0.1441
	ST	100					0.1189	0.1201

Table 5.2: Summary of the number of identified clusters (K) and the external and internal clustering metrics. "F" stands for F-measure and "C" stands for conductance. "NB" refers to the nonbacktracking matrix method, and "ST" refers to the self-tuning method. "-" means "not available" due to lack of ground-truth cluster labels. For each dataset, the method that leads the best clustering metric is highlighted in bold face. AMOS is shown to outperform most clustering methods for all datasets.

where $\text{NC}_k = \frac{W_k^{out}}{2\cdot W_k^{in}+W_k^{out}} + \frac{W_k^{out}}{2\cdot(W_k^{all}-W_k^{in})+W_k^{out}}$, and W_k^{in}, W_k^{out} and W_k^{all} are the sum of within-cluster, between-cluster and total edge weights of cluster $\mathcal{C}_k$, respectively. Lower conductance means better clustering performance.

Table 5.2 summarizes the external and internal clustering metrics of the four methods for the single-layer graph datasets listed in Table 5.1. It is observed from Table 6.2 that AMOS outperforms most clustering metrics for all datasets, which demonstrates its robustness and reliability.

CHAPTER VI

Multilayer Spectral Graph Clustering via Convex Layer Aggregation

Extending the phase transition analysis in single-layer graphs in Chapter IV and the AMOS algorithm in Chapter V, this chapter studies multilayer spectral graph clustering (SGC) via convex layer aggregation for multilayer graphs. Multilayer graphs are useful methods for representing and handling heterogeneous data, where each layer describes a specific type of connectivity pattern among a common node set across layers. For example, in multi-relational social networks, each layer corresponds to one social relation. In temporal networks, each layer corresponds to the snapshot of the entire network at a sampled time instance. Multilayer graphs have been applied to many signal processing and data mining techniques, including inference of mixture models [115, 171], tensor decomposition [43], information extraction [116], multi-view learning and processing [170], graph wavelet transform [95], principal component analysis and dictionary learning [15, 34], and community detection [81, 84], among others.

In particular, the task of multilayer graph clustering is to find a consensus cluster assignment on each node in the common node set by inspecting the connectivity pattern in each layer. Different from clustering in single-layer graphs, clustering in multilayer graph faces new challenges due to (1) information aggregation from multiple layers, and (2) lack of a theoretical framework on clustering reliability assessment. This chapter aims to tackle these challenges by proposing a multilayer SGC method via convex layer aggregation. Specifically, we propose to perform SGC on an aggregated graph via convex layer combination, where each layer is assigned with a nonnegative weight for aggregation. We first analyze the performance of multilayer SGC via convex layer aggregation under a novel multilayer signal plus noise model, where the signal and noise refer to within-cluster and between-cluster edge connections, respectively. Numerical experiments are conducted to verify its performance. We then propose MIMOSA, a multilayer iterative model order selection algorithm featuring automated layer weight adaptation and cluster count selection for multilayer SGC. Experimental results on real-world multilayer graphs show that MIMOSA has superior clustering performance over the baseline approach of assigning uniform layer weight, and the greedy multilayer modularity maximization method [101].

In summary, the contributions of this chapter are twofolds. First, under a general multilayer signal plus noise model, we establish a phase transition analysis on the performance of multilayer SGC via convex layer aggregation. Fixing the within-cluster edges (signals) and varying the parameters governing the between-cluster edges (noises), we show that the accuracy of multilayer SGC can be separated into two regimes: a reliable regime where high clustering accuracy can be guaranteed, and an unreliable regime where high clustering accuracy is impossible. Moreover, we specify the critical value that separates these two regimes, which is an analytical function of the signal strength, the number of clusters, the cluster size distributions, and the layer weight vector for convex layer aggregation. As a result, we establish a complete theoretic framework that specifies the interplay between the layer weights, the multilayer graph connectivity structure, and the performance of multilayer SGC via convex layer aggregation. This theoretic framework also provides a novel criterion for assessing the quality of clustering results.

Second, leveraging the established clustering reliability criterion under the multilayer signal plus noise model, we propose a multilayer iteration model order selection algorithm (MIMOSA) for multilayer SGC via convex layer aggregation. MIMOSA is a multilayer SGC algorithm that features automated model order selection for determining the number of clusters and the layer weights. It is an iterative SGC algorithm on the aggregated graph that incrementally increases the number of clusters, adapts layer weight assignment based on the noise level estimates from each layer, and adopts a series of statistical clustering reliability tests. As a result, MIMOSA finds the minimal number of clusters and the optimal layer weight assignment such that the identified clusters are estimated to be in the reliable regime as supported by the established theoretic framework. We apply MIMOSA to several real-world multilayer graphs and find that the clusters identified by MIMOSA indeed result in better clustering performance than the other two methods in terms of multiple external and internal clustering metrics.

6.1 Multilayer Graph Model and Spectral Graph Clustering via Convex Layer Aggregation

6.1.1 Multilayer graph model

Throughout this chapter, we consider the multilayer graph model of L layers representing different relationships among a common node set $\mathcal{V}$ of n nodes. The graph in the ℓ-th layer is an undirected graph with nonnegative edge wights, which is denoted by $G_\ell = (\mathcal{V}, \mathcal{E}_\ell)$, where $\mathcal{E}_\ell$ is the set of weighted edges in the ℓ-th layer. The $n \times n$ binary symmetric adjacency matrix $\mathbf{A}^{(\ell)}$ is used to represent the connectivity structure of G_ℓ. The entry $[\mathbf{A}^{(\ell)}]_{uv} = 1$ if nodes u and v are connected in the ℓ-th layer, and $[\mathbf{A}^{(\ell)}]_{uv} = 0$ otherwise. Similarly, the $n \times n$ nonnegative symmetric weight matrix $\mathbf{W}^{(\ell)}$ is used to represent the edge weights in G_ℓ, where $\mathbf{W}^{(\ell)}$ and $\mathbf{A}^{(\ell)}$ have the same zero structure.

We assume each layer in the multilayer graph is a (possibly correlated) representation of common K clusters that partitions the node set $\mathcal{V}$, where the k-th cluster has cluster size n_k such that $\sum_{k=1}^{K} n_k = n$. $n_{\min} = \min_{k \in \{1,\ldots,K\}} n_k$ and $n_{\max} = \max_{k \in \{1,\ldots,K\}} n_k$ denote the largest and smallest cluster size, respectively. Specifically, the adjacency matrix $\mathbf{A}^{(\ell)}$ of G_ℓ in the ℓ-th layer can be represented as

$$\mathbf{A}^{(\ell)} = \begin{bmatrix} \mathbf{A}_1^{(\ell)} & \mathbf{C}_{12}^{(\ell)} & \mathbf{C}_{13}^{(\ell)} & \cdots & \mathbf{C}_{1K}^{(\ell)} \\ \mathbf{C}_{21}^{(\ell)} & \mathbf{A}_2^{(\ell)} & \mathbf{C}_{23}^{(\ell)} & \cdots & \mathbf{C}_{2K}^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{K1}^{(\ell)} & \mathbf{C}_{K2}^{(\ell)} & \cdots & \cdots & \mathbf{A}_K^{(\ell)} \end{bmatrix}, \tag{6.1}$$

where $\mathbf{A}_k^{(\ell)}$ is an $n_k \times n_k$ binary symmetric matrix denoting the adjacency matrix of within-cluster edges of the k-th cluster in the ℓ-th layer, and $\mathbf{C}_{ij}^{(\ell)}$ is an $n_i \times n_j$ binary rectangular matrix denoting the adjacency matrix of between-cluster edges of clusters i and j in the ℓ-th layer, $1 \leq i, j \leq K$, $i \neq j$, and $\mathbf{C}_{ij}^{(\ell)} = \mathbf{C}_{ij}^{(\ell)T}$.

Similarly, the edge weight matrix $\mathbf{W}^{(\ell)}$ of the ℓ-th layer can be represented as

$$\mathbf{W}^{(\ell)} = \begin{bmatrix} \mathbf{W}_1^{(\ell)} & \mathbf{F}_{12}^{(\ell)} & \mathbf{F}_{13}^{(\ell)} & \cdots & \mathbf{F}_{1K}^{(\ell)} \\ \mathbf{F}_{21}^{(\ell)} & \mathbf{W}_2^{(\ell)} & \mathbf{F}_{23}^{(\ell)} & \cdots & \mathbf{F}_{2K}^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{K1}^{(\ell)} & \mathbf{F}_{K2}^{(\ell)} & \cdots & \cdots & \mathbf{W}_K^{(\ell)} \end{bmatrix}, \tag{6.2}$$

where $\mathbf{W}_k^{(\ell)}$ is an $n_k \times n_k$ nonnegative symmetric matrix denoting the edge weights of within-cluster edges of the k-th cluster in the ℓ-th layer, and $\mathbf{F}_{ij}^{(\ell)}$ is an $n_i \times n_j$

nonnegative rectangular matrix denoting the edge weights of between-cluster edges of clusters i and j in the ℓ-th layer, $1 \leq i, j \leq K$, $i \neq j$, and $\mathbf{F}_{ij}^{(\ell)} = \mathbf{F}_{ij}^{(\ell)^T}$.

6.1.2 Multilayer signal plus noise model

Using the cluster-wise block representations of the adjacency and edge weight matrices for the multilayer graph model described in (6.1) and (6.2), we propose a signal plus noise model for $\mathbf{A}^{(\ell)}$ and $\mathbf{W}^{(\ell)}$ to analyze the effect of convex layer aggregation on graph clustering. Specifically, for each layer we assume the connectivity structure and edge weight distributions follow the random interconnection model (RIM) in Sec. 4.1. In RIM the signal of the k-th cluster in the ℓ-th layer is the connectivity structure and weights of the within-cluster edges represented by the matrices $\mathbf{A}_k^{(\ell)}$ and $\mathbf{W}_k^{(\ell)}$, respectively. In particular, analogous to the formulation of many detection problems in signal processing, the signal can be arbitrary in the sense that we impose no distributional assumption for the within-cluster edges. The noise between clusters i and j in the ℓ-th layer is the connectivity structure and weights of the between-cluster edges represented by the matrices $\mathbf{C}_{ij}^{(\ell)}$ and $\mathbf{F}_{ij}^{(\ell)}$, respectively.

Throughout this chapter, we assume the connectivity of a between-cluster edge (i.e., the noise) in each layer is independently drawn from a layer-wise and block-wise independent common Bernoulli distribution. Specifically, each entry in $\mathbf{C}_{ij}^{(\ell)}$ representing the existence of an edge between clusters i and j in the ℓ-layer is an independent realization of a Bernoulli random variable with edge connection probability $p_{ij}^{(\ell)} \in [0, 1]$ that is layer-wise and block-wise independent. In addition, given the existence of an edge (u, v) between clusters i and j in the ℓ-layer, the entry $[\mathbf{F}_{ij}^{(\ell)}]_{uv}$ representing the corresponding edge weight is independently drawn from a nonnegative distribution with mean $\overline{W}_{ij}^{(\ell)}$ and bounded fourth moment that is layer-wise and block-wise independent.

For the ℓ-th layer, the noise accounting for the between-cluster edges is said to be *block-wise identical* if the noise parameters $p_{ij}^{(\ell)} = p^{(\ell)}$ and $\overline{W}_{ij}^{(\ell)} = \overline{W}^{(\ell)}$ for every cluster pair i and j, $i \neq j$. Otherwise it is said to be *block-wise non-identical.* The effect of these two noise models on multilayer spectral graph clustering will be studied in Sec. 6.2.

6.1.3 Multilayer spectral graph clustering via convex layer aggregation

Let $\mathbf{w} = [w_1, \ldots, w_L]^T \in \mathcal{W}_L$ be an $L \times 1$ column vector representing the layer weight vector for convex layer aggregation, where $\mathcal{W}_L = \{\mathbf{w} : w_\ell \geq 0,\ \sum_{\ell=1}^{L} w_\ell = 1\}$ is the set of feasible layer weight vectors. The single-layer graph obtained via convex layer aggregation with layer weight vector $\mathbf{w}$ is denoted by $G^{\mathbf{w}}$. The (weighted) adjacency matrix $\mathbf{A}^{\mathbf{w}}$ and the edge weight matrix $\mathbf{W}^{\mathbf{w}}$ of $G^{\mathbf{w}}$ have the relation $\mathbf{A}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \mathbf{A}^{(\ell)}$ and $\mathbf{W}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \mathbf{W}^{(\ell)}$. The graph Laplacian matrix $\mathbf{L}^{\mathbf{w}}$ of $G^{\mathbf{w}}$ is defined as $\mathbf{L}^{\mathbf{w}} = \mathbf{S}^{\mathbf{w}} - \mathbf{W}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \mathbf{L}^{(\ell)}$, where $\mathbf{S}^{\mathbf{w}} = \mathrm{diag}(\mathbf{s}^{\mathbf{w}})$ is a diagonal matrix, $\mathbf{s}^{\mathbf{w}} = \mathbf{W}^{\mathbf{w}} \mathbf{1}_n$ is the vector of nodal strength of $G^{\mathbf{w}}$, $\mathbf{1}_n$ is the $n \times 1$ column vector of ones, and $\mathbf{L}^{(\ell)}$ is the graph Laplacian matrix of G_ℓ. Similarly, the graph Laplacian matrix $\mathbf{L}_k^{\mathbf{w}}$ accounting for the within-cluster edges of the k-th cluster in $G^{\mathbf{w}}$ is defined as $\mathbf{L}_k^{\mathbf{w}} = \mathbf{S}_k^{\mathbf{w}} - \mathbf{W}_k^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \mathbf{L}_k^{(\ell)}$, where $\mathbf{W}_k^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \mathbf{W}_k^{(\ell)}$, $\mathbf{S}_k^{\mathbf{w}} = \mathrm{diag}(\mathbf{W}_k^{\mathbf{w}} \mathbf{1}_{n_k})$, and $\mathbf{L}_k^{(\ell)} = \mathbf{S}_k^{(\ell)} - \mathbf{W}_k^{(\ell)}$. The i-th smallest eigenvalue of $\mathbf{L}^{\mathbf{w}}$ is denoted by $\lambda_i(\mathbf{L}^{\mathbf{w}})$. Based on the definition of $\mathbf{L}^{\mathbf{w}}$, the smallest eigenvalue $\lambda_1(\mathbf{L}^{\mathbf{w}})$ of $\mathbf{L}^{\mathbf{w}}$ is 0, since $\mathbf{L}^{\mathbf{w}} \mathbf{1}_n = \mathbf{0}_n$, where $\mathbf{0}_n$ is the $n \times 1$ column vector of zeros.

Spectral graph clustering [97] partitions the nodes in $G^{\mathbf{w}}$ into K ($K \geq 2$) clusters based on the K eigenvectors associated with the K smallest eigenvalues of $\mathbf{L}^{\mathbf{w}}$. Specifically, spectral graph clustering first transforms a node in $G^{\mathbf{w}}$ to a K-dimensional vector in the subspace spanned by these eigenvectors, and then implements K-means clustering [72] on the K-dimensional vector space representation to group the nodes in $G^{\mathbf{w}}$ into K clusters based on their distances. For analysis purposes, throughout this chapter we assume $G^{\mathbf{w}}$ is a connected graph. In practice if $G^{\mathbf{w}}$ is a disconnected graph, spectral graph clustering can be applied to each connected component in $G^{\mathbf{w}}$. Moreover, if $G^{\mathbf{w}}$ is a connected graph, $\lambda_i(\mathbf{L}^{\mathbf{w}}) > 0$ for all $i \geq 2$. That is, the second to the n-th smallest eigenvalue of $\mathbf{L}^{\mathbf{w}}$ are all positive [55]. The eigenvector associated with the smallest eigenvalue $\lambda_1(\mathbf{L}^{\mathbf{w}})$ provides no information about graph clustering since it is proportional to a constant vector $\mathbf{1}_n$.

Written in a mathematical expression, let $\mathbf{Y} \in \mathbb{R}^{n \times (K-1)}$ denote the eigenvector matrix where its k-th column is the $(k+1)$-th eigenvector associated with $\lambda_{k+1}(\mathbf{L}^{\mathbf{w}})$, $1 \leq k \leq K-1$. By the Courant-Fischer theorem [78], $\mathbf{Y}$ is the solution of the minimization problem

$$\begin{aligned} & S_{2:K}(\mathbf{L}^{\mathbf{w}}) = \min_{\mathbf{X} \in \mathbb{R}^{n \times (K-1)}} \operatorname{trace}(\mathbf{X}^T \mathbf{L}^{\mathbf{w}} \mathbf{X}), \\ & \text{subjec to } \mathbf{X}^T \mathbf{X} = \mathbf{I}_{K-1},\ \mathbf{X}^T \mathbf{1}_n = \mathbf{0}_{K-1}, \end{aligned} \tag{6.3}$$

where the optimal value $S_{2:K}(\mathbf{L}^{\mathbf{w}}) = \operatorname{trace}(\mathbf{Y}^T \mathbf{L}^{\mathbf{w}} \mathbf{Y})$ in (6.3) is the partial eigenvalue sum $S_{2:K}(\mathbf{L}^{\mathbf{w}}) = \sum_{k=2}^{K} \lambda_k(\mathbf{L}^{\mathbf{w}})$, $\mathbf{I}_{K-1}$ is the $(K-1) \times (K-1)$ identity matrix, and the constraints in (6.3) impose orthonormality and centrality on the eigenvectors. In summary, multilayer spectral graph cluster via convex layer aggregation works by computing the eigenvector matrix $\mathbf{Y}$ from $\mathbf{L}^{\mathbf{w}}$ of $G^{\mathbf{w}}$, and implementing K-means clustering on the rows of $\mathbf{Y}$ to group the nodes into K clusters.

6.2 Performance Analysis of Multilayer Spectral Graph Clustering via Convex Layer Aggregation

In this section, we establish three theorems for performance analysis of multilayer spectral graph clustering (SGC) via convex layer aggregation. The analysis provides a theoretic framework for multilayer SGC and allows us to evaluate the quality of clustering results in terms of a novel signal-to-noise (SNR) ratio that falls out of the established theorems, which is then used for determining the number of clusters and selecting layer weights in the multilayer SGC algorithm proposed in Sec. 6.3.

The first theorem (Theorem 6.1) specifies the interplay between layer weights and the success of SGC by establishing a condition under which multilayer SGC via convex layer aggregation fails to correctly identify clusters under the multilayer signal plus noise model in Sec. 6.1.2.

The second theorem (Theorem 6.2) establishes phase transitions in the success of multilayer SGC under the block-wise identical noise model for a given layer weight vector $\mathbf{w}$. Under the block-wise identical noise model, define $t^{(\ell)} = p^{(\ell)} \cdot \overline{W}^{(\ell)}$ to be the noise level of the ℓ-th layer and let $t^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \cdot t^{(\ell)}$ be the aggregated noise level via convex layer aggregation. We show that for each $\mathbf{w}$ there exists a critical value $t^{\mathbf{w}*}$ of $t^{\mathbf{w}}$ such that if $t^{\mathbf{w}} < t^{\mathbf{w}*}$, multi-layer SGC can correctly identify the clusters, and if $t^{\mathbf{w}} > t^{\mathbf{w}*}$, multi-layer SGC is in vain.

The third theorem (Theorem 6.3) extends the phase transition analysis of the block-wise identical noise model to the block-wise non-identical noise model. Under the block-wise non-identical noise model, define $t_{\max}^{(\ell)} = \max_{i,j,i \neq j} p_{ij}^{(\ell)} \cdot \overline{W}_{ij}^{(\ell)}$ to be the maximum noise level of the ℓ-th layer and let $t_{\max}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \cdot t_{\max}^{(\ell)}$. Then for each $\mathbf{w}$ we show that good clustering results can be guaranteed provided that $t_{\max}^{\mathbf{w}} < t^{\mathbf{w}*}$, where $t^{\mathbf{w}*}$ is the critical value for phase transition under the block-wise identical noise model.

In the sequel, there are a number of limit theorems stated about the behavior of random matrices and vectors whose dimensions go to infinity as the sizes $\{n_k\}_{k=1}^{K}$ of the clusters go to infinity while their relative sizes $n_k/n_{k'}$ are held constant. For simplicity and convenience, the limit theorems are often stated in terms of the the finite, but arbitrarily large, dimensions n_k, $k = 1, 2, \ldots, K$.

6.2.1 Breakdown condition for multilayer SGC via convex layer aggregation

Under the multilayer signal plus noise model in Sec. 6.1.2, let $t_{ij}^{(\ell)} = \overline{W}_{ij}^{(\ell)} \cdot p_{ij}^{(\ell)}$ be the noise level between clusters i and j in the ℓ-layer, $1 \leq i, j \leq K$, $i \neq j$, and $1 \leq \ell \leq L$. The following theorem establishes a general breakdown condition under which multilayer SGC fails to correctly identify the clusters.

Theorem 6.1 (general breakdown condition)**.**
Let $\widetilde{\mathbf{W}}^{\mathbf{w}}$ be the $(K-1) \times (K-1)$ matrix with (i,j)-th element

$$[\widetilde{\mathbf{W}}^{\mathbf{w}}]_{ij} = \begin{cases} \sum_{\ell=1}^{L} w_\ell \left[(n_i + n_K)\, t_{iK}^{(\ell)} + \sum_{z=1, z\neq i}^{K-1} n_z t_{iz}^{(\ell)} \right], & \text{if } i = j; \\ \sum_{\ell=1}^{L} w_\ell n_i \cdot \left(t_{iK}^{(\ell)} - t_{ij}^{(\ell)} \right), & \text{if } i \neq j. \end{cases}$$

The following holds almost surely as $n_k \to \infty\ \forall\ k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. If for any layer weight vector $\mathbf{w} \in \mathcal{W}_L$, $\lambda_i \left(\frac{\widetilde{\mathbf{W}}^{\mathbf{w}}}{n} \right) \neq \lambda_j \left(\frac{\mathbf{L}^{\mathbf{w}}}{n} \right)$ for all $i = 1, 2, \ldots, K-1$ and $j = 2, 3, \ldots, K$, then multilayer SGC cannot be successful.

Proof. The proof is given in Appendix E.1. □

Theorem 6.1 specifies the interplay between the layer weight vector $\mathbf{w}$ and the accuracy of multilayer SGC. Different from the case of single-layer graphs (i.e., $L = 1$ and hence $\mathbf{w} = 1$) such that the layer weight has no effect on the performance of SGC, Theorem 6.1 states that multilayer SGC cannot be successful if every possible layer weight vector $\mathbf{w} \in \mathcal{W}_L$ leads to distinct $K-1$ smallest nonzero eigenvalues of the matrices $\frac{\widetilde{\mathbf{W}}^{\mathbf{w}}}{n}$ and $\frac{\mathbf{L}^{\mathbf{w}}}{n}$. It also suggests that the selection of layer weight vector does affect the performance of multilayer SGC.

6.2.2 Phase transitions in multilayer SGC under block-wise identical noise

Under the multilayer signal plus noise model in Sec. 6.1.2, if we further assume the between-cluster edges in each layer follow a block-wise identical distribution, then the noise level in the ℓ-th layer can be characterized by the parameter $t^{(\ell)} = p^{(\ell)} \cdot \overline{W}^{(\ell)}$, where $p^{(\ell)} \in [0,1]$ is the edge connection parameter and $\overline{W}^{(\ell)} > 0$ is the mean of the between-cluster edge weights in the ℓ-th layer. Under the block-wise identical noise model and given a layer weight vector $\mathbf{w} \in \mathcal{W}_L$, let $t^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell t^{(\ell)}$ denote the aggregated noise level of the graph $G^{\mathbf{w}}$. Theorem 6.2 below establishes phase transitions in the eigendecomposition of the graph Laplacian matrix $\mathbf{L}^{\mathbf{w}}$ of the graph $G^{\mathbf{w}}$. We show that there exists a critical value $t^{\mathbf{w}*}$ such that the K smallest eigenpairs of $\mathbf{L}^{\mathbf{w}}$ that are used for multilayer SGC have different characteristics when $t^{\mathbf{w}} < t^{\mathbf{w}*}$ and $t^{\mathbf{w}} > t^{\mathbf{w}*}$. In particular, we show that the solution to the minimization problem in (6.3), the eigenvector matrix $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T \in \mathbb{R}^{n \times (K-1)}$ represented by the cluster partitioned form, where $\mathbf{Y}_k \in \mathbb{R}^{n_k \times (K-1)}$ with its rows indexing the nodes in cluster k, has cluster-wise separability when $t^{\mathbf{w}} < t^{\mathbf{w}*}$ in the sense that the matrices $\{\mathbf{Y}_k\}_{k=1}^{K}$ are row-wise identical and cluster-wise distinct, whereas when $t^{\mathbf{w}} > t^{\mathbf{w}*}$ the row-wise average of each matrix $\mathbf{Y}_k$ is a zero vector and hence the clusters are not separable by inspecting the eigenvector matrix $\mathbf{Y}$.

Theorem 6.2 (block-wise identical noise)**.**

Let $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T$ *be the solution of the minimization problem in (6.3) and let* $c^{\mathbf{w}*} = \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{n} \right\}$*, where* $\mathbf{L}_k^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \mathbf{L}_k^{(\ell)}$*. Given a layer weight vector* $\mathbf{w} \in \mathcal{W}_L$*, under the block-wise identical noise model with aggregated noise level* $t^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell t^{(\ell)} = \sum_{\ell=1}^{L} w_\ell p^{(\ell)} \overline{W}^{(\ell)}$*, there exists a critical value* $t^{\mathbf{w}*}$ *such that the following holds almost surely as* $n_k \to \infty \ \forall\ k$ *and* $\frac{n_{\min}}{n_{\max}} \to c > 0$*:*

$$\text{(a)} \begin{cases} \text{If } t^{\mathbf{w}} \leq t^{\mathbf{w}*}, \ \frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n} = (K-1)t^{\mathbf{w}}; \\ \text{If } t^{\mathbf{w}} > t^{\mathbf{w}*}, \ c^{\mathbf{w}*} + (K-1)\left(1 - \frac{n_{\max}}{n}\right) t^{\mathbf{w}} \leq \frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n} \leq c^{\mathbf{w}*} + (K-1)\left(1 - \frac{n_{\min}}{n}\right) t^{\mathbf{w}}. \end{cases}$$

In particular, if $t^{\mathbf{w}} > t^{\mathbf{w}*}$ *and* $c = 1$, $\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n} = c^{\mathbf{w}*} + \frac{(K-1)^2}{K} t^{\mathbf{w}}$.

Furthermore,

$$
\text{(b)} \begin{cases} \text{If } t^{\mathbf{w}} < t^{\mathbf{w}*},\ \mathbf{Y}_k = \mathbf{1}_{n_k}\mathbf{1}_{K-1}^T\mathbf{V}_k = \left[v_1^k\mathbf{1}_{n_k}, v_2^k\mathbf{1}_{n_k}, \ldots, v_{K-1}^k\mathbf{1}_{n_k}\right],\ \forall\ k \in \{1,2,\ldots,K\}; \\ \text{If } t^{\mathbf{w}} > t^{\mathbf{w}*},\ \mathbf{Y}_k^T\mathbf{1}_{n_k} = \mathbf{0}_{K-1},\ \forall\ k \in \{1,2,\ldots,K\}; \\ \text{If } t^{\mathbf{w}} = t^{\mathbf{w}*},\ \forall\ k \in \{1,2,\ldots,K\},\ \mathbf{Y}_k = \mathbf{1}_{n_k}\mathbf{1}_{K-1}^T\mathbf{V}_k \text{ or } \mathbf{Y}_k^T\mathbf{1}_{n_k} = \mathbf{0}_{K-1}, \end{cases}
$$

where $\mathbf{V}_k = \operatorname{diag}(v_1^k, v_2^k, \ldots, v_{K-1}^k) \in \mathbb{R}^{(K-1)\times(K-1)}$ *is a diagonal matrix.*

In particular, when $t^{\mathbf{w}} < t^{\mathbf{w}*}$, $\mathbf{Y}$ *has the following properties:*

(b-1) *The columns of* $\mathbf{Y}_k$ *are constant vectors.*

(b-2) *Each column of* $\mathbf{Y}$ *has at least two nonzero cluster-wise constant components, and these constants have alternating signs such that their weighted sum equals* 0 *(i.e.,* $\sum_k n_k v_j^k = 0,\ \forall\ j \in \{1,2,\ldots,K-1\}$*).*

(b-3) *No two columns of* $\mathbf{Y}$ *have the same sign on the cluster-wise nonzero components. Finally,* $t^{\mathbf{w}*}$ *satisfies:*

(c) $t_{LB}^{\mathbf{w}} \leq t^{\mathbf{w}*} \leq t_{UB}^{\mathbf{w}}$, *where*

$$
\begin{cases} t_{LB}^{\mathbf{w}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{(K-1)n_{\max}}; \\ t_{UB}^{\mathbf{w}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{(K-1)n_{\min}}. \end{cases}
$$

In particular, $t_{LB}^{\mathbf{w}} = t_{UB}^{\mathbf{w}}$ *when* $c = 1$.

Proof. The proof is given in Appendix E.2. □

Theorem 6.2 (a) establishes a phase transition in the increase of the normalized partial eigenvalue sum $\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n}$ with respect to the aggregated noise level $t^{\mathbf{w}}$. When $t^{\mathbf{w}} \leq t^{\mathbf{w}*}$ the quantity $\frac{S_{2:K}(\mathbf{L})}{n}$ is exactly $(K-1)t^{\mathbf{w}}$. When $t^{\mathbf{w}} > t^{\mathbf{w}*}$ the slope in $t^{\mathbf{w}}$ of $\frac{S_{2:K}(\mathbf{L})}{n}$ changes and the intercept $c^* = \min_{k\in\{1,2,\ldots,K\}}\left\{\frac{S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{n}\right\} = \min_{k\in\{1,2,\ldots,K\}}\left\{\frac{\sum_{\ell=1}^{L} w_\ell S_{2:K}(\mathbf{L}_k^{(\ell)})}{n}\right\}$ depends on the cluster having the smallest aggregated partial eigenvalue sum given a layer weight vector $\mathbf{w}$. In particular, when all clusters have the same size (i.e., $n_{\max} = n_{\min} = \frac{n}{K}$) so that $c = 1$, $\frac{S_{2:K}(\mathbf{L})}{n}$ undergoes a slope change from $K-1$ to $\frac{(K-1)^2}{K}$ at the critical value $t^{\mathbf{w}} = t^{\mathbf{w}*}$.

Theorem 6.2 (b) establishes a phase transition in cluster-wise separability of the eigenvector matrix $\mathbf{Y}$ for multilayer SGC. When $t^{\mathbf{w}} < t^{\mathbf{w}*}$, the conditions (b-1) to (b-3) imply that the rows of the cluster-wise components $\{\mathbf{Y}_k\}_{k=1}^{K}$ are coherent, and hence the row vectors in $\mathbf{Y}$ possess cluster-wise separability. On the other hand, when $t^{\mathbf{w}} > t^{\mathbf{w}*}$, the row sum of each $\mathbf{Y}_k$ is a zero vector, making $\mathbf{Y}_k$ incoherent. This means that the entries of each column in $\mathbf{Y}_k$ have alternating signs and hence K-means clustering on the rows of $\mathbf{Y}$ yields incorrect clusters.

Theorem 6.2 (c) establishes upper and lower bounds on the critical threshold value $t^{\mathbf{w}*}$ of the aggregated noise level $t^{\mathbf{w}}$ given a layer weight vector $\mathbf{w}$. These bounds are determined by the cluster having the smallest aggregated partial eigenvalue sum $S_{2:K}(\mathbf{L}_k^{\mathbf{w}}) = \sum_{\ell=1}^{L} w_\ell S_{2:K}(\mathbf{L}_k^{(\ell)})$, the number of clusters K, and the largest and smallest cluster size ($n_{\max}$ and $n_{\min}$). When all cluster sizes are identical (i.e., $c = 1$), these bounds become tight (i.e., $t_{\mathrm{LB}}^{\mathbf{w}} = t_{\mathrm{UB}}^{\mathbf{w}}$). Moreover, by the nonnegativity of the layer weights we can obtain a universal lower bound on $t_{\mathrm{LB}}^{\mathbf{w}}$ for any $\mathbf{w} \in \mathcal{W}_L$, which is

$$\begin{aligned} t_{\mathrm{LB}}^{\mathbf{w}} &= \frac{\min_{k\in\{1,2,\dots,K\}} S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{(K-1)n_{\max}} \\ &\geq \frac{\min_{k\in\{1,2,\dots,K\}} \min_{\ell\in\{1,2,\dots,L\}} S_{2:K}(\mathbf{L}_k^{(\ell)})}{(K-1)n_{\max}}. \end{aligned} \tag{6.4}$$

Since $S_{2:K}(\mathbf{L}_k^{(\ell)})$ is a measure of connectivity for cluster k in the ℓ-th layer, the lower bound of $t_{\mathrm{LB}}^{\mathbf{w}}$ in (6.4) implies that the performance of multilayer SGC is indeed affected by the least connected cluster among all K clusters and across L layers. Specifically, if the graph in each layer is unweighted and $K = 2$, then $S_{2:K}(\mathbf{L}_k^{(\ell)}) = \lambda_2(\mathbf{L}_k^{(\ell)})$ reduces to the algebraic connectivity of cluster k in the ℓ-th layer. Similarly, we can obtain a universal upper bound on $t_{\mathrm{UB}}^{\mathbf{w}}$ for any $\mathbf{w} \in \mathcal{W}_L$, which is

$$t_{\mathrm{UB}}^{\mathbf{w}} \leq \frac{\min_{k\in\{1,2,\dots,K\}} \max_{\ell\in\{1,2,\dots,L\}} S_{2:K}(\mathbf{L}_k^{(\ell)})}{(K-1)n_{\min}}. \tag{6.5}$$

6.2.3 Phase transitions in multilayer SGC under block-wise non-identical noise

Under the block-wise non-identical noise model, the noise level of between-cluster edges between clusters i and j in the ℓ-th layer is characterized by the parameter $t_{ij}^{(\ell)} = p_{ij}^{(\ell)} \cdot \overline{W}_{ij}^{(\ell)}$, $1 \leq i,j \leq K$, $i \neq j$, and $1 \leq \ell \leq L$. Let $t_{\max}^{(\ell)} = \max_{1 \leq i,j \leq K,\, i \neq j} t_{ij}^{(\ell)}$ be the maximum noise level in the ℓ-th layer and let $t_{\max}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell t_{\max}^{(\ell)}$ denote the aggregated maximum noise level given a layer weight vector $\mathbf{w} \in \mathcal{W}_L$.

Let $\mathbf{Y} \in \mathbb{R}^{n \times (K-1)}$ be the eigenvector matrix of $\mathbf{L}^{\mathbf{w}}$ under the block-wise non-identical noise model, and let $\widetilde{\mathbf{Y}} \in \mathbb{R}^{n \times (K-1)}$ be the eigenvector matrix of the graph Laplacian $\widetilde{\mathbf{L}}^{\mathbf{w}}$ of another random graph generated under the block-wise identical noise model with aggregated noise level $t^{\mathbf{w}}$, which is independent of $\mathbf{L}$. Theorem 6.3 below specifies the distance between the subspaces spanned by the columns of $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ by inspecting their principal angles [97]. Specifically, since $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ both have orthonormal columns, the vector $\mathbf{v}$ of $K-1$ principal angles between their column spaces is $\mathbf{v} = [\cos^{-1} \sigma_1(\mathbf{Y}^T \widetilde{\mathbf{Y}}), \ldots, \cos^{-1} \sigma_{K-1}(\mathbf{Y}^T \widetilde{\mathbf{Y}})]^T$, where $\sigma_k(\mathbf{M})$ is the k-th largest singular value of a real rectangular matrix $\mathbf{M}$. Let $\mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}) = \operatorname{diag}(\mathbf{v})$, and let $\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})$ be defined entrywise. When $t^{\mathbf{w}} < t^{\mathbf{w}*}$, Theorem 6.3 provides an upper bound on the Frobenius norm of $\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})$, which is denoted by $\| \sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}) \|_F$. Moreover, if $t_{\max}^{\mathbf{w}} < t^{\mathbf{w}*}$, where $t^{\mathbf{w}*}$ is the critical threshold value for the block-wise identical noise model as specified in Theorem 6.2, then $\| \sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}) \|_F$ can be further bounded.

Theorem 6.3 (block-wise non-identical noise)**.**

Under the multilayer signal plus noise model in Sec. 6.1.2 with maximum noise level $\{t_{\max}^{(\ell)}\}_{\ell=1}^{L}$ *for each layer, given a layer weight vector* $\mathbf{w} \in \mathcal{W}_L$*, let* $t^{\mathbf{w}*}$ *be be the critical threshold value for the block-wise identical noise model specified by Theorem 6.2, and define* $\delta_{t^{\mathbf{w}},n} = \min\{t^{\mathbf{w}}, |\lambda_{K+1}(\frac{\mathbf{L}^{\mathbf{w}}}{n}) - t^{\mathbf{w}}|\}$*. For a fixed* $t^{\mathbf{w}}$*, if* $t^{\mathbf{w}} < t^{\mathbf{w}*}$ *and* $\delta_{t^{\mathbf{w}},n} \to \delta_{t^{\mathbf{w}}} > 0$ *as* $n_k \to \infty$ $\forall$ k*, the following statement holds almost surely as*

$n_k \to \infty\ \forall\ k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$:

$$\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F \leq \frac{\|\mathbf{L}^{\mathbf{w}} - \widetilde{\mathbf{L}}^{\mathbf{w}}\|_F}{n\delta_{t^{\mathbf{w}}}}. \tag{6.6}$$

Furthermore, let $t^{\mathbf{w}}_{\max} = \sum_{\ell=1}^{L} w_\ell t^{(\ell)}_{\max}$. If $t^{\mathbf{w}}_{\max} < t^{\mathbf{w}*}$,

$$\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F \leq \min_{t^{\mathbf{w}} \leq t^{\mathbf{w}}_{\max}} \frac{\|\mathbf{L}^{\mathbf{w}} - \widetilde{\mathbf{L}}^{\mathbf{w}}\|_F}{n\delta_{t^{\mathbf{w}}}}. \tag{6.7}$$

Proof. The proof is given in Appendix E.3. □

Theorem 6.3 shows that the subspace distance $\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F$ is upper bounded by (6.6), where $\widetilde{\mathbf{Y}}$ is the eigenvector matrix of $\widetilde{\mathbf{L}}^{\mathbf{w}}$ under the block-wise identical noise model when its aggregated noise level $t^{\mathbf{w}} < t^{\mathbf{w}*}$. Furthermore, if the aggregated maximum noise level $t^{\mathbf{w}}_{\max} < t^{\mathbf{w}*}$, then a tight upper bound on $\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F$ can be obtained by (6.7). Therefore, using the phase transition results of the cluster-wise separability in $\widetilde{\mathbf{Y}}$ as established in Theorem 6.2 (b), when $t^{\mathbf{w}}_{\max} < t^{\mathbf{w}*}$, cluster-wise separability in $\mathbf{Y}$ can be expected provided that $\|\sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}})\|_F$ is small.

6.3 MIMOSA: Multilayer Iterative Model Order Selection Algorithm

The phase transition analysis established in Sec. 6.2 shows that under the multilayer signal plus noise model in Sec. 6.1.2, the performance of multilayer spectral graph clustering (SGC) via convex layer aggregation can be separated into two regimes: a reliable regime where high clustering accuracy is guaranteed, and an unreliable regime where high clustering accuracy is impossible. We have specified the critical threshold value of the aggregated noise level that separates these two regimes, and have shown that the assigned layer weight vector $\mathbf{w}$ for convex layer aggregation indeed affects the accuracy of multilayer SGC.

In this section, we use the established phase transition criterion to propose a multilayer SGC algorithm, for which we call multilayer iterative model order selection algorithm (MIMOSA). MIMOSA is a multilayer SGC algorithm that features automated model order selection for determining the number of clusters (K) and the layer weight vector $\mathbf{w}$. It works by incrementally partitioning the aggregated graph $G^{\mathbf{w}}$ into K clusters, adjusting the layer weight vector, and finding the minimal number of clusters such that the output clusters are estimated to be in the reliable regime. The flow diagram of MIMOSA is displayed in Fig. 6.1, and the complete algorithm is summarized in Algorithm 6.2. The details of MIMOSA are discussed as follows.

6.3.1 Input data

The input data for MIMOSA is summarized as follows. (1) a multilayer graph $\{G_\ell\}_{\ell=1}^{L}$ of L layers, where each layer G_ℓ is an undirected weighted graph. (2) an initial layer weight vector $\mathbf{w}^{\text{ini}} \in \mathcal{W}_L$. $\mathbf{w}^{\text{ini}}$ can be specified according to domain knowledge, or it can be a uniform vector such that $w_\ell = \frac{1}{L}\ \forall\ \ell$. (3) a layer weight adaptation coefficient set $\mathcal{T} = \{\tau_z\}_{z=1}^{|\mathcal{T}|}$. The coefficients in $\mathcal{T}$ play a role in the process of layer weight adaptation in Sec. 6.3.2. (4) a p-value significance level η that is used for the block-wise homogeneity test in Sec. 6.3.3. (5) confidence interval parameters $\{\alpha_\ell\}_{\ell=1}^{L}$ of each layer under the block-wise identical noise model for clustering reliability evaluation in Sec. 6.3.4. (6) confidence interval parameters $\{\alpha'_\ell\}_{\ell=1}^{L}$ of each layer under the block-wise non-identical noise model for clustering reliability evaluation in Sec. 6.3.5.

6.3.2 Layer weight adaptation

Given an initial layer weight vector $\mathbf{w}^{\text{ini}}$ and the number of clusters K in the iterative process (step 4) of MIMOSA, we propose to adjust the layer weight vector $\mathbf{w}$ for convex layer aggregation by estimating the noise level $\{\widehat{t}_{\text{ini}}^{(\ell)}\}_{\ell=1}^{L}$ under the block-

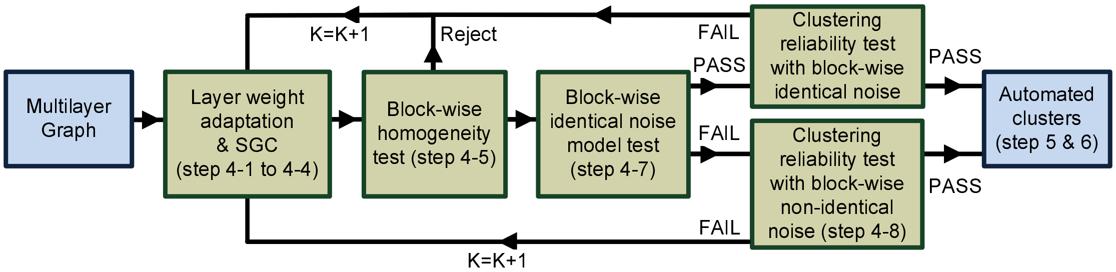

Multilayer Graph
Layer weight adaptation & SGC (step 4-1 to 4-4)
Block-wise homogeneity test (step 4-5)
Block-wise identical noise model test (step 4-7)
Clustering reliability test with block-wise identical noise
Clustering reliability test with block-wise non-identical noise (step 4-8)
Automated clusters (step 5 & 6)
K=K+1
Reject
FAIL
PASS
FAIL
PASS
PASS
FAIL
K=K+1

Figure 6.1: Flow diagram of the proposed multilayer iterative model order selection algorithm (MIMOSA) for multilayer spectral graph clustering (SGC).

wise identical noise model in Sec. 6.1.2. Specifically, given K clusters $\{\mathcal{C}_k^{\mathbf{w}^{\text{ini}}}\}_{k=1}^K$ of size $\{\widehat{n}_k\}_{k=1}^K$ via multilayer SGC with $\mathbf{w}^{\text{ini}}$, let $\{\widehat{\mathbf{C}}_{ij}^{(\ell)}\}$ and $\{\widehat{\mathbf{F}}_{ij}^{(\ell)}\}$ be the interconnection matrix and edge weight matrix of $\{\mathcal{C}_k^{\mathbf{w}^{\text{ini}}}\}_{k=1}^K$, respectively, for $1 \leq i,j \leq K$, $i \neq j$, and $1 \leq \ell \leq L$. Then the noise level estimator under the block-wise identical noise model is

$$\widehat{t}_{\text{ini}}^{(\ell)} = \widehat{p}^{(\ell)} \cdot \widehat{\overline{W}}^{(\ell)}, \tag{6.8}$$

for $\ell \in \{1, 2, \ldots, L\}$, where $\widehat{p}^{(\ell)} = \frac{\sum_{i=1}^K \sum_{j=i+1}^K \widehat{m}_{ij}^{(\ell)}}{\sum_{i=1}^K \sum_{j=i+1}^K \widehat{n}_i \widehat{n}_j}$ is the maximum likelihood estimator (MLE) of $p^{(\ell)}$, $\widehat{m}_{ij}^{(\ell)} = \mathbf{1}_{\widehat{n}_i}^T \widehat{\mathbf{C}}_{ij}^{(\ell)} \mathbf{1}_{\widehat{n}_j}$ is the number of between-cluster edges of clusters i and j in the ℓ-th layer, and $\widehat{\overline{W}}^{(\ell)}$ is the average of between-cluster edge weights in the ℓ-th layer.

Since the estimates $\{\widehat{t}_{\text{ini}}^{(\ell)}\}_{\ell=1}^L$ reflect the noise level in each layer, we propose to adjust the layer weight vector $\mathbf{w} \in \mathcal{W}_L$ with a nonnegative regularization parameter $\tau \in \mathcal{T}$. The adjusted $\mathbf{w}$ layer weight vector is inversely proportional to the estimated noise level, which is defined as

$$w_\ell \propto \frac{w_\ell^{\text{ini}}}{1 + \tau \cdot \widehat{t}_{\text{ini}}^{(\ell)}}, \tag{6.9}$$

for $\ell \in \{1, 2, \ldots, L\}$. Note that if $\tau = 0$, then $\mathbf{w}$ reduces to $\mathbf{w}^{\text{ini}}$. In addition, larger τ further penalizes the layers of high noise level by assigning less weight for convex layer aggregation.

6.3.3 Block-wise homogeneity test

Given K clusters $\{\mathcal{C}_k^{\mathbf{w}}\}_{k=1}^K$ with respect to a layer weight vector $\mathbf{w}$ in the iterative process (step 4) of MIMOSA, we implement a block-wise homogeneity test for each block $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ accounting for the interconnection matrix of clusters i and j in the ℓ-th layer, in order to test the assumption of the block-wise homogeneity noise model as assumed in Sec. 6.1.2, which is the cornerstone of the phase transition results established in Sec. 6.2.

More specifically, we are testing the null hypothesis that $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ *is a realization of a random matrix with i.i.d. Bernoulli entries* and the alternative hypothesis that $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ *is a realization of a random matrix with independent Bernoulli entries*, for every (i, j, ℓ) such that $1 \leq i, j \leq K$, $i \neq j$, and $1 \leq \ell \leq L$. We use the V-test [129] for homogeneity testing of the row sums or column sums of $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ as described in Algorithm 6.1. Given a set of independent binomial random variables, the V-test tests that they are all identically distributed. For concreteness, here we apply the V-test to the row sums of each $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ independently.

For each $\widehat{\mathbf{C}}_{ij}^{(\ell)}$, the test statistic Z of the V-test converges to a standard normal distribution as $n_i, n_j \to \infty$, and the p-value for the hypothesis that the row sums of $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ are i.i.d. is p-value$(i, j, \ell) = 2 \cdot \min\{\Phi(Z), 1 - \Phi(Z)\}$, where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard normal distribution. The block-wise homogeneity test on $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ rejects the null hypothesis if p-value$(i, j, \ell) \leq \eta$, where η is the desired single comparison significance level. In step 4-5 of MIMOSA, the layer weight vector $\mathbf{w}$ and the corresponding clusters $\{\mathcal{C}_k^{\mathbf{w}}\}_{k=1}^K$ are deemed unreliable if there exists some $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ such its p-value does not exceed the significance level.

6.3.4 Clustering reliability test under the block-wise identical noise model

In the iterative process of step 4 in MIMOSA, if every interconnection matrix $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ passes the block-wise homogeneity test in Sec. 6.3.3, the identified clusters $\{\mathcal{C}_k^{\mathbf{w}}\}_{k=1}^K$

Algorithm 6.1 p-value computation from V-test for block-wise homogeneity test

```
Input: An $n_i \times n_j$ interconnection matrix $\widehat{\mathbf{C}}_{ij}^{(\ell)}$
Output: p-value$(i, j, \ell)$
$\mathbf{x} = \widehat{\mathbf{C}}_{ij}^{(\ell)} \mathbf{1}_{n_j}$ (# of nonzero entries of each row in $\widehat{\mathbf{C}}_{ij}^{(\ell)}$)
$\mathbf{y} = n_j \mathbf{1}_{n_i} - \mathbf{x}$ (# of zero entries of each row in $\widehat{\mathbf{C}}_{ij}^{(\ell)}$)
$X = \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{1}_{n_i}$ and $Y = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{1}_{n_i}$.
$N = n_i n_j (n_j - 1)$ and $V = \left(\sqrt{X} + \sqrt{Y}\right)^2$.
Compute test statistic $Z = \frac{V-N}{\sqrt{2N}}$
Compute p-value$(i, j, \ell)$= $2 \cdot \min\{\Phi(Z), 1 - \Phi(Z)\}$
```

are then used to test the clustering reliability under the block-wise identical noise model in Sec. 6.1.2. In particular, for each layer ℓ, we first estimate the noise level parameter $\widehat{p}_{ij}^{(\ell)}$ for each every cluster pair i and j as $\widehat{p}_{ij}^{(\ell)} = \frac{\widehat{m}_{ij}^{(\ell)}}{\widehat{n}_i \widehat{n}_j}$, where $\widehat{p}_{ij}^{(\ell)}$ is an MLE of $p_{ij}^{(\ell)}$. We then use a generalized log-likelihood ratio test (GLRT) developed in Appendix D.1 based on the Wilk's theorem [167] to specify an asymptotic $100(1-\alpha_\ell)\%$ confidence interval for $p^{(\ell)}$ accounting for the block-wise identical noise level parameter for each layer, which is

$$\Bigg\{ p^{(\ell)} : \xi_{\binom{K}{2}-1, 1-\frac{\alpha_\ell}{2}} \leq 2 \sum_{i=1}^{K} \sum_{j=i+1}^{K} \mathbb{I}_{\{\widehat{p}_{ij}^{(\ell)} \in (0,1)\}} \left[\widehat{m}_{ij}^{(\ell)} \ln \widehat{p}_{ij}^{(\ell)} + (\widehat{n}_i \widehat{n}_j - \widehat{m}_{ij}^{(\ell)}) \ln(1 - \widehat{p}_{ij}^{(\ell)}) \right]$$
$$-2 \left(m^{(\ell)} - \sum_{k=1}^{K} \widehat{m}_k^{(\ell)} \right) \ln p^{(\ell)} - \left[n^2 - \sum_{k=1}^{K} \widehat{n}_k^2 - 2 \left(m^{(\ell)} - \sum_{k=1}^{K} \widehat{m}_k^{(\ell)} \right) \right] \ln(1 - p^{(\ell)})$$
$$\leq \xi_{\binom{K}{2}-1, \frac{\alpha_\ell}{2}} \Bigg\}, \tag{6.10}$$

where $\xi_{q,\alpha}$ is the upper α-th quantile of the central chi-square distribution with degree of freedom q, $\mathbb{I}_E$ is the indicator function of the event E, $m^{(\ell)}$ is the total number of edges in the ℓ-th layer, and $\widehat{m}_k^{(\ell)}$ is the number of within-cluster edges of cluster k in the ℓ-th layer.

If the estimated block-wise identical noise level parameter $\widehat{p}^{(\ell)} = \frac{\sum_{i=1}^{K} \sum_{j=i+1}^{K} \widehat{m}_{ij}^{(\ell)}}{\sum_{i=1}^{K} \sum_{j=i+1}^{K} \widehat{n}_i \widehat{n}_j}$ is within the confidence interval in (6.10) for every ℓ, then the clusters $\{\mathcal{C}_k^{\mathbf{w}}\}$ satisfy

Algorithm 6.2 Multilayer iterative model order selection algorithm (MIMOSA)

```
Input:
(1) a multilayer graph $\{G_\ell\}_{\ell=1}^{L}$
(2) an initial layer weight vector $\mathbf{w}^{\mathrm{ini}} \in \mathcal{W}_L$
(3) a layer weight adaptation coefficient set $\mathcal{T} = \{\tau_z\}_{z=1}^{|\mathcal{T}|}$
(4) a p-value significance level $\eta$
(5) confidence interval parameters $\{\alpha_\ell\}_{\ell=1}^{L}$ under the block-wise identical noise model for each layer
(6) confidence interval parameters $\{\alpha'_\ell\}_{\ell=1}^{L}$ under the block-wise non-identical noise model for each layer
Output: $K$ clusters $\{\mathcal{C}_k\}_{k=1}^{K}$
Initialization: $K = 2$. Flag $= 1$. $\mathcal{W}_{\mathrm{reliable}} = \varnothing$.
while Flag$= 1$ do
    1. Compute $\mathbf{Y} \in \mathbb{R}^{n \times (K-1)}$ of $\mathbf{L}^{\mathbf{w}^{\mathrm{ini}}}$
    2. Obtain $K$ clusters $\{\mathcal{C}_k^{\mathbf{w}^{\mathrm{ini}}}\}_{k=1}^{K}$ by K-means algorithm on the rows of $\mathbf{Y}$
    3. Estimate the noise level $\{\widehat{t}_{\mathrm{ini}}^{(\ell)}\}_{\ell=1}^{L}$ from (6.8)
    4. Layer weight adaptation and multilayer SGC reliability tests:
    for $z = 1$ to $|\mathcal{T}|$ do
        4-1. Layer weight adaptation: $w_\ell \leftarrow w_\ell^{\mathrm{ini}} \cdot (1 + \tau_z \cdot \widehat{t}^{(\ell)})^{-1}$, $\forall\, \ell \in \{1, 2, \ldots, L\}$
        4-2. Layer weight normalization: $w_\ell \leftarrow w_\ell \cdot (\sum_{\ell'=1}^{L} w_{\ell'})^{-1}$, $\forall\, \ell \in \{1, 2, \ldots, L\}$
        4-3. Compute $\mathbf{Y} \in \mathbb{R}^{n \times (K-1)}$ of $\mathbf{L}^{\mathbf{w}}$
        4-4. Obtain $K$ clusters $\{\mathcal{C}_k^{\mathbf{w}}\}_{k=1}^{K}$ by K-means algorithm on the rows of $\mathbf{Y}$
        4-5. Block-wise homogeneity test:
        calculate p-value$(i, j, \ell)$ by Algorithm 6.1, $1 \leq i, j \leq K$, $i \neq j$, and $1 \leq \ell \leq L$
        if p-value$(i, j, \ell) \leq \eta$ for some $(i, j, \ell)$ then
            Go back to step 4-1 with $z = z + 1$
        end if
        4-6. Estimate the noise level $\{\widehat{t}_{ij}^{(\ell)}\}$ for all $i, j, \ell$ and estimate $\widehat{t}_{\mathrm{LB}}^{\mathbf{w}}$ from (6.11)
        4-7. Block-wise identical noise test:
        estimate the aggregated noise level $\widehat{t}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \cdot \widehat{t}^{(\ell)}$
        if $\widehat{t}^{(\ell)}$ lies in the confidence interval (6.10) $\forall\, \ell$ then
            if $\widehat{t}^{\mathbf{w}} < \widehat{t}_{\mathrm{LB}}^{\mathbf{w}}$ then
                Flag$= 0$. $\mathcal{W}_{\mathrm{reliable}} = \mathcal{W}_{\mathrm{reliable}} \cup \{\mathbf{w}\}$.
            end if
        else if $\widehat{t}^{(\ell)}$ does not lie in the confidence interval (6.10) for some $\ell$ then
            4-8. Block-wise non-identical noise test:
            estimate the aggregated maximum noise level $\widehat{t}_{\max}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \widehat{t}_{\max}^{(\ell)}$
            if $\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij}(\frac{\widehat{t}_{\mathrm{LB}}}{\widehat{\overline{W}}_{ij}^{(\ell)}}, \widehat{p}_{ij}^{(\ell)}) \geq 1 - \alpha'_\ell \;\forall\, \ell$ then
                if $\widehat{t}_{\max}^{\mathbf{w}} < \widehat{t}_{\mathrm{LB}}^{\mathbf{w}}$ then
                    Flag$= 0$. $\mathcal{W}_{\mathrm{reliable}} = \mathcal{W}_{\mathrm{reliable}} \cup \{\mathbf{w}\}$.
                end if
            end if
        end if
        Go back to step 4-1 with $z = z + 1$
    end for
```

Algorithm 6.2 MIMOSA (continued)

```
        if Flag$= 1$ then
            Go back to step 1 with $K = K + 1$
        end if
    end while
    5. SNR criterion: select $\mathbf{w}^* = \arg\max_{\mathbf{w}\in\mathcal{W}_{\text{reliable}}} \frac{\widehat{t}^{\mathbf{w}}_{\text{LB}}}{\widehat{t}^{\mathbf{w}}}$
    6. Output final clustering result: $\{\mathcal{C}_k\}_{k=1}^K \leftarrow \{\mathcal{C}_k^{\mathbf{w}^*}\}_{k=1}^K$
```

the block-wise identical noise model, and therefore we can apply the phase transition results in Theorem 6.2 to evaluate the clustering reliability. In particular, we compare the estimated aggregated noise level $\widehat{t}^{\mathbf{w}}$ with the estimated phase transition lower bound $\widehat{t}^{\mathbf{w}}_{\text{LB}}$ of $t^{\mathbf{w}}_{\text{LB}}$ in Theorem 6.2 (c), where $\widehat{t}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \widehat{t}^{(\ell)} = \sum_{\ell=1}^{L} w_\ell \cdot \widehat{p}^{(\ell)} \cdot \widehat{\overline{\overline{W}}}^{(\ell)}$, and

$$\widehat{t}^{\mathbf{w}}_{\text{LB}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\sum_{\ell=1}^{L} w_\ell \cdot \widehat{\mathbf{L}}_k^{(\ell)})}{(K-1)\cdot \widehat{n}_{\min}}, \tag{6.11}$$

where $\widehat{\mathbf{L}}_k^{(\ell)}$ is the graph Laplacian matrix of within-cluster edges of cluster $\mathcal{C}_k^{\mathbf{w}}$ in the ℓ-layer, $S_{2:K}(\sum_{\ell=1}^{L} w_\ell \cdot \widehat{\mathbf{L}}_k^{(\ell)}) = \sum_{z=2}^{K} \lambda_z(\sum_{\ell=1}^{L} w_\ell \cdot \widehat{\mathbf{L}}_k^{(\ell)})$, and $\widehat{n}_{\min} = \min_{k\in\{1,2,\ldots,K\}} \widehat{n}_k$. Therefore, using Theorem 6.2, the clusters $\{\mathcal{C}_k^{\mathbf{w}}\}_{k=1}^K$ are deemed reliable if $\widehat{t}^{\mathbf{w}} < \widehat{t}^{\mathbf{w}}_{\text{LB}}$, since the eigenvector matrix $\mathbf{Y}$ used for multilayer SGC possesses cluster-wise separability.

6.3.5 Clustering reliability test under the block-wise non-identical noise model

In the iterative process of step 4 in MIMOSA, if every interconnection matrix $\widehat{\mathbf{C}}_{ij}^{(\ell)}$ passes the block-wise homogeneity test in Sec. 6.3.3, but some layers fail the clustering reliability test under the block-wise identical noise model in Sec. 6.3.4, the identified clusters $\{\mathcal{C}_k^{\mathbf{w}}\}_{k=1}^K$ are then used to test the clustering reliability under the block-wise non-identical noise model in Sec. 6.1.2 based on Theorem 6.3. Given a layer weight vector $\mathbf{w}$, the noise level estimates $\{\widehat{t}_{ij}^{(\ell)}\}$, and the estimate $\widehat{t}^{\mathbf{w}}_{\text{LB}}$ of the phase transition lower bound in (6.11), we compare the maximum noise level $\widehat{t}^{(\ell)}_{\max} = \max_{1\le i,j\le K, i\neq j} \widehat{t}^{(\ell)}_{ij}$ with $\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}$ for each layer ℓ. More specifically, for each layer ℓ, we use $\widehat{t}^{(\ell)}_{\max}$ to test the null hypothesis $H_0^{(\ell)}$: $t^{(\ell)}_{\max} < t^{\mathbf{w}}_{LB}$ against the alternative hypothesis $H_1^{(\ell)}$: $t^{(\ell)}_{\max} \geq t^{\mathbf{w}}_{LB}$. The test accepts $H_0^{(\ell)}$ if the condition in (6.12) holds, and rejects $H_0^{(\ell)}$ otherwise.

Using the Anscombe transformation on $\{\widehat{t}^{(\ell)}_{ij}\}$ for variance stabilization [8], testing whether $\widehat{t}^{(\ell)}_{\max}$ lies within an asymptotic $100(1-\alpha'_\ell)\%$ confidence interval under $H_0^{(\ell)}$ is equivalent to testing the condition

$$\prod_{i=1}^{K}\prod_{j=i+1}^{K} F_{ij}\left(\frac{\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}}{\widehat{\overline{W}}^{(\ell)}_{ij}}, \widehat{p}^{(\ell)}_{ij}\right) \geq 1-\alpha'_\ell, \tag{6.12}$$

where

$$F_{ij}\left(\frac{\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}}{\widehat{\overline{W}}^{(\ell)}_{ij}}, \widehat{p}^{(\ell)}_{ij}\right) = \Phi\left(\sqrt{4\widehat{n}_i\widehat{n}_j+2}\cdot\left(A_{ij}\left(\frac{\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}}{\widehat{\overline{W}}^{(\ell)}_{ij}}\right) - A_{ij}(\widehat{p}^{(\ell)}_{ij})\right)\right)\cdot \mathbb{I}_{\{\widehat{p}^{(\ell)}_{ij}\in(0,1)\}}$$
$$+\,\mathbb{I}_{\{\widehat{t}^{(\ell)}_{ij}<\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}\}}\mathbb{I}_{\{\widehat{p}^{(\ell)}_{ij}\in\{0,1\}\}}. \tag{6.13}$$

The proof of the condition in (6.12) is given in Appendix E.4.

Therefore, if the estimated maximum noise level $\widehat{t}^{(\ell)}_{\max}$ satisfies the condition in (6.12) for each layer ℓ, then if the aggregated maximum noise level $\widehat{t}^{\mathbf{w}}_{\max} = \sum_{\ell=1}^{L} w_\ell \widehat{t}^{(\ell)}_{\max} < \widehat{t}^{\mathbf{w}}_{\mathrm{LB}}$, by Theorem 6.3 the identified clusters $\{\mathcal{C}_k\}_{k=1}^{K}$ are deemed reliable since the eigenvector matrix $\mathbf{Y}$ possesses good cluster-wise separability.

6.3.6 A signal-to-noise ratio criterion for final clustering results

In step 4 of MIMOSA, given the number of clusters K, if MIMOSA finds any feasible layer weight vector that passes the clustering reliability tests in Sec. 6.3.4 or Sec. 6.3.5, it then stores the vector in the set $\mathcal{W}_{\mathrm{feasible}}$, and stops increasing K. This means that MIMOSA has identified a set of clustering results of the same number of clusters K that are deemed reliable based on the clustering reliability tests. To select the best clustering result from the feasible set, in step 5 we use the phase transition results established in Sec. 6.2 to define a signal-to-noise ratio (SNR) for each clustering result, which is

$$\mathrm{SNR}^{\mathbf{w}} = \frac{\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}}{\widehat{t}^{\mathbf{w}}}. \tag{6.14}$$

$\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}$ can be viewed as the aggregated strength of within-cluster edges, and $\widehat{t}^{\mathbf{w}}$ is the the aggregated noise level across layers. Therefore, the final clustering result is the clusters $\{\mathcal{C}_k^{\mathbf{w}^*}\}_{k=1}^K$, where $\mathbf{w}^* = \arg\max_{\mathbf{w}\in\mathcal{W}_{\mathrm{feasible}}} \mathrm{SNR}^{\mathbf{w}}$ is the layer weight vector having the largest SNR in the set $\mathcal{W}_{\mathrm{feasible}}$.

6.3.7 Computational complexity analysis

Let n and m be the number of nodes and edges in the aggregated graph $G^{\mathbf{w}}$, respectively. Fixing a model order K (i.e., the number of clusters) in the iteration of MIMOSA as displayed in Fig. 6.1, the computational complexity of MIMOSA consists of three parts.

1. Based on the incremental eigenpair computation method in CH. II, acquiring an additional smallest eigenvector for spectral graph clustering takes $O(m+n)$ iterations via power iteration approach, since the number of nonzero entries in the graph Laplacian matrix $\mathbf{L}^{\mathbf{w}}$ of $G^{\mathbf{w}}$ is $m+n$.

2. The estimation of the multilayer RIM parameters $\{t_{ij}^{(\ell)}\}$ takes $O(Lm)$ operations since for each layer they only depend on the number of edges and edge weights. The estimation of $t^{\mathbf{w}}_{\mathrm{LB}}$ takes $O(K(m+n)\cdot K) = O(K^2(m+n))$ iterations for computing the least partial eigenvalue sum among K clusters.

3. K-means clustering takes $O(nK^2)$ operations [174] for clustering n data points of dimension $K-1$ into K groups.

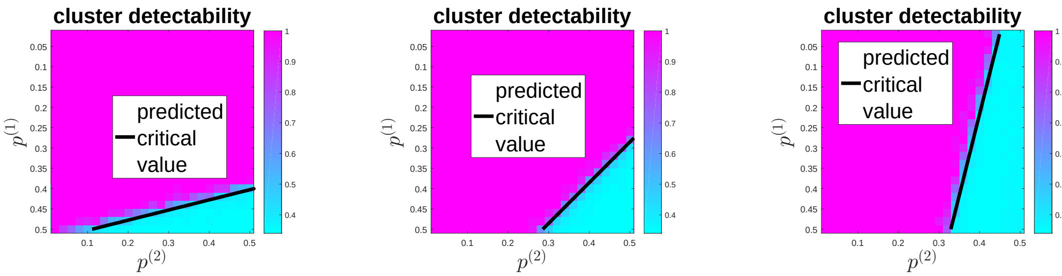
As a result, if MIMOSA outputs K clusters, then the iterative process leads to total computational complexity of $O(|\mathcal{T}|K^3(m+n)+|\mathcal{T}|Lm)$ operations.

6.4 Numerical Experiments

To validate the phase transition results in the accuracy of multilayer SGC via convex layer aggregation in Sec. 6.2, we generate synthetic multilayer graphs from a two-layer correlated multilayer graph model. Specifically, we generate edge connections within and between $K = 3$ equally-sized ground-truth clusters on $L = 2$ layers G_1 and G_2. The two layers G_1 and G_2 are correlated since their edge connections are generated in the following manner. For every node pair (u, v) of the same cluster, with probability q_{11} there is a within-cluster edge (u, v) in G_1 and G_2, with probability q_{10} there is a within-cluster edge (u, v) in G_1 but not in G_2, with probability q_{01} there is a within-cluster edge (u, v) in G_2 but not in G_1, and with probability q_{00} there is no edge (u, v) in G_1 and G_2. These four parameters are nonnegative and sum to 1. For between-cluster edges, we adopt the block-wise identical noise model in Sec. 6.1.2 such that for each layer ℓ, the edge connection between every node pair from different clusters is an i.i.d. Bernoulli random variable with parameter $p^{(\ell)}$.

6.4.1 Phase transitions in multilayer SGC via convex layer aggregation

By varying the noise level $\{p^{(\ell)}\}_{\ell=1}^2$, Fig. 6.2 shows the accuracy of multilayer SGC with respect to different layer weight vector $\mathbf{w} = [w_1\ w_2]^T$, where the accuracy is evaluated in terms of cluster detectability, i.e., the fraction of correctly identified nodes in the same cluster. Given a fixed $\mathbf{w}$, as proved in Theorem 6.2, there is indeed a phase transition in cluster detectability that separates the noise level $\{p^{(\ell)}\}_{\ell=1}^2$ into two regimes: a reliable regime where high clustering accuracy is guaranteed, and an unreliable regime where high clustering accuracy is impossible. Furthermore, the critical value of $\{p^{(\ell)}\}_{\ell=1}^2$ that separates these two regimes are successfully predicted



(a) $(w_1, w_2) = (0.8, 0.2)$ (b) $(w_1, w_2) = (0.5, 0.5)$ (c) $(w_1, w_2) = (0.2, 0.8)$

Figure 6.2: Phase transitions in the accuracy of multilayer SGC with respect to different layer weight vector $\mathbf{w} = [w_1 \ w_2]^T$ for the two-layer correlated graph model. $n_1 = n_2 = n_3 = 1000$, $q_{11} = 0.3$, $q_{10} = 0.2$, $q_{01} = 0.1$, and $q_{00} = 0.4$. The results are averaged over 10 runs. For a given $\mathbf{w}$, the variations in the noise level $\{p^{(\ell)}\}_{\ell=1}^{2}$ indeed separates the accuracy of multilayer SGC into a reliable regime and an unreliable regime. Furthermore, the critical value that separates these two regimes are successfully predicted by Theorem 6.2.

by Theorem 6.2 (c), which validates the phase transition analysis. Fig. 6.3 shows the geometric mean of cluster detectability from multilayer SGC via convex layer aggregation for the two-layer correlated graph model, where w_1 is uniformly sampled from the interval $[0, 1]$. It can be observed that the universal phase transition lower bound predicted by (6.4) indeed specifies a regime where any layer weight vector $\mathbf{w} \in \mathcal{W}_2$ can lead to correct clustering results. Similarly, the universal phase transition upper bound predicted by (6.5) specifies a regime where any layer weight vector $\mathbf{w} \in \mathcal{W}_2$ leads to incorrect clustering results.

6.4.2 The effect of layer weight vector on multilayer SGC via convex layer aggregation

Next we investigate the effect of layer weight vector $\mathbf{w}$ on multilayer SGC via convex layer aggregation given a fixed noise level. In the two-layer graph setting, since by definition $w_2 = 1 - w_1$, it suffices to study the effect of w_1 on clustering accuracy. Fig. 6.4 shows the clustering accuracy by varying w_1 under the two-layer correlated graph model. As shown in Fig. 6.4 (a), if each layer has low noise level,

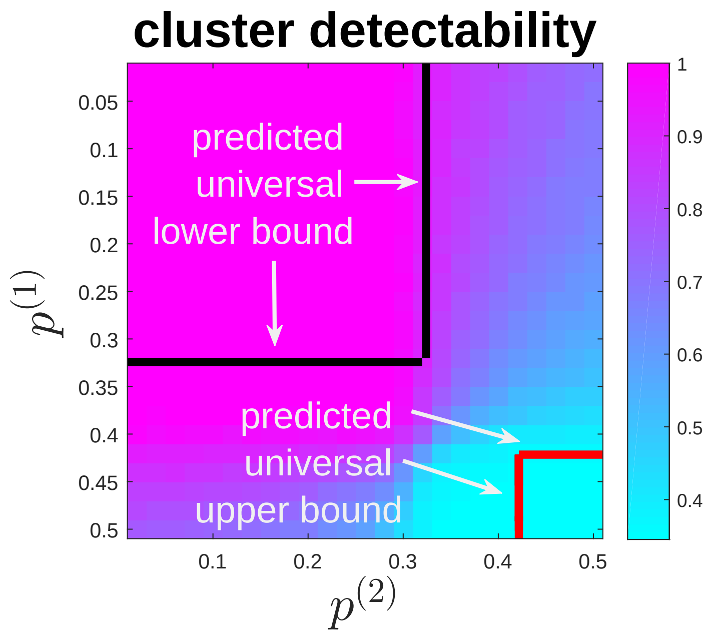
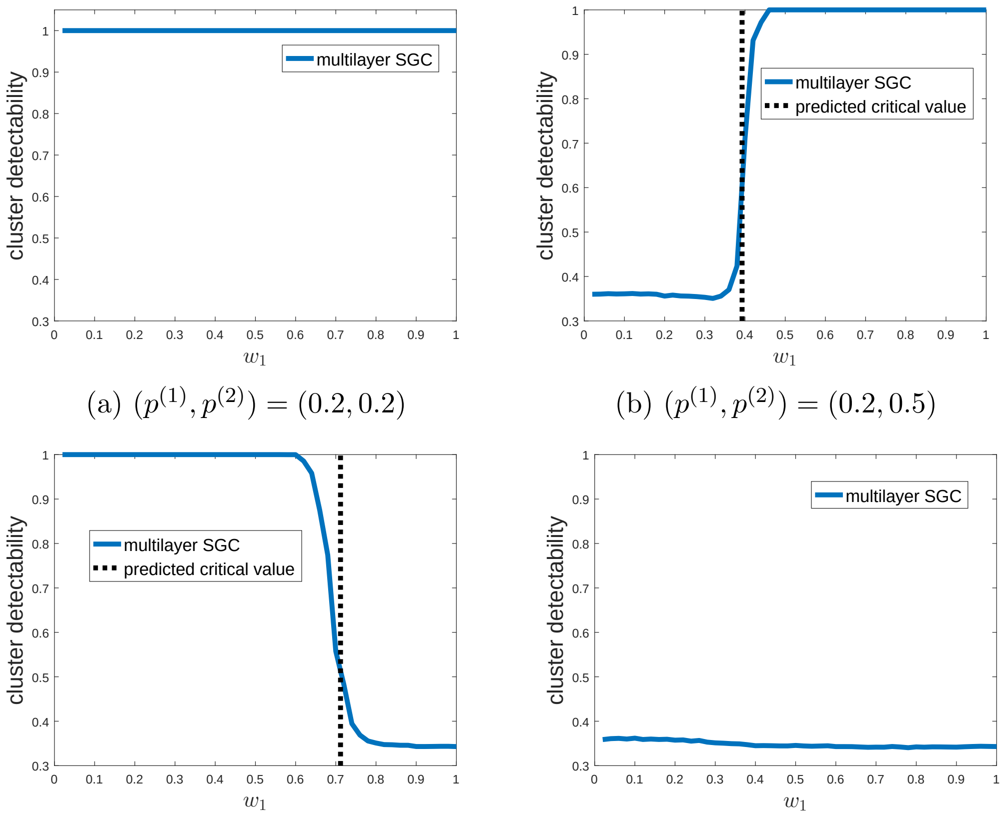


Figure 6.3: Phase transitions in the geometric mean of cluster detectability from multilayer SGC via convex layer aggregation for the two-layer correlated graph model, where w_1 is uniformly sampled from the $[0,1]$ with unit interval 0.1. $n_1 = n_2 = n_3 = 500$, $q_{11} = 0.3$, $q_{10} = 0.2$, $q_{01} = 0.1$, and $q_{00} = 0.4$. The results are averaged over 10 runs. It can be observed that the universal phase transition lower bound predicted by (6.4) indeed specifies a regime where any layer weight vector $\mathbf{w} \in \mathcal{W}_2$ can lead to correct clustering results. Similarly for the universal phase transition upper bound predicted by (6.5).

then any layer weight vector $\mathbf{w} \in \mathcal{W}_2$ can lead to correct clustering result. If one layer has high noise level, Fig. 6.4 (b) and (c) show that there exists a critical value $w_1^\star \in [0,1]$ that separates the cluster detectability into a reliable regime and an unreliable regime. In particular, Theorem 6.2 implies that the critical value $w_1^\star$, if existed, satisfies the condition $t^{\mathbf{w}} = t^{\mathbf{w}^*}$ when $\mathbf{w} = [w_1^\star, 1 - w_1^\star]^T = \mathbf{w}^*$, which is equivalent to

$$\frac{K-1}{K}\left[w_1^\star p^{(1)} + (1-w_1^\star) p^{(2)}\right] = w_1^\star \cdot \min_{k \in \{1,2,\ldots,K\}} S_{2:K}\left(\frac{\mathbf{L}_k^{(1)}}{n}\right) + (1-w_1^\star) \cdot \min_{k \in \{1,2,\ldots,K\}} S_{2:K}\left(\frac{\mathbf{L}_k^{(2)}}{n}\right). \tag{6.15}$$



(a) $(p^{(1)}, p^{(2)}) = (0.2, 0.2)$

(b) $(p^{(1)}, p^{(2)}) = (0.2, 0.5)$

(c) $(p^{(1)}, p^{(2)}) = (0.5, 0.2)$

(d) $(p^{(1)}, p^{(2)}) = (0.5, 0.5)$

Figure 6.4: The effect of the layer weight vector $\mathbf{w} = [w_1 \ w_2]^T$ on the accuracy of multilayer SGC with respect to difference noise level $\{p^{(\ell)}\}_{\ell=1}^2$ for the two-layer correlated graph model. $n_1 = n_2 = n_3 = 1000$, $q_{11} = 0.3$, $q_{10} = 0.2$, $q_{01} = 0.1$, and $q_{00} = 0.4$. The results are averaged over 50 runs. Fig. 6.4 (a) shows that in the case of low noise level for each layer, any layer weight vector $\mathbf{w} \in \mathcal{W}_2$ can lead to correct clustering result. Fig. 6.4 (b) and (c) show that if one layer has high noise level, then there may exist a critical value $w_1^* \in [0, 1]$ that separates the cluster detectability into a reliable regime and an unreliable regime. Furthermore, the critical value w_1^* is shown to satisfy the equation in (6.15) derived from Theorem 6.2. Fig. 6.4 (d) shows that in the case of high noise level for each layer, no layer weight vector can lead to correct clustering result, and the cluster detectability is similar to random guessing of clustering accuracy 33.33%.

It is observed that the empirical critical value $w_1^\star$ matches the predicted value from (6.15). Lastly, as shown in Fig. 6.4 (d), if each layer has high noise level, then no layer weight vector can lead to correct clustering result, and the corresponding cluster detectability is similar to random guessing of clustering accuracy $\frac{1}{K} \approx 33.33\%$.

6.5 MIMOSA on Real-World Multi-Layer Graphs

6.5.1 Dataset descriptions

In this section, we apply MIMOSA to 7 real-world multilayer graphs and compute the external and internal clustering metrics for quality assessment. The statistics of the 7 real-world multilayer graphs are summarized in Table 6.1, and the details are described as follows.

- **VC 7th grader social network** [162]: This dataset is based a survey of social relations among 29 7th grade students in Victoria, Australia. There are 12 boys and 17 girls in this dataset. A 3-layer graph is created based on different relationships, including "friends you get on with", "your best friends", and "friends you prefer to work with" in the class. For each layer we only retain the edge of mutual agreement among every student pair.

- **Leskovec-Ng collaboration network**[1]: We collected the coauthors of Prof. Jure Leskovec or Prof. Andrew Ng at Stanford University from ArnetMiner [151] from year 1995 to year 2014. In total, there are 191 researchers in this dataset. We separate the coauthorship of 20 years by a 5-year interval and hence create a 4-layer multilayer graph. For each layer, there is an edge between two researchers if they coauthored at least one paper in the 5-year interval. For every edge in each layer, we adopt the temporal collaboration strength as the edge weight [141, 177]. Notably, although Prof. Leskovec and Prof. Ng both

[1] The dataset can be downloaded from https://sites.google.com/site/pinyuchenpage/datasets

worked at the same department, there is no record of coauthorship between them on ArnetMiner. Nonetheless, the entire collaboration network among 191 researchers is a connected graph so that the graph clustering task is nontrivial. We manually label each researcher by either "Leskovec's collaborator" or "Ng's collaborator" based on the collaboration frequency, and use the labels as the ground-truth cluster assignment. The ground-truth clusters with researcher names are displayed in Fig. 6.5.

- **109th Congress votes**: We collected the votes of 100 senators of the 109th U.S. Congress to create 3 multilayer graph datasets based on bill subjects, including "Budget", "Energy", and "Security". Only bills that every senator has voting records are considered in these datasets. For each bill subject (a multilayer graph), each layer corresponds to one bill. In each layer, there is an edge between two senators if they both have the same vote. We use the party (Democratic or Republican) as the ground-truth cluster label. In addition, we label the independent senator as Democratic since he caucused with the Democrats.

- **Reality mining** [123]: The reality mining dataset contains mobile and social traces among 94 MIT students. We extract the largest connected component of students from this dataset to form a 2-layer graph, where one layer represents user connection via text messaging, and the other layer represents user connection via proximity (Bluetooth scanning). For each layer we only retain the edge of mutual contact among every student pair.

- **London transportation network** [44]: The London transportation network dataset contains different transportation routes of stations in London. We extract the largest connected component of stations that are either connected by Overground transportation or by Docklands Light Railway (DLR) to form a 2-

Dataset	# of layers	Ground-truth cluster labels
VC 7th grader social network	3	boy girl
Leskovec-Ng collaboration network	4	Leskovec's collaborator Ng's collaborator
109th Congress votes - Budget	4	Democratic Republican
109th Congress votes - Energy	2	Democratic Republican
109th Congress votes - Security	2	Democratic Republican
Reality mining	2	None
London transportation network	2	None

Table 6.1: Summary of real-world multilayer graph datasets.

layer graph, where one layer represents overground connectivity, and the other layer represents DLR connectivity.

Since MIMOSA allows the input multilayer graph to be weighted, for each layer G_ℓ, if G_ℓ is unweighted, we adopt the degree normalization technique [97] such that the (u,v)-th entry in the edge weight matrix $\mathbf{W}^{(\ell)}$ is $[\mathbf{W}^{(\ell)}]_{uv} = \frac{[\mathbf{A}^{(\ell)}]_{uv}}{\sqrt{d_u^{(\ell)} \cdot d_v^{(\ell)}}}$ if $d_u^{(\ell)} > 0$ and $d_v^{(\ell)} > 0$, and $[\mathbf{W}^{(\ell)}]_{uv} = 0$ otherwise, where $\mathbf{A}^{(\ell)}$ is the adjacency matrix of G_ℓ and $d_u^{(\ell)}$ is the degree of node u in G_ℓ.

6.5.2 Performance evaluation

Using the multilayer graphs datasets listed in Table 6.1, we compare the clustering performance of MIMOSA with other two methods that also feature automated cluster assignment without specifying the number of clusters K *a priori*. The first method is the baseline approach that assigns uniform weight on each layer for layer aggregation (i.e., $w_\ell = \frac{1}{L} \ \forall \ \ell$). Since this baseline approach is equivalent to MIMOSA with the setting $\mathbf{w}^{\text{ini}} = \frac{\mathbf{1}_L}{L}$ and $\mathcal{T} = \{0\}$, we call this method *MIMOSA-uniform*. The second method is a greedy multilayer modularity maximization approach that extends the

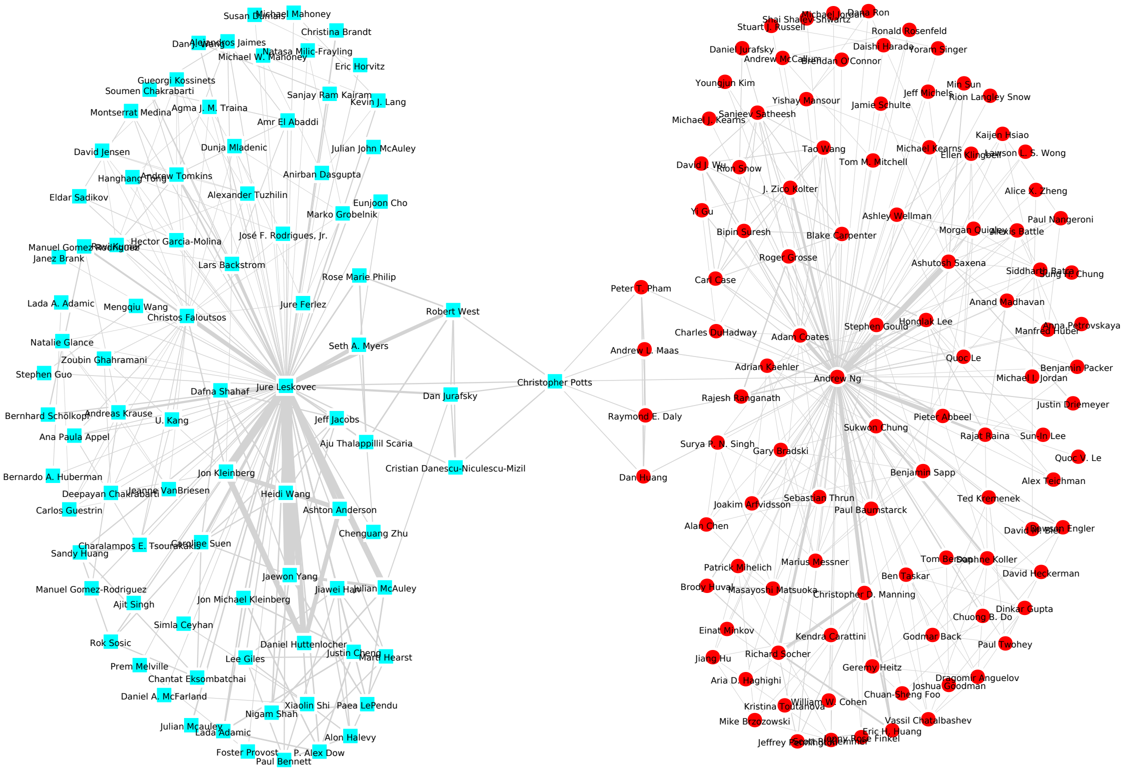


Figure 6.5: Ground-truth clusters of the collected Leskovec-Ng collaboration network. Nodes represent researchers, edges represent the strength of coauthorship [141, 177], and colors and shapes represent two clusters - "Leskovec's collaborator" (cyan square) or "Ng's collaborator" (red circle).

Louvain method for clustering in single-layer graphs to multilayer graphs, which is called *GenLouvain*[2]. GenLouvain aims to merge the nodes to maximize the multilayer modularity defined in [101] in a greedy manner. For all datasets, we set the resolution parameter $\gamma = 1$ and the latent inter-layer coupling parameter $\omega = 1$ for GenLouvain. For MIMOSA, we set $\mathbf{w}^{\text{ini}} = \frac{\mathbf{1}_L}{L}$ to be a uniform vector, $\eta = 10^{-5}$, $\alpha_\ell = \alpha'_\ell = 0.05 \; \forall \; \ell$, and the regularization set $\mathcal{T} = \{0, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}\}$.

We use the external and internal clustering metrics introduced in Sec. 5.2.1 to evaluate the performance of different methods. Since these metrics are designed for single-layer graphs, we extend these metrics to multilayer graphs by summing the

[2] http://netwiki.amath.unc.edu/GenLouvain/GenLouvain

metrics from each layer.

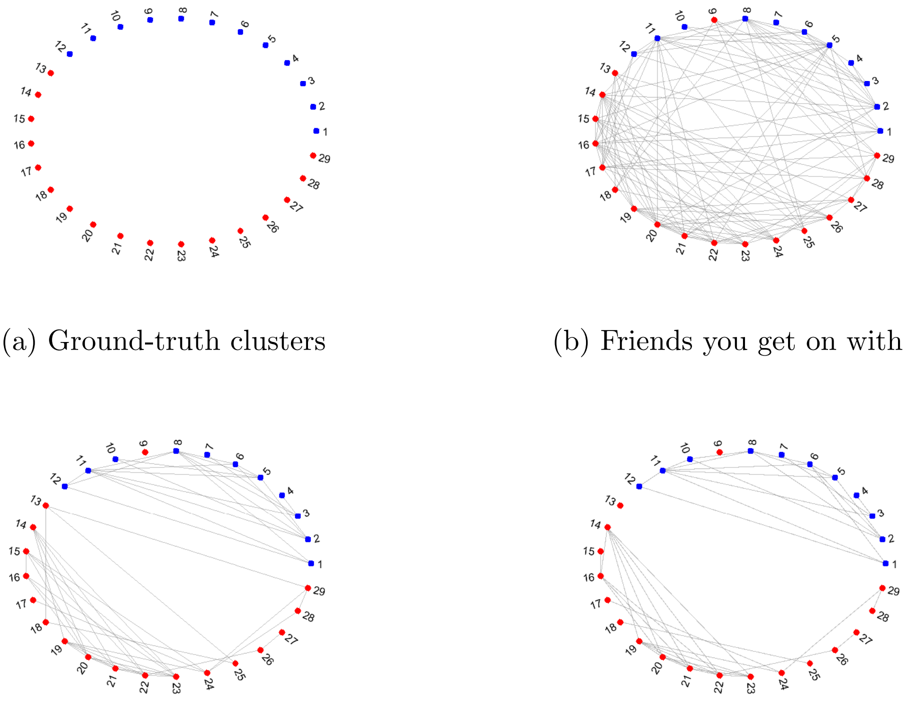
Table 6.2 summarizes the external and internal clustering metrics of the three methods for the multilayer graph datasets listed in Table 6.1. For MIMOSA and MIMOSA-uniform, we terminate the iterative process and report the clustering result as “not applicable” (NA) when the number of clusters K exceeds $\frac{n}{2}$, where n is the number of nodes. As a result, NA means that before termination no clustering results have passed the clustering reliability tests.

It is observed from Table 6.2 that MIMOSA has the best clustering performance among 5 out of 7 datasets. For the Congress votes-Budget and Congress votes-Security datasets, MIMOSA still has comparable performance to the best method. For the VC 7th grader social network and Leskovec-Ng collaboration network datasets, MIMOSA-uniform fails to find a reliable clustering result, whereas MIMOSA has superior clustering metrics over other methods. The robustness of MIMOSA implies the utility of layer weight adaptation, and it also suggests that assigning uniform weight to every layer regardless of the noise level may lead to unreliable clustering results. In addition, we also observe that GenLouvain tends to identify more clusters than the number of ground-truth clusters.

As a visual illustration, Fig. 6.6 displays the ground-truth clusters and the clusters identified by MIMOSA for each layer of the VC 7th grader social network dataset. The number of clusters identified by MIMOSA is 2, which is consistent with the ground truth. The optimal layer weight vector obtained from step 5 of MIMOSA in Algorithm 6.2 is $\mathbf{w}^* = [0.0531\ 0.1608\ 0.7861]^T$. Comparing each layer with the ground-truth clusters, it can be observed that the connectivity patterns in Fig. 6.6 (c) and (d) are more consistent with the ground truth, whereas the connectivity pattern in Fig. 6.6 (a) is less informative, which explains why MIMOSA adapts more weights to the second and the third layers. It is worth noting that MIMOSA correctly groups all nodes into 2 clusters except node 9. However, we also observe that node 9 has no

Dataset	Method	K	NMI	RI	F-measure	Conductance	NC
VC 7th grader social network	MIMOSA	2	**0.8123**	**0.9310**	**0.9317**	**0.2649**	**0.4330**
	MIMOSA-uniform	NA	NA	NA	NA	NA	NA
	GenLouvain	3	0.6495	0.7833	0.7333	0.4487	0.6051
Leskovec-Ng collaboration network	MIMOSA	2	**1**	**1**	**1**	**0.0213**	**0.0415**
	MIMOSA-uniform	NA	NA	NA	NA	NA	NA
	GenLouvain	16	0.4972	0.7156	0.6055	0.3054	0.3702
109th Congress votes - Budget	MIMOSA	2	0.7959	0.9224	0.9220	0.2713	0.4975
	MIMOSA-uniform	2	**0.8778**	**0.9604**	**0.9603**	**0.2702**	**0.5055**
	GenLouvain	2	0.7959	0.9224	0.9220	0.2713	0.4978
109th Congress votes - Energy	MIMOSA	2	**0.7290**	**0.8861**	**0.8855**	**0.1151**	**0.2086**
	MIMOSA-uniform	2	0.6716	0.8513	0.8508	0.1154	0.2178
	GenLouvain	2	0.5403	0.8182	0.8173	**0.1151**	**0.2086**
109th Congress votes - Security	MIMOSA	2	0.6105	0.8513	0.8506	**0.04**	0.0785
	MIMOSA-uniform	2	0.6304	0.8513	0.8506	**0.04**	0.0785
	GenLouvain	2	**0.6598**	**0.8685**	**0.8678**	**0.04**	**0.0770**
Reality mining	MIMOSA	2	-	-	-	**0.0819**	**0.1573**
	MIMOSA-uniform	2	-	-	-	**0.0819**	**0.1573**
	GenLouvain	3	-	-	-	0.2239	0.3165
London transportation network	MIMOSA	5	-	-	-	**0.0553**	**0.0801**
	MIMOSA-uniform	5	-	-	-	**0.0553**	**0.0801**
	GenLouvain	14	-	-	-	0.1558	0.1763

Table 6.2: Summary of the number of identified clusters (K) and the external and internal clustering metrics. "NA" means "not applicable", and "-" means "not available" due to lack of ground-truth cluster labels. For each dataset, the method that leads the highest clustering metric is highlighted in bold face.



(a) Ground-truth clusters

(b) Friends you get on with

(c) Your best friends

(d) Friends you work with

Figure 6.6: Illustration of ground-truth clusters and the clusters found by MIMOSA for the VC 7th grader social network dataset. Fig. 6.6 (a) displays the ground-truth clusters, where nodes 1 to 12 are boys (labeled by blue color) and nodes 13 to 29 are girls (labeled by red color). Fig. 6.6 (b) to (d) display the clusters (labeled by different colors) found by MIMOSA on each layer. Comparing to the ground-truth clusters, MIMOSA correctly group all nodes into 2 clusters except node 9, since node 9 has no edge connections in Fig. 6.6 (c) and (d), and has more connections to girls than boys in Fig. 6.6 (a).

edge connections in the two informative layers as shown in Fig. 6.6 (c) and (d), and indeed has more connections to girls than boys in the first layer as shown in Fig. 6.6 (a).

CHAPTER VII

Local Fiedler Vector Centrality and Deep Community Detection

In this chapter we specify how insertion and deletion of nodes or edges on a graph can be mapped to certain matrix operations associated with the graph. We then define a new centrality measure based on the matrix operation and show its application for deep community detection. A deep community in a graph is a connected component that can only be seen after removal of nodes or edges from the rest of the graph. We formulate the problem of detecting deep communities as multi-stage node removal that maximizes a new centrality measure, called the local Fiedler vector centrality (LFVC), at each stage. The LFVC is associated with the sensitivity of algebraic connectivity to node or edge removals. We prove that a greedy node/edge removal strategy, based on successive maximization of LFVC, has bounded performance loss relative to the optimal, but intractable, combinatorial batch removal strategy. Under a stochastic block model framework, we show that the greedy LFVC strategy can extract deep communities with probability one as the number of observations becomes large. We apply the greedy LFVC strategy to real-world network datasets. Compared with conventional community detection methods we demonstrate improved ability to identify important communities and key members in the network.

Many community detection methods are based on detecting nodes or edges with high centrality. Some examples of commonly used centrality measures are summarized in Sec. 1.5.]In [165], a node removal strategy based on targeting high degree nodes is proposed to improve the performance of the modularity method. The authors of [165] argue that high-degree nodes incur more noisy connections than low-degree nodes, and it is experimentally demonstrated that removing high-degree nodes can better reveal the community structure.

Nonparametric community detection methods, such as the the edge betweenness method [63] and the modularity method [105], can be viewed as edge removal strategies that aim to maximize a centrality measure, e.g., the modularity or betweenness measures. It is worth noting that these methods presume that each node in the graph is affiliated with a community. However, in some community detection applications it often occurs that the graphs contain spurious edges connecting to irrelevant "noisy" nodes that are not members of any single community. In such cases, noisy nodes and edges mask the true communities in the graph. Detection of these masked communities is a difficult problem that we call "deep community detection". The formal definition of a deep community is given in Sec. 7.2. Due to the presence of noisy nodes and spurious edges [9, 57], deep communities elude detection when conventional community detection methods methods are applied.

In this chapter, a new partitioning strategy is applied to detect deep communities. This strategy uses a new local measure of centrality that is specifically designed to unmask communities in the presence of spurious edges. The new partitioning strategy is based on a novel spectral measure [38] of centrality called local Fiedler vector centrality (LFVC). LFVC is associated with the sensitivity of algebraic connectivity [55] when a subset of nodes or edges are removed from a graph [27, 28]. We show that LFVC relates to a monotonic submodular set function which ensures that greedy node or edge removals based on LFVC are nearly as effective as the optimal combinatorial batch removal strategy.

Our approach utilizes LFVC to iteratively remove nodes in the graph to reveal deep communities. A removed node that connects multiple deep communities is assigned mixed membership: it is shared among these communities. We illustrate the proposed deep community detection method on several real-world networks. When our proposed greedy LFVC approach is applied to the network scientist coauthorship dataset [106], it reveals deep communities that are not identified by conventional community detection methods. When applied to social media, the Last.fm online music dataset, we show that LFVC has the best performance in detecting users with similar interest in artists.

7.1 Algebraic Connectivity and Fiedler Vector

7.1.1 Algebraic connectivity

The algebraic connectivity of G is defined as the second smallest eigenvalue of $\mathbf{L}$, i.e., $\lambda_2(\mathbf{L})$. G is connected if and only if $\lambda_2(\mathbf{L}) > 0$. Moreover, it is a well-known property [55] that for any non-complete graph,

$$\lambda_2(\mathbf{L}) \leq \text{node connectivity} \leq \text{edge connectivity}, \tag{7.1}$$

where node/edge connectivity is the least number of node/edge removals that disconnects the graph. (7.1) is the main motivation for our proposed node/edge pruning approach. A graph with larger algebraic connectivity is more resilient to node and edge removals. In addition, let $d_{\min}$ be the minimum degree of G, it is also well-known [38, 40] that $\lambda_2(\mathbf{L}) \leq 1$ if and only if $d_{\min} = 1$. That is, a graph with a leaf node (i.e., a node with a single edge) cannot have algebraic connectivity larger than 1. For any connected graph, we can represent the algebraic connectivity as

$$\lambda_2(\mathbf{L}) = \min_{\|\mathbf{x}\|_2=1,\ \mathbf{x}\perp\mathbf{1}} \mathbf{x}^T\mathbf{L}\mathbf{x} \tag{7.2}$$

by the Courant-Fischer theorem [76] and the fact that the constant vector is the eigenvector associated with $\lambda_1(\mathbf{L}) = 0$.

7.1.2 Fiedler vector

The Fiedler vector of a graph is the eigenvector associated with the second smallest eigenvalue $\lambda_2(\mathbf{L})$ of the graph Laplacian matrix $\mathbf{L}$ [55]. The Fiedler vector has been widely used in graph partitioning, image segmentation and data clustering [97, 127, 143, 144, 148]. Analogously to modularity partitioning, the Fiedler vector performs community detection by separating the nodes in the graph according to the signs of the corresponding Fiedler vector elements. Similarly, hierarchical community structure can be detected by recursive partitioning with the Fiedler vector. In this chapter, we use the Fiedler vector to define a new centrality measure. One advantage of using the Fiedler vector over other global centrality measures is that it can be computed in a distributed manner via local information exchange over the graph [16].

7.2 Deep Community Detection

A deep community is defined in terms of an additive signal (community) plus noise model. Let $\mathbf{A}_1, \ldots, \mathbf{A}_g$ denote the $n \times n$ mutually orthogonal binary adjacency matrices associated with g non-singleton connected components in a noiseless graph G_0 over n nodes. Assume the nodes have been permuted so that $\mathbf{A}_1, \ldots, \mathbf{A}_g$ are block diagonal with non-overlapping block indices $\mathcal{I}_1, \ldots, \mathcal{I}_g$. The observed graph G is a noise corrupted version of G_0 where random edges have been inserted between the connected components of G_0. More specifically, let $\mathbf{A}_{\text{nse}}$ be a random adjacency

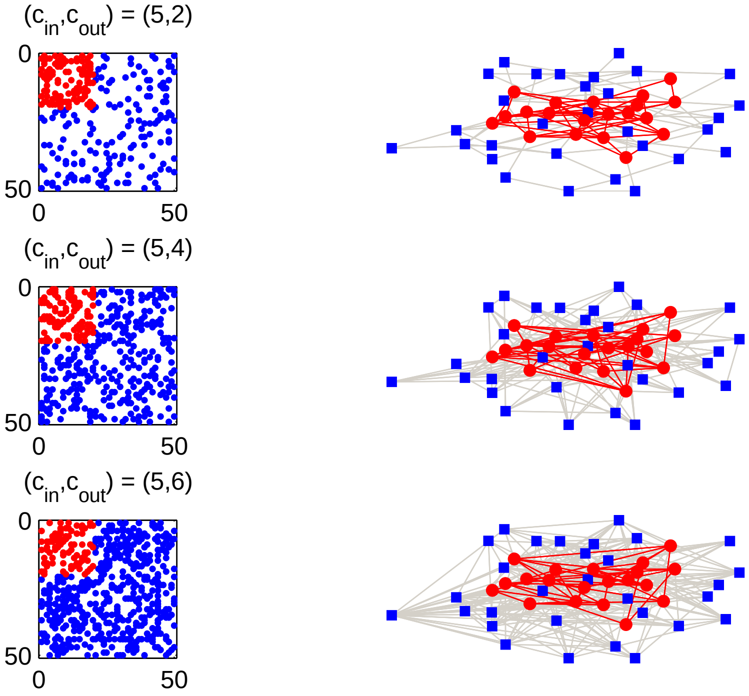


Figure 7.1: An illustration of deep community detection. The entire network is a realization of the two-community stochastic block model with $p_2 = p$. That is, the first block is the deep community and the second block only contains spurious edges. The network size $n = 50$ and deep community size $n_1 = n_{\text{deep}} = 20$. The parameters $c_{\text{in}} = n_{\text{deep}} \cdot p_1$ and $c_{\text{out}} = n_2 \cdot p$. The nodes in the deep community are marked by red solid circle, and the other nodes are marked by blue solid rectangles. The left and right columns represent adjacency matrices and their corresponding graphs, respectively. It is observed when c_{in} is fixed, the deep community is more difficult to be detected as c_{out} increases.

matrix with the property that $\mathbf{A}_{\text{nse}}(i,j) = 0$, $i,j \in \mathcal{I}_k$, for $k = 1, \ldots, g$ and where the rest of the elements of $\mathbf{A}_{\text{nse}}$ are Bernoulli i.i.d random variables. Then the adjacency matrix $\mathbf{A}$ of G satisfies the signal plus noise model

$$\mathbf{A} = \sum_{k=1}^{g} \mathbf{A}_k + \mathbf{A}_{\text{nse}}. \tag{7.3}$$

The deep community detection problem is to recover the embedded connected components $\mathbf{A}_1, \ldots, \mathbf{A}_g$ from the noise corrupted observations $\mathbf{A}$. An illustrative visual example of deep community detection is shown in Fig. 7.1. Deep community detection is equivalent to the planted clique problem [7] in the special case that $g = 1$ and the non-zero block of $\mathbf{A}_1$ corresponds to a complete graph, i.e., all off-diagonal elements of this block are equal to one. Models similar to (7.3) have also been used for hypothesis testing on the existence of dense subgraphs embedded in random graphs [99, 100]. The null hypothesis is the noise only model (i.e., $\mathbf{A}_k = \mathbf{0}\ \forall k$). The alternative hypothesis is the signal plus noise model (7.3) with $\mathbf{A}_k \neq \mathbf{0}$.

We propose an iterative denoising algorithm for recovering deep communities that is based on either node or edge removals. The proposed algorithm uses a spectral centrality measure, defined in Sec. 7.3, to determine the nodes/edges to be pruned from the observed graph with adjacency matrix $\mathbf{A}$.

Let $\widetilde{\mathbf{L}}$ be the resulting $n \times n$ graph Laplacian matrix after removing a subset of nodes or edges from the graph. The following theorem provides an upper bound on the number of deep communities in the remaining graph $\widetilde{G}$.

Theorem 7.1. *For any node removal set $\mathcal{R}$ of G with $|\mathcal{R}| = q$, let r be the rank of the resulting graph Laplacian matrix $\widetilde{\mathbf{L}}$ and let $\|\widetilde{\mathbf{L}}\|_* = \sum_i \lambda_i(\widetilde{\mathbf{L}})$ denote its nuclear norm. The number ϵ of remaining non-singleton connected components in $\widetilde{G}$ has the upper bound*

$$\begin{aligned}\epsilon &\leq n - q - r \\ &\leq n - q - \frac{\|\widetilde{\mathbf{L}}\|_*}{\lambda_n(\widetilde{\mathbf{L}})} \\ &= n - q - \frac{2\widetilde{m}}{\lambda_n(\widetilde{\mathbf{L}})},\end{aligned} \tag{7.4}$$

where $\widetilde{m}$ is the number of edges in $\widetilde{G}$. The first inequality in (7.4) becomes an equality if all connected components in $\widetilde{G}$ are non-singletons. The second inequality in (7.4) becomes an equality if all non-singleton connected components are complete subgraphs of the same size. Similarly, for any edge removal set of G, let r be the rank of *the resulting graph Laplacian matrix* $\widetilde{\mathbf{L}}$. *The number* ϵ *of remaining non-singleton connected components in* $\widetilde{G}$ *has the upper bound* $\epsilon \leq n - r \leq n - \frac{\|\widetilde{\mathbf{L}}\|_*}{\lambda_n(\widetilde{\mathbf{L}})} = n - \frac{2\widetilde{m}}{\lambda_n(\widetilde{\mathbf{L}})}$.

Proof. The proof can be found in Appendix F.1. □

The upper bound in **Theorem** 7.1 can be further relaxed by applying the inequality $\lambda_n(\widetilde{\mathbf{L}}) \leq 2\widetilde{d}_{\max}$ [38], where $\widetilde{d}_{\max}$ is the maximum degree of $\widetilde{G}$. Other bounds on $\lambda_n(\widetilde{\mathbf{L}})$ can be found in [69].

The next theorem shows that the largest non-singleton connected component size can be represented as a norm of a matrix whose column vectors are orthogonal and sparsest among all binary vectors that form a basis of the null space of $\widetilde{\mathbf{L}}$.

Theorem 7.2. *Define the sparsity of a vector to be the number of zero entries in the vector. Let* $\mathrm{null}(\widetilde{\mathbf{L}})$ *denote the null space of* $\widetilde{\mathbf{L}}$ *and let* $\mathbf{X}$ *denote the matrix whose columns are orthogonal and they form the sparsest basis of* $\mathrm{null}(\widetilde{\mathbf{L}})$ *among binary vectors. Let* $\psi(\widetilde{G})$ *be the largest non-singleton connected component size of* $\widetilde{G}$. *Then* $\psi(\widetilde{G}) = \|\mathbf{X}\|_1 = \max_i \|\mathbf{x}_i\|_1$, *where* $\mathbf{x}_i$ *is the i-th column vector of binary matrix* $\mathbf{X}$.

Proof. The proof can be found in Appendix F.2. □

Theorems 7.1 and 7.2 are key results that motivate and theoretically justify the proposed local Fiedler vector centrality measure introduced below. **Theorem** 7.1 establishes that the number of deep communities is closely related to the number of edge/node removals that are required to reveal them. **Theorem** 7.2 establishes that L_1 norm of the sparsest basis for the null space of the graph Laplacian matrix can be used to estimate the size of the largest deep community in the network.

7.3 The Proposed Node and Edge Centrality: Local Fiedler Vector Centrality (LFVC)

The proposed deep community detection algorithm (Algorithm 7.1) is based on removal of nodes or edges according to how the removals affect a measure of algebraic connectivity. This measure, called the local Fiedler vector centrality (LFVC), is computed from the graph Laplacian matrix. In particular, the LFVC is motivated by the fact that node/edge removals result in low rank perturbations to the graph Laplacian matrix when $n \gg d_{\max}$, where $d_{\max}$ is the maximum degree. The node and edge LFVC are then defined to correspond to an upper bound on algebraic connectivity.

7.3.1 Edge-LFVC

Considering the graph $\widetilde{G}(i,j) = (\mathcal{V}, \mathcal{E} \cup (i,j))$ by adding an edge $(i,j) \notin \mathcal{E}$ to G, we have $\widetilde{\mathbf{L}} = \mathbf{L} + \Delta\mathbf{L}$ and $\Delta\mathbf{L} = \Delta\mathbf{D} - \Delta\mathbf{A}$, where $\Delta\mathbf{D}$ and $\Delta\mathbf{A}$ are the augmented degree and adjacency matrices, respectively. Denote the resulting graph Laplacian matrix by $\widetilde{\mathbf{L}}(i,j)$. Let $\mathbf{e}_i$ be a zero vector except that its i-th element is equal to 1. Then

$$\Delta\mathbf{D} = \mathrm{diag}(\mathbf{e}_i) + \mathrm{diag}(\mathbf{e}_j) = \mathbf{e}_i\mathbf{e}_i^T + \mathbf{e}_j\mathbf{e}_j^T; \tag{7.5}$$

$$\Delta\mathbf{A} = \mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T, \tag{7.6}$$

and therefore

$$\widetilde{\mathbf{L}}(i,j) = \mathbf{L} + (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T. \tag{7.7}$$

Thus, the resulting graph Laplacian matrix $\widetilde{\mathbf{L}}(i,j)$ after adding an edge (i,j) to G is the original graph Laplacian matrix $\mathbf{L}$ perturbed by a rank one matrix $(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i -$

$\mathbf{e}_j)^T$. Similarly, when an edge $(i,j) \in \mathcal{E}$ is removed from G, we have $\widetilde{\mathbf{L}}(i,j) = \mathbf{L} - (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T$.

Consider removing an edge $(i,j) \in \mathcal{E}$ from G resulting in $\widetilde{\mathbf{L}}(i,j)$ above. Let $\mathbf{y}$ denote the Fiedler vector of $\mathbf{L}$, computing $\mathbf{y}^T\widetilde{\mathbf{L}}(i,j)\mathbf{y}$ gives an upper bound on $\lambda_2(\widetilde{\mathbf{L}}(i,j))$ as

$$\begin{aligned}\lambda_2(\widetilde{\mathbf{L}}(i,j)) &\leq \mathbf{y}^T\widetilde{\mathbf{L}}(i,j)\mathbf{y} \\ &= \mathbf{y}^T(\mathbf{L} - (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T)\mathbf{y} \\ &= \lambda_2(\mathbf{L}) - (y_i - y_j)^2 \qquad (7.8)\end{aligned}$$

following the definition of $\lambda_2(\mathbf{L}) = \min_{\|\mathbf{x}\|_2=1, \mathbf{x}\perp\mathbf{1}} \mathbf{x}^T\mathbf{L}\mathbf{x}$ in (7.2). It is worth mentioning that for any connected graph G there exists at least one edge removal such that the inequality $\lambda_2(\widetilde{\mathbf{L}}(i,j)) < \lambda_2(\mathbf{L})$ holds, otherwise $y_i = y_j$ for all $i,j \in \mathcal{V}$ and this violates the constraints that $\|\mathbf{y}\|_2 = 1$ and $\sum_{i=1}^n y_i = 0$. Consequently, there exists at least one edge removal that leads to a decrease in algebraic connectivity.

Similarly, when we remove a subset of edges $\mathcal{E}_\mathcal{R} \subset \mathcal{E}$ from G, where $|\mathcal{E}_\mathcal{R}| = h$. We obtain an upper bound

$$\lambda_2(\widetilde{\mathbf{L}}(\mathcal{E}_\mathcal{R})) \leq \lambda_2(\mathbf{L}) - \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} (y_i - y_j)^2. \qquad (7.9)$$

Correspondingly, we define the local Fiedler vector edge centrality as

$$\text{edge-LFVC}(i,j) = (y_i - y_j)^2. \qquad (7.10)$$

Edge-LFVC is a measure of centrality as it associates the sensitivity of algebraic connectivity to edge removal as described in (7.9). The top h edge removals which lead to the largest decrease on the right hand side of (7.9) are the h edges with the highest edge-LFVC.

7.3.2 Node-LFVC

When a node $i \in \mathcal{V}$ is removed from G, all the edges attached to i will also be removed from G. Similar to (7.8), the resulting graph Laplacian matrix $\widetilde{\mathbf{L}}(i)$ can be regarded as a rank d_i matrix perturbation of $\mathbf{L}$. Since $\mathbf{L} - \widetilde{\mathbf{L}}(i) = \sum_{j \in \mathcal{N}_i} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T$, where $\mathcal{N}_i$ is the set of neighboring nodes of node i, we obtain an upper bound

$$\begin{aligned} \lambda_2(\widetilde{\mathbf{L}}(i)) &\leq \mathbf{y}^T \widetilde{\mathbf{L}}(i) \mathbf{y} \\ &= \mathbf{y}^T (\mathbf{L} + \widetilde{\mathbf{L}}(i) - \mathbf{L}) y \\ &= \lambda_2(\mathbf{L}) - \sum_{j \in \mathcal{N}_i} (y_i - y_j)^2. \end{aligned} \tag{7.11}$$

Similar to edge removal, for any connected graph, there exists at least one node removal that leads to a decrease in algebraic connectivity.

If a subset of nodes $\mathcal{R} \subset \mathcal{V}$ are removed from G, where $|\mathcal{R}| = q$, then

$$\mathbf{L} - \widetilde{\mathbf{L}}(\mathcal{R}) = \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T - \frac{1}{2} \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{R}} \mathbf{A}_{ij} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T, \tag{7.12}$$

where the last term accounts for the edges that are attached to the removed nodes at both ends. Consequently, similar to (7.9), we obtain an upper bound for multiple node removals

$$\lambda_2(\widetilde{\mathbf{L}}(\mathcal{R})) \leq \lambda_2(\mathbf{L}) - \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} (y_i - y_j)^2 + \frac{1}{2} \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{R}} \mathbf{A}_{ij} (y_i - y_j)^2. \tag{7.13}$$

We define the local Fiedler vector node centrality as

$$\text{node-LFVC}(i) = \sum_{j \in \mathcal{N}_i} (y_i - y_j)^2, \tag{7.14}$$

which is the sum of the square terms of the Fiedler vector elementwise differences between node i and its neighboring nodes, and it is also the sum of edge-LFVC of

$i's$ neighboring nodes. From (7.11) and (7.13), node-LFVC is associated with the upper bound on the resulting algebraic connectivity for node removal when $|\mathcal{R}| = 1$. A node with higher centrality implies that it plays a more important role in the network connectivity structure.

7.3.3 Monotonic submodularity and greedy removals

Fixing $|\mathcal{R}| = q$, consider the problem of finding the optimal node removal set $\mathcal{R}_{\text{opt}}$ that maximizes the decrease in the upper bound on algebraic connectivity in (7.13). The computational complexity of this batch removal problem is of combinatorial order $\binom{n}{q}$. Here we show that the greedy LFVC removal procedure, shown in Algorithm 7.1, and whose computation is only linear in n, has bounded performance loss relative to the combinatorial algorithm in terms of achieving, within a multiplicative constant $(1 - 1/e)$, an upper bound on algebraic connectivity, where e is Euler's constant. Let

$$f(\mathcal{R}) = \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} (y_i - y_j)^2 - \frac{1}{2} \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{R}} \mathbf{A}_{ij} (y_i - y_j)^2 \tag{7.15}$$

and recall from (7.13) that $\lambda_2(\widetilde{\mathbf{L}}(\mathcal{R})) \leq \lambda_2(\mathbf{L}) - f(\mathcal{R})$. Note that when $|\mathcal{R}| = 1$, $f(\mathcal{R})$ reduces to node-LFVC as $\mathbf{A}_{ii} = 0$. The following lemma provides the cornerstone to **Theorem** 7.4.

Lemma 7.3. *The function $f(\mathcal{R})$ in (7.15) is equal to*

$$f(\mathcal{R}) = \frac{1}{2} \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{N}_i} (y_i - y_j)^2 + \frac{1}{2} \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{V}/\mathcal{R}} \mathbf{A}_{ij} (y_i - y_j)^2.$$

Furthermore, $f(\mathcal{R}) \geq 0$ and $f(\varnothing) = 0$, where $\varnothing$ is the empty set.

Proof. The proof can be found in Appendix F.3. □

The following theorem establishes monotonic submodularity [86] of $f(\mathcal{R})$. Monotonicity means $f(\mathcal{R})$ is a non-decreasing function: for any subsets $\mathcal{R}_1, \mathcal{R}_2$ of the node set $\mathcal{V}$ satisfying $\mathcal{R}_1 \subset \mathcal{R}_2$ we have $f(\mathcal{R}_1) \leq f(\mathcal{R}_2)$. Submodularity means $f(\mathcal{R})$ has diminishing gain: for any $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \mathcal{V}$ and $v \in \mathcal{V} \setminus \mathcal{R}_2$ the discrete derivative $\Delta f(v|\mathcal{R}) = f(\mathcal{R} \cup \{v\}) - f(\mathcal{R})$ satisfies $\Delta f(v|\mathcal{R}_2) \leq \Delta f(v|\mathcal{R}_1)$. As will be seen below (see (F.9)), this implies that greedy node removal based on LFVC is almost as effective as the combinatorially complex batch algorithm that searches over all possible removal sets $\mathcal{R}$.

Theorem 7.4. *$f(\mathcal{R})$ is a monotonic submodular set function.*

Proof. The proof can be found in Appendix F.4. □

Based on **Theorem** 7.4, we propose a greedy node-LFVC based node removal algorithm for deep community detection as summarized in Algorithm 7.1. Algorithm 7.1 yields an adjacency matrix $\hat{\mathbf{A}}$ that corresponds to the remaining edges after node removal. In addition to a list of the q removed nodes, the deep communities are defined by the non-singleton connected components in $\hat{\mathbf{A}}$ supplemented by the nodes that were removed, where the membership of these nodes is defined by the connected components in $\hat{\mathbf{A}}$ to which they connect. More specifically, if $\hat{S} = (\mathcal{V}_{\hat{S}}, \mathcal{E}_{\hat{S}})$ denotes one of these non-singleton connected components, the set $\mathcal{V}_{\hat{S}} \cup \left\{ i \in \mathcal{R} : \mathbf{A}_{ij} = 1 \text{ for some } j \in \hat{S} \right\}$ is called a deep community. This definition means that some of the removed nodes may be shared by more than one deep community. The following theorem shows that this greedy algorithm has bounded performance loss no worse than 0.63 as compared with the optimal combinatorial batch removal strategy.

Theorem 7.5. *Fix the target number of nodes to be removed as $|\mathcal{R}| = q$. Let $\mathcal{R}_{\text{opt}}$ be the optimal node removal set that maximizes $f(\mathcal{R})$ and let $\mathcal{R}_k$ be the greedy node removal set at the k-th stage of Algorithm 7.1, where $|\mathcal{R}_k| = k$. Then*

$$f(\mathcal{R}_{\text{opt}}) - f(\mathcal{R}_q) \leq \left(1 - \frac{1}{q}\right)^q f(\mathcal{R}_{\text{opt}}) \leq \frac{1}{e} f(\mathcal{R}_{\text{opt}}).$$

Algorithm 7.1 Deep Community Detection by greedy node-LFVC

```
**Input:** Adjacency matrix $\mathbf{A}$, number of removed nodes $q$
**Output:** Deep communities
$\mathcal{R} = \varnothing$
for $i = 1$ to $q$ do
    Find the largest connected component
    Compute the corresponding Fiedler vector $\mathbf{y}$
    Find $i^* = \arg\max_i \sum_{j \in \mathcal{N}_i} (y_i - y_j)^2$
    $\mathcal{R} = \mathcal{R} \cup i^*$
    Remove $i^*$ and its edges from the graph
end for
Find $\hat{S}$, one of the non-singleton connected components.
The set $\mathcal{V}_{\hat{S}} \cup \left\{ i \in \mathcal{R} : \mathbf{A}_{ij} = 1 \text{ for some } j \in \hat{S} \right\}$ is a deep community.
```

Furthermore,

$$\lambda_2(\widetilde{\mathbf{L}}(\mathcal{R}_q)) \leq \lambda_2(\mathbf{L}) - \left(1 - e^{-1}\right) f(\mathcal{R}_{\text{opt}}). \tag{7.16}$$

Proof. The proof can be found in Appendix F.5. □

The submodularity of the function f implies that after q greedy iterations the performance loss is within a factor $1/e$ of optimal batch removal [103]. In other words, when removing $\mathcal{R}_q$ from G, the algebraic connectivity is guaranteed to decrease by at least $(1 - e^{-1})f(\mathcal{R}_{\text{opt}})$ of its original value. Consequently, identifying the top q nodes affecting algebraic connectivity can be regarded as a monotonic submodular set function maximization problem, and the greedy algorithm can be applied iteratively to remove the node with the highest node-LFVC. Similarly, we can use edge-LFVC to detect deep communities by successively remove the edge with the highest edge-LFVC from the graph, and it is easy to show that the term $\sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} (y_i - y_j)^2$ in (7.9) is a monotonic submodular set function of the edge removal set $\mathcal{E}_{\mathcal{R}}$.

7.4 Deep Community Detection on Real-world Datasets

In this section, we use the proposed node and edge centrality measures to perform deep community detection on several datasets collected from real-world networks. In the implementations of the community detection methods below, the number of removed nodes or edges is a user-specified free parameter. For LFVC (Algorithm 7.1) this parameter can be selected based on the bounds established in **Theorem** 7.1. We define h the number of edge removals, q the number of node removals and g the number of deep communities. The results are compared with the modularity method and other node centralities discussed in Sec. 1.5. For data visualization, vertex shapes and colors represent different communities, and edges attached to the removed nodes are retained in the figures in comparison with other methods. Nodes with cross labels (black X labels) are singleton survivors that do not belong to any deep communities using LFVC (Algorithm 7.1).

7.4.1 Dolphin social network

It is shown in [96] that there are tight social structures in dolphin populations. Most dolphins interact with other dolphins of the same group and only a few dolphins can interact with dolphins from different groups. In terms of the proposed LFVC algorithm, these latter Dolphins introduce ”noisy” edges connecting the two communities. Figure 7.2 shows that they can therefore be detected by LFVC. In Fig. 7.2 we compare the results of separating 62 dolphins into two communities as proposed in [96]. For this dataset, community detections based on modularity, edge-LFVC and node-LFVC have high concordance on the community structures. To partition the graph into two communities, we need to remove 6 edges based on edge-LFVC or remove 4 nodes based on node-LFVC. The four dolphins that are able to communicate between these two communities are further identified by node-LFVC.

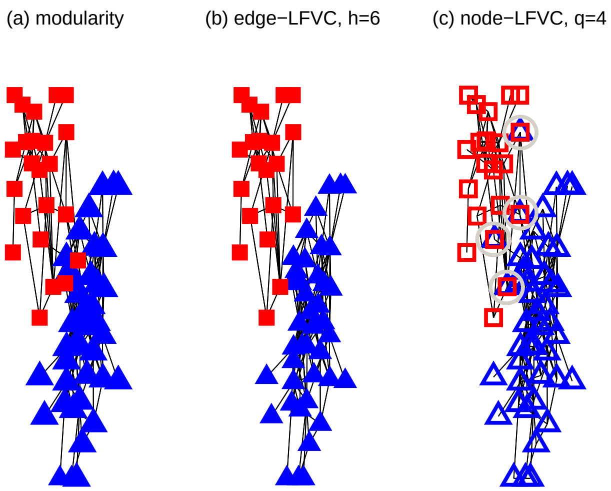


Figure 7.2: Dolphin social network [96] with $n = 62$ nodes and $m = 159$ edges. (a) The modularity method. (b) Edge-LFVC community detection with $h = 6$ edge removals. (c) Node-LFVC community detection with $q = 4$ node removals. Using node-LFVC, we are able to identify the four dolphins that interact with two groups as marked by nodes in gray circles. This algorithm, defined by Algorithm 7.1, detects that these four nodes are members of the two communities. The result of spectral clustering is shown in the supplementary file[1]. Spectral clustering results in the same discovered communities as the proposed edge-LFVC community detection method. However, unlike the proposed node-LFVC method it does not explicitly identify the four mixed membership dolphins that connect the two communities.

7.4.2 Zachary's karate club

Zachary's karate club [173] is a widely used example for social network analysis, which contains interactions among 34 karate students. Based on the student activities, Zachary determines the ground-truth community structure for $g = 2$, which coincides with the result of the modularity method in Fig. 7.3 (a). However, the visualization indicates that there are some deep communities embedded in these two communities, such as the five-node community in the upper left corner. Indeed, the modularity will keep increaseing if we further divide communities into 3 and 4 small communities as shown in Fig. 7.4 (b) and (c), respectively.

As shown in Fig. 7.4 (a), using edge-LFVC, the five-node community in the left upper corner is revealed when we partition the graph into two connected subgraphs.

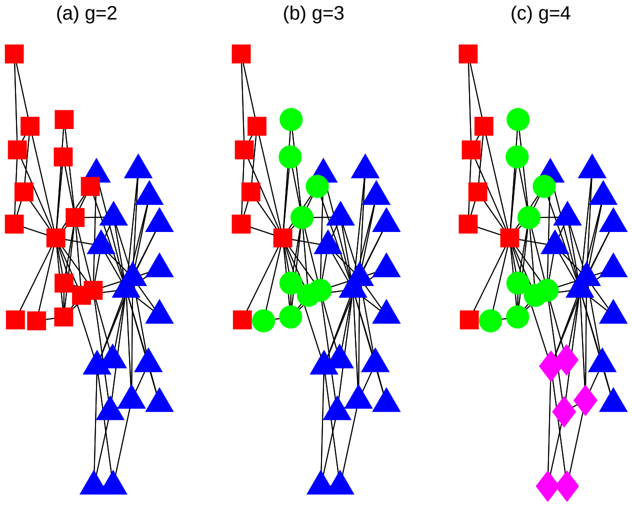


Figure 7.3: The modularity method on Zachary's karate club [173] with $n = 34$ nodes and $m = 78$ edges.

In Fig. 7.4 (b), three communities are revealed and the only node with a single acquaintance is excluded from any deep community. Excluding this node makes the community structure more tightly connected compared with Fig. 7.3 (b). For $g = 4$, the community structure in Fig. 7.4 (c) much resembles Fig. 7.3 (c) except that we exclude the node having a single acquaintance.

Using node-LFVC, we are able to extract important communities and key members as shown in Fig. 7.5. For $g = 2$, only one node removal is required to partition the graph into two connected subgraphs, which implies that this node is common to the two communities according to the proposed Algorithm 7.1. For $g = 3$, two deep communities (green circle and blue triangle) are discovered in the largest community (the blue triangle community in Fig. 7.5 (a)), where these two deep communities have dense internal connections compared with the external connections to other members in the largest community. These discovered deep communities are important communities embedded in the network since they play an important role in connecting the singleton survivors indicated by black X labels. Similar observations hold for $g = 4$ in Fig. 7.5 (c).

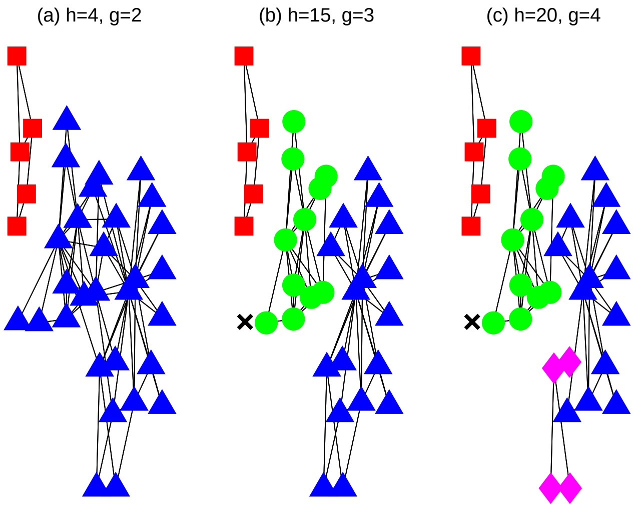


Figure 7.4: Edge-LFVC community detection on Zachary's karate club [173] with $n = 34$ nodes and $m = 78$ edges. For $g = 3$ and 4, the only node with a single acquaintance is excluded from any deep community.

7.4.3 Coauthorship among network scientists

We next examine the coauthorship network studied by Newman [106]. Nodes represent network scientists and edges represent the existence of coauthorship. Multiple memberships are expected to occur in this dataset since a network scientist may collaborate with other network scientists across different regions all the while having many collaborations with his/her colleagues and students at the same institution. As a result, one would expect, as implemented by Algorithm 7.1, node-LFVC to be advantageous for identifying authors who with multiple memberships and detecting deep communities.

As shown in Fig. 7.6, the first node with the highest node-LFVC is Yamir Moreno, who is a network scientist in Spain but has many collaborators outside Spain. The local (two-hop) coauthorship network of Yamir Moreno is shown in Fig. 7.6. The red square community represents the network scientists in Spain and Europe, whereas the blue triangle community represents the rest of the network scientists.

After removing Yamir Moreno from the network, the node with the highest node-

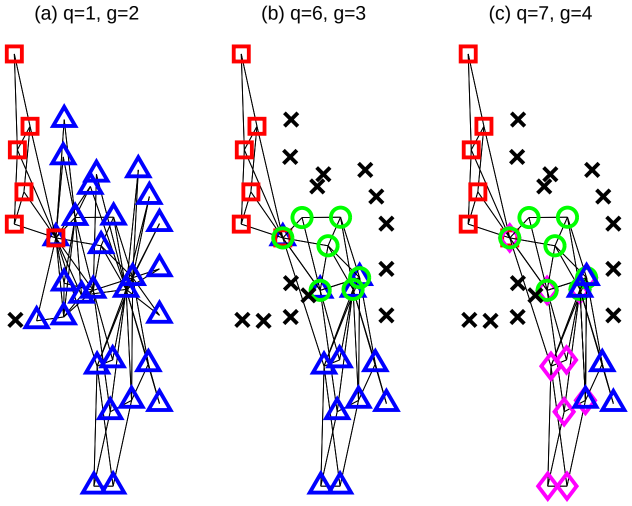


Figure 7.5: Node-LFVC community detection on Zachary's karate club [173] with $n = 34$ nodes and $m = 78$ edges. Important communities and key members are discovered using node-LFVC. This also demonstrates how the singleton survivors (nodes with black X labels) interact through the deep communities. The result of spectral clustering is shown in the supplementary file[1]. When $g = 4$, spectral clustering yields imbalanced communities (one community has single node).

LFVC in the remaining largest community is Mark Newman, who is associated with 5 community memberships and 3 singleton survivors as shown in Fig. 7.7. Each community can be related to certain relationship such as colleagues, students and research institutions. Notably, Lusseau is detected as a singleton survivor in the deep community detection process in Fig. 7.7. This can be explained by the fact that although Lusseau has coauthorship with Newman, his research area is primarily in zoology and he has no interactions with other network scientists in the dataset since other network scientists are mainly specialists in physics. Also note that the modularity method (gray dashed box) fails to detect these deep communities and it detects 25 out of 28 network scientists in Fig. 7.7 as one big community.

7.4.4 Last.fm online music system

Last.fm is an online music system which allows users to tag their favorite songs and artists and make friends with other users. We use the friendship dataset collected

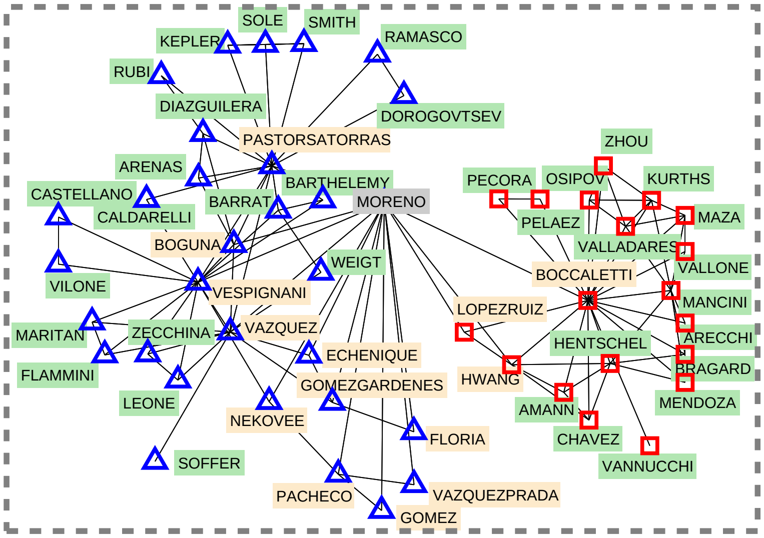


Figure 7.6: Yamir Moreno's local 2-hop coauthorship network (from part of the network of coauthorship among network scientists [106] having $n = 379$ nodes and $m = 914$ edges). Moreno has 14 coauthors (marked by light orange color) and his coauthors have 35 coauthors. The modularity method [106] detects that Moreno is a member of only one large community (dashed box in gray). The proposed LFVC method detects Moreno as belonging to two separate communities indicated by red and blue nodes, respectively.

in [21] for deep community detection based on node-LFVC and the other centralities introduced in Sec. 1.5. Two quantities, the normalized largest community size and the number of discovered communities with respect to node removals, are used to evaluate the performance of community detection when different node centralities are applied. These two quantities reflect the effectiveness of graph partitioning. The number of removed nodes is the number of stages for performing deep community detection and removing more nodes reveals more deep communities and key members in the network.

As shown in Fig. 7.8 (a), the normalized largest community size decays linearly with respect to the number of node removals. Among all node centralities, node-LFVC has the steepest decaying rate. Furthermore, using node-LFVC discovers more deep communities, as shown in Fig. 7.8 (b) during the first 50 node removals. The only

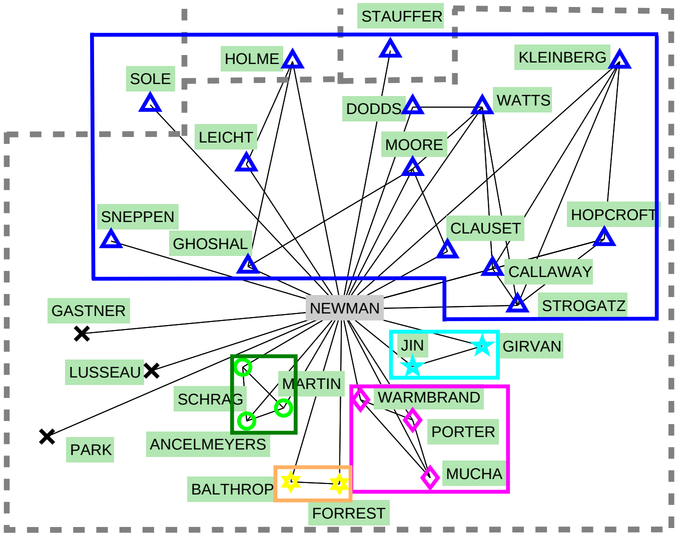


Figure 7.7: Mark Newman's local 1-hop coauthor network in the network scientist coauthorship graph [106]. The proposed LFVC method detects Newman as belonging to 5 communities (marked by different vertex shapes and colors in solid boxes) and being associated with 3 singleton survivors (marked by black X label). Notably, Lusseau is detected as singleton survivor since his research area is primarily in zoology. As shown in gray dashed box, the modularity method [106] detects 25 out of 28 scholars as being in a single community, and the top left 3 scholars as belonging to 3 different communities.

node centrality that is comparable to node-LFVC is betweenness centrality.

To validate the effectiveness of deep community detection, we use the user-artists dataset in [21] to compute the listening similarity in each discovered community. The dataset contains 17632 artists and records the number of times each user has listened to an artist. Let $\mathbf{w}_i$ be a 17632-by-1 vector with its j-th entry being the number of times the i-th user has listened to the j-th artist. The residual community similarity (RCS) is defined as the sum of cosine similarity between each user in the same community excluding the nodes that have been removed and the singleton survivors. The residual community similarity of a deep community C_k is defined as

$$\text{RCS}(C_k) = \sum_{i \in C_k, i \notin \mathcal{R}} \sum_{j \in C_k, j > i, j \notin \mathcal{R}} \frac{\mathbf{w}_i^T \mathbf{w}_j}{\|\mathbf{w}_i\|_2 \|\mathbf{w}_i\|_2}. \tag{7.17}$$

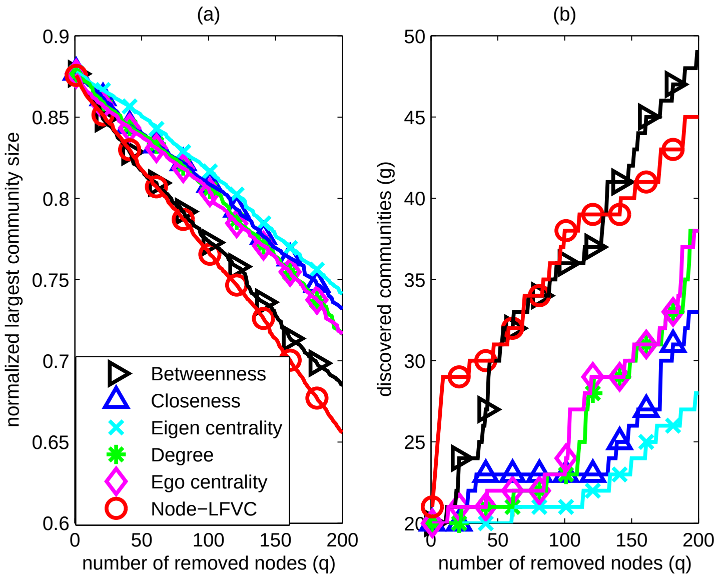


Figure 7.8: Friendship in Last.fm online music system [21] with $n = 1843$ nodes and $m = 12668$ edges. (a) Normalized largest community size decreases in the number of node removals at different rates under different node centralities. (b) Discovered communities with respect to node removals using different node centralities. Node-LFVC outperforms other node centralities in terms of minimizing the largest community size, and while being capable of detecting more communities in the network for the first 50 removals.

The residual sum of community similarity (RSCS) is defined as the sum of RCS of each discovered community. That is,

$$\text{RSCS} = \sum_{k=1}^{g} \text{RCS}(C_k). \tag{7.18}$$

As shown in Fig. 7.9, the residual sum of community similarity based on node-LFVC is larger than that for other centralities. This suggests that node removals based on node-LFVC can best detect friendship communities that share common interest in artists. Note that although betweenness may detect more communities in Fig. 7.8 (b), Fig. 7.9 shows that the residual sum of community listening similarity based on betweenness is smaller than that based on node-LFVC, which indicate that node-LFVC reveals more accurate community structure than betweenness. The residual sum of community similarity decreases with respect to the number of discov-

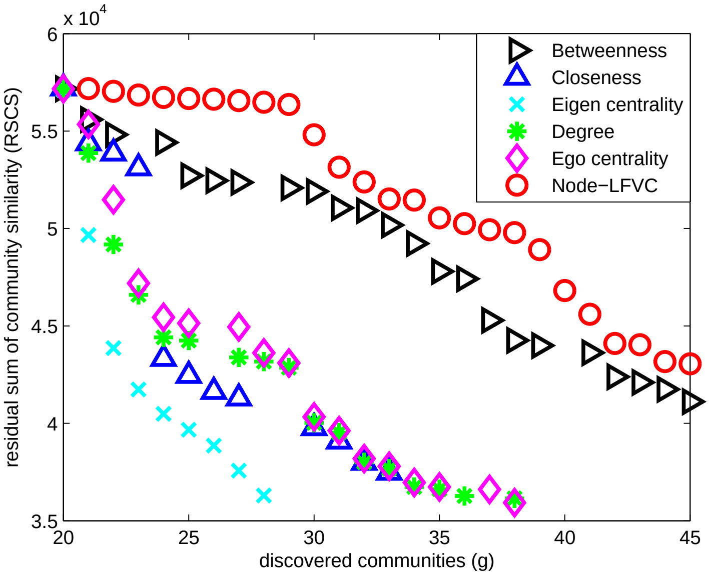


Figure 7.9: Residual sum of community similarity (RSCS) in Last.fm network. The residual sum of community similarity based on node-LFVC outperforms other centralities, which indicates that node removals based on node-LFVC can best detect deep communities that share common interest in artists.

ered communities due to the fact that the removed nodes and singleton survivors are excluded for similarity computation.

CHAPTER VIII

Identifying Influential Links for Event Propagation on Twitter: A Network of Networks Approach

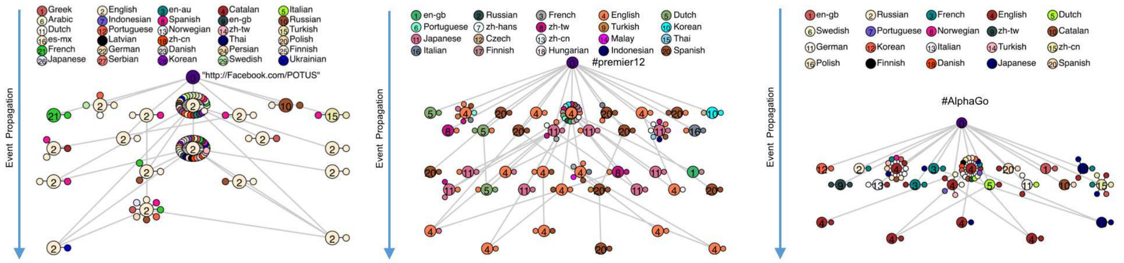
Patterns of event propagation in online social networks are closely related to the modeling and analysis of information dissemination over certain networks and physical systems. Examples include epidemic processes in contact networks [113, 120], information diffusion in social networks and social media [41, 49, 82, 83, 134, 172], and malware propagation in technological networks [37, 39, 60, 182], among others. This chapter studies the importance of follower links for event propagation on Twitter, where the importance of a follower link is associated with the consequence of its removal to event propagation. Three recent event propagation traces are collected with the user languages being used to identify the Network of Networks (NoN) structure embedded in the Twitter follower networks.

Specifically, this chapter exploits the network structure embedded in online social networks for identifying influential links for event propagation. Specifically, we use Twitter follower networks to study and develop an effective link score function that reflects the importance of a follower link in event propagation. An event on a Twitter follower network can be a uniform resource locator (URL) of a web address or a hashtag in a tweet. A follower who has seen a tweet and decided (not) to retweet the event is called a retweeter (non-retweeter). A typical example of event propagation on Twitter is the announcement of the discovery of a Higgs boson-like particle in July 2012 [42]. Given a Twitter follower network, our proposed method effectively identifies important follower links affecting event propagation based on the network connectivity structure without requiring prior knowledge such as where the event is originally posted and how the event is retweeted.

We model event propagation using an iterative state equation, and then propose a Left Eigenvector Score (LES) for each follower link. We show that LES is able to identify influential follower links for event propagation in the sense that the removal of those links is effective in reducing the event propagation. Although our method requires only the information of the network's connectivity structure, it can be easily extended to incorporate certain additional user information to further improve the effectiveness of the proposed method. Specifically, we utilize the Network of Networks (NoN) structure in Twitter follower networks as additional user information. The NoN model is a general approach for characterizing a network at different scales. A large-scale network is often composed of several sub-networks, and the interconnectivity and interdependency between these sub-networks are known to be crucial to information dissemination and network robustness [19, 61, 109, 119, 133, 142].

To validate the effectiveness of LES and the NoN structure of event propagation on Twitter, we collect three recent event propagation traces on Twitter using the Application Programming Interface (API)[1] provided by Twitter for public data retrieval, which in turn offers new platforms for tracking and collecting real-world event propagation traces on Twitter at large scales. The user language is used to identify the sub-networks within the Twitter follower network under consideration. We find that the between-network links play an important role in event propagation, as they account for information dissemination from one user language to another. Experimental results demonstrate that link removals based on LES can successfully reduces

[1] Twitter REST APIs. Available at https://dev.twitter.com/rest/public



(a) Obama FB (b) Premier 12 (c) AlphaGo

Figure 8.1: The three collected retweeter networks with user language identifying the Network of Networks (NoN) structure. A retweeter is represented by a node with language setting denoted by its color/number. An edge between two nodes indicates that the event is retweeted from one node to another. The node 0 represents the virtual source of event propagation. For succinct representation, we grouped all the same-language leaf retweeters of a single node into a small node. It is observed that an event is first disseminated by some seed nodes, and other nodes tend to retweet the event from a same-language node.

event propagation in real-world traces, especially when the between-network follower links are used for LES calculation.

8.1 The NoN Structure of Event Propagation on Twitter

To study event propagation, we collected the traces of three recent events on Twitter during a period of two weeks through the Twitter API. These events include URLs and hashtags specified as follows[2].

- **Obama FB:** A URL that links to U.S. President Obama's personal Facebook page created in 2015.
- **Premier 12:** A hashtag of an international baseball tournament in 2015.
- **AlphaGo:** A hashtag about a board game algorithm defeating a European Go champion in 2016 [146].

[2] More details about the collected event propagation traces on Twitter are given in Appendix G.1. The collected datasets can be downloaded from https://sites.google.com/site/pinyuchenpage/datasets

We also collected each user's language setting on Twitter, which is used as the network identity. The source of an event need not to be unique. For example, the same URL can be independently posted by some users and then be retweeted by their followers. Fig. 8.1 displays the Network of Networks (NoN) structure in the retweeter network of the aforementioned events. It is observed that the propagation patterns of these events share some common features. (i) For each event, there are some hub users such that their posts are retweeted by many other users. For the Obama FB event, one hub user is President Obama's personal Twitter account, and another hub user is White House's Twitter account. For the Premier 12 event, one hub user is the tournament organizer's official Twitter account. For the AlphaGo event, one hub user is Google's Twitter account. (ii) The events are originally posted by some "seed users" of different languages, and other users tend to retweet the event from a user of the same language. Take Premier 12 as an example, the tweets regarding Premier 12 are first tweeted by some seed users of different languages, including Dutch, English, Spanish, Korean, zh-TW and Italian. Then most of the tweets are retweeted by users of the same language.

8.2 Methodology

8.2.1 Event propagation model

Consider a Twitter follower network consisting of n users and m follower links. Let $\mathbf{A}$ be an $n \times n$ binary adjacency matrix representing the follower relationship in the network, where its entry of the i-th row and the j-th column $[\mathbf{A}]_{ij} = 1$ if user i follows user j, and $[\mathbf{A}]_{ij} = 0$ otherwise. We divide the time period of event propagation into F non-overlapping frames, and let $\mathbf{A}_t$ be an $n \times n$ binary adjacency matrix indicating the follower links that have been activated for event propagation during the t-th time frame, $t = 1, 2, \ldots F$. In other words, $[\mathbf{A}]_{ij} = 1$ indicates that user i follows user j, while $[\mathbf{A}_t]_{ij} = 1$ indicates that user i retweets user j during the t-th time frame. Let $\mathbf{r}_t$ be an n-dimensional binary vector indicating the event propagation status of every user, where $\mathbf{r}_t$'s i-th entry $[\mathbf{r}_t]_i = 1$ if the event has ever been posted or retweeted by the i-th user since the beginning to the t-th time frame, and $[\mathbf{r}_t]_i = 0$ otherwise. In addition, let $\mathbf{r}_0$ be a binary vector such that its nonzero entries indicate the set of users who first post the event. Then the event propagation model can be written as an iterative state equation

$$\mathbf{r}_{t+1} = \mathbb{T}\left(\mathbf{r}_t + \mathbb{T}\left(\mathbf{A}_{t+1}^T \mathbf{r}_t\right)\right), \ \forall\, t = 0, 1, 2, \ldots, F-1, \tag{8.1}$$

where $\mathbf{A}_{t+1}^T$ is the matrix transpose of $\mathbf{A}_{t+1}$, and $\mathbb{T}(\cdot)$ is an entry-wise threshold function defined as $[\mathbb{T}(\mathbf{x})]_i = 1$ if $[\mathbf{x}]_i > 1$ and $[\mathbb{T}(\mathbf{x})]_i = [\mathbf{x}]_i$ if $0 \leq [\mathbf{x}]_i \leq 1$, for any nonnegative vector $\mathbf{x}$. The term $\mathbb{T}\left(\mathbf{A}_{t+1}^T \mathbf{r}_t\right)$ can be viewed as the increment vector for event propagation in the $t+1$-th time frame. The derivation of (8.1) is given in Appendix G.2.

The event propagation model in (8.1) can be easily adapted to incorporate the NoN structure of a Twitter follower network. Let $\mathbf{A}^{\text{bet}}$ and $\mathbf{A}^{\text{wit}}$ denote the adjacency matrix of the between-network and within-network follower links, respectively. The event propagation model can be rewritten as

$$\mathbf{r}_{t+1} = \mathbb{T}\left(\mathbf{r}_t + \mathbb{T}\left({\mathbf{A}_{t+1}^{\text{bet}}}^T \mathbf{r}_t\right) + \mathbb{T}\left({\mathbf{A}_{t+1}^{\text{wit}}}^T \mathbf{r}_t\right)\right) \tag{8.2}$$

for all $t = 0, 1, 2, \ldots, F-1$. The matrices $\mathbf{A}_t^{\text{bet}}$ and $\mathbf{A}_t^{\text{wit}}$ are defined similarly as $\mathbf{A}_t$ such that $\mathbf{A}_t = \mathbf{A}_t^{\text{bet}} + \mathbf{A}_t^{\text{wit}}$. The terms $\mathbb{T}\left({\mathbf{A}_{t+1}^{\text{bet}}}^T \mathbf{r}_t\right)$ and $\mathbb{T}\left({\mathbf{A}_{t+1}^{\text{wit}}}^T \mathbf{r}_t\right)$ in (8.2) account for the event propagation increment caused by between-network and within-network follower links, respectively.

8.2.2 Surrogate function for event propagation

Since we are interested in investigating the effect of link removals on a Twitter follower network prior to actual event propagation, in practice only the adjacency matrix $\mathbf{A}$ of the Twitter follower network is modeled as known, whereas the event propagation status vector $\mathbf{r}_t$ and the adjacency matrix $\mathbf{A}_t$ affecting actual event propagation are unknown. Nonetheless, we will show that the largest eigenvalue of $\mathbf{A}$, denoted by $\lambda_{\max}(\mathbf{A})$, can be used as a surrogate function for the containment of event propagation, as it is associated with an upper bound on the increment of event propagation. In addition, $\lambda_{\max}(\mathbf{A})$ is known to be related to the information dissemination threshold of some parametric epidemic models [120, 130].

Specifically, let $\|\mathbf{x}\|_0$ denote the number of nonzero entries of an n-dimensional vector $\mathbf{x}$, which is also known as the ℓ_0 norm or the sparsity level of $\mathbf{x}$. Under the sparsity assumption that $\|\mathbf{r}_F\|_0 \leq s$, where $s \leq n$ is a trivial upper bound on s, we can obtain a surrogate function of the increment $\|\mathbb{T}\left(\mathbf{A}_{t+1}^T \mathbf{r}_t\right)\|_0$ in terms of $\lambda_{\max}(\mathbf{A})$, s and n, which is

$$\left\|\mathbb{T}\left(\mathbf{A}_{t+1}^T \mathbf{r}_t\right)\right\|_0 \leq s \cdot \lambda_{\max}(\mathbf{A}) + \sqrt{ns} \tag{8.3}$$

for all $t = 0, 1, 2, \ldots, F-1$. The derivation is given in Appendix G.3. It is clear from (8.3) that minimizing the largest eigenvalue $\lambda_{\max}(\mathbf{A})$ of the adjacency matrix $\mathbf{A}$ can be effective in containing event propagation, since $\lambda_{\max}(\mathbf{A})$ is associated with an upper bound on the event propagation increment $\|\mathbb{T}\left(\mathbf{A}_{t+1}^T \mathbf{r}_t\right)\|_0$ for each iteration in t.

Applying the results in (8.3) to the event propagation model with NoN structure in (8.2), we can obtain upper bounds on the increments $\mathbb{T}\left({\mathbf{A}_{t+1}^{\text{bet}}}^T \mathbf{r}_t\right)$ and $\mathbb{T}\left({\mathbf{A}_{t+1}^{\text{wit}}}^T \mathbf{r}_t\right)$ associated with between-network and within-network follower links in terms of $\lambda_{\max}(\mathbf{A}^{\text{bet}})$

and $\lambda_{\max}(\mathbf{A}^{\text{wit}})$, which are

$$\left\| \mathbb{T}\left({\mathbf{A}_{t+1}^{\text{bet}}}^T \mathbf{r}_t \right) \right\|_0 \leq s \cdot \lambda_{\max}(\mathbf{A}^{\text{bet}}) + \sqrt{ns}; \tag{8.4}$$

$$\left\| \mathbb{T}\left({\mathbf{A}_{t+1}^{\text{wit}}}^T \mathbf{r}_t \right) \right\|_0 \leq s \cdot \lambda_{\max}(\mathbf{A}^{\text{wit}}) + \sqrt{ns}. \tag{8.5}$$

8.2.3 LES: left eigenvector score

Since in Sec. 8.2.2 the largest eigenvalue of the adjacency matrix of a Twitter follower network, $\lambda_{\max}(\mathbf{A})$, is shown to be an important factor affecting event propagation, we propose a score function on follower links such that link removals based on the score function of decreasing order become an effective reducer in the largest eigenvalue. Specifically, we use the left eigenvector $\mathbf{y}$ of the adjacency matrix $\mathbf{A}$ to define a score for each follower link for evaluating every follower link's importance in event propagation. By the Perron-Frobenius theorem [76], the largest eigenvalue of an adjacency matrix is always real and nonnegative, and its associated left eigenvector $\mathbf{y}$ has nonnegative entries and unit Euclidean norm, i.e., $[\mathbf{y}]_i \geq 0$ for all i and $(\sum_i [\mathbf{y}]_i^2)^{1/2} = 1$. Since $\mathbf{y}$ satisfies the eigenfunction $\mathbf{A}^T\mathbf{y} = \lambda_{\max}(\mathbf{A})\mathbf{y}$, it can be viewed as the vector of eigenvector centrality of each user based on every user's follower connectivity patten in the Twitter follower network, for which the eigenvector centrality is a measure of importance among influential nodes in a network [107].

Let (i, j) denote a follower link in the Twitter follower network representing the relation that user i follows user j. The follower link score we propose for assessing the influence in event propagation, which we call the Left Eigenvector Score (LES), is defined as

$$\text{LES}(i, j) = [\mathbf{y}]_i \cdot [\mathbf{y}]_j. \tag{8.6}$$

Since $\mathbf{y}$ is the vector of eigenvector centrality based on each user's follower connectivity pattern, high LES for a follower link (i, j) means that the followers of both user i and user j play an important role in the Twitter follower network, and hence the follower link (i, j) is crucial to event propagation.

Moreover, we show that removing the follower links of top LES can be effective in reducing the largest eigenvalue $\lambda_{\max}(\mathbf{A})$, and hence is able to contain event propagation increment according to (8.3). Let $\mathcal{E}_{\mathcal{R}}$ denote a subset of follower links in a Twitter follower network such that $(i, j) \in \mathcal{E}_{\mathcal{R}}$ if the follower link (i, j) will be removed from the Twitter follower network. For any follower link removal set $\mathcal{E}_{\mathcal{R}}$ with cardinality $|\mathcal{E}_{\mathcal{R}}| = q \geq 1$, let $\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})$ be the adjacency matrix after removing the follower links in $\mathcal{E}_{\mathcal{R}}$ from the Twitter follower network. If $\sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{y}]_i [\mathbf{y}]_j > 0$, then

$$\lambda_{\max}(\mathbf{A}) - \sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{y}]_i [\mathbf{y}]_j \leq \lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})); \tag{8.7}$$

$$\lambda_{\max}(\mathbf{A}) - c \cdot \sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{y}]_i [\mathbf{y}]_j \geq \lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})), \tag{8.8}$$

where $c = \frac{\epsilon}{q}$, $\epsilon = \sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} [\widetilde{\mathbf{y}}]_i [\widetilde{\mathbf{y}}]_j$, and $\widetilde{\mathbf{y}}$ is the left leading eigenvector of $\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})$. The proof of (8.7) and (8.8) is given in Appendix G.4. Since the number of nonzero entries in $\mathbf{A}$ is the total number of follower links m, computing the left eigenvector $\mathbf{y}$ takes $O(m)$ time by power iteration methods, and reporting the top q follower links of LES take $O(mq)$ time. Therefore, the overall computation time complexity for finding the removal set $\mathcal{E}_{\mathcal{R}}$ of cardinality q is $O(mq)$.

Similar analysis in (8.7) and (8.8) can be directly applied to the largest eigenvalues $\lambda_{\max}(\mathbf{A}^{\text{bet}})$ and $\lambda_{\max}(\mathbf{A}^{\text{wit}})$ in (8.4) and (8.5) by using their corresponding left leading eigenvectors. As a result, the proposed LES can be easily adapted to the NoN structure in the Twitter follower network.

Dataset	Retweet Event	Duration	Networks (Languages)	Users	Follower Links	Between-Network Follower Links	Within-Network Follower Links
Obama FB	http://Facebook.com/POTUS	Nov. 9th-Nov. 23rd, 2015	117	5,169,477	7,272,858	19.74%	80.26%
Premier 12	#premier12	Nov 19th-Dec. 3rd, 2015	90	7,557,534	9,702,942	22.11%	77.89%
AlphaGo	#AlphaGo	Jan 27th-Feb.10th, 2016	141	9,259,187	9,794,702	29.35%	70.65%

Table 8.1: Statistics of the collected events and Twitter follower networks

8.3 Experiments on Twitter Traces

8.3.1 Experiment setup and dataset description

To study the effect of follower link removals on event propagation, we collected three real-world event propagation traces and user languages from Twitter as described in Sec. 8.1. We also collected the users who have seen but have not retweeted the event (i.e., non-retweeters) and their user languages to form a Twitter follower network for testing the effect of link removals on event propagation. In other words, the collected Twitter follower networks include the follower connectivity structure of retweeters and non-retweeters of an event, and their user languages are used to identify the NoN structure. The statistics of the collected datasets are summarized in Table 8.1. One notable NoN feature of these Twitter follower networks is that the between-network follower links only account for a portion of roughly 20% to 30% of total follower links.

Evaluation Metric. When designing a link score function for assessing the influence in event propagation, the available information are the adjacency matrix and the NoN identities of a Twitter follower network. The actual event propagation traces are only used to compare the performance of different link scores. Specifically, we use the *event reachability* as the performance metric, which is defined as the fraction of users who can still post or retweet the events after some follower links are removed from the original Twitter follower network. The event fails to propagate further to a user's follower if the corresponding follower link has been removed. As a result, link removals that lead to lower event reachability means that these links are more influential to event propagation.

Follower Link Scores. We compare the effect of removing top q follower links on event reachability based on different link score functions, for which the score function of a follower link (i,j) takes the form

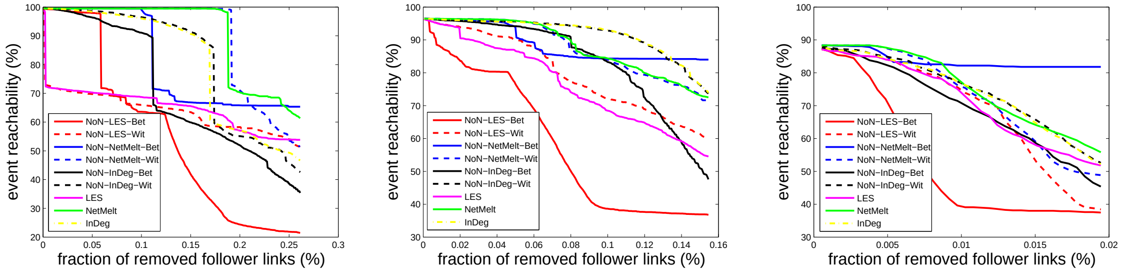
$$\text{score}(i,j) = [\mathbf{x}]_i \cdot [\widetilde{\mathbf{x}}]_j, \tag{8.9}$$

where $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ are nonnegative n-dimensional vectors [3].

The following summarizes different score functions for performance comparison, including the scenario where the network identity of every user is known and the NoN model is applied such that the between-network and within-network follower links are used separately for link score computation. The implementation and computation time complexity of returning top q follower links for different follower link score functions are given in Appendix G.5.

- **LES:** LES uses the left leading eigenvector of the adjacency matrix $\mathbf{A}$ for score computation.
- **InDeg:** InDeg uses the in-degree (number of followers) of each user for score

[3] The score function can be easily incorporated with centrality measures on users based on the Twitter follower network topology. However, since the Twitter follower network is often not a connected graph, i.e., there is not a path connecting any two users in the network, centrality measures defined on connected graphs, such as the closeness and betweenness centrality measures, cannot be used as a score function.



(a) Obama FB (b) Premier 12 (c) AlphaGo

Figure 8.2: The effect of removing top-score follower links on the collected Twitter datasets in Table 8.1. Event reachability is the fraction of users who can still post or retweet the event after some follower links are removed from the original Twitter follower network. It is observed that using the proposed LES and exploiting the NoN structure, the NoN-LES-Bet method can achieve remarkable reduction in event reachability. The results suggest that LES indeed reflects the level of importance of a follower link for event propagation, and between-network follower links are crucial to event propagation.

computation.

- **NetMelt:** NetMelt [157] is an edge removal algorithm proposed to decrease the largest eigenvalue $\lambda_{\max}(\mathbf{A})$ by using the left and right leading eigenvectors of $\mathbf{A}$.

- **NoN-LES-Bet (NoN-LES-Wit):** NoN-LES-Bet (NoN-LES-Wit) exploits the NoN structure and evaluates the score function using the left leading eigenvector of the between-network (within-network) adjacency matrix $\mathbf{A}^{\text{bet}}$ ($\mathbf{A}^{\text{wit}}$).

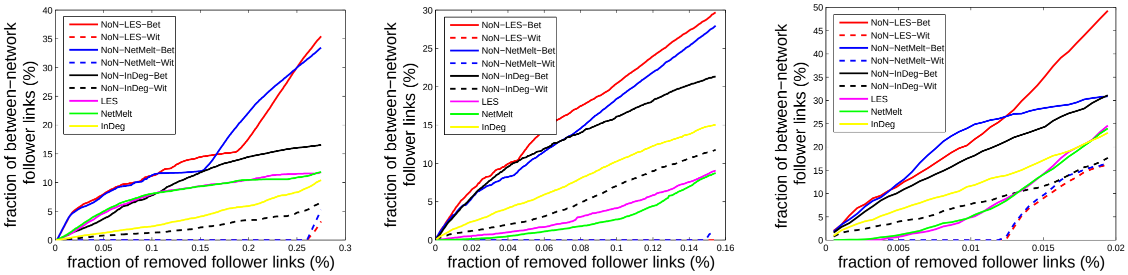
- **NoN-InDeg-Bet (NoN-InDeg-Wit):** NoN-InDeg-Bet and NoN-InDeg-Wit are extensions of the InDeg score tailored to the NoN structure.

- **NoN-NetMelt-Bet (NoN-NetMelt-Wit):** Non-NetMelt-Bet and NoN-NetMelt-Wit are NetMelt algorithms that incorporate the NoN structure.

8.3.2 Performance evaluation

Fig. 8.2 displays the event reachability with respect to different link removal methods as described in Sec. 8.3.1. Comparing to the link removal methods without using the NoN structure (LES, InDeg and NetMelt), it can be observed that incorporating the NoN structure (user languages) can further reduce event reachability. In particular, the NoN-LES-Bet method is shown to outperform other methods in the Premier 12 and AlphaGo datasets. For the Obama FB dataset, LES and NoN-LES-Wit can be more effective than other methods for the first few follower link removals. However, as the number of removals increases these two methods soon lose their appeals, and NoN-LES-Bet is shown to significantly outperform other methods. For example, if we are able to remove 0.25% of follower links from the Obama FB dataset, NoN-LES-Bet can reduce the event reachability to roughly 20%, whereas the second best method (NoN-InDeg-Bet) only reduces the event reachability to roughly 35%, which means that NoN-LES-Bet achieves at least 15% performance improvement compared with other methods. These results suggest that LES can better reflect the level of importance of a follower link for event propagation. More interestingly, the success of NoN-LES-Bet in reducing event propagation on Twitter leads to the finding that although between-network follower links only take roughly 20% to 30% of total follower links in these datasets, they are crucial to event propagation.

The effectiveness of LES in reducing event propagation can be explained by the fact it is a minimizer of an upper bound on the increment of event propagation as established in Sec. 8.2. On the contrary, in these experiments link score functions based on in-degrees or NetMelt are less effective in containing event propagation when compared with the LES-based methods, as they are not specifically designed for identifying influential links for event propagation. The finding that the LES-based methods are superior over the InDeg-based methods suggests that event propagation not only depends on the number of followers, but also on the role of each user's



(a) Obama FB (b) Premier 12 (c) AlphaGo

Figure 8.3: Fraction of between-network follower links in different link removal methods. Comparing to Fig. 8.2, although the fraction of removed between-network follower links of NoN-LES-Bet and NoN-NetMelt-Bet are similar, the follower links identified by NoN-LES-Bet are more influential in event propagation as their removals result in lower event reachability.

followers in event propagation. This is also consistent with the finding of strong/weak social ties for event propagation in online social networks [53, 152].

As shown in Fig. 8.3, we also find that although NoN-LES-Bet and NoN-NetMelt-Bet lead to similar fraction of between-network follower link removals, NoN-LES-Bet achieves lower event reachability than NoN-NetMelt-Bet as shown in Fig. 8.2. This implies that the proposed LES is indeed more effective in identifying influential follower links for event propagation.

CHAPTER IX

Assessing and Safeguarding Network Resilience to Nodal Attacks

This chapter introduces new methods for evaluating and improving resilience of network connectivity to attacks on nodes of the network. Network connectivity is evaluated using a centrality measure that quantifies sensitivity of the size of the largest connected component to node removals. Based on the local Fiedler vector centrality (LFVC) proposed in Chapter VII, a new method for improving resilience is introduced called *edge rewiring*. In terms of actions on graphs, rewiring an edge on a graph can be viewed as simultaneously inserting a new edge and removing an existing edge.

The topology of the power grid of western US states is used to illustrate the proposed method. Using the proposed centrality measure, we show that the power grid topology is especially vulnerable to nodal attacks. In particular, by using the proposed centrality measure, an attacker could reduce the largest component size by nearly a factor of two by only targeting 0.2% of the nodes. More importantly, we show that network resilience can be greatly improved via a few edge rewires without introducing additional edges in the network.

The problem of establishing resilience of network connectivity to node removals has received much recent attention [6, 26, 32, 33, 169]. Resilience is closely related to reliability of networks when a subset of nodes are inactivated. It arises in applications including service disruption in communication systems caused by router failures, and blackout in power systems caused by power station shutdowns, among others. In these applications network functionality can be disrupted by targeted attacks, e.g., denial of service (DoS) or jamming attacks, or by natural occurrences, e.g., weather-related link failures and power outages. In this chapter we introduce a new method for assessing the resilience of networks to node removals and preventive approaches to desensitize numerous connectivity attacks.

A resilient network has global connectivity and largest component size that are only minimally disrupted by limited attacks on nodes or edges. For example, a fully connected network allows communication between all pairs of nodes and its largest component is the entire set of nodes in the network. One measure of network connectivity is given by the standard graph-theoretic *k-connectivity* definition: a graph is k-connected if any set of k-1 node removals does not disconnect the graph. However, this definition does not account for the number of communication paths between nodes that are disrupted, which is more relevant to the functioning of the network. A more relevant measure of connectivity is proposed here: the minimum number of node removals necessary to reduce the size of the largest component by a fixed proportion, e.g. 10% or 50%, of its original size.

To illustrate consider a large network where one of its nodes is connected to the rest of the network by a single edge (i.e., node degree one). Removing this edge (or the adjacent node) will reduce both the number of communication paths and the largest component size by one. However, if the network is composed of two cliques of equal size connected by a single edge then removal of this edge will reduce the number of paths and the largest component size by a factor of two.

A node centrality measure is a quantity that measures the level of importance of a node in a network. The utility of centrality measures is that they can break the combinatorial bottleneck of searching through all the possible permutations and combinations of nodes that might reduce largest component size. An attack that removes nodes according to a measure of centrality, such as the one introduced in Sec. 1.5, will be referred to as a *centrality attack*. For example, the authors of [6, 32, 33, 169] study the effectiveness of degree centrality attacks, i.e., removing the largest hub nodes, as a way to reduce the size of the largest component of the network. However, it has been shown in [26] that node degree is not the most effective centrality measure for minimizing largest component size. For different network topologies, investigating resilience of network connectivity to centrality attacks provides a unified metric for evaluating network vulnerabilities.

Quantitative network resilience measures can also be used to assess the effectiveness of preventive approaches for hardening a network against attacks. Two preventive approaches are discussed in this chapter. The first method is the *edge addition* method [62], where edges are added to the network to enhance network resilience. The second method is the proposed *edge rewiring* method, where new edges are introduced by swapping a subset of existing edges.

One possible advantage of edge rewiring over edge addition is that edge rewiring requires no additional edges to enhance network resilience. The edge rewiring method might be preferable to the edge addition method in the following aspects:

- **Lower operational and maintenance costs**: for power grids, power dissipation and facility maintenance costs are proportional to the total number of edges in the network.

- **Easier link monitoring for network security**: in large-scale systems such as Internet and cellular infrastructures, introducing additional edges inevitably raises the security risks to information exposure, and it also incurs extra burden for system administration and monitoring.

- **Reduced provisioning budget**: in networking applications with stringent energy/bandwidth constraints, such as sensor networks and peer-to-peer (P2P) networks, introducing additional edges consumes more networking resources.

To illustrate resilience of network connectivity to different centrality attacks, and effectiveness of preventive approaches, we consider the power grid network for western US states [163]. We show that different centrality measures differ significantly in their ability to assess resilience of this real-world network. If the proposed centrality measure is used by an attacker, the largest component size can be reduced to nearly half of its original size by removing only 0.2% of nodes in the network. Attacks using other types of centrality measures are less effective in reducing largest component size. In particular, even if as many as 1% of the nodes are removed, less than 6% reduction in largest component size is achieved by other types of centrality attacks. In addition, we show that the proposed edge rewiring method can greatly improve network resilience via only a few edge rewires while achieving the same performance as the edge addition method.

9.1 Resilience of Western US States Power Grid Topology to Centrality Attacks

Nodal centrality attack on a network incapacitates the nodes that have highest centrality measure. The resilience of a network to centrality attacks is defined as the decrease in the size of the largest component that results from the attack. Throughout this chapter we adopt a greedy node removal strategy that sequentially removes the node with highest centrality measure from the remaining largest component. The centrality measure is recalculated after node removals. It has been shown in [75] that greedy node removal strategies can be effective reducers of the largest component size as compared with batch node removal strategies based on the same centrality

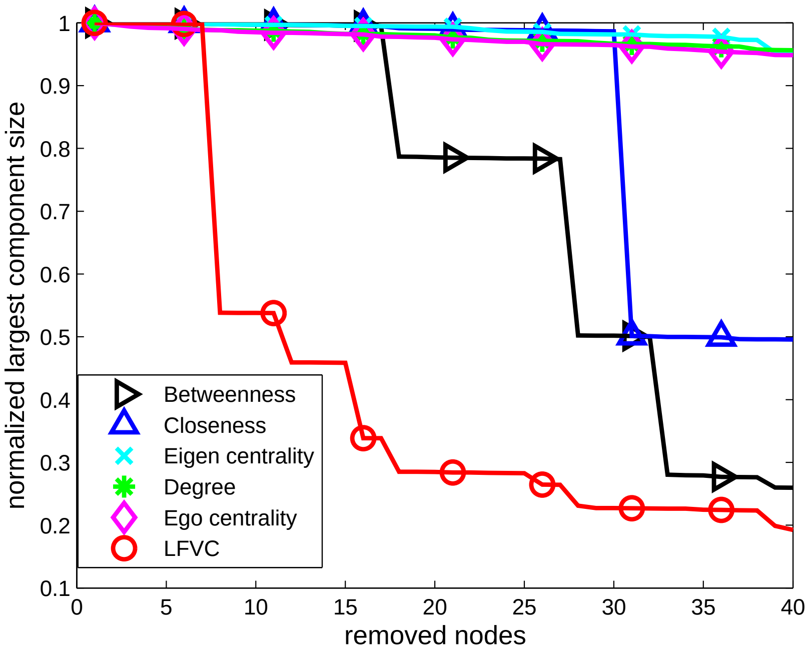


Figure 9.1: Resilience of network connectivity to different centrality attacks on the power grid topology of western US states [163]. This network contains 4941 nodes and 6594 edges, where nodes represent power stations and edges represent power lines. By removing roughly 0.2% of the nodes in the network based on an LFVC attack, the largest component size is reduced to nearly half of its original size.

measure. For general centrality measures there is no performance guarantee relating the greedy node removal strategy and the optimal batch removal strategy. However, using submodularity of the LFVC measure, it is proved in [28] that greedy node removal based on LFVC comes within at least $1 - 1/e$ of the performance of an optimal batch node removal strategy, where e is the Euler constant. Therefore one might expect that greedy LFVC attacks are almost as effective as batch LFVC attacks in terms of severe impact on network connectivity.

We use the topology of the power grid of western US states [163] to illustrate network vulnerability to different types of centrality attacks. The results are shown in Fig. 9.1. This network contains 4941 nodes and 6594 edges, where nodes represent power stations and edges represent power lines. More network topology information can be found in the supplementary file. One can see from Fig. 9.1 that an LFVC attack is capable of reducing the largest component size to roughly 54% of its original size by removing only 8 nodes from the network. On the other hand, betweenness and closeness attacks require 28 and 31 node removals, respectively, to achieve the same reduction. Equivalently, the LFVC attack requires removal of only 0.2% of the nodes in order to severely disrupt communications between nearly half of the nodes in the network. Furthermore, degree, eigen centrality, and ego centrality attacks fail to as significantly disrupt the network (less than 6% reduction in largest component) even when 1% of the nodes are attacked. By inspecting the adjacency matrix $\mathbf{A}$ in [163], it is observed that the adjacency matrix has apparent blockwise structure where blocks are densely connected subgrids that are interconnected by relatively a few inter-subgrid edges (see supplementary file). Since the high-degree nodes are not connected to those interconnected edges and each subgrid is densely connected, greedy degree attacks do not result in severe connectivity loss. We conclude that LFVC attacks do significantly more damage than other types of centrality attacks. Therefore, LFVC is a more reliable measure of resilience of the network.

9.2 Preventive Approaches to Centrality Attacks

Here we discuss two preventive approaches to protect against centrality attacks, namely the *edge addition* method and the *edge rewiring* method.

9.2.1 Edge addition method

Edge addition is perhaps the most intuitive method for enhancing resilience of network connectivity since it adds edges that are not already present in G. Let $\widehat{\mathbf{L}}$ be the resulting graph Laplacian matrix after adding an edge $(i,j) \notin \mathcal{E}$ to G and let $\mathbf{1}$ be a vector of all ones. Recalling the definition of the graph Laplacian matrix $\mathbf{L}$ in Sec. 1.3.3, $\widehat{\mathbf{L}} - \mathbf{L} = (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T$, where $\mathbf{e}_i$ is an all-zero vector except that its i-th entry is equal to 1. The term $(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T$ corresponds to the graph Laplacian matrix of the removed edge (i, j) alone. Since the algebraic connectivity $\mu(\mathbf{L})$ is the second smallest eigenvalue of $\mathbf{L}$ and the smallest eigenvalue of $\mathbf{L}$ is 0 with associated eigenvector $\mathbf{1}$, we have the representation $\mu(\mathbf{L}) = \min_{\|\mathbf{x}\|_2=1, \mathbf{x}^T\mathbf{1}=0} \mathbf{x}^T\mathbf{L}\mathbf{x}$ [55]. It is proved in [62] that

$$\mu(\widehat{\mathbf{L}}) \geq \mu(\mathbf{L}) + c_1 \cdot (y_i - y_j)^2, \tag{9.1}$$

where $\mathbf{y}$ is the eigenvector of $\mu(\mathbf{L})$ and $c_1 > 0$ is a positive constant.

Since algebraic connectivity is a lower bound on node connectivity and edge connectivity, it is proposed in [62] that one should iteratively add an edge that maximizes the quantity $(y_i - y_j)^2$ to the graph. For each iteration, the edge that maximizes $(y_i - y_j)^2$ maximizes the lower bound on the resulting algebraic connectivity, and therefore enhances network resilience to centrality attacks. The edge addition method will serve as the baseline for comparison to the proposed edge rewiring method.

9.2.2 Edge rewiring method

Edge rewiring aims to rewire the edges in the graph in order to enhance the resilience of network connectivity to attacks. In particular, edge rewiring method does not change the total number of edges in the graph. The proposed edge rewiring algorithm is summarized as follows.

For each rewire, the edge rewiring method consists of two stages: an *edge addition* stage and an *edge deletion* stage. In the edge addition stage, similar to the edge addition method, an edge $(i, j) \notin \mathcal{E}$ that maximizes $(y_i - y_j)^2$ is selected to maximize the lower bound (9.1) on the resulting algebraic connectivity. Let $\phi(\mathbf{L})$ denote the largest eigenvalue of $\mathbf{L}$ and let $\mathbf{z}$ denote the associated eigenvector of $\phi(\mathbf{L})$. In the edge deletion stage, an edge $(k, \ell) \in \mathcal{E}$ that maximizes $(z_k - z_\ell)^2$ is removed. The intuition is as follows. Let $\widetilde{\mathbf{L}}$ denote the graph Laplacian matrix after removing an

Algorithm 9.1 Edge rewiring method

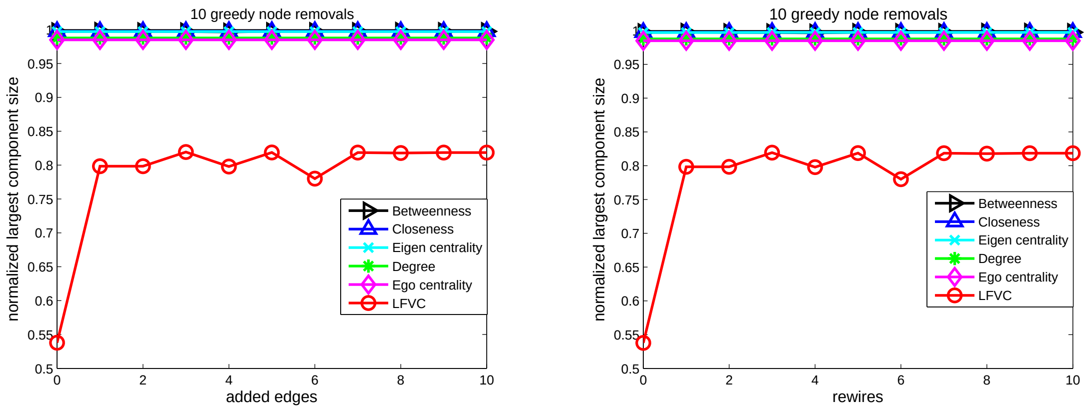
```
Input: number of rewires $r$, graph $G = (\mathcal{V}, \mathcal{E})$
Output: rewired graph $\widetilde{G} = (\mathcal{V}, \widetilde{\mathcal{E}})$
for $i = 1$ to $r$ do
    Compute the second smallest eigenvector $\mathbf{y}$ of $\mathbf{L}$
    Compute the largest eigenvector $\mathbf{z}$ of $\mathbf{L}$
    Find $(i^*, j^*) = \arg\max_{(i,j) \notin \mathcal{E}} (y_i - y_j)^2$
    Find $(k^*, \ell^*) = \arg\max_{(k,\ell) \in \mathcal{E}} (z_k - z_\ell)^2$
    Edge addition stage: $\widetilde{\mathcal{E}} \leftarrow \mathcal{E} \cup (i^*, j^*)$
    Edge deletion stage: $\widetilde{\mathcal{E}} \leftarrow \widetilde{\mathcal{E}} / (k^*, \ell^*)$
    $G \leftarrow \widetilde{G}$
end for
```

edge from G. Since $\text{trace}(\mathbf{L}) - \text{trace}(\widetilde{\mathbf{L}}) = 2$, i.e., 2 times the number of edge removals, and by Cauchy's eigenvalue interlacing property [76], $\phi(\mathbf{L}) \geq \phi(\widetilde{\mathbf{L}})$ and $\mu(\mathbf{L}) \geq \mu(\widetilde{\mathbf{L}})$, we have

$$\mu(\widetilde{\mathbf{L}}) \geq \mu(\mathbf{L}) + \phi(\mathbf{L}) - \phi(\widetilde{\mathbf{L}}) - 2. \tag{9.2}$$

Consequently, for maximum effect, the edge rewiring algorithm should remove the edge that maximizes $\phi(\mathbf{L}) - \phi(\widetilde{\mathbf{L}})$ such that the lower bound on the resulting algebraic connectivity in (9.2) is maximized. By definition, $\phi(\mathbf{L}) = \max_{\|\mathbf{x}\|_2 = 1} \mathbf{x}^T \mathbf{L} \mathbf{x}$, and $\mathbf{L} - \widetilde{\mathbf{L}} = (\mathbf{e}_k - \mathbf{e}_\ell)(\mathbf{e}_k - \mathbf{e}_\ell)^T$ when the edge $(k, \ell) \in \mathcal{E}$ is removed. Therefore, computing $\mathbf{z}^T \widetilde{\mathbf{L}} \mathbf{z}$, we have $\phi(\mathbf{L}) - \phi(\widetilde{\mathbf{L}}) \leq (z_k - z_\ell)^2$. Moreover, by the eigenvector property that $\mathbf{z}$ is orthogonal to $\mathbf{1}$ (i.e., $\mathbf{z}^T \mathbf{1} = 0$), it is easy to verify that there exists an edge $(k, \ell) \in \mathcal{E}$ and a constant $c_2 > 0$ such that $\phi(\mathbf{L}) - \phi(\widetilde{\mathbf{L}}) \geq c_2 \cdot (z_k - z_\ell)^2$.

Note that since the eigenvector $\mathbf{y}$ associated with $\mu(\mathbf{L})$ can be computed in a distributed manner [16], the eigenvector $\mathbf{z}$ associated with $\phi(\mathbf{L})$ can also be obtained using distributed local computations and message passing.



(a) The edge addition method.

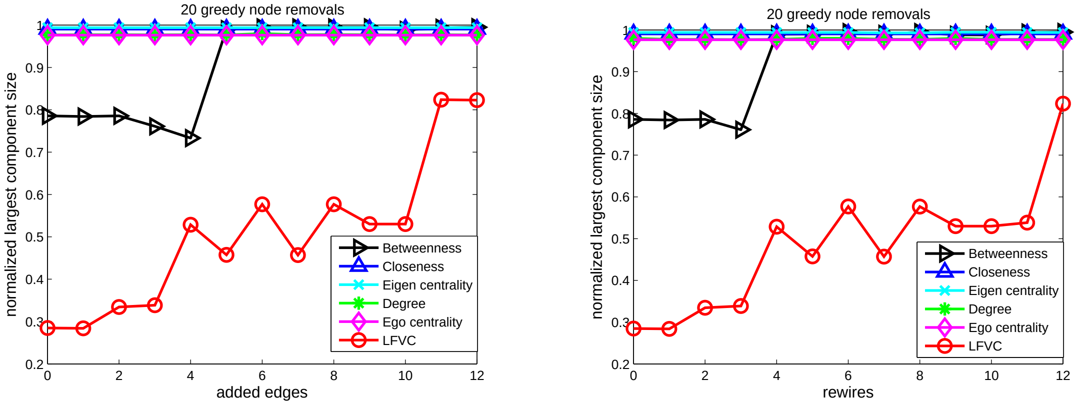
(b) The proposed edge rewiring method.

Figure 9.2: Network connectivity when restricted to 10 greedy node removals on the power grid topology of western US states [163]. For the edge addition method, the network connectivity can be enhanced from 54% to 80% under LFVC attacks by adding one edge. The proposed edge rewiring method can perform as well as the edge addition method without introducing additional edges in the network.

9.3 Performance Evaluation

In this section, we evaluate the effectiveness of the edge addition and edge rewiring methods on protecting the power grid topology [163] from centrality attacks. When 10 nodes are removed from the network by LFVC attacks, Fig. 9.1 shows that the network connectivity is reduced to 54%. In contrast, under other types of centrality attacks there is almost no loss in connectivity when 10 nodes are removed. Fig. 9.2 (a) illustrates the effect of edge addition as a preventive approach against centrality attacks. It is observed that by adding one edge, the network connectivity can be increased from 54% to 80% under LFVC attack. Fig. 9.2 (b) illustrates the proposed edge rewiring method. Similar to the edge addition method, one edge rewire is capable of enhancing the network connectivity from 54% to 80%. Thus using the edge rewiring method with only one edge rewire can protect the network as well as the edge addition method even though the latter introduces additional edges in the network.

When 20 nodes are removed from the network, as shown in Fig. 9.3 (a), 11 edge additions are required to increase network connectivity from 29% to 82%. In



(a) The edge addition method.

(b) The proposed edge rewiring method.

Figure 9.3: Network connectivity when restricted to 20 greedy node removals on the power grid topology of western US states [163]. For the edge addition method, 11 additional edges are required to enhance the network connectivity from 29% to 82%. The proposed edge rewiring method requires only 12 edge rewires to achieve the same performance as the edge addition method, which means that we only need to rewire fewer than 0.4% of edges to make it resilient to centrality attacks.

comparison, as shown in Fig. 9.3 (b), the proposed edge rewiring method requires only 12 edge rewires to achieve the same performance, which means that we only need to rewire fewer than 0.4% of the edges to make it resilient to centrality attacks. This performance advantage is explainable since, for the same number of edge additions or rewiring actions, edge rewiring changes twice as many edges in the network as edge addition.

CHAPTER X

Graph-Theoretic Action Recommendations for Cyber Resiliency

This chapter presents a theoretical framework for modeling lateral movement attacks in enterprise networks and proposes a graph-based methodology for designing mission critical cyber systems that are resilient against such attacks by mapping feasible preventative actions to operations on graph matrices. The enterprise is modeled as a tripartite network capturing the interaction between users, machines and applications, and a set of procedures is proposed to harden the network by increasing the cost of lateral movement.

Cyber security is one of the most critical problems of our time. Notwithstanding the enormous strides that researchers and practitioners have made in modeling, analyzing and mitigating cyber attacks, black hats find newer and newer methods for launching attacks requiring white hats to revisit the problem with a new perspective. One of the major ways[1] that attackers launch an attack against an enterprise is by what is known as *lateral movement via privilege escalation.* This attack cycle, shown in Fig. 10.1, begins with the compromise of a single user account (not necessarily a privileged one) in the targeted organization typically via phishing email, spear phishing or other social engineering techniques. From this initial foothold and with time

[1] http://www.verizonenterprise.com/DBIR

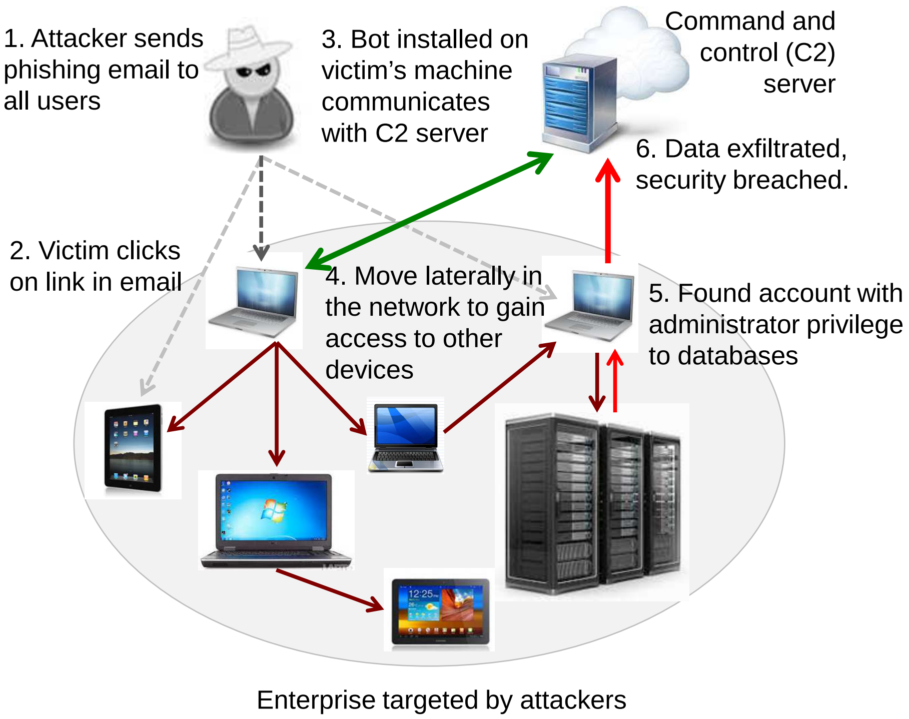
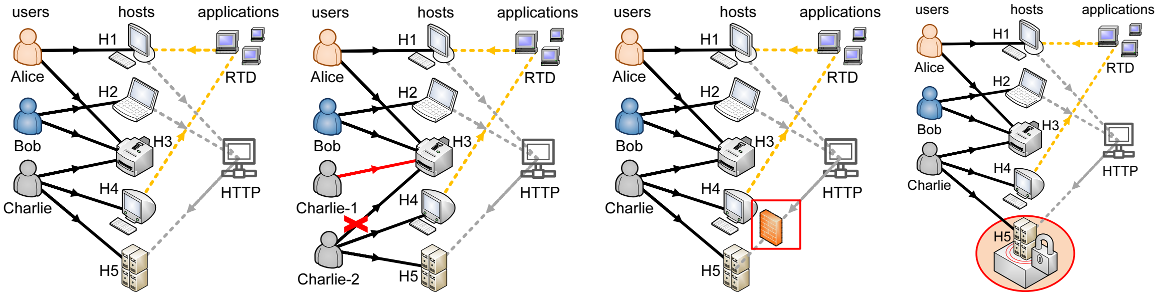

1. Attacker sends phishing email to all users
3. Bot installed on victim's machine communicates with C2 server
Command and control (C2) server
6. Data exfiltrated, security breached.
2. Victim clicks on link in email
4. Move laterally in the network to gain access to other devices
5. Found account with administrator privilege to databases
Enterprise targeted by attackers

Figure 10.1: An illustration of a cyber attack using privilege escalation techniques.

on his side, the attacker begins to explore the network, possibly compromising other user accounts until he gains access to a user account with administrative privileges to the coveted resource: files containing intellectual property, employee or customer databases or credentials to manage the network itself. Typically the attacker compromises multiple intermediate user accounts, each granting him increasing privileges. Skilled attackers frequently camouflage their lateral movements into the normal network traffic making these attacks particularly difficult to detect and insidious. Since the authorized user plays the role of an unwitting accomplice in these attacks, there is an increasing consensus that designing large enterprises to be resilient against such attacks is the preferred defensive approach.

Resilient systems accept that not all attacks can be detected and prevented; nonetheless, the system should be able to continue operation even in the face of cyber attacks and provide its core services or *missions* even if in a degraded manner [66]. To build such a resilient system it is important to be proactive in understanding



(a) Tripartite network (b) Segmentation (c) Edge hardening (d) Node hardening

Figure 10.2: (a) Illustration of a tripartite network consisting of a set of users, a set of hosts and a set of applications. (b) Segmentation in user-host access graph. The user Charlie modifies his access configuration by disabling the access of the existing account (Charlie-2) to host H3 and by creating a new user account (Charlie-1) for accessing H3 such that an attacker cannot reach the data server H5 though the printer H3 if Charlie-2 is compromised. (c) Edge hardening in host-application graph via additional firewall rules on all network flows to H5 through HTTP. (d) Node hardening in host-application graph via system update or security patch installation on H5.

and reasoning about lateral movement in an enterprise network, its potential effects on the organization, and identify ways to best defend against these threats. Unfortunately, a theoretical framework for such risk analysis is currently missing. Our goal in this chapter is to establish the theoretical foundations of a systematic framework for building networks resilient to lateral movement attacks.

We model lateral movement attack on a mission as a graph-walk problem on a tripartite *user-host-application* network that logically comprises two subgraphs: a *user-host* graph and a *host-application* graph. Fig. 10.2 illustrates the model and our methodology. The user-host-application paradigm allows us to develop an abstraction of a mission in terms of concrete entities whose behavior can be monitored and controlled. Note that, a mission is more than just the IT network or infrastructure that it is executing on. At an operational level a mission captures interactions between diverse categories of users, software and hardware resources (e.g., virtual machines, workstations, mobile devices) and applications, and we use these entities to abstract a mission.

Defining lateral movements as graph walks allows us to determine which nodes in the tripartite graph can be *reached* starting at a given node. From an attacker's perspective, these nodes that can be "reached" are exactly those mission components that can be attacked and compromised via exploits. The more number of nodes that can be reached by the attacker, the more "damage" he/she can render to the mission. Given a system snapshot and a compromised workstation or mobile device, we can define the "Attacker's Reachability" as a measure that estimates the number of hosts at risk through a given number of system exploits. Now, from a defender's perspective, putting some defensive control on one of these nodes (or edges) allows the walk to be broken at that point. Intuitively, then such walk can also be used to identify mission hardening strategies that reduce risk. This central idea is illustrated in Fig. 10.2. The heterogeneity of a cyber system entails a network of networks (NoN) representation of entities in the system as displayed in Fig. 10.2, allowing us to devise effective hardening strategies from different perspectives, which differs from works focusing on manipulating the network topology under the assumption that the graph is homogeneous, that is, all nodes have an identical role in a cyber system.

As our model considers the heterogeneity of a cyber system and incorporates several defensive actions for enhancing the resilience to lateral movement attacks, to assist reading the utility of the proposed approaches and the established theoretical results are summarized in Table 10.1, and the proofs of the established mathematical results are placed in the appendix (Appendix H).

System Heterogeneity	Attack Reachability	Proposed Approaches	Theoretical Guarantees
User-Host	Algorithm 10.1	Algorithm 10.4 Algorithm 10.5	Theorem 10.3 Corollary 10.4
Host-Application	Algorithm 10.2	Algorithm 10.6 Algorithm 10.7	Theorem 10.7 Corollary 10.8
User-Host-Application	Algorithm 10.3	all of the above	all of the above

Table 10.1: Utility of the proposed algorithms and established theoretical results in Chapter X.

10.1 Tripartite Network Model and Iterative Reachability Computation of Lateral Movement

10.1.1 Notations and tripartite graph model

The expression e denotes the Euler's number, i.e., the base of the natural logarithm. The expression $\mathbf{e}_i^x$ denotes the $x \times 1$ canonical vector of zero entries except its i-th entry being 1. The expression $\mathbf{I}_n$ denotes the $n \times n$ identify matrix. The expression $\mathbf{1}_n$ denotes the $n \times 1$ column vector of ones. The expression $\mathrm{col}_x(\mathbf{X})$ denotes the x-th column of $\mathbf{X}$. The expression $\lambda_{\max}(\mathbf{X})$ denotes the largest eigenvalue (in magnitude) of a square matrix $\mathbf{X}$. The operation $\cdot^T$ denotes matrix or vector transpose. The operation $\otimes$ denotes the Kronecker product which is defined in Appendix H.1. The operation $\odot$ denotes the Hadamard (entry-wise) product of matrices. The operator $\mathbb{T} : \mathbb{R}_+^n \mapsto [0,1]^n$ is a thresholding function such that $[\mathbb{T}(\mathbf{x})]_i = [\mathbf{x}]_i$ if $0 \leq [\mathbf{x}]_i \leq 1$ and $[\mathbb{T}(\mathbf{x})]_i = 1$ if $[\mathbf{x}]_i > 1$. The operator $\mathbb{H}_\mathbf{a} : [0,1]^n \mapsto \{0,1\}^n$ is an entry-wise indicator function such that $[\mathbb{H}_\mathbf{a}(\mathbf{x})]_i = 1$ if $[\mathbf{x}]_i > [\mathbf{a}]_i$, and $[\mathbb{H}_\mathbf{a}(\mathbf{x})]_i = 0$ otherwise.

The tripartite graph in Fig. 10.2 can be characterized by a set of users $\mathcal{V}_{user}$, a set of hosts $\mathcal{V}_{host}$, a set of applications $\mathcal{V}_{app}$, a set of user-host accesses $\mathcal{E} \subset \mathcal{V}_{user} \times \mathcal{V}_{host}$, and a set of host-application-host activities $\mathcal{T} \subset \mathcal{V}_{host} \times \mathcal{V}_{app} \times \mathcal{V}_{host}$. The cardinality of $\mathcal{V}_{user}$, $\mathcal{V}_{host}$ and $\mathcal{V}_{app}$ are denoted by U, N and K, respectively. The list of main notations and symbols used in this chapter are listed in Table 10.2.

$U/N/K$	number of users / hosts / applications
$\lambda_{\max}(\mathbf{X})$	largest eigenvalue of matrix $\mathbf{X}$
$\otimes$	Kronecker product
$\mathbf{1}_n$	$n \times 1$ column vector of ones
$\mathbf{A}_C$	User-host graph matrix
$\mathbf{A}$	Host-application graph matrix
$\mathbf{P}$	Compromise probability matrix
$\mathbf{B}$	$\mathbf{A}_C^T \mathbf{A}_C$
$\mathbf{J}$	$(\mathbf{P} \otimes \mathbf{1}_N)^T \mathbf{A}^T$
$\mathbf{r}$ / $\mathbf{a}$	Reachability / Hardening level vector
$\mathbb{T}(\mathbf{x})$	Threshold function on vector $\mathbf{x}$
$\mathbb{H}_{\mathbf{a}}(\mathbf{x})$	Comparator function of $\mathbf{x}$ and $\mathbf{a}$

Table 10.2: List of main notations and symbols in Chapter X.

10.1.2 Reachability of lateral movement on user-host access graph

Let $G_C = (\mathcal{V}_{user}, \mathcal{V}_{host}, \mathcal{E})$ with $\mathcal{E} \subset \mathcal{V}_{user} \times \mathcal{V}_{host}$ denote the user-host bipartite graph. The access privileges between users and hosts are represented by a binary $U \times N$ adjacency matrix $\mathbf{A}_C$, where $[\mathbf{A}_C]_{ij} = 1$ if user i can access host j, and $[\mathbf{A}_C]_{ij} = 0$ otherwise. Let $\mathbf{r}_0$ be an $N \times 1$ binary vector indicating the initial host compromise status, where $[\mathbf{r}_0]_j = 1$ if host j is initially being compromised, and $[\mathbf{r}_0]_j = 0$ otherwise. Given $\mathbf{r}_0$, we are interested in computing the final binary compromise vector $\mathbf{r}_\infty$ when attackers leverage user access privileges to compromise other accessible hosts. $\mathbf{r}_\infty$ specifies the reachability of a lateral movement attack, where reachability is defined as the fraction of hosts that can be reached via graph walks on G_C starting from $\mathbf{r}_0$. Therefore, reachability is used as a quantitative measure of network vulnerability to lateral movement attacks. Furthermore, studying $\mathbf{r}_\infty$ allows us to investigate the dominant factor that leads to high reachability and more efficient countermeasures.

The computation of $\mathbf{r}_\infty$ can be viewed as a cascading process of repetitive walks on G_C starting from a set of compromised hosts. Let $\mathbf{r}_t$ denote the binary compromise vector after t-hop walks and let $\mathbf{w}_h$ be the number of h-hop walks starting from $\mathbf{r}_0$ and $\mathbf{w}_0 = \mathbf{r}_0$. The hop count of a walk between two hosts in G_C is defined as the

Algorithm 10.1 Iterative reachability computation for lateral movement on user-host graph

```
Input: $\mathbf{r}_0$, $\mathbf{B}$
Output: $\mathbf{r}_\infty$
Initialization: $\mathbf{r}_{old} = \mathbf{r}_0$. Flag = 1.
while Flag= 1 do
    $\mathbf{r}_{new} = \mathbb{T}\left(\mathbf{r}_{old} + \mathbf{B}\mathbf{r}_{old}\right)$.
    if $\mathbf{r}_{new} = \mathbf{r}_{old}$ then
        Flag= 0. $\mathbf{r}_\infty = \mathbf{r}_{new}$.
    else
        $\mathbf{r}_{old} = \mathbf{r}_{new}$
    end if
end while
```

number of traversed users. We begin by computing $\mathbf{r}_1$ from $\mathbf{r}_0$: the number of 1-hop walk from $\mathbf{r}_0$ to host j is $[\mathbf{w}_1]_j = \sum_{i=1}^{U} \sum_{k=1}^{N} [\mathbf{A}_C]_{ij} [\mathbf{A}_C]_{ik} [\mathbf{r}_0]_k = \mathbf{e}_j^{N^T} \mathbf{A}_C^T \mathbf{A}_C \mathbf{r}_0$. Let $\mathbf{B} = \mathbf{A}_C^T \mathbf{A}_C$, an induced adjacency matrix of hosts in G_C, where $[\mathbf{B}]_{ij}$ is the number of common users that can access hosts i and j. Then we have $\mathbf{w}_1 = \mathbf{B}\mathbf{r}_0$ and $\mathbf{r}_1 = \mathbb{T}(\mathbf{w}_1)$. Generalizing this result, we have

$$\mathbf{w}_{h+1} = \mathbf{B}\mathbf{w}_h = \mathbf{B}^{h+1}\mathbf{r}_0; \tag{10.1}$$

$$\mathbf{r}_{t+1} = \mathbb{T}\left(\sum_{h=1}^{t+1} \mathbf{w}_h\right). \tag{10.2}$$

The term in (10.2) accounts for the accumulation of compromised hosts up to $t+1$ hops. Note that based on the property of $\mathbb{T}$, (10.2) can be simplified as

$$\mathbf{r}_{t+1} = \mathbb{T}\left(\mathbf{r}_t + \mathbf{B}\mathbf{r}_t\right). \text{ (Appendix H.2)} \tag{10.3}$$

(10.3) suggests that the term $\mathbf{B}$ is the dominant factor affecting the propagation of lateral movement, and we obtain an efficient iterative algorithm (Algorithm 10.1) for computing $\mathbf{r}_\infty$ that involves successive matrix-vector multiplications until $\mathbf{r}_t$ converges.

10.1.3 Reachability of lateral movement on host-application graph

The host-application graph contains the information of one host communicating with another host through an application. Let $\mathbf{A}_k$ be an $N \times N$ binary matrix representing the host-to-host communication through application k, where $[\mathbf{A}_k]_{ij} = 1$ means host i communicates with j through application k; and $[\mathbf{A}_k]_{ij} = 0$ otherwise. The $N \times KN$ binary matrix $\mathbf{A} = [\mathbf{A}_1 \ \mathbf{A}_2 \ \cdots \ \mathbf{A}_K]$ is the concatenated matrix of K host-application-host matrices $\mathbf{A}_k$ for $k = 1, 2, \ldots, K$. Let $\mathbf{P}$ be a $K \times N$ matrix where its entry $[\mathbf{P}]_{kj}$ specifies the probability of compromising host j through application k. Each host is assigned with a hardening value $[\mathbf{a}]_j \in [0, 1]$ indicating its security level.

Similar to Sec. 10.1.2, we are interested in computing the reachability of lateral movement on the host-application graph. The hop count of a walk between two hosts in the host-application graph is defined as the average number of paths between the two hosts through applications. Let $\mathbf{W}$ be an $N \times N$ matrix where $[\mathbf{W}]_{ij}$ is the average number of one-hop walk from host i to host j. Then we have $[\mathbf{W}]_{ij} = \sum_{k=1}^{K} [\mathbf{A}_k]_{ij} \mathbf{P}_{kj}$. Let $\mathbf{w}_h$ be an $N \times 1$ vector representing the average number of h-hop walks of hosts and $\mathbf{w}_0 = \mathbf{r}_0$. Then the j-th entry of the 1-hop vector $\mathbf{w}_1$ is

$$[\mathbf{w}_1]_j = \mathbf{e}_j^T \left[\mathrm{col}_j(\mathbf{P})^T \otimes \mathbf{I}_n \right] \mathbf{A}^T \mathbf{r}_0. \text{ (Appendix H.3)} \tag{10.4}$$

Stacking (10.4) as a column vector gives

$$\mathbf{w}_1 = \left(\mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T \mathbf{r}_0. \text{ (Appendix H.4)} \tag{10.5}$$

The 1-hop compromise vector $\mathbf{r}_1$ is defined as $\mathbf{r}_1 = \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\mathbf{w}_1 \right) \right)$. In effect the operator $\mathbb{H}_{\mathbf{a}}$ compares the thresholded average number of walks with the hardening level for each host, which means a host j can be compromised only when the thresholded average number of 1-hop walk $[\mathbb{T} \left(\mathbf{w}_1 \right)]_j$ is greater than its hardening level $[\mathbf{a}]_j$.

Algorithm 10.2 Iterative reachability computation for lateral movement on host-application graph

```
Input: $\mathbf{r}_0$, $\mathbf{A}$, $\mathbf{P}$ and $\mathbf{a}$
Output: $\mathbf{r}_\infty$
Initialization: $\mathbf{r}_{old} = \mathbf{r}_0$. Flag = 1.
while Flag= 1 do
    $\mathbf{r}_{new} = \mathbb{H}_{\mathbf{a}} \left( \mathbb{T} \left( \mathbf{r}_{old} + \left( \mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T \mathbf{r}_{old} \right) \right)$.
    if $\mathbf{r}_{new} = \mathbf{r}_{old}$ then
        Flag= 0. $\mathbf{r}_\infty = \mathbf{r}_{new}$.
    else
        $\mathbf{r}_{old} = \mathbf{r}_{new}$
    end if
end while
```

Algorithm 10.3 Iterative reachability computation for lateral movement on user-host-application graph

```
Input: $\mathbf{r}_0$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{B}$ and $\mathbf{a}$
Output: $\mathbf{r}_\infty$
Initialization: $\mathbf{r}_{old} = \mathbf{r}_0$. Flag = 1.
while Flag= 1 do
    $\mathbf{r}_{new} = \mathbb{H}_{\mathbf{a}} \left( \mathbb{T} \left( \mathbf{r}_{old} + \left[ \mathbf{B} + \left( \mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T \right] \mathbf{r}_{old} \right) \right)$.
    if $\mathbf{r}_{new} = \mathbf{r}_{old}$ then
        Flag= 0. $\mathbf{r}_\infty = \mathbf{r}_{new}$.
    else
        $\mathbf{r}_{old} = \mathbf{r}_{new}$
    end if
end while
```

Generalizing this result to h-hop, we have

$$\mathbf{w}_{h+1} = \left(\mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T \mathbf{w}_h; \tag{10.6}$$

$$\mathbf{r}_{t+1} = \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\sum_{h=1}^{t+1} \mathbf{w}_h \right) \right). \tag{10.7}$$

The term in (10.7) has an equivalent expression

$$\mathbf{r}_{t+1} = \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\mathbf{r}_t + \left(\mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T \mathbf{r}_t \right) \right). \text{ (Appendix H.5)} \tag{10.8}$$

As a result, the matrix $\mathbf{J} =: \left(\mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T$ is the dominant factor for lateral movement on the host-application graph, and (10.8) leads to an iterative algorithm (Algorithm 10.2) for reachability computation, which is similar to the methodology of Algorithm 10.1.

10.1.4 Reachability of lateral movement on tripartite user-host-application graph

Utilizing the developed results in Sec. 10.1.2 and Sec. 10.1.3, the cascading process of lateral movement on the tripartite user-host-application graph can be modeled by

$$\mathbf{r}_{t+1} \equiv \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\mathbf{r}_t + \left[\mathbf{B} + \left(\mathbf{P} \otimes \mathbf{1}_N \right)^T \mathbf{A}^T \mathbf{r}_t \right) \right] \right).$$

The corresponding iterative algorithm for reachability computation is summarized in Algorithm 10.3.

Moreover, in cases when attackers only know partial information (network topology) of the tripartite graph, one can apply binary (potentially probabilistic) masking functions on **B** or **A** and evaluate the corresponding reachability using the proposed algorithms.

10.2 Segmentation on User-Host Graph

In this section we investigate segmentation on user-host graph as a countermeasure for suppressing lateral movement attacks. Segmentation works by creating new user accounts to separate user-host in order to alleviate the reachability of lateral movement, as illustrated in Fig. 10.2 (b). In principle segmentation removes some edges from the access graph G_C and then merge these removed edges to create new user accounts. Therefore, segmentation retains the same access functionality and constrains lateral movement attacks at the price of additional user accounts. The following analysis provides a theoretical framework of different segmentation strategies.

Recall from (10.3) that the matrix $\mathbf{B}$ is the key factor affecting the reachability of lateral movement on G_C. Therefore, an effective edge removal approach for segmentation is reducing the spectral radius of $\mathbf{B}$ (i.e., $\lambda_{\max}(\mathbf{B})$) by removing some edges from G_C. Note that by definition $\mathbf{B} = \mathbf{A}_C^T \mathbf{A}_C$ so that $\mathbf{B}$ is a positive semidefinite (PSD) matrix, and all entries of $\mathbf{B}$ are nonnegative. Therefore, by the Perron-Frobenious theorem [76] the entries of $\mathbf{B}$'s largest eigenvector $\mathbf{u}$ (i.e., the eigenvector such that $\mathbf{B}\mathbf{u} = \lambda_{\max}(\mathbf{B})\mathbf{u}$) are nonnegative.

Here we investigate the change in $\lambda_{\max}(\mathbf{B})$ when an edge is removed from G_C in order to define an edge score function that is associated with spectral radius reduction. If an edge $(i,j) \in \mathcal{E}$ is removed from G_C, then the resulting adjacency matrix of $G_C \setminus (i,j)$ is $\widetilde{\mathbf{A}}_C\left((i,j)\right) = \mathbf{A}_C - \mathbf{e}_i^U \mathbf{e}_j^{NT}$. The corresponding induced adjacency matrix is

$$\begin{aligned}\widetilde{\mathbf{B}}\left((i,j)\right) &= \widetilde{\mathbf{A}}_C\left((i,j)\right)^T \widetilde{\mathbf{A}}_C\left((i,j)\right) \\ &= \mathbf{B} - \mathbf{A}_C^T \mathbf{e}_i^U \mathbf{e}_j^{NT} - \mathbf{e}_j^N \mathbf{e}_i^{UT} \mathbf{A}_C + \mathbf{e}_j^N \mathbf{e}_j^{NT}. \qquad (10.9)\end{aligned}$$

By the Courant-Fischer theorem [76] we have

$$\begin{aligned}\lambda_{\max}\left(\widetilde{\mathbf{B}}\left((i,j)\right)\right) &\geq \mathbf{u}^T \widetilde{\mathbf{B}}\left((i,j)\right)\mathbf{u} \\ &= \lambda_{\max}(\mathbf{B}) - 2\mathbf{u}^T \mathbf{A}_C^T \mathbf{e}_i^U [\mathbf{u}]_j + [\mathbf{u}]_j^2. \qquad (10.10)\end{aligned}$$

(10.10) leads to a greedy removal strategy that finds the edge $(i,j) \in \mathcal{E}$ that maximizes the edge score function $2\mathbf{u}^T\mathbf{A}_C^T\mathbf{e}_i^U[\mathbf{u}]_j - [\mathbf{u}]_j^2$, in order to minimize a lower bound on the spectral radius of $\widetilde{\mathbf{B}}\left((i,j)\right)$. Moreover, Lemma 10.1 below shows that the edge score function is also associated with an upper bound on the spectral radius of $\widetilde{\mathbf{B}}\left((i,j)\right)$. Following similar methodology, when a subset of edges $\mathcal{E}_{\mathcal{R}} \subset \mathcal{E}$ are removed from G_C, we have

$$\lambda_{\max}\left(\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)\right) \geq \lambda_{\max}(\mathbf{B}) - f(\mathcal{E}_{\mathcal{R}}),\ \text{(Appendix H.6)} \tag{10.11}$$

where the function

$$f(\mathcal{E}_{\mathcal{R}}) = 2\sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \mathbf{u}^T\mathbf{A}_C^T\mathbf{e}_i^U[\mathbf{u}]_j - \sum_{i\in\mathcal{V}_{user}}\ \sum_{j\in\mathcal{V}_{host},(i,j)\in\mathcal{E}_{\mathcal{R}}}\ \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j[\mathbf{u}]_s. \tag{10.12}$$

In a nutshell, the function $f(\mathcal{E}_{\mathcal{R}})$ provides a score that evaluates the effect of edge removal set $\mathcal{E}_{\mathcal{R}}$ on the spectral radius of $\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)$. The lemma presented in Appendix H.7 shows $f(\mathcal{E}_{\mathcal{R}})$ is nonnegative as it can be represented as a sum of nonnegative terms. The following lemma shows that $f(\mathcal{E}_{\mathcal{R}})$ is associated with an upper bound on the spectral radius of $\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)$. Therefore, maximizing $f(\mathcal{E}_{\mathcal{R}})$ can be an effective strategy for spectral radius reduction.

Lemma 10.1. *For any edge removal set of cardinality $q \geq 1$, if there exits one edge removal set $\mathcal{E}_{\mathcal{R}} \subset \mathcal{E}$ with $|\mathcal{E}_{\mathcal{R}}| = q$ such that $f(\mathcal{E}_{\mathcal{R}}) > 0$, then there exists some constant $c > 0$ such that*

$$\lambda_{\max}(\mathbf{B}) - c \cdot f(\mathcal{E}_{\mathcal{R}}) \geq \lambda_{\max}\left(\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)\right). \tag{10.13}$$

Proof. The proof can be found in Appendix H.8. □

Moreover, the lemma presented in Appendix H.9 shows that $f(\mathcal{E}_{\mathcal{R}})$ is a monotonic increasing set function. A monotonic increasing set function means that for any two subsets $\mathcal{E}_{\mathcal{R}1}, \mathcal{E}_{\mathcal{R}2} \subset \mathcal{E}$ satisfying $\mathcal{E}_{\mathcal{R}1} \subset \mathcal{E}_{\mathcal{R}2}$, $f(\mathcal{E}_{\mathcal{R}2}) \geq f(\mathcal{E}_{\mathcal{R}1})$. In addition, the

Algorithm 10.4 Greedy score segmentation algorithm

```
**Input:** $\mathbf{A}_C$, number of segmented edges $q$
**Output:** modified access adjacency matrix $\mathbf{A}_C^q$
**if** recalculating score **then**
    Initialization: $\mathbf{A}_C^{old} = \mathbf{A}_C$. $\mathcal{E}_{old} \leftarrow \mathcal{E}$. $\mathcal{E}_{\mathcal{R}} \leftarrow \varnothing$.
    **for** $z = 1$ to $q$ **do**
        1. Compute the leading eigenvector $\mathbf{u}$ of $\mathbf{B} = \mathbf{A}_C^{old^T}\mathbf{A}_C^{old}$
        2. Compute score $f\left((i,j)\right) = 2\mathbf{u}^T\mathbf{A}_C^{old^T}\mathbf{e}_i^U[\mathbf{u}]_j - [\mathbf{u}]_j^2$ for all $(i,j) \in \mathcal{E}_{old}$
        3. Remove the highest scored edge $(i^*, j^*) \in \mathcal{E}_{old}$ from $\mathbf{A}_C^{old}$
        4. $\mathbf{A}_C^{old} = \mathbf{A}_C^{old} - \mathbf{e}_{i^*}^U\mathbf{e}_{j^*}^{N^T}$. $\mathcal{E}_{old} \leftarrow \mathcal{E}_{old} \setminus (i^*, j^*)$. $\mathcal{E}_{\mathcal{R}} \leftarrow \mathcal{E}_{\mathcal{R}} \cup (i^*, j^*)$.
    **end for**
**else**
    1. Compute the leading eigenvector $\mathbf{u}$ of $\mathbf{B} = \mathbf{A}_C^T\mathbf{A}_C$
    2. Compute score $f\left((i,j)\right) = 2\mathbf{u}^T\mathbf{A}_C^T\mathbf{e}_i^U[\mathbf{u}]_j - [\mathbf{u}]_j^2$ for all $(i,j) \in \mathcal{E}$
    3. Remove the $q$ edges of highest scores from $\mathbf{A}_C$
    4. Store this set of $q$ edges in $\mathcal{E}_{\mathcal{R}}$
**end if**
5. Segment the removed edges in $\mathcal{E}_{\mathcal{R}}$ to create new users. A new user $u$ has access
to a set of hosts $\{s : (u,s) \in \mathcal{E}_{\mathcal{R}}\}$
6. Obtain the modified access adjacency matrix $\mathbf{A}_C^q$ from segmentation
```

following theorem shows that $f(\mathcal{E}_{\mathcal{R}})$ is a monotone submodular set function [59], which establishes performance guarantee of greedy edge removal on reducing the spectral radius of $\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)$. Submodularity means $f(\mathcal{E}_{\mathcal{R}})$ has diminishing gain: for any $\mathcal{E}_{\mathcal{R}1} \subset \mathcal{E}_{\mathcal{R}2} \subset \mathcal{E}$ and $e \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{R}2}$, the discrete derivative $\Delta f(e|\mathcal{E}_{\mathcal{R}}) = f(\mathcal{E}_{\mathcal{R}} \cup e) - f(\mathcal{E}_{\mathcal{R}})$ satisfies $\Delta f(e|\mathcal{E}_{\mathcal{R}2}) \leq \Delta f(e|\mathcal{E}_{\mathcal{R}1})$.

Theorem 10.2. *$f(\mathcal{E}_{\mathcal{R}})$ is a monotone submodular set function.*

Proof. The proof can be found in Appendix H.10. □

With the established results, a greedy segmentation algorithm (Algorithm 10.4) is proposed that computes the edge score function $f\left((i,j)\right) = 2\mathbf{u}^T\mathbf{A}_C^T\mathbf{e}_i^U[\mathbf{u}]_j - [\mathbf{u}]_j^2$ for every edge $(i,j) \in \mathcal{E}$ and segments q edges of highest scores to create new user accounts. For efficient computation step 2 of Algorithm 10.4 can be represented by the matrix form $\mathbf{F} = \left[2\mathbf{A}_C^T\mathbf{u}\mathbf{u}^T - \mathbf{1}_U\left(\mathbf{u} \odot \mathbf{u}\right)^T\right] \odot \mathbf{A}_C$, where $[\mathbf{F}]_{ij} = f\left((i,j)\right)$ if $(i,j) \in \mathcal{E}_{\mathcal{R}}$, and $[\mathbf{F}]_{ij} = 0$ otherwise.

Using the monotonic submodularity of $f(\mathcal{E}_{\mathcal{R}})$ in Theorem 10.2, the following theorem shows that this greedy algorithm (Algorithm 10.4 without score recalculation) has performance guarantee on spectral radius reduction relative to the optimal batch edge removal strategy of combinatorial computation complexity for selecting the best q edges.

Theorem 10.3. (Greedy segmentation without score recalculation) *Let $\mathcal{E}_{\mathcal{R}}^{opt}$ be the optimal batch edge removal set of cardinality $q \geq 1$ that maximizes $f(\mathcal{E}_{\mathcal{R}})$ and let $\mathcal{E}_{\mathcal{R}}^{q}$ with $|\mathcal{E}_{\mathcal{R}}^{q}| = q$ be the greedy edge removal set obtained from Algorithm 10.4. If $f(\mathcal{E}_{\mathcal{R}}^{q}) > 0$, then there exists some constant $c' > 0$ such that*

$$f(\mathcal{E}_{\mathcal{R}}^{opt}) - f(\mathcal{E}_{\mathcal{R}}^{q}) \leq \left(1 - \frac{1}{q}\right)^{q} f(\mathcal{E}_{\mathcal{R}}^{opt}) \leq \frac{1}{\mathsf{e}} f(\mathcal{E}_{\mathcal{R}}^{opt});$$

$$\lambda_{\max}(\mathbf{B}) - f(\mathcal{E}_{\mathcal{R}}^{opt}) \leq \lambda_{\max}\left(\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}^{q}\right)\right) \leq \lambda_{\max}(\mathbf{B}) - c' \cdot f(\mathcal{E}_{\mathcal{R}}^{opt}).$$

Proof. The proof can be found in Appendix H.11. □

As a variant of Algorithm 10.4 without score recalculation, for better traceability one may desire to successively recalculate the largest eigenvector $\mathbf{u}$ and update the edge score function $f(i, j)$ after each edge removal. The following corollary provides a theoretical analysis of the greedy segmentation algorithm with score recalculation (Algorithm 10.4 with score recalculation), which shows that score recalculation can successively reduce the spectral radius of $\mathbf{B}$.

Corollary 10.4. (Greedy segmentation with score recalculation) *Let $\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})$ denote the adjacency matrix of $G_C \setminus \mathcal{E}_{\mathcal{R}}$ for some $\mathcal{E}_{\mathcal{R}} \subset \mathcal{E}$, and let $\mathbf{u}_{\mathcal{E}_{\mathcal{R}}}$ denote the largest eigenvector of $\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}})$. For any edge removal set $\mathcal{E}_{\mathcal{R}} \subset \mathcal{E}$, let $f_{\mathcal{E}_{\mathcal{R}}}(i,j) = 2\mathbf{u}_{\mathcal{E}_{\mathcal{R}}}^T \widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})^T \mathbf{e}_i^U [\mathbf{u}_{\mathcal{E}_{\mathcal{R}}}]_j - [\mathbf{u}_{\mathcal{E}_{\mathcal{R}}}]_j^2$, and let (i^*, j^*) be a maximizer of $f_{\mathcal{E}_{\mathcal{R}}}(i,j)$. Then $\lambda_{\max}\left(\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}})\right) \geq \lambda_{\max}\left(\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}} \cup (i^*, j^*))\right)$. Furthermore, if $f_{\mathcal{E}_{\mathcal{R}}}(i^*, j^*) > 0$, then $\lambda_{\max}\left(\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}})\right) > \lambda_{\max}\left(\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}} \cup (i^*, j^*))\right)$.*

Proof. The proof can be found in Appendix H.12. □

In addition to establishing the performance guarantee of the greedy score segmentation algorithm in Algorithm 10.4 for reducing the spectral radius of $\mathbf{B}$, the following theorem shows that the two intuitive greedy segmentation algorithms proposed in Algorithm 10.5, with an aim of successively segmenting the edge connecting to the most connected user or host, are also effectively reducing an upper bound on the spectral radius of $\mathbf{B}$. The terms $\mathbf{d}^U = \mathbf{A}_C \mathbf{1}_N$ and $\mathbf{d}^N = \mathbf{A}_C^T \mathbf{1}_U$ denote the degree vector of users and hosts, respectively, and the terms $d_{\max}^{user}$ and $d_{\max}^{host}$ denote the maximum degree of users and hosts in G_C, respectively.

Theorem 10.5. (Greedy user-(host-)first segmentation) *If an edge* (i, j) *is removed from* G_C *and* $\widetilde{\mathbf{B}}(i, j)$ *is irreducible, then*

$$\lambda_{\max}\left(\widetilde{\mathbf{B}}(i,j)\right) \leq d_{\max}^{user} \cdot d_{\max}^{host} - \max_{s \in \{1,2,\ldots,N\}} \left[\left([\mathbf{d}^U]_i - 1 \right) \mathbf{e}_j^N - \mathbf{A}_C^T \mathbf{e}_i^U \right]_s .$$

Proof. The proof and the case when $\widetilde{\mathbf{B}}(i, j)$ is reducible can be found in Appendix H.13. □

Since the term $\mathbf{A}_C^T \mathbf{e}_i^U$ in Theorem 10.5 is a vector of access connections of user i, Theorem 10.5 suggests a greedy user-first segmentation approach that segments the edge between the user of maximum degree and the corresponding accessible host of maximum degree in order to reduce the upper bound on spectral radius in Theorem 10.5. Similar analysis apples to the greedy host-first segmentation approach in Algorithm 10.5.

10.3 Hardening on Host-Application Graph

In this section we discuss two countermeasures for constraining lateral movement on the host-application graph. Edge hardening refers to securing access from appli-

Algorithm 10.5 Greedy user-(host-)first segmentation algorithm

```
**Input:** $\mathbf{A}_C$, number of segmented edges $q$
**Output:** modified access adjacency matrix $\mathbf{A}_C^q$
Initialization: $\mathbf{A}_C^{old} = \mathbf{A}_C$. $\mathcal{E}_{old} \leftarrow \mathcal{E}$. $\mathcal{E}_{\mathcal{R}} \leftarrow \varnothing$.
**for** $z = 1$ to $q$ **do**
    1. Compute user (host) degree vector $\mathbf{d}^U = \mathbf{A}_C^{old}\mathbf{1}_N$ ($\mathbf{d}^N = {\mathbf{A}_C^{old}}^T \mathbf{1}_U$)
    2. Obtain $i^* = \arg\max_i [\mathbf{d}^U]_i$ and $j^* = \arg\max_{j:[\mathbf{A}_C^{old}]_{i^*j}>0} [\mathbf{d}^N]_h$ ($j^* = \arg\max_j [\mathbf{d}^N]_j$ and $i^* = \arg\max_{i:[\mathbf{A}_C^{old}]_{ij^*}>0} [\mathbf{d}^U]_i$)
    3. Remove the edge $(i^*, j^*) \in \mathcal{E}_{old}$ from $\mathbf{A}_C^{old}$.
    4. $\mathbf{A}_C^{old} = \mathbf{A}_C^{old} - \mathbf{e}_{i^*}^U {\mathbf{e}_{j^*}^N}^T$. $\mathcal{E}_{old} \leftarrow \mathcal{E}_{old} \setminus (i^*, j^*)$. $\mathcal{E}_{\mathcal{R}} \leftarrow \mathcal{E}_{\mathcal{R}} \cup (i^*, j^*)$
**end for**
5. Segment the removed edges in $\mathcal{E}_{\mathcal{R}}$ to create new users. A new user $u$ has access to a set of hosts $\{s : (u, s) \in \mathcal{E}_{\mathcal{R}}\}$
6. Obtain the modified access adjacency matrix $\mathbf{A}_C^q$ from segmentation
```

cation k to host j, and in effect reducing the compromise probability $[\mathbf{P}]_{kj}$. Node hardening refers to securing a particular host and in effect increasing its hardening level.

Recall from (10.8) that the reachability of lateral movement on host-application graph is governed by the matrix $\mathbf{J} = (\mathbf{P} \otimes \mathbf{1}_N)^T \mathbf{A}^T$. Note that although $\mathbf{J}$ is in general not a symmetric matrix, it is a matrix of nonnegative entries and hence by the Perron-Frobenious theorem [76] $\lambda_{\max}(\mathbf{J})$ is real and nonnegative, and the entries of its largest eigenvector are nonnegative.

Hardening a host j for an application k means that after hardening the compromise probability $[\mathbf{P}]_{kj}$ is reduced to some value ϵ_{kj} such that $[\mathbf{P}]_{kj} > \epsilon_{kj} \geq 0$. Let $\mathcal{H}$ denote the set of hardened edges and let $\widetilde{\mathbf{P}}_{\mathcal{H}}$ be the compromise probability matrix after edge hardening. Then we have $\widetilde{\mathbf{P}}_{\mathcal{H}} = \mathbf{P} - \sum_{(k,j)\in\mathcal{H}} \left([\mathbf{P}]_{kj} - \epsilon_{kj}\right) \mathbf{e}_k^K {\mathbf{e}_j^N}^T$. Let $\widetilde{\mathbf{J}}(\mathcal{H}) = (\widetilde{\mathbf{P}}_{\mathcal{H}} \otimes \mathbf{1}_N)^T \mathbf{A}^T$ and let $\mathbf{y}$ be the largest eigenvector of $\mathbf{J}$. We can show that

$$\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right) \geq \lambda_{\max}(\mathbf{J}) - \mathbf{y}^T \Delta\mathbf{J}_{\mathcal{H}} \mathbf{y}; \tag{10.14}$$

$$\Delta\mathbf{J}_{\mathcal{H}} = \left[\left(\sum_{(k,j)\in\mathcal{H}} \left([\mathbf{P}]_{kj} - \epsilon_{kj}\right) \mathbf{e}_k^K {\mathbf{e}_j^N}^T\right) \otimes \mathbf{1}_N\right]^T \mathbf{A}^T.$$

Algorithm 10.6 Greedy edge hardening algorithm

```
Input: $\mathbf{J} = (\mathbf{P} \otimes \mathbf{1}_N)^T \mathbf{A}^T$, number of hardened edges $\eta$, $\{\epsilon_{kj}\}_{k \in \{1,2,\ldots,K\}, j \in \{1,2,\ldots,N\}}$
Output: modified compromise probability matrix $\mathbf{P}^\eta$
if recalculating score then
    Initialization: $\mathbf{P}^\eta = \mathbf{P}$. $\mathbf{J}^{old} = \mathbf{J}$.
    for $z = 1$ to $\eta$ do
        1. Compute the leading eigenvector $\mathbf{y}$ of $\mathbf{J}^{old}$
        2. Compute score $\phi\big((k,j)\big) = \mathbf{y}^T \Delta \mathbf{J}^{old}_{(k,j)} \mathbf{y}$
        3. Obtain $(k^*, j^*) = \arg\max_{k,j} \phi\big((k,j)\big)$
        4. Edge hardening: $[\mathbf{P}^\eta]_{k^* j^*} = \epsilon_{k^* j^*}$
        5. $\mathbf{J}^{old} = (\mathbf{P}^\eta \otimes \mathbf{1}_N)^T \mathbf{A}^T$ (see Appendix H.16)
    end for
else
    Initialization: $\mathbf{P}^\eta = \mathbf{P}$
    1. Compute the leading eigenvector $\mathbf{y}$ of $\mathbf{J}$
    2. Compute score $\phi\big((k,j)\big) = \mathbf{y}^T \Delta \mathbf{J}_{(k,j)} \mathbf{y}$
    3. Find the $\eta$ edges of highest scores
    4. Store this set of $\eta$ edges in $\mathcal{H}$
    5. Edge hardening: $[\mathbf{P}^\eta]_{kj} = \epsilon_{kj}$ for all $(k,j) \in \mathcal{H}$
end if
```

The proof of (10.14) can be found in Appendix H.14.

Let $\phi(\mathcal{H}) = \mathbf{y}^T \Delta \mathbf{J}_{\mathcal{H}} \mathbf{y}$ be a score function that reflects the effect of the edge hardening set $\mathcal{H}$ on spectral radius reduction of $\mathbf{J}$. The lemma presented in Appendix H.15 shows that $\phi(\mathcal{H})$ is a monotonic increasing set function of $\mathcal{H}$. The following analysis shows that $\phi(\mathcal{H})$ is associated with a pair of upper and lower bounds on the spectral radius of $\mathbf{J}$ after edge hardening.

The edge hardening algorithm proposed in Algorithm 10.6 is a greedy algorithm that hardens the η edges of highest scores between applications and hosts, where the per-edge hardening score is defined as $\phi\big((k,j)\big) = \mathbf{y}^T \Delta \mathbf{J}_{(k,j)} \mathbf{y}$. Step 5 in Algorithm 10.6 with score recalculation can be updated efficiently by tracking the changes in the matrix $\mathbf{J}$ caused by Step 4 (see Appendix H.16). The following theorem shows that the hardened edge set obtained from Algorithm 10.6 without score recalculation is a maximizer of $\phi(\mathcal{H})$.

Theorem 10.6. (Greedy edge hardening without score recalculation) *For any hard-*

Algorithm 10.7 Greedy node hardening algorithm

```
**Input:** edge score $\phi((k,j))$, number of hardened nodes $\zeta$, $\{\alpha_j\}_{j=1}^{N}$
**Output:** modified node hardening vector $\tilde{\mathbf{a}}$
Initialization: $\tilde{\mathbf{a}} = \mathbf{a}$
1. Compute edge hardening score $\phi\big((k,j)\big)$ for all $k \in \{1,2,\ldots,K\}$ and $j \in \{1,2,\ldots,N\}$
2. Compute node hardening score $\rho(j) = \sum_{k=1}^{K} \phi((k,j))$ for all $j \in \{1,2,\ldots,N\}$
3. Find the first $\zeta$ nodes of highest scores and store this set of $\zeta$ nodes in $\mathcal{H}^{node}$
4. Node hardening: $[\tilde{\mathbf{a}}]_j = \alpha_j$ for all $j \in \mathcal{H}^{node}$
```

ening set $\mathcal{H}$ of cardinality $|\mathcal{H}| = \eta \geq 1$, let $\mathcal{H}^{\eta}$ with $|\mathcal{H}^{\eta}| = \eta$ be the greedy hardening set obtained from Algorithm 10.6. Then $\mathcal{H}^{\eta}$ is a maximizer of $\phi(\mathcal{H})$.

Proof. The proof can be found in Appendix H.17. □

Furthermore, the following theorem shows that Algorithm 10.6 without score recalculation has bounded performance guarantee on spectral radius reduction of $\mathbf{J}$ relative to that of the optimal batch edge hardening set for which the computation complexity is combinatorial.

Theorem 10.7. (Performance garantee of greedy edge hardening without score recalculation) *For any hardening set $\mathcal{H}$ of cardinality $|\mathcal{H}| = \eta \geq 1$, $\lambda_{\max}(\mathbf{J}) \geq \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right)$. Furthermore, let $\mathcal{H}^{opt}$ with $|\mathcal{H}^{opt}| = \eta$ be the optimal hardening set that minimizes $\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right)$ and let $\mathcal{H}^{\eta}$ with $|\mathcal{H}^{\eta}| = \eta$ be the hardening set that maximizes $\phi(\mathcal{H})$. If $\lambda_{\max}(\mathbf{J}) > 0$ and $\phi(\mathcal{H}^{\eta}) > 0$, then there exists some constant $c'' > 0$ such that*

$$\lambda_{\max}(\mathbf{J}) - \phi(\mathcal{H}^{\eta}) \leq \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{opt})\right) \leq \lambda_{\max}(\mathbf{J}) - c'' \cdot \phi(\mathcal{H}^{\eta}).$$

Proof. The proof can be found in Appendix H.19. □

Moreover, the corollary below shows that Algorithm 10.6 with score recalculation can successively reduce the spectral radius of $\mathbf{J}$.

Corollary 10.8. (Greedy edge hardening with score recalculation) *Let $\mathbf{y}_{\mathcal{H}}$ denote the largest eigenvector of $\widetilde{\mathbf{J}}(\mathcal{H})$ and let $\phi_{\mathcal{H}}\big((k,j)\big) = \mathbf{y}_{\mathcal{H}}^T \widetilde{\mathbf{J}}\left(\mathcal{H} \cup (k,j)\right) \mathbf{y}_{\mathcal{H}}$. For any edge* *hardening set $\mathcal{H}$, let (k^*, j^*) be a maximizer of $\phi_{\mathcal{H}}\left((k,j)\right)$. Then $\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right) \geq \lambda_{\max}\left(\widetilde{\mathbf{J}}\left(\mathcal{H} \cup (k^*, j^*)\right)\right)$. Furthermore, if $\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right) > 0$ and $\phi_{\mathcal{H}}\left((k^*, j^*)\right) > 0$, then $\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right) > \lambda_{\max}\left(\widetilde{\mathbf{J}}\left(\mathcal{H} \cup (k^*, j^*)\right)\right)$.*

Proof. The proof can be found in Appendix H.18. □

Lastly, for node hardening we use the edge hardening score $\phi((k,j))$ to define the node hardening score $\rho(j)$ for host j, where $\rho(j) = \sum_{k=1}^{K} \phi((k,j))$. In effect, node hardening on host j enhances its hardening level from $[\mathbf{a}]_j$ to a value $\alpha_j \in [[\mathbf{a}]_j, 1]$. A greedy node hardening algorithm based on the node hardening score is summarized in Algorithm 10.7. In Sec. 10.4 we also investigate the performance of two other node score functions based on $\mathbf{a}$ and $\mathbf{J}$ for greedy node hardening, namely $\rho^{\mathbf{a}}(j) = 1/[\mathbf{a}]_j$ and $\rho^{\mathbf{J}}(j) = \sum_{s=1}^{N} [\mathbf{J}]_{js}$.

10.4 Experimental Results

10.4.1 Dataset description and experiment setup

To demonstrate the effectiveness of the proposed segmentation and hardening strategies against lateral movement attacks, we use the event logs and network flows collected from a large enterprise to create a tripartite user-host-application graph as in Fig. 10.2 (a) for performance analysis. This graph contains 5863 users, 4474 hosts, 3 applications, 8413 user-host access records and 6230 host-application-host network flows. All experiments assume that the defender has no knowledge of which nodes are compromised and the defender only uses the given tripartite network configuration for segmentation and hardening.

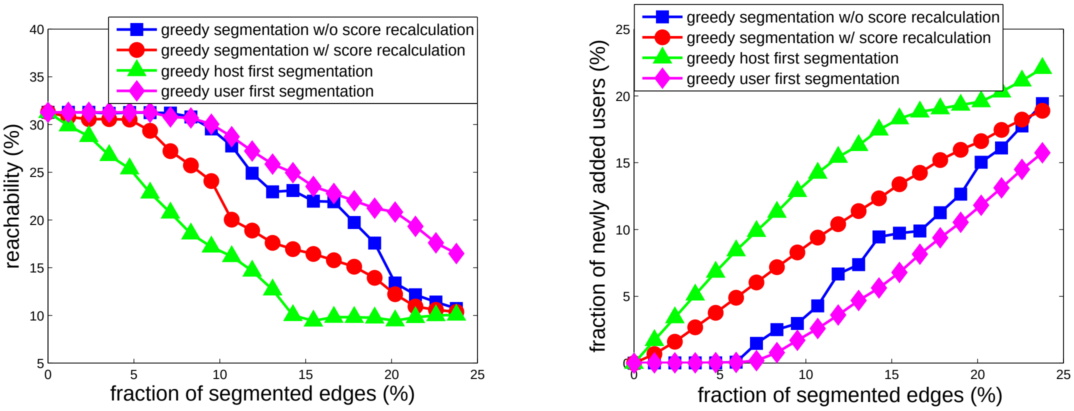
To simulate a lateral movement attack we randomly select 5 hosts (approximates 0.1% of total host number) as the initially compromised hosts and use the algorithms developed in Sec. 10.1 to evaluate the reachability, which is defined as the fraction of reachable hosts by propagating on the tripartite graph from the initially compromised hosts. The initial node hardening level of each host is independently and uniformly drawn from the unit interval between 0 and 1. The compromise probability matrix $\mathbf{P}$ is a random matrix where the fraction of nonzero entries is set to be 10% and each nonzero entry is independently and uniformly drawn from the unit internal between 0 and 1. The compromise probability after hardening, ϵ_{kj}, is set to be 10^{-5} for all k and j. All experimental results are averaged over 10 trials.

10.4.2 Lateral movement and segmentation on user-host graph

Fig. 10.3 shows the effect of different segmentation strategies proposed in Sec. 10.2 on the user-host graph. In particular, Fig. 10.3 (a) shows that greedy host-first segmentation strategy is the most effective approach to constrain reachability given the same number of segmented edges because accesses to high connectivity hosts (i.e., hubs) are segmented. For example, segmenting 15% of user-host accesses can reduce the reachability to nearly one third of its initial value. Greedy segmentation with score recalculation is shown to be more effective than that without score recalculation since it is adaptive to user-host access modification during segmentation. Greedy user-first segmentation strategy is not as effective as the other strategies since segmentation does not enforce any user-host access reduction and therefore after segmentation a user can still access the hosts but with different accounts.

Fig. 10.3 (b) shows the fraction of newly created accounts with respect to different segmentation strategies. There is clearly a trade-off between network robustness and implementation practicality since Fig. 10.3 suggests that segmentation strategies with better reachability reduction capability also lead to more additional accounts. However, in practice a user might be reluctant to use many accounts to pursue his/her daily jobs even though doing so can greatly mitigate the risk from lateral movement attacks.

We also investigate the impact of user-host access information asymmetry on lat-



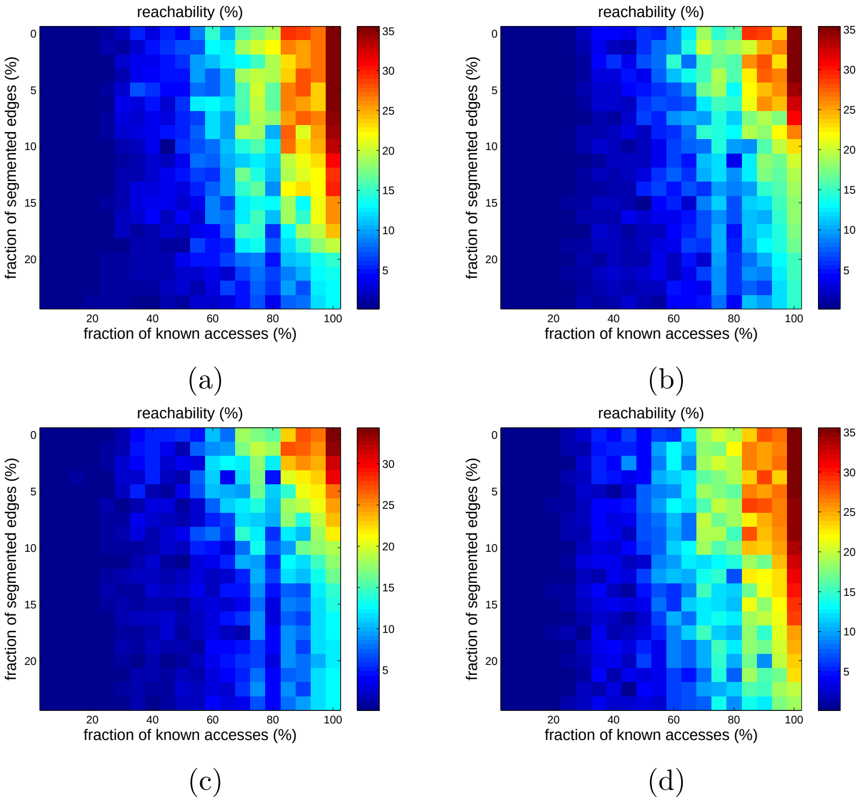
(a) (b)

Figure 10.3: The effect of segmentation on the user-host access graph. (a) Reachability with respect to different segmentation strategies. (b) Fraction of newly created user accounts from segmentation.

eral movement attacks. Information asymmetry means that the defender uses complete user-host access information for segmentation whereas the attacker launching a lateral movement attack can only leverage the known user-host access information. Fig. 10.4 shows that lateral movement attacks can be constrained when sufficient segmentation is implemented and the user-host access information is limited to an attacker, otherwise a surge in reachability is expected.

10.4.3 Lateral movement and hardening on host-application graph

Fig. 10.5 shows the effect of different hardening strategies proposed in Sec. 10.3 on the host-application graph. As shown in Fig. 10.5 (a), the proposed greedy edge hardening strategies with and without score recalculation have similar performance in reachability reduction, and they outperform the greedy heuristic strategy that hardens edges of highest compromise probability. This suggest that the proposed edge hardening strategies indeed finds the nontrivial edges affecting lateral movement. Fig. 10.5 (b) shows that the node hardening strategies using the node score function ρ and $\rho^{\mathbf{J}}$ lead to similar performance in reachability reduction, and they outperform the greedy heuristic strategy that hardens nodes of lowest hardening level. These



(a)

(b)

(c)

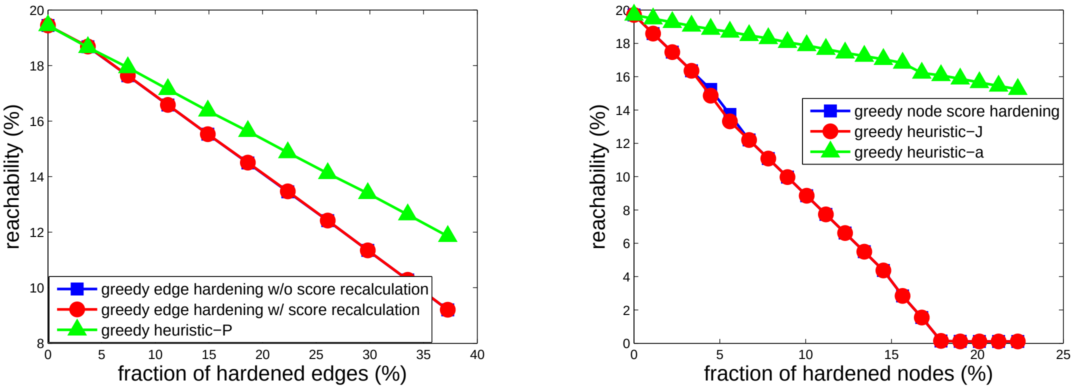
(d)

Figure 10.4: The effect of known user-host access information on lateral movement attacks. (a) Greedy segmentation without score recalculation. (b) Greedy segmentation with score recalculation. (c) Greedy host first segmentation. (d) Greedy user first segmentation. Lateral movement attacks can be constrained in terms of reachability when sufficient segmentation is implemented and the user-host access information is limited to an attacker.

results suggest the intuition of hardening the host of lowest security level might not be the best strategy for constraining lateral movement.

10.4.4 Lateral movement, segmentation and hardening on tripartite graph

Lastly, we investigate the joint effect of segmentation and hardening on constraining lateral movement attacks on the user-host-application tripartite graph. Fig. 10.6 shows the lateral movement reachability under a selected combination of segmentation and hardening strategies, namely greedy segmentation w/ score recalculation, greedy edge hardening w/ score recalculation, and greedy node hardening with score ρ. For clarity we only plot representative points to demonstrate the effectiveness. It



(a) (b)

Figure 10.5: The effect of hardening on host-application graph. (a) Reachability with respect to different edge hardening strategies. (b) Reachability with respect to different node hardening strategies.

can be observed that originally more than half of hosts can be compromised if no preventative actions are taken. Nonetheless, the proposed segmentation and hardening strategies can greatly mitigate the reachability of lateral movements to secure the network.

10.5 Benchmark: Performance Evaluation on Actual Lateral Movement Attacks

This section demonstrates the importance of incorporating the heterogeneity of a cyber system for enhancing the resilience to lateral movement attacks. Specifically, real lateral movement attacks taking place in an enterprise network are collected as a performance benchmark[2]. This dataset contains the communication patterns between 2010 hosts via 2 communication protocols, and therefore the enterprise network can be summarized as a bipartite host-application graph. It also contains lateral movements originated from a single compromised host, and in total includes 2001 propagation paths. The experiment in this section differs from the analysis in Sec. 10.4, as

[2] The dataset can be downloaded from https://sites.google.com/site/pinyuchenpage/datasets

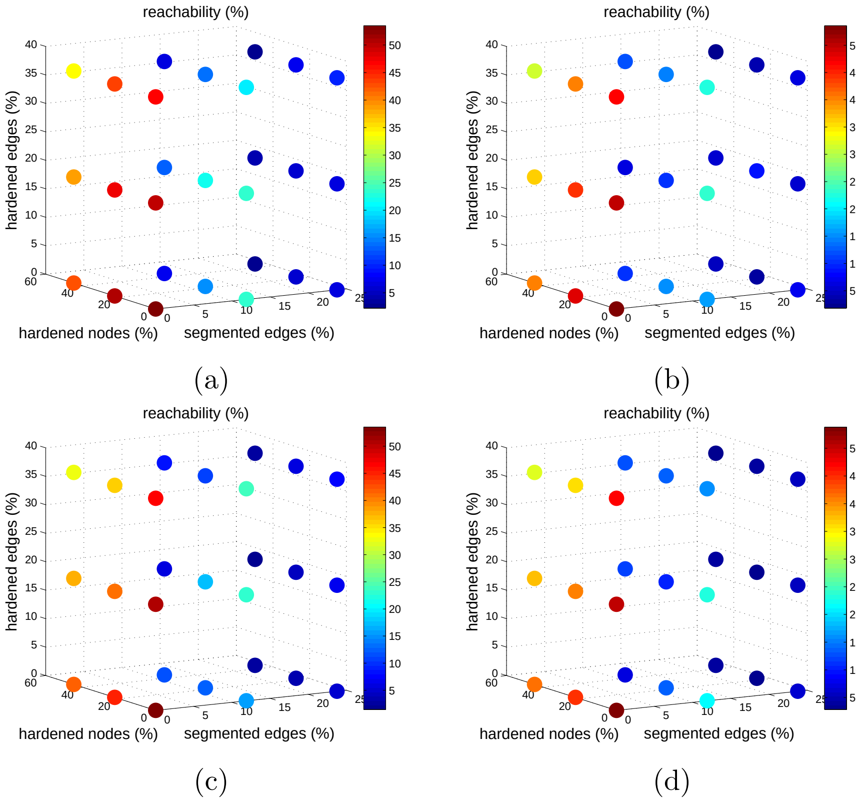


Figure 10.6: The effect of segmentation and hardening on lateral movement attack in user-host-application tripartite graph. (a) Greedy segmentation w/o score recalculation, greedy edge hardening w/o score recalculation, and greedy node hardening with score ρ. (b) Greedy segmentation w/ score recalculation, greedy edge hardening w/ score recalculation, and greedy node hardening with score ρ. (c) Greedy host first segmentation, greedy edge hardening w/o score recalculation, and greedy node hardening with score $\rho^{\mathbf{J}}$. (d) Greedy host first segmentation, greedy edge hardening w/ score recalculation, and greedy node hardening with score $\rho^{\mathbf{J}}$.

this dataset contains actual lateral movement traces on the host-application graph, whereas in Sec. 10.4 we have a complete user-host-application tripartite graph of an enterprise, but without the actual attack traces.

This benchmark dataset was collected from the network traffic of a cyber testbed running inside a OpenStack-based cloud with nearly 2000 virtual machine instances. Starting from a known machine (host), the attack involved logging from one machine to another using SSH. Implemented by automated scripts, on each machine the attack replicated to four other machines at the beginning of every hour. This process

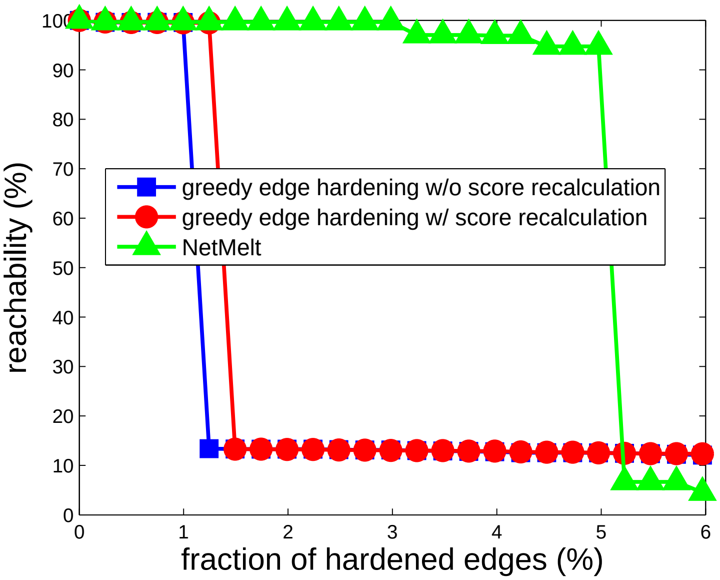


Figure 10.7: Performance evaluation on the collected benchmark dataset. Using our proposed approaches, the lateral movement attacks can be further restrained by incorporating the heterogeneity of the cyber system.

continued for 8 hours. We collected network traffic flows from each virtual machine and combined to produce a 16 GB packet capture dataset. Each packet information was further aggregated to produce "flow" level information, which can be interpreted as a "communication session" between two machines. As an example, when a client connects to the server, the client may send 5 packets and receive 10 packets of data from the server. The "flow" level data will combine these 15 data packets into a single "flow" to represent one interaction between the machines. Each flow record has the following elements: IP address and port information for both source and destination devices, protocol, flow start time, duration and message size. We infer the application by considering the protocol and destination address pair. As an example, a flow to destination port 22 over TCP protocol implies an SSH connection. To apply our proposed method to the cyber system against lateral movement attacks, we select the source and destination IP address, and the applications to build the host-application graph.

We compare the performance of our proposed edge hardening method (Algorithm

10.6) to the NetMelt algorithm [158], which is a well-known edge removal method for containing information diffusion on a homogeneous graph. The edges in the host-application bipartite graph are hardened sequentially according to the computed scores, and the initial compromise probability matrix $\mathbf{P}$ is set to be a matrix of ones. For every propagation path, the lateral movement will be contained if the edge it attempts to leverage is hardened. Since NetMelt can only deal with homogeneous graphs (in this case, the host-host graph), its recommendation on hardening a host pair is equivalent to hardening K corresponding host-application edges (in this case, $K=2$), whereas our method has better granularity for edge hardening by considering the connectivity structure of the host-application bipartite graph. The computational time complexity of NetMelt is $O(m\eta + N)$ [158], where m is the number of edges in the host-host graph, η is the number of hardened edges, and N is the number of hosts. Since the operation of leading eigenpair computation in Algorithm 10.6 is similar to NetMelt, the computation time complexity for Algorithm 10.6 without score recalculation is $O(m'\eta + N)$, where m' is the number of nonzero entries in the matrix $\mathbf{J}$. For Algorithm 10.6 with score recalculation, the computational time complexity is $O(m'\eta^2 + N\eta)$.

Fig. 10.7 shows the reachability of lateral movements with respect to the fraction of hardened edges. Initially the reachability is nearly 100%, suggesting that almost every host is vulnerable to lateral movement attacks without edge hardening. It can be observed that the proposed method (both with or without score recalculation) can restrain the reachability to roughly 10% by hardening less than 1.5% of edges, whereas NetMelt requires to harden more than 5% of edges to achieve comparable reachability, since it does not exploit the heterogeneity of the cyber system. Consequently, the results demonstrate the utility of incorporating heterogeneity for building resilient systems.

CHAPTER XI

Conclusion and Future Work

Many data mining and inference techniques are built upon certain actions on the associated graph representations. This dissertation has focused on two types of actions on graphs, namely *graph spectral decompositions* and *insertions and removals of nodes or edges*, for understanding their fundamental principles in graph data analytics, and proposing novel and efficient algorithms for spectral graph clustering in data science and for network resilience in cyber security. In addition to establishing theoretical foundations and providing performance guarantees on the developed algorithms, we have provided numerical experiments on both synthetic and real-world datasets to validate and complement the theory.

In Chapter II, we have proposed an efficient incremental eigenpair computation method for graph Laplacian matrices, which works by transforming a batch eigenvalue decomposition problem into a sequential leading eigenpair computation problem. The proposed method is elegant, robust and easy to implement using a scientific programming language, such as Matlab. We provided analytical proof of its correctness and demonstrated that it achieves significant reduction in computation time when compared with the batch computation method. It also serves as the cornerstone for incremental model order selection for graph clustering in single-layer and multilayer graphs in the following chapters.

In Chapter III, we have established a universal phase transition threshold on community detectability using the spectral modularity method for a general stochastic block model in single-layer graphs. The critical phase transition is universal in the sense that it does not depend on the ratio of community sizes as long as the community sized grow with a comparable rate.

In Chapter IV, we have established a phase transition analysis of spectral graph clustering (SGC) under the random interconnection model (RIM) in single-layer graphs. Under the RIM, we proved that there exists a critical value of the inter-cluster edge connection probability that separates SGC into two regimes: a regime where SGC is successful, and a regime where SGC is unsuccessful. We also provided analytical upper and lower bounds on the critical value for phase transition, and extended this framework to single-layer weighted graphs.

In Chapter V, we have proposed a framework for automated model order selection (AMOS) for SGC, which are applicable to unweighted and weighted single-layer graphs. The proposed AMOS algorithm is based on the phase transition analysis of SGC established in Chapter IV. It works by iterative SGC and finds the minimal model order (i.e., the number of clusters) that satisfies the phase transition criterion for clustering reliability. In addition, AMOS also provides statistical clustering reliability guarantees. Numerical results on real-world network data showed that the clusters found by AMOS are consistent with the ground truth meta information.

In Chapter VI, we have extended the phase transition analysis for SGC to multilayer graphs via convex layer aggregation under a multilayer signal plus noise model based on the RIM. A multilayer iterative model order selection algorithm (MIMOSA) has been proposed for SGC in multilayer graphs. MIMOSA features automated model order selection and layer weight adaption for finding common clusters shared among different layers. Numerical results on simulated multilayer graphs validate the phase transition analysis, and the experiments on real-world multilayer graphs show that the clusters found by MIMOSA result in improved performance in terms of multiple external and internal clustering metrics.

In Chapter VII, we proposed a centrality measure called local Fiedler vector centrality (LFVC) based on bounds on the sensitivity of algebraic connectivity to node or edge removals. We proved that LFVC relates to a monotonic submodular set function such that greedy node removals based on LFVC can be applied to identify the most vulnerable nodes or edges with bounded performance loss compared to the optimal combinatorial batch removal strategy. We also applied LFVC to deep community detection for discovering embedded clusters in graphs via edge or node removals. The proposed LFVC method provides better resolution for discovering important communities and key members in the studied real-world social network datasets.

In Chapter VIII, we have developed a novel method for identifying influential links for event propagation on Twitter. We utilized the network of networks (NoN) structure in real-world event propagation patterns on Twitter and proposed a left eigenvector score (LES) to identify the level of importance in event propagation for every follower link. Experiments on reducing event reachability via link removals show that exploiting the NoN structure and LES leads to superior performance over trivial methods using the number of followers for score calculation. Consequently, LES successfully identifies influential links for event propagation and offers new insights on modeling information dissemination in general networks.

In Chapter IX, we have studied network resilience to centrality attacks, where the resilience is evaluated in terms of its network connectivity after node or edge removals. We also proposed an edge rewiring method to enhance network resilience without introducing additional edges to the network. The results on the power grid of western US states show that the network is particularly vulnerable to LFVC attacks, and that the edge rewiring method can significantly improve network resilience with only a few edge rewires. Moreover, the proposed edge rewiring method can be implemented in a distributed fashion via in-network computation.

In Chapter X, we have developed a graph-theoretic framework for joint modeling of multiple dimensions of cyber behavior (user access control, application traffic) for enhancing cyber enterprise resiliency in an unified, tripartite graph model. Our experiments performed on a real dataset demonstrate the value and powerful insights from this unified model with respect to analysis performed on a single dimensional dataset. Through the tripartite graph model, the dominant factors affecting lateral movement are identified and effective algorithms are proposed to constrain the reachability with theoretical performance guarantees. We also synthesized a benchmark dataset containing traces of actual lateral movement attacks. The results showed that our proposed approach can effectively contain lateral movements by incorporating the heterogeneity of the system.

11.1 Future work

There are many interesting directions that are worthy of future study:

First, the phase transition analysis for spectral graph clustering in single-layer and multilayer graphs in Chapters IV and VI are developed upon the random interconnection model (RIM). Although the RIM includes popular block models such as the stochastic block model, the inferential limits of graph clustering under the RIM has not been explored. In addition, the RIM assumes Bernoulli-type noisy inter-cluster connections. The generalization and extension to more complicated inter-cluster connection models will be very helpful toward complete understanding of graph spectral decompositions for graph clustering. Moreover, nonlinear layer aggregation for multilayer graphs, and phase transition analysis of eigendecomposition on tensors, are two directions that are worthy of further investigation.

Second, it has been known that graph spectral decompositions may not possess nice concentration properties in sparse graphs due to sparsity in the matrix representation. However, recent works [92, 93] have shown promising results that the concentration properties of graph spectral decompositions hold in sparse graphs via simple regularization techniques. As a result, how regularization can improve the performance of graph clustering techniques developed in this dissertation, such as AMOS in Chapter V, MIMOSA in Chapter VI, and LFVC deep community detection in Chapter VII, are worthy of further study.

Third, many empirical results on graph clustering have reported that joint actions on graphs via graph spectral decompositions and node or edge removals can significantly improve its performance, especially for graphs possessing heterogeneous connectivity structure such as the power-law networks or overlapping communities. However, the effect of these joint actions on graph clustering are not fully understood and the theoretical analysis is still lacking. Furthermore, the established theoretic framework can be readily applied to analyzing the performance of graph summarization techniques including graph sparsification and random sampling for graph clustering and other tasks for graph data analytics.

Lastly, the algorithms developed for event propagation on Twitter and network resilience for cyber security in Chapters VIII, IX, and X are based on a static graph setting. In practice, a cyber system or an online social network may involve network dynamics and only partial information may be given for inference and decision making. Online algorithms and adaptive methods on graphs that take into account the network dynamics and incomplete network information are interesting and challenging directions.

APPENDICES

APPENDIX A

Appendix of Chapter II

A.1 Proof of Lemma 2.1

Since $\mathbf{L}$ is a positive semidefinite (PSD) matrix, $\lambda_i(\mathbf{L}) \geq 0$ for all i. Since G is a connected graph, by (1.2) $\mathbf{L}\mathbf{1}_n = (\mathbf{S} - \mathbf{W})\mathbf{1}_n = \mathbf{0}_n$. Therefore, by the PSD property we have $(\lambda_1(\mathbf{L}), \mathbf{v}_1(\mathbf{L})) = (0, \frac{\mathbf{1}_n}{\sqrt{n}})$. Moreover, since $\mathbf{L}$ is a symmetric real-valued square matrix, from (1.2) we have

$$\mathrm{trace}(\mathbf{L}) = \sum_{i=1}^{n} \mathbf{L}_{ii} = \sum_{i=1}^{n} \lambda_i(\mathbf{L}) = \sum_{i=1}^{n} s_i = s. \tag{A.1}$$

By the PSD property of $\mathbf{L}$, we have $\lambda_n(\mathbf{L}) < s$ since $\lambda_2(\mathbf{L}) > 0$ for any connected graph. Therefore, by the orthogonality of eigenvectors of $\mathbf{L}$ (i.e., $\mathbf{1}_n^T \mathbf{v}_i(\mathbf{L}) = 0$ for all $i \geq 2$) the eigenvalue decomposition of $\widetilde{\mathbf{L}}$ can be represented as

$$\begin{aligned} \widetilde{\mathbf{L}} &= \sum_{i=2}^{n} \lambda_i(\mathbf{L}) \mathbf{v}_i(\mathbf{L}) \mathbf{v}_i^T(\mathbf{L}) + \frac{s}{n} \mathbf{1}_n \mathbf{1}_n^T \\ &= \sum_{i=1}^{n} \lambda_i(\widetilde{\mathbf{L}}) \mathbf{v}_i(\widetilde{\mathbf{L}}) \mathbf{v}_i^T(\widetilde{\mathbf{L}}), \end{aligned} \tag{A.2}$$

where $(\lambda_n(\widetilde{\mathbf{L}}), \mathbf{v}_n(\widetilde{\mathbf{L}})) = (s, \frac{\mathbf{1}_n}{\sqrt{n}})$ and $(\lambda_i(\widetilde{\mathbf{L}}), \mathbf{v}_i(\widetilde{\mathbf{L}})) = (\lambda_{i+1}(\mathbf{L}), \mathbf{v}_{i+1}(\mathbf{L}))$ for $1 \leq i \leq n-1$.

A.2 Proof of Lemma 2.3

The graph Laplacian matrix of a disconnected graph consisting of δ connected components can be represented as a matrix with diagonal block structure, where each block in the diagonal corresponds to one connected component in G[26], i.e.,

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{L}_2 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \ddots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{L}_\delta \end{bmatrix}, \tag{A.3}$$

where $\mathbf{L}_k$ is the graph Laplacian matrix of k-th connected component in G. From the proof of Lemma 2.1 each connected component contributes to exactly one zero eigenvalue for $\mathbf{L}$, and

$$\lambda_n(\mathbf{L}) < \sum_{k=1}^{\delta} \sum_{i \in \text{component } k} \lambda_i(\mathbf{L}_k) = \sum_{k=1}^{\delta} \sum_{i \in \text{component } k} s_i = s. \tag{A.4}$$

Therefore, we have the results in Lemma 2.3.

A.3 Proof of Corollary 2.2

Recall from (1.3) that $\mathbf{L}_\mathcal{N} = \mathbf{S}^{-\frac{1}{2}}\mathbf{L}\mathbf{S}^{-\frac{1}{2}}$, and also we have $\mathbf{L}_\mathcal{N}\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n = \mathbf{S}^{-\frac{1}{2}}\mathbf{L}\mathbf{1}_n = \mathbf{0}_n$. Moreover, it can be shown that $0 \leq \lambda_1(\mathbf{L}_\mathcal{N}) \leq \lambda_2(\mathbf{L}_\mathcal{N}) \leq \ldots \leq \lambda_n(\mathbf{L}_\mathcal{N}) \leq 2$ [98], and $\lambda_2(\mathbf{L}_\mathcal{N}) > 0$ if G is connected. Following the same derivation for Lemma 2.1 we obtain the corollary. Note that $\mathbf{S}^{\frac{1}{2}} = \text{diag}(\sqrt{s_1}, \sqrt{s_2}, \ldots, \sqrt{s_n})$ and $(\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n)^T\mathbf{S}^{\frac{1}{2}}\mathbf{1}_n = \mathbf{1}_n^T\mathbf{S}\mathbf{1}_n = s$.

A.4 Proof of Corollary 2.4

The results can be obtained by following the same derivation procedure in Sec. A.2 and the fact that $\lambda_n(\mathbf{L}_{\mathcal{N}}) \leq 2$ [98].

A.5 Proof of Theorem 2.6

From Lemma 2.1,

$$\mathbf{L} + \frac{s}{n}\mathbf{1}_n\mathbf{1}_n^T + \mathbf{V}_K\mathbf{\Lambda}_K\mathbf{V}_K^T = \sum_{i=K+1}^{n} \lambda_i(\mathbf{L})\mathbf{v}_i(\mathbf{L})\mathbf{v}_i^T(\mathbf{L}) + \sum_{i=2}^{K} s \cdot \mathbf{v}_i(\mathbf{L})\mathbf{v}_i^T(\mathbf{L}) + \frac{s}{n}\mathbf{1}_n\mathbf{1}_n^T, \tag{A.5}$$

which is a valid eigenpair decomposition that can be seen by inflating the K smallest eigenvalues of $\mathbf{L}$ to s with the originally paired eigenvectors. Using (A.5) we obtain the eigenvalue decomposition of $\widetilde{\mathbf{L}}$ as

$$\begin{aligned}\widetilde{\mathbf{L}} &= \mathbf{L} + \mathbf{V}_K\mathbf{\Lambda}_K\mathbf{V}_K^T + \frac{s}{n}\mathbf{1}_n\mathbf{1}_n^T - s\mathbf{I} \\ &= \sum_{i=K+1}^{n} (\lambda_i(\mathbf{L}) - s)\mathbf{v}_i(\mathbf{L})\mathbf{v}_i^T(\mathbf{L}).\end{aligned} \tag{A.6}$$

Since $0 \leq \lambda_{K+1}(\mathbf{L}) \leq \lambda_{K+2}(\mathbf{L}) \leq \ldots \leq \lambda_n(\mathbf{L})$, we have $|\lambda_{K+1}(\mathbf{L}) - s| \geq |\lambda_{K+2}(\mathbf{L}) - s| \geq \ldots \geq |\lambda_n(\mathbf{L}) - s|$. Therefore, the eigenpair $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L}))$ can be obtained by computing the leading eigenpair of $\widetilde{\mathbf{L}}$. In particular, if $\mathbf{L}$ has distinct eigenvalues, then the leading eigenpair of $\widetilde{\mathbf{L}}$ is unique. Therefore, by (A.6) we have the relation

$$(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L})) = (\lambda_1(\widetilde{\mathbf{L}}) + s, \mathbf{v}_1(\widetilde{\mathbf{L}})). \tag{A.7}$$

A.6 Proof of Tandheorem 2.7

First observe from (A.3) that $\mathbf{L}$ has δ zero eigenvalues since each connected component contributes to exactly one zero eigenvalue for $\mathbf{L}$. Following the same derivation procedure in the proof of Theorem 2.6 and using Lemma 2.3, we have

$$\begin{aligned}\widetilde{\mathbf{L}} &= \mathbf{L} + \mathbf{V}_{K,\delta}\mathbf{\Lambda}_{K,\delta}\mathbf{V}_{K,\delta}^T + s\mathbf{V}_\delta\mathbf{V}_\delta^T - s\mathbf{I} \\ &= \sum_{i=K+1,K\geq\delta}^{n} (\lambda_i(\mathbf{L}) - s)\mathbf{v}_i(\mathbf{L})\mathbf{v}_i^T(\mathbf{L}). \end{aligned} \tag{A.8}$$

Therefore, the eigenpair $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L}))$ can be obtained by computing the leading eigenpair of $\widetilde{\mathbf{L}}$. If $\mathbf{L}$ has distinct nonzero eigenvalues (i.e, $\lambda_{\delta+1}(\mathbf{L}) < \lambda_{\delta+2}(\mathbf{L}) < \ldots < \lambda_n(\mathbf{L})$), we obtain the relation $(\lambda_{K+1}(\mathbf{L}), \mathbf{v}_{K+1}(\mathbf{L})) = (\lambda_1(\widetilde{\mathbf{L}}) + s, \mathbf{v}_1(\widetilde{\mathbf{L}}))$.

APPENDIX B

Appendix of Chapter III

B.1 Proof of (3.22)

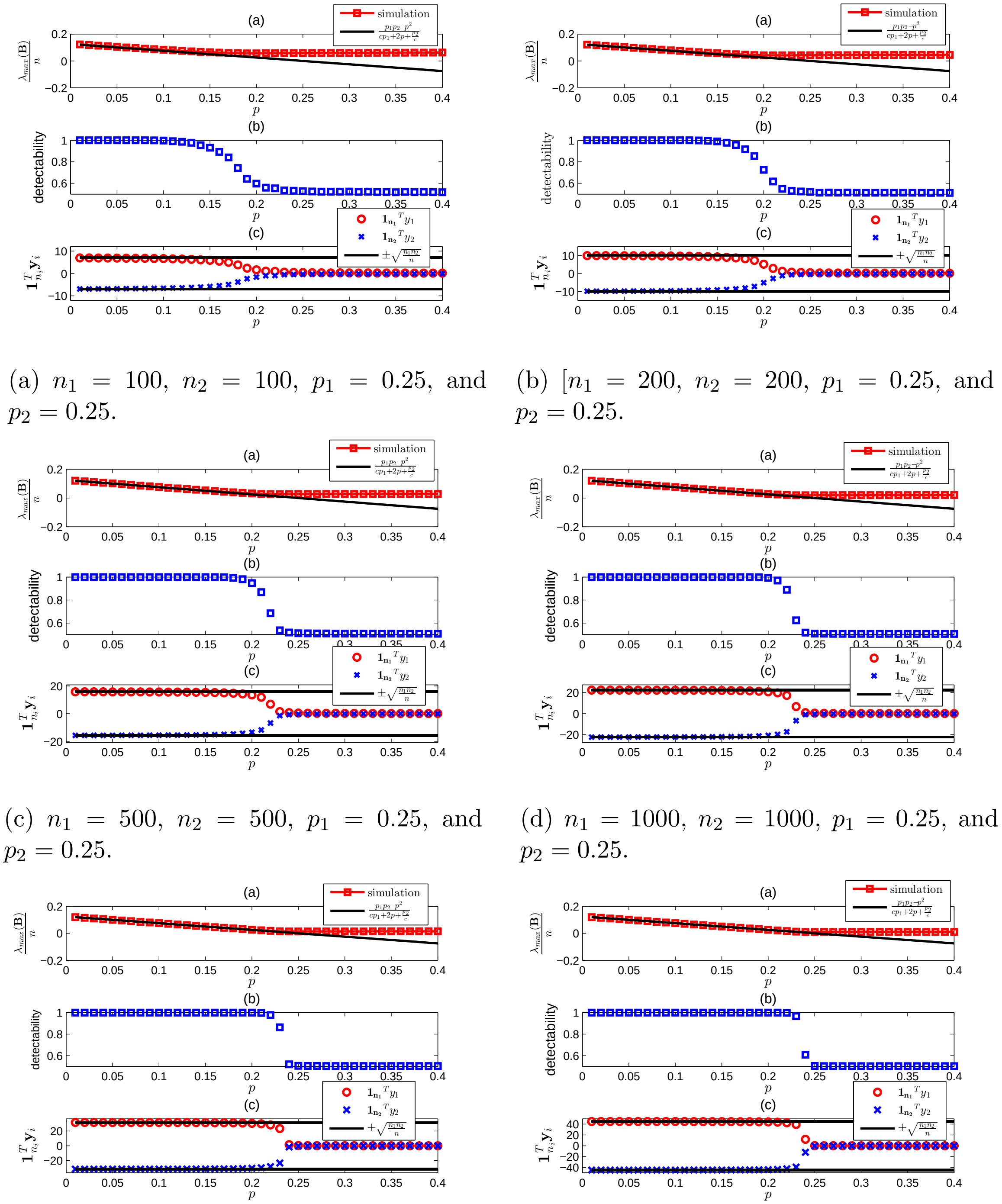
We prove the result by showing $\frac{\mathbf{y}_1^T\mathbf{B}_1\mathbf{y}_1}{n} \xrightarrow{\text{a.s.}} 0$ and $\frac{\mathbf{y}_2^T\mathbf{B}_2\mathbf{y}_2}{n} \xrightarrow{\text{a.s.}} 0$ such that $\sqrt{\frac{nn_1}{n_2}}\mathbf{y}_1 \xrightarrow{\text{a.s.}} \pm\mathbf{1}_{n_1}$ and $\sqrt{\frac{nn_2}{n_1}}\mathbf{y}_2 \xrightarrow{\text{a.s.}} \mp\mathbf{1}_{n_2}$ due to the facts that the vector of all ones is always in the null space of a modularity matrix and $\mathbf{y}_1^T\mathbf{1}_{n_1} + \mathbf{y}_2^T\mathbf{1}_{n_2} = 0$. We prove this statement by contradiction. Assume $\mathbf{y}_1$ and $\mathbf{y}_2$ converge almost surely to other vectors such that $\frac{\mathbf{y}_1^T\mathbf{B}_1\mathbf{y}_1}{n} \to c_4 \neq 0$ and $\frac{\mathbf{y}_2^T\mathbf{B}_2\mathbf{y}_2}{n} \to c_5 \neq 0$ and $c_4 + c_5 = 0$ in order to satisfy (3.21). By the concentration results in (3.12) and (3.13), we have

$$\begin{aligned}
\frac{\mathbf{y}_1^T\mathbf{B}_1\mathbf{y}_1}{n} &= \frac{\mathbf{y}_1^T\left(\mathbf{A}_1 - b_1\widetilde{\mathbf{d}}_1\widetilde{\mathbf{d}}_1^T\right)\mathbf{y}_1}{n} \\
&\xrightarrow{\text{a.s.}} \frac{\mathbf{y}_1^T\left(p_1\mathbf{1}_{n_1}\mathbf{1}_{n_1}^T - \frac{1}{{n_1}^2p_1}\cdot {n_1}^2p_1^2\mathbf{1}_{n_1}\mathbf{1}_{n_1}^T\right)\mathbf{y}_1}{n} \\
&= 0, \qquad\qquad \text{(B.1)}
\end{aligned}$$

and similarly $\frac{\mathbf{y}_2^T\mathbf{B}_2\mathbf{y}_2}{n} \xrightarrow{\text{a.s.}} 0$, which contradicts the assumption that $\frac{\mathbf{y}_1^T\mathbf{B}_1\mathbf{y}_1}{n} \xrightarrow{\text{a.s.}} c_4 \neq 0$ and $\frac{\mathbf{y}_2^T\mathbf{B}_2\mathbf{y}_2}{n} \xrightarrow{\text{a.s.}} c_5 \neq 0$. Therefore $\sqrt{\frac{nn_1}{n_2}}\mathbf{y}_1 \xrightarrow{\text{a.s.}} \pm\mathbf{1}_{n_1}$ and $\sqrt{\frac{nn_2}{n_1}}\mathbf{y}_2 \xrightarrow{\text{a.s.}} \mp\mathbf{1}_{n_2}$.

B.2 The Effect of Community Size on Phase Transition

To investigate the effect of community size on phase transition, we generate synthetic communities from the stochastic block model with different community sizes by fixing $c = 1$ and $p_1 = p_2 = 0.25$. The predicted phase transition threshold in (3.24) is $p^* = 0.25$. The results (averaged for 100 runs) are shown in Fig. B.1. The phase transition is apparent for small community size in the sense that the spectral modularity method fails to detect the communities in the super-critical regime (i.e., the $p > p^*$ regime). In the sub-critical regime (i.e., the $p \leq p^*$ regime), we observe an intermediate regime of community detectability for small community size, and this intermediate regime vanishes as we increase the community size. This can be explained by the fluctuation of finite community size on the concentration results in (3.18), (3.19), (3.22), and (3.24). By concentration theory the fluctuation decreases with the increase of community size, and an abrupt transition occurs at the phase transition threshold p^* when $n_1, n_2 \to \infty$ and $\frac{n_1}{n_2} \to c > 0$.



(a) $n_1 = 100$, $n_2 = 100$, $p_1 = 0.25$, and $p_2 = 0.25$.

(b) [$n_1 = 200$, $n_2 = 200$, $p_1 = 0.25$, and $p_2 = 0.25$.

(c) $n_1 = 500$, $n_2 = 500$, $p_1 = 0.25$, and $p_2 = 0.25$.

(d) $n_1 = 1000$, $n_2 = 1000$, $p_1 = 0.25$, and $p_2 = 0.25$.

(e) $n_1 = 2000$, $n_2 = 2000$, $p_1 = 0.25$, and $p_2 = 0.25$.

(f) $n_1 = 4000$, $n_2 = 4000$, $p_1 = 0.25$, and $p_2 = 0.25$.

Figure B.1: The effect of community size on phase transition. The phase transition phenomenon hold for communities of small sizes, and the empirical phase transition threshold gets closer to the predicted asymptotic threshold $p^* = 0.25$ as the community size increases.

APPENDIX C

Appendix of Chapter IV

C.1 Proof of Theorem 4.1

Based on the partitioned matrix representation of $\mathbf{A}$ in (1.1), define the induced graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$. In particular, the (i,j)-th block is a matrix $\mathbf{L}_{ij}$ of dimension $n_i \times n_j$ satisfying

$$\mathbf{L}_{ij} = \begin{cases} \mathbf{L}_i + \sum_{z=1,\ z \neq i}^{K} \mathbf{D}_{iz}, & \text{if } i = j, \\ -\mathbf{C}_{ij}, & \text{if } i \neq j, \end{cases} \tag{C.1}$$

where $\mathbf{L}_i$ is the graph Laplacian matrix of $\mathbf{A}_i$, $\mathbf{D}_{ij} = \text{diag}(\mathbf{C}_{ij}\mathbf{1}_{n_j})$ is the diagonal degree matrix contributed by the inter-cluster edges between clusters i and j. Applying (C.1) to (4.1), let $\boldsymbol{\nu} \in \mathbb{R}^{(K-1)}$ and $\mathbf{U} \in \mathbb{R}^{(K-1)\times(K-1)}$ with $\mathbf{U} = \mathbf{U}^T$ be the Lagrange multiplier of the constraints $\mathbf{X}^T\mathbf{1}_n = \mathbf{0}_{K-1}$ and $\mathbf{X}^T\mathbf{X} = \mathbf{I}_{K-1}$, respectively. The Lagrangian function is

$$\Gamma(\mathbf{X}) = \text{trace}(\mathbf{X}^T\mathbf{L}\mathbf{X}) - \boldsymbol{\nu}^T\mathbf{X}^T\mathbf{1}_n - \text{trace}\Big(\mathbf{U}(\mathbf{X}^T\mathbf{X} - \mathbf{I}_{K-1})\Big). \tag{C.2}$$

Let $\mathbf{Y} \in \mathbb{R}^{n\times(K-1)}$ be the solution of (4.1). Differentiating (C.2) with respect to $\mathbf{X}$ and substituting $\mathbf{Y}$ into the equations, we obtain the optimality condition

$$2\mathbf{L}\mathbf{Y} - \mathbf{1}_n\boldsymbol{\nu}^T - 2\mathbf{Y}\mathbf{U} = \mathbf{O}, \tag{C.3}$$

where $\mathbf{O}$ is a matrix of zero entries. Left multiplying (C.3) by $\mathbf{1}_n^T$, we obtain

$$\boldsymbol{\nu} = \mathbf{0}_{K-1}. \tag{C.4}$$

Left multiplying (C.3) by $\mathbf{Y}^T$ and using (C.4) we have

$$\mathbf{U} = \mathbf{Y}^T\mathbf{L}\mathbf{Y} = \mathrm{diag}(\lambda_2(\mathbf{L}), \lambda_3(\mathbf{L}), \ldots, \lambda_K(\mathbf{L})), \tag{C.5}$$

which we denote by the diagonal mateix $\boldsymbol{\Lambda}$. Hence by (4.1) we have

$$S_{2:K}(\mathbf{L}) = \mathrm{trace}(\mathbf{U}). \tag{C.6}$$

Now let $\mathbf{X} = [\mathbf{X}_1^T, \mathbf{X}_2^T, \ldots, \mathbf{X}_K^T]^T$ and $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T$, where $\mathbf{X}_k \in \mathbb{R}^{n_k\times(K-1)}$ and $\mathbf{Y}_k \in \mathbb{R}^{n_k\times(K-1)}$. With representation (C.5), the Lagrangian function in (C.2) can be written as

$$\begin{aligned}\Gamma(\mathbf{X}) = &\sum_{k=1}^{K}\mathrm{trace}(\mathbf{X}_k^T\mathbf{L}_k\mathbf{X}_k) + \sum_{k=1}^{K}\sum_{j=1,j\neq k}^{K}\mathrm{trace}(\mathbf{X}_k^T\mathbf{D}_{kj}\mathbf{X}_k) - \sum_{k=1}^{K}\sum_{j=1,j\neq k}^{K}\mathrm{trace}(\mathbf{X}_k^T\mathbf{C}_{kj}\mathbf{X}_j)\\ &- \sum_{k=1}^{K}\mathrm{trace}(\mathbf{U}\mathbf{X}_k^T\mathbf{X}_k) + \mathrm{trace}(\mathbf{U}).\end{aligned} \tag{C.7}$$

Differentiating (C.7) with respect to $\mathbf{X}_k$ and substituting $\mathbf{Y}_k$ into the equation, we obtain the optimality condition that for all $k \in \{1, 2, \ldots, K\}$,

$$\mathbf{L}_k \mathbf{Y}_k + \sum_{j=1, j \neq k}^{K} \mathbf{D}_{kj} \mathbf{Y}_k - \sum_{j=1, j \neq k}^{K} \mathbf{C}_{kj} \mathbf{Y}_j - \mathbf{Y}_k \mathbf{U} = \mathbf{O}. \tag{C.8}$$

Using results from the Talagrand's concentration theorem [150], the Latala's theorem [91] and the fact that each entry in $\mathbf{C}_{ij}$ is a Bernoulli random variable we can show that

$$\frac{\mathbf{C}_{ij}}{\sqrt{n_i n_j}} \xrightarrow{\text{a.s.}} p_{ij} \mathbf{1}\mathbf{1}^T \tag{C.9}$$

as $n_i, n_j \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, where $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence and $\mathbf{1}$ is the constant vector of unit norm. The proof of (C.9) can be found in [25]. Hence we have

$$\frac{\mathbf{D}_{ij}}{n_j} = \frac{\mathrm{diag}(\mathbf{C}_{ij} \mathbf{1}_{n_j})}{n_j} \xrightarrow{\text{a.s.}} p_{ij} \mathbf{I}. \tag{C.10}$$

The condition that $\frac{n_{\min}}{n_{\max}} \to c > 0$ guarantees that the cluster sizes grow at comparable rates so that (C.9) holds for all $\mathbf{C}_{ij}$. Using (C.10) and left multiplying (C.8) by $\frac{\mathbf{1}_{n_k}^T}{n}$ gives

$$\frac{1}{n} \left[\sum_{j=1, j \neq k}^{K} n_j p_{kj} \mathbf{1}_{n_k}^T \mathbf{Y}_k - \sum_{j=1, j \neq k}^{K} n_k p_{kj} \mathbf{1}_{n_j}^T \mathbf{Y}_j - \mathbf{1}_{n_k}^T \mathbf{Y}_k \mathbf{U} \right] \xrightarrow{\text{a.s.}} \mathbf{0}_{K-1}^T, \ \forall\ k. \tag{C.11}$$

Using the relation $\mathbf{1}_{n_K}^T \mathbf{Y}_{n_K} = -\sum_{j=1}^{K-1} \mathbf{1}_{n_j}^T \mathbf{Y}_{n_j}$, (C.11) can be represented as an asymptotic form of Sylvester's equation

$$\frac{1}{n} \left(\widetilde{\mathbf{A}} \mathbf{Z} - \mathbf{Z} \mathbf{\Lambda} \right) \xrightarrow{\text{a.s.}} \mathbf{O}, \tag{C.12}$$

where $\mathbf{Z} = [\mathbf{Y}_{n_1}^T \mathbf{1}_{n_1}, \mathbf{Y}_{n_2}^T \mathbf{1}_{n_2}, \ldots, \mathbf{Y}_{n_{K-1}}^T \mathbf{1}_{n_{K-1}}]^T \in \mathbb{R}^{(K-1) \times (K-1)}$, $\mathbf{\Lambda} = \mathrm{diag}(\lambda_2(\mathbf{L}), \lambda_3(\mathbf{L}),$

$\ldots, \lambda_K(\mathbf{L}))$, $\widetilde{\mathbf{A}}$ is the matrix specified in Theorem 4.1, and we use the relation $\mathbf{U} = \mathbf{Y}^T \mathbf{L} \mathbf{Y} = \mathbf{\Lambda}$ from (C.5).

Let $\otimes$ denote the Kronecker product defined in Appendix H.1 and let $\mathbf{vec}(\mathbf{Z})$ denote the vectorization operation of $\mathbf{Z}$ by stacking the columns of $\mathbf{Z}$ into a vector. (C.12) can be represented as

$$\frac{1}{n}(\mathbf{I}_{K-1} \otimes \widetilde{\mathbf{A}} - \mathbf{\Lambda} \otimes \mathbf{I}_{K-1})\mathbf{vec}(\mathbf{Z}) \xrightarrow{\text{a.s.}} \mathbf{0}, \tag{C.13}$$

where the matrix $\mathbf{I}_{K-1} \otimes \widetilde{\mathbf{A}} - \mathbf{\Lambda} \otimes \mathbf{I}_{K-1}$ is the Kronecker sum, denoted by $\widetilde{\mathbf{A}} \oplus -\mathbf{\Lambda}$. Observe that $\mathbf{vec}(\mathbf{Z}) = \mathbf{0}$ is always a trivial solution to (C.13), and if $\widetilde{\mathbf{A}} \oplus -\mathbf{\Lambda}$ is non-singular, $\mathbf{vec}(\mathbf{Z}) = \mathbf{0}$ is the unique solution to (C.13). Since $\mathbf{vec}(\mathbf{Z}) = \mathbf{0}$ implies $\mathbf{1}_{n_k}^T \mathbf{Y}_{n_k} = \mathbf{0}_{K-1}^T$ for all $k = 1, 2, \ldots, K$, the centroid $\frac{\mathbf{1}_{n_k}^T \mathbf{Y}_{n_k}}{n_k}$ of each cluster in the eigenspace is a zero vector, the clusters are not separable, and therefore accurate clustering is not possible. Therefore a sufficient condition for spectral graph clustering on the RIM to fail is that the matrix $\mathbf{I}_{K-1} \otimes \widetilde{\mathbf{A}} - \mathbf{\Lambda} \otimes \mathbf{I}_{K-1}$ be non-singular. Moreover, using the property of the Kronecker sum that the eigenvalues of $\widetilde{\mathbf{A}} \oplus -\mathbf{\Lambda}$ satisfy $\{\lambda_\ell(\widetilde{\mathbf{A}} \oplus -\mathbf{\Lambda})\}_{\ell=1}^{(K-1)^2} = \{\lambda_i(\widetilde{\mathbf{A}}) - \lambda_j(\mathbf{\Lambda})\}_{i,j=1}^{K-1}$, the sufficient condition on failure of spectral graph clustering on the RIM is that $\lambda_i\left(\frac{\widetilde{\mathbf{A}}}{n}\right) \neq \lambda_j\left(\frac{\mathbf{L}}{n}\right)$ for all $i = 1, 2, \ldots, K-1$ and $j = 2, 3, \ldots, K$.

C.2 Proof of Theorem 4.2

Following the derivations in AppendixC.1, since $\mathbf{1}_{n_k}^T \mathbf{Y}_k = -\sum_{j=1, j\neq k}^{K} \mathbf{1}_{n_j}^T \mathbf{Y}_j$, under the homogeneous RIM (i.e., $p_{ij} = p$), (C.11) can be simplified to

$$\left(p\mathbf{I}_{K-1} - \frac{\mathbf{U}}{n}\right) \mathbf{Y}_k^T \mathbf{1}_{n_k} \xrightarrow{\text{a.s.}} \mathbf{0}_{K-1}, \ \forall\ k. \tag{C.14}$$

This implies that one of the two cases below has to hold:

$$\text{Case 1: } \frac{\mathbf{U}}{n} \xrightarrow{\text{a.s.}} p\mathbf{I}_{K-1}; \tag{C.15}$$

$$\text{Case 2: } \mathbf{Y}_k^T\mathbf{1}_{n_k} \xrightarrow{\text{a.s.}} \mathbf{0}_{K-1}, \ \forall\, k. \tag{C.16}$$

Note that with (C.6) Case 1 implies

$$\frac{S_{2:K}(\mathbf{L})}{n} = \frac{\text{trace}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})}{n} = \frac{\text{trace}(\mathbf{U})}{n} \xrightarrow{\text{a.s.}} (K-1)p. \tag{C.17}$$

In Case 1, left multiplying (C.8) by $\frac{\mathbf{Y}_k^T}{n}$ and using (C.9) and (C.10) gives

$$\begin{aligned}\frac{1}{n}\Bigg[&\mathbf{Y}_k^T\mathbf{L}_k\mathbf{Y}_k + \sum_{j=1,j\neq k}^{K} n_j p \mathbf{Y}_k^T\mathbf{Y}_k \\ &- \sum_{j=1,j\neq k}^{K} p\mathbf{Y}_k^T\mathbf{1}_{n_k}\mathbf{1}_{n_j}^T\mathbf{Y}_j - \mathbf{Y}_k^T\mathbf{Y}_k\mathbf{U}\Bigg] \xrightarrow{\text{a.s.}} \mathbf{O}, \ \forall\, k.\end{aligned} \tag{C.18}$$

Since $\mathbf{1}_{n_k}^T\mathbf{Y}_k = -\sum_{j=1,j\neq k}^{K}\mathbf{1}_{n_j}^T\mathbf{Y}_j$, (C.18) can be simplified as

$$\begin{aligned}\frac{1}{n}\Big[&\mathbf{Y}_k^T\mathbf{L}_k\mathbf{Y}_k + (n-n_k)p\mathbf{Y}_k^T\mathbf{Y}_k + p\mathbf{Y}_k^T\mathbf{1}_{n_k}\mathbf{1}_{n_k}^T\mathbf{Y}_k \\ &-\mathbf{Y}_k^T\mathbf{Y}_k\mathbf{U}\Big] \xrightarrow{\text{a.s.}} \mathbf{O}, \ \forall\, k.\end{aligned} \tag{C.19}$$

Taking the trace of (C.19) and using (C.15), we have

$$\frac{1}{n}\left[\text{trace}(\mathbf{Y}_k^T\mathbf{L}_k\mathbf{Y}_k)\right] + \frac{p}{n}\left[\text{trace}(\mathbf{Y}_k^T\mathbf{1}_{n_k}\mathbf{1}_{n_k}^T\mathbf{Y}_k) - n_k\text{trace}(\mathbf{Y}_k^T\mathbf{Y}_k)\right] \xrightarrow{\text{a.s.}} 0, \ \forall\, k \tag{C.20}$$

Since (C.20) has to be satisfied for all values of p in Case 1, this implies the following two conditions have to hold simultaneously:

$$\begin{aligned}&\tfrac{1}{n}\left[\text{trace}(\mathbf{Y}_k^T\mathbf{L}_k\mathbf{Y}_k)\right]\xrightarrow{\text{a.s.}}0,\ \forall\ k;\\&\tfrac{1}{n}\left[\text{trace}(\mathbf{Y}_k^T\mathbf{1}_{n_k}\mathbf{1}_{n_k}^T\mathbf{Y}_k)-n_k\text{trace}(\mathbf{Y}_k^T\mathbf{Y}_k)\right]\xrightarrow{\text{a.s.}}0,\ \forall\ k.\end{aligned}\tag{C.21}$$

Since $\mathbf{L}_k$ is a positive semidefinite (PSD) matrix, $\mathbf{L}_k\mathbf{1}_{n_k}=\mathbf{0}_{n_k}$, and $\lambda_2(\mathbf{L}_k)>0$, $\frac{1}{n}\left[\text{trace}(\mathbf{Y}_k^T\mathbf{L}_k\mathbf{Y}_k)\right]\xrightarrow{\text{a.s.}}0$ implies that every column of $\mathbf{L}_k$ is a constant vector. Therefore, (C.21) implies that in Case 1,

$$\mathbf{Y}_k\xrightarrow{\text{a.s.}}\mathbf{1}_{n_k}\mathbf{1}_{K-1}^T\mathbf{V}_k=\left[v_1^k\mathbf{1}_{n_k},v_2^k\mathbf{1}_{n_k},\ldots,v_{K-1}^k\mathbf{1}_{n_k}\right],\tag{C.22}$$

where $\mathbf{V}=\text{diag}(v_1^k,v_2^k,\ldots,v_{K-1}^k)$ is a diagonal matrix.

Let $\mathcal{S}=\{\mathbf{X}\in\mathbb{R}^{n\times(K-1)}:\ \mathbf{X}^T\mathbf{X}=\mathbf{I}_{K-1},\ \mathbf{X}^T\mathbf{1}_n=\mathbf{0}_{K-1}\}$. In Case 2, since $\mathbf{Y}_k^T\mathbf{1}_{n_k}\xrightarrow{\text{a.s.}}\mathbf{0}_{K-1}\ \forall\ k$, we have

$$\begin{aligned}\frac{S_{2:K}(\mathbf{L})}{n}&\xrightarrow{\text{a.s.}}\min_{\mathbf{X}\in\mathcal{S}}\left\{\frac{1}{n}\left[\sum_{k=1}^K\text{trace}(\mathbf{X}_k^T\mathbf{L}_k\mathbf{X}_k)+p\sum_{k=1}^K(n-n_k)\text{trace}(\mathbf{X}_k^T\mathbf{X}_k)\right]\right\}&&\text{(C.23)}\\&\geq\min_{\mathbf{X}\in\mathcal{S}}\left\{\frac{1}{n}\sum_{k=1}^K\text{trace}(\mathbf{X}_k^T\mathbf{L}_k\mathbf{X}_k)\right\}+\min_{\mathbf{X}\in\mathcal{S}}\left\{\frac{p}{n}\sum_{k=1}^K(n-n_k)\text{trace}(\mathbf{X}_k^T\mathbf{X}_k)\right\}&&\text{(C.24)}\\&=\min_{k\in\{1,2,\ldots,K\}}\left\{\frac{S_{2:K}(\mathbf{L}_k)}{n}\right\}+\frac{(K-1)p}{n}\min_{k\in\{1,2,\ldots,K\}}(n-n_k)&&\text{(C.25)}\\&=\min_{k\in\{1,2,\ldots,K\}}\left\{\frac{S_{2:K}(\mathbf{L}_k)}{n}\right\}+\frac{(K-1)(n-n_{\max})p}{n},&&\text{(C.26)}\end{aligned}$$

where $n_{\max}=\max_{k\in\{1,2,\ldots,K\}}n_k$.

Let $\mathcal{S}_k=\{\mathbf{X}\in\mathbb{R}^{n\times(K-1)}:\ \mathbf{X}_k^T\mathbf{X}_k=\mathbf{I}_{K-1},\ \mathbf{X}_j=\mathbf{O}_{n_j\times(K-1)}\ \forall\ j\neq k,\ \mathbf{X}^T\mathbf{1}_n=$

$\mathbf{0}_{K-1}\}$. Since $\mathcal{S}_k \subseteq \mathcal{S}$, in Case 2, we have

$$\frac{S_{2:K}(\mathbf{L})}{n} \xrightarrow{\text{a.s.}} \min_{\mathbf{X}\in\mathcal{S}} \left\{ \frac{1}{n} \left[\sum_{k=1}^{K} \text{trace}(\mathbf{X}_k^T \mathbf{L}_k \mathbf{X}_k) + p \sum_{k=1}^{K} (n-n_k)\text{trace}(\mathbf{X}_k^T \mathbf{X}_k) \right] \right\} \tag{C.27}$$

$$\leq \min_{k\in\{1,2,\ldots,K\}} \min_{\mathbf{X}\in\mathcal{S}_k} \left\{ \frac{1}{n} \left[\sum_{k=1}^{K} \text{trace}(\mathbf{X}_k^T \mathbf{L}_k \mathbf{X}_k) + p \sum_{k=1}^{K} (n-n_k)\text{trace}(\mathbf{X}_k^T \mathbf{X}_k) \right] \right\} \tag{C.28}$$

$$= \min_{k\in\{1,2,\ldots,K\}} \left\{ \frac{1}{n} \left[S_{2:K}(\mathbf{L}_k) + (K-1)(n-n_k)p \right] \right\} \tag{C.29}$$

$$\leq \min_{k\in\{1,2,\ldots,K\}} \left\{ \frac{1}{n} \left[S_{2:K}(\mathbf{L}_k) + (K-1)(n-n_{\min})p \right] \right\} \tag{C.30}$$

$$= \min_{k\in\{1,2,\ldots,K\}} \left\{ \frac{S_{2:K}(\mathbf{L}_k)}{n} \right\} + \frac{(K-1)(n-n_{\min})p}{n}, \tag{C.31}$$

where $n_{\min} = \min_{k\in\{1,2,\ldots,K\}} n_k$.

Comparing (C.17) with (C.26) and (C.31), as a function of p the slope of $\frac{S_{2:K}(\mathbf{L})}{n}$ changes at some critical value p^* that separates Case 1 and Case 2, and by the continuity of $\frac{S_{2:K}(\mathbf{L})}{n}$ a lower bound on p^* is

$$p_{\text{LB}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1)n_{\max}}, \tag{C.32}$$

and an upper bound of p^* is

$$p_{\text{UB}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1)n_{\min}}. \tag{C.33}$$

C.3 Proof of Corollary 4.3

Recall the eigenvector matrix $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T$, where $\mathbf{Y}_k$ is the $n_k \times (K-1)$ matrix with row vectors representing the nodes from cluster k. Since $\mathbf{Y}^T\mathbf{Y} = \sum_{k=1}^{K} \mathbf{Y}_k^T \mathbf{Y}_k = \mathbf{I}_{(K-1)\times(K-1)}$, $\mathbf{Y}^T \mathbf{1}_n = \sum_{k=1}^{K} \mathbf{Y}_k^T \mathbf{1}_{n_k} = \mathbf{0}_{K-1}$, and from (C.22) when $p < p^*$ the matrix $\mathbf{Y}_k \xrightarrow{\text{a.s.}} \mathbf{1}_{n_k} \mathbf{1}_{K-1}^T \mathbf{V}_k = \left[v_1^k \mathbf{1}_{n_k}, v_2^k \mathbf{1}_{n_k}, \ldots, v_{K-1}^k \mathbf{1}_{n_k} \right]$ as $n_k \to \infty$ and

$\frac{n_{\min}}{n_{\max}} \to c > 0$, we have

$$\begin{aligned} &\textstyle\sum_{k=1}^{K} n_k \mathbf{v_k}\mathbf{v_k}^T = \mathbf{I}_{(K-1)\times(K-1)}; \\ &\textstyle\sum_{k=1}^{K} n_k \mathbf{v_k} = \mathbf{0}_{K-1}, \end{aligned} \tag{C.34}$$

where $\mathbf{v_k} = \mathbf{V}_k \mathbf{1}_{n_k} = [v_1^k, v_2^k, \ldots, v_{K-1}^k]^T$. (C.34) suggests that some $\mathbf{v_k}$ cannot be a zero vector since $\sum_{k=1}^{K} n_k (v_j^k)^2 = 1$ for all $j \in \{1, 2, \ldots, K-1\}$, and from (C.34) we have

$$\begin{aligned} &\textstyle\sum_{k:v_j^k>0} n_k v_j^k = -\sum_{k:v_j^k<0} n_k v_j^k, \ \forall\, j \in \{1, 2, \ldots, K-1\}; \\ &\textstyle\sum_{k:v_i^k v_j^k>0} n_k v_i^k v_j^k = -\sum_{k:v_i^k v_j^k<0} n_k v_i^k v_j^k, \ \forall\, i, j \in \{1, 2, \ldots, K-1\}, i \neq j. \end{aligned} \tag{C.35}$$

This concludes the properties in Corollary 4.3.

C.4 Proof of Corollary 4.4

If $\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k) = \Omega(n_{\max})$, then by Theorem 4.2 (c) $p_{\text{LB}} > 0$. Therefore $p^* \geq p_{\text{LB}} > 0$. Similarly, If $\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k) = o(n_{\min})$, then by Theorem 4.2 (c) $p_{\text{UB}} = 0$. Therefore $p^* = 0$. Finally, since $S_{2:K}(\mathbf{L}_k) = \sum_{i=2}^{K} \lambda_i(\mathbf{L}_k) \geq (K-1)\lambda_2(\mathbf{L}_k)$ and $S_{2:K}(\mathbf{L}_k) = \sum_{i=2}^{K} \lambda_i(\mathbf{L}_k) \leq (K-1)\lambda_K(\mathbf{L}_k)$, applying these two inequalities to Theorem 4.2 (c) gives Corollary 4.4 (c).

C.5 Proof of Corollary 4.5

If cluster k is a complete graph, then $\lambda_i(\mathbf{L}_k) = n_k$ for $2 \leq i \leq n_k$ [161]. Therefore $p_{\text{LB}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1)n_{\max}} = \frac{\min_{k\in\{1,2,\ldots,K\}} n_k}{n_{\max}} = \frac{n_{\min}}{n_{\max}} = c$, and $p_{\text{UB}} = \frac{\min_{k\in\{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1)n_{\min}} = \frac{\min_{k\in\{1,2,\ldots,K\}} n_k}{n_{\min}} = 1$. If cluster k is a star graph, then $\lambda_i(\mathbf{L}_k) = 1$ for $2 \leq i \leq n_k - 1$ [161]. Hence if $K < n_{\min}$, then $S_{2:K}(\mathbf{L}_k) = o(n_{\min})$ and by Corollary 4.4 (b) $p^* = 0$.

C.6 Proof of (4.2)

If cluster k is a Erdos-Renyi random graph with edge connection probability p_k, then $\frac{\lambda_i(\mathbf{L}_k)}{n_k} \xrightarrow{\text{a.s.}} p_k$ for $2 \leq i \leq n_k$ [25] as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, where p_k is a constant. Therefore $p_{\text{LB}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1) n_{\max}} = \frac{\min_{k \in \{1,2,\ldots,K\}} n_k p_k}{n_{\max}} \geq c \cdot \min_{k \in \{1,2,\ldots,K\}} p_k$, and $p_{\text{UB}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k)}{(K-1) n_{\min}} = \frac{\min_{k \in \{1,2,\ldots,K\}} n_k p_k}{n_{\min}} \leq \frac{n_{\max} \cdot \min_{k \in \{1,2,\ldots,K\}} p_k}{n_{\min}} = \frac{1}{c} \cdot \min_{k \in \{1,2,\ldots,K\}} p_k$.

C.7 Proof of Corollary 4.6

Corollary 4.6 (a) is a direct result from Theorem 4.2 (a), with $K = 2$ and the fact that $\min\{a, b\} = \frac{a+b-|a-b|}{2}$ for all $a, b \geq 0$. Corollary 4.6 (b) is a direct result from Theorem 4.2 (b) and Corollary 4.3, with the orthonormality constraints that $\mathbf{y}_1^T \mathbf{1}_{n_1} + \mathbf{y}_2^T \mathbf{1}_{n_2} = 0$ and $\mathbf{y}_1^T \mathbf{y}_1 + \mathbf{y}_2^T \mathbf{y}_2 = 1$. Corollary 4.6 (c) is a direct result from Corollary 4.4 (c), with $\max\{a, b\} = \frac{a+b+|a-b|}{2}$ for all $a, b \geq 0$.

C.8 Proof of Corollary 4.7

We first show that when $p_{\max} < p^*$, the second eigenvalue of $\frac{\mathbf{L}}{n}$, $\lambda_2(\frac{\mathbf{L}}{n})$, lies within the interval $[p_{\min}, p_{\max}]$ almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Consider a graph generated by the inhomogeneous RIM with parameter $\{p_{ij}\}$. By (C.9) with proper scaling the entries of each interconnection matrix $\mathbf{C}_{ij}$ converge to p_{ij} almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Let $\mathbf{A}(p)$ be the adjacency matrix under the homogeneous RIM with parameter p. Then the adjacency matrix $\mathbf{A}$ of the inhomogeneous RIM can be written as $\mathbf{A} = \mathbf{A}(p_{\min}) + \mathbf{\Delta A}$, and the graph Laplacian matrix associated with $\mathbf{A}$ can be written as $\mathbf{L} = \mathbf{L}(p_{\min}) + \mathbf{\Delta L}$, where $\mathbf{L}(p_{\min})$ and $\mathbf{\Delta L}$ are associated with $\mathbf{A}(p_{\min})$ and $\mathbf{\Delta A}$, respectively. Since $p_{\min} = \min_{i \neq j} p_{ij}$, as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, $\frac{\mathbf{\Delta A}}{n}$ is a symmetric nonnegative matrix almost surely, and $\frac{\mathbf{\Delta L}}{n}$ is a graph Laplacian matrix almost surely. By the PSD property of a graph Laplacian matrix and Corollary 4.6 (a), we obtain $\lambda_2(\frac{\mathbf{L}}{n}) \geq p_{\min}$ almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Similarly, following the same procedure we can show that $\lambda_2(\frac{\mathbf{L}}{n}) \leq p_{\max}$ almost surely as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Lastly, when $p < p^*$, using the fact from (C.15) that $\lambda_j(\frac{\widetilde{\mathbf{L}}}{n}) \xrightarrow{\text{a.s.}} p$, and $\lambda_j(\frac{\mathbf{L}(p_{\min})}{n}) \leq \lambda_j(\frac{\mathbf{L}}{n}) \leq \lambda_j(\frac{\mathbf{L}(p_{\max})}{n})$ almost surely for all $j \in \{2, 3, \ldots, K\}$ as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, we obtain the results.

C.9 Proof of Theorem 4.8

Applying the Davis-Kahan $\sin\theta$ theorem [22] to the eigenvector matrices $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ associated with the graph Laplacian matrices $\frac{\mathbf{L}}{n}$ and $\frac{\widetilde{\mathbf{L}}}{n}$, respectively, we obtain an upper bound on the distance of column spaces spanned by $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$, which is $\| \sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}\|_F \leq \frac{\|\mathbf{L}-\widetilde{\mathbf{L}}\|_F}{n\delta}$, where $\delta = \inf\{|x-y| : x \in \{0\} \cup [\lambda_{K+1}(\frac{\mathbf{L}}{n}), \infty),~ y \in [\lambda_2(\frac{\widetilde{\mathbf{L}}}{n}), \lambda_K(\frac{\widetilde{\mathbf{L}}}{n})]\}$. If $p < p^*$, using the fact from (C.15) that $\lambda_j(\frac{\widetilde{\mathbf{L}}}{n}) \xrightarrow{\text{a.s.}} p$ for all $j \in \{2, 3, \ldots, K\}$ as $n_k \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, the interval $[\lambda_2(\frac{\widetilde{\mathbf{L}}}{n}), \lambda_K(\frac{\widetilde{\mathbf{L}}}{n})]$ reduces to a point p almost surely. Therefore, δ reduces to δ_p as defined in Theorem 4.8. Furthermore, if $p_{\max} \leq p^*$, then (4.4) holds for all $p \leq p_{\max}$. Taking the minimum of all upper bounds in (4.4) for $p \leq p_{\max}$ completes the theorem.

C.10 Proof of Theorem 4.9

Similar to the proof of Theorem 4.2, for undirected weighted graphs under the homogeneous RIM we need to show

$$\frac{\mathbf{W}_{ij}}{\sqrt{n_i n_j}} \xrightarrow{\text{a.s.}} p\overline{W}\mathbf{1}\mathbf{1}^T \tag{C.36}$$

for all $i, j \in \{1, 2, \ldots, K\}$ as $n_i, n_j \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, where $\mathbf{W}_{ij}$ is the weight matrix of inter-cluster edges between the cluster pair (i, j) and $\overline{W}$ is the mean of the common nonnegative inter-cluster edge weight distribution. By the smoothing property we have the mean of $\mathbf{W}_{ij}$ to be

$$\mathbb{E}\mathbf{W}_{ij} = \mathbb{E}\left[\mathbb{E}\left[\mathbf{W}_{ij}\mathbf{C}_{ij}|\mathbf{C}_{ij}\right]\right] = \mathbb{E}\mathbf{C}_{ij}\mathbb{E}\left[\mathbf{W}_{ij}|\mathbf{C}_{ij}\right] = p\cdot\overline{W}.$$

Let $\mathbf{\Delta} = \mathbf{W}_{ij} - \overline{\mathbf{W}}_{ij}$, where $\overline{\mathbf{W}}_{ij} = p\overline{W}\mathbf{1}_{n_i}\mathbf{1}_{n_j}^T$ is a matrix whose elements are the means of entries in $\mathbf{W}_{ij}$. Then $[\mathbf{\Delta}]_{uv} = [\mathbf{W}_{ij}]_{uv} - p\overline{W}$ with probability p and $[\mathbf{\Delta}]_{uv} = -p\overline{W}$ with probability $1-p$. Let $\sigma_i(\mathbf{M})$ denote the i-th largest singular value of a real rectangular matrix $\mathbf{M}$. The Latala's theorem [91] states that for any random matrix $\mathbf{M}$ with statistically independent and zero mean entries, there exists a positive constant c_1 such that

$$\mathbb{E}\left[\sigma_1(\mathbf{M})\right] \leq c_1\left(\max_u \sqrt{\sum_v \mathbb{E}\left[[\mathbf{M}]_{uv}^2\right]} + \max_v \sqrt{\sum_u \mathbb{E}\left[[\mathbf{M}]_{uv}^2\right]} + \sqrt[4]{\sum_{u,v}\mathbb{E}\left[[\mathbf{M}]_{uv}^4\right]}\right). \tag{C.37}$$

It is clear that $\mathbb{E}\left[[\mathbf{\Delta}]_{uv}\right] = 0$ and each entry in $\mathbf{\Delta}$ is independent. Substituting $\mathbf{M} = \frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}$ into the Latala's theorem, since $p \in [0,1]$ and the common inter-cluster edge weight distribution has finite fourth moment, by the smoothing property we have $\max_u \sqrt{\sum_v \mathbb{E}\left[[\mathbf{M}]_{uv}^2\right]} = O(\frac{1}{\sqrt{n_i}})$, $\max_v \sqrt{\sum_u \mathbb{E}\left[[\mathbf{M}]_{uv}^2\right]} = O(\frac{1}{\sqrt{n_j}})$, and $\sqrt[4]{\sum_{u,v}\mathbb{E}\left[[\mathbf{M}]_{uv}^4\right]} = O(\frac{1}{\sqrt[4]{n_i n_j}})$. Therefore $\mathbb{E}\left[\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}\right)\right] \to 0$ for all $i,j \in \{1,2,\ldots,K\}$ as $n_i, n_j \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$.

Next we use the Talagrand's concentration theorem stated as follows. Let $g : \mathbb{R}^k \mapsto \mathbb{R}$ be a convex and Lipschitz function. Let $\mathbf{x} \in \mathbb{R}^k$ be a random vector and assume that every element of $\mathbf{x}$ satisfies $|\mathbf{x}_i| \leq \phi$ for all $i = 1,2,\ldots,k$ and some constant ϕ, with probability one. Then there exist positive constants c_2 and c_3 such that for any $\epsilon > 0$,

$$\Pr\left(\left|g(\mathbf{x}) - \mathbb{E}\left[g(\mathbf{x})\right]\right| \geq \epsilon\right) \leq c_2 \exp\left(\frac{-c_3\epsilon^2}{\phi^2}\right). \tag{C.38}$$

It is well-known that the largest singular value of a matrix $\mathbf{M}$ can be represented as $\sigma_1(\mathbf{M}) = \max_{\mathbf{z}^T\mathbf{z}=1} ||\mathbf{M}\mathbf{z}||_2$ [76] so that $\sigma_1(\mathbf{M})$ is a convex and Lipschitz function. Applying the Talagrand's theorem by substituting $\mathbf{M} = \frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}$ and using the facts that $\mathbb{E}\left[\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}\right)\right] \to 0$ and $\frac{[\mathbf{\Delta}]_{uv}}{\sqrt{n_i n_j}} \leq \frac{[\mathbf{W}]_{uv}}{\sqrt{n_i n_j}}$, we have

$$\Pr\left(\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}\right) \geq \epsilon\right) \leq c_2 \exp\left(-c_3 n_i n_j \epsilon^2\right). \tag{C.39}$$

Since for any positive integer $n_i, n_j > 0$ $n_i n_j \geq \frac{n_i+n_j}{2}$, $\sum_{n_i,n_j} c_2 \exp\left(-c_3 n_i n_j \epsilon^2\right) < \infty$. By Borel-Cantelli lemma [137], $\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}\right) \xrightarrow{\text{a.s.}} 0$ when $n_i, n_j \to \infty$. Finally, a standard matrix perturbation theory result [76] is $|\sigma_i(\overline{\mathbf{W}}_{ij} + \mathbf{\Delta}) - \sigma_i(\overline{\mathbf{W}}_{ij})| \leq \sigma_1(\mathbf{\Delta})$ for all i, and as $\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}\right) \xrightarrow{\text{a.s.}} 0$, we have as $n_i, n_j \to \infty$,

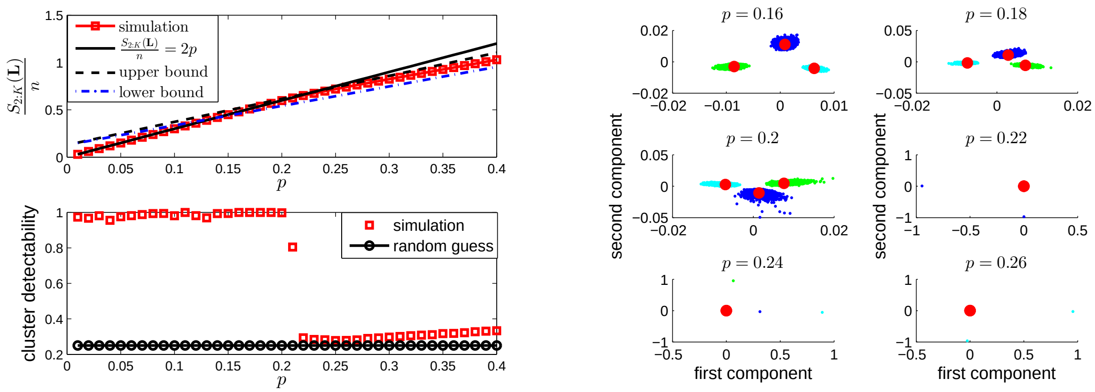
$$\sigma_1\left(\frac{\mathbf{W}_{ij}}{\sqrt{n_i n_j}}\right) = \sigma_1\left(\frac{\overline{\mathbf{W}}_{ij} + \mathbf{\Delta}}{\sqrt{n_i n_j}}\right) \xrightarrow{\text{a.s.}} \sigma_1\left(\frac{\overline{\mathbf{W}}_{ij}}{\sqrt{n_i n_j}}\right) = p\overline{W}; \tag{C.40}$$

$$\sigma_i\left(\frac{\mathbf{W}_{ij}}{\sqrt{n_i n_j}}\right) \xrightarrow{\text{a.s.}} 0,\ \forall\ i \geq 2. \tag{C.41}$$

Furthermore, by Wedin's $\sin\theta$ theorem [164], the singular vectors of $\mathbf{W}_{ij}$ and $\overline{\mathbf{W}}_{ij}$ are close to each other in the sense that the square of inner product of their left/right singular vectors converges to 1 almost surely when $\sigma_1\left(\frac{\mathbf{\Delta}}{\sqrt{n_i n_j}}\right) \xrightarrow{\text{a.s.}} 0$. Therefore $\frac{\mathbf{W}_{ij}}{\sqrt{n_i n_j}} \xrightarrow{\text{a.s.}} p\overline{W}\mathbf{1}\mathbf{1}^T$. Lastly, following the same proof procedure in Appendix C.2, we obtain Theorem 4.9.

C.11 Additional phase transition results in simulated networks

Fig. C.1 (a) shows the phase transition in normalized partial eigenvalue sum $\frac{S_{2:K}(\mathbf{L})}{n}$ and cluster detectability for clusters generated by Erdos-Renyi random graphs



(a) Phase transition in normalized partial sum of eigenvalues $\frac{S_{2:K}(\mathbf{L})}{n}$ and cluster detectability.

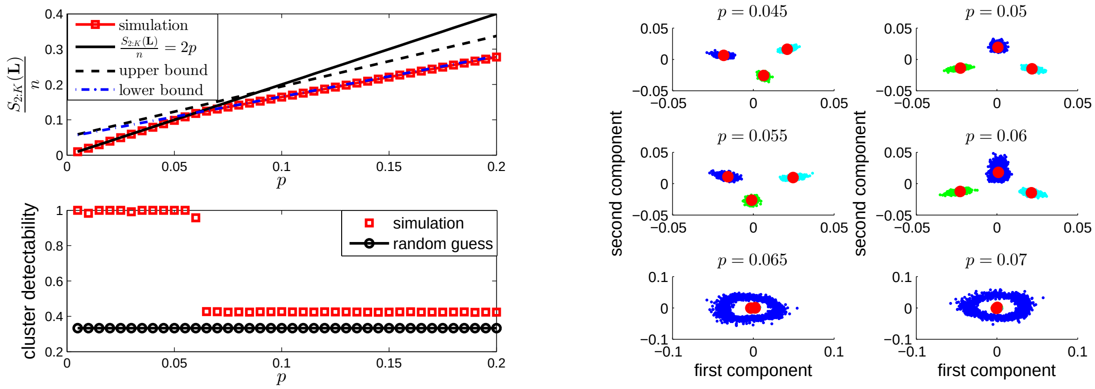
(b) Row vectors in $\mathbf{Y}$ with respect to different p. Colors and red solid circles represent clusters and cluster-wise centroids.

Figure C.1: Phase transition of clusters generated by Erdos-Renyi random graphs. $K = 3$, $(n_1, n_2, n_3) = (6000, 8000, 10000)$, and $p_1 = p_2 = p_3 = 0.25$. The empirical lower bound $p_{\text{LB}} = 0.1373$ and the empirical upper bound $p_{\text{UB}} = 0.2288$. The results in (a) are averaged over 50 trials.

with different network sizes. As predicted by Theorem 4.2 (a), the slope of $\frac{S_{2:K}(\mathbf{L})}{n}$ undergoes a phase transition at some critical threshold value p^*. When $p < p^*$, $\frac{S_{2:K}(\mathbf{L})}{n}$ is exactly $2p$. When $p > p^*$, $\frac{S_{2:K}(\mathbf{L})}{n}$ is upper and lower bounded by the derived bounds. Fig. C.1 (b) shows the row vectors of $\mathbf{Y}$ that verifies Theorem 4.2 (b) and Corollary 4.3. Similar phase transition can be found for clusters generated by the Watts-Strogatz small world network model [163] with different cluster sizes in Fig. C.2.

Next we investigate the sensitivity of cluster detectability to the inhomogeneous RIM. We consider the perturbation model $p_{ij} = p_0 + \text{unif}(-a, a)$, where p_0 is the base edge connection probability and $\text{unif}(-a, a)$ is an uniform random variable with support $(-a, a)$. The simulation results in Figs. C.3 (a) and (b) show that almost perfect cluster detectability is still valid when p_{ij} is within certain perturbation of p_0. The sensitivity of cluster detectability to inhomogeneous RIM also implies the accuracy of SGC under the inhomogeneous RIM in Theorem 4.8.

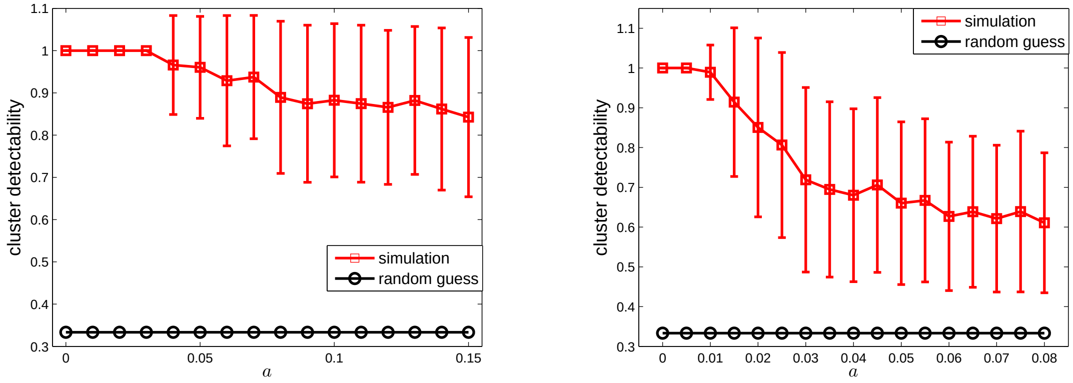
Note that Theorem 4.1 also explains the effect of the perturbation model $p_{ij} =$



(a) Phase transition in normalized partial sum of eigenvalues $\frac{S_{2:K}(\mathbf{L})}{n}$ and cluster detectability.

(b) Row vectors in $\mathbf{Y}$ with respect to different p. Colors and red solid circles represent clusters and cluster-wise centroids.

Figure C.2: Phase transition of clusters generated by the Watts-Strogatz small world network model. $K = 3$, $(n_1, n_2, n_3) = (1500, 1000, 1000)$, average number of neighbors $= 200$, and rewire probability for each cluster is 0.4, 0.4, and 0.6. The empirical lower and upper bounds are $p_{\text{LB}} = 0.0602$ and $p_{\text{UB}} = 0.0902$. The results in (a) are averaged over 50 trials.



(a)

(b)

Figure C.3: Sensitivity of cluster detectability to the inhomogeneous RIM. The results are average over 50 trials and error bars represent standard deviation. (a) Clusters generated by Erdos-Renyi random graphs. $K = 3$, $n_1 = n_2 = n_3 = 8000$, $p_1 = p_2 = p_3 = 0.25$, and $p_0 = 0.15$. (b) Clusters generated by the Watts-Strogatz small world network model. $K = 3$, $n_1 = n_2 = n_3 = 1000$, average number of neighbors $= 200$, and rewire probability for each cluster is 0.4, 0.4, 0.6, and $p_0 = 0.08$.

$p_0 + \mathrm{unif}(-a, a)$ on cluster detectability. As a increases the off-diagonal entries in $\widetilde{\mathbf{A}}$ further deviate from 0 and the matrix $\widetilde{\mathbf{A}} \oplus -\mathbf{\Lambda}$ in AppendixC.1 gradually becomes non-singular, resulting in the degradation of cluster detectability. Furthermore, using Theorem 4.1 and the Gershgorin circle theorem [76], each eigenvalue of $\frac{\widetilde{\mathbf{A}}}{n}$ lies within at least one of the closed disc centered at $\frac{[\widetilde{\mathbf{A}}]_{ii}}{n}$ with radius R_i, where $R_i = \frac{n_i}{n} \sum_{j=1, j \neq i}^{K-1} |p_{iK} - p_{ij}|$. Therefore larger inhomogeneity in p_{ij} further drives the matrix $\widetilde{\mathbf{A}} \oplus -\mathbf{\Lambda}$ away from singularity.

APPENDIX D

Appendix of Chapter V

D.1 Asymptotic confidence interval for the homogeneous RIM

Here we define the generalized log-likelihood ratio test (GLRT) under the RIM for the hypothesis $H_0 : p_{ij} = p \; \forall i, j, i \neq j$, against its alternative hypothesis $H_1 : p_{ij} \neq p$, for at least one $i, j,\ i \neq j$. Let $f_{ij}^h(x, \theta | \{\widehat{G}_k\}_{k=1}^K)$ denote the likelihood function of observing x edges between $\widehat{G}_i$ and $\widehat{G}_j$ under hypothesis H_h, and θ is the edge interconnection probability. $\widehat{n}_k$ is the number of nodes in cluster k, and $\widehat{m}_{ij}$ is the number of edges between clusters i and j. Then under the RIM,

$$
\begin{aligned}
f_{ij}^1(\widehat{m}_{ij}, p_{ij} | \{\widehat{G}_k\}_{k=1}^K) &= \binom{\widehat{n}_i \widehat{n}_j}{\widehat{m}_{ij}} p_{ij}^{\widehat{m}_{ij}} (1 - p_{ij})^{\widehat{n}_i \widehat{n}_j - \widehat{m}_{ij}}; \\
f_{ij}^0(\widehat{m}_{ij}, p | \{\widehat{G}_k\}_{k=1}^K) &= \binom{\widehat{n}_i \widehat{n}_j}{\widehat{m}_{ij}} p^{\widehat{m}_{ij}} (1 - p)^{\widehat{n}_i \widehat{n}_j - \widehat{m}_{ij}}. \qquad \text{(D.1)}
\end{aligned}
$$

Since $\widehat{p}_{ij}$ is the MLE of p_{ij} under H_1 and $\widehat{p}$ is the MLE of p under H_0, the GLRT statistic is

$$\begin{aligned}
\mathrm{GLRT} &= 2\ln \frac{\sup_{p_{ij}} \prod_{i=1}^{K} \prod_{j>i}^{K} f_{ij}^{1}(\widehat{m}_{ij}, p_{ij} | \{\widehat{G}_k\}_{k=1}^{K})}{\sup_{p_{ij}=p} \prod_{i=1}^{K} \prod_{j>i}^{K} f_{ij}^{0}(\widehat{m}_{ij}, p_{ij} | \{\widehat{G}_k\}_{k=1}^{K})} \\
&= 2\ln \frac{\prod_{i=1}^{K} \prod_{j=i+1}^{K} f_{ij}^{1}(\widehat{m}_{ij}, \widehat{p}_{ij} | \{\widehat{G}_k\}_{k=1}^{K})}{\prod_{i=1}^{K} \prod_{j=i+1}^{K} f_{ij}^{0}(\widehat{m}_{ij}, \widehat{p} | \{\widehat{G}_k\}_{k=1}^{K})} \\
&= 2 \left\{ \sum_{i=1}^{K} \sum_{j=i+1}^{K} \mathbb{I}_{\{\widehat{p}_{ij} \in (0,1)\}} \left[\widehat{m}_{ij} \ln \widehat{p}_{ij} + (\widehat{n}_i \widehat{n}_j - \widehat{m}_{ij}) \ln(1-\widehat{p}_{ij}) \right] \right. \\
&\quad \left. - \left(m - \sum_{k=1}^{K} \widehat{m}_i \right) \ln \widehat{p} - \left[\frac{1}{2} \left(n^2 - \sum_{k=1}^{K} \widehat{n}_k^2 \right) - \left(m - \sum_{k=1}^{K} \widehat{m}_k \right) \right] \ln(1-\widehat{p}) \right\},
\end{aligned} \tag{D.2}$$

where we use the relations that $\sum_{i=1}^{K} \sum_{j=i+1}^{K} \widehat{m}_{ij} = m - \sum_{k=1}^{K} \widehat{m}_k$ and $\sum_{i=1}^{K} \sum_{j=i+1}^{K} \widehat{n}_i \widehat{n}_j = \frac{n^2 - \sum_{k=1}^{K} \widehat{n}_k^2}{2}$. By the Wilk's theorem [167], as $n_k \to \infty \ \forall \ k$, this statistic converges in law to the chi-square distribution, denoted by χ_ν^2, with $\nu = \binom{K}{2} - 1$ degrees of freedom. Therefore, we obtain the asymptotic $100(1-\alpha)\%$ confidence interval for p in (5.1).

D.2 Phase transition tests for undirected weighted graphs

Given clusters $\{\widehat{G}_k\}_{k=1}^{K}$ of an undirected weighted graph obtained from spectral clustering with model order K, let $\widehat{\overline{W}}_{ij}$ be the average weight of the inter-cluster edges between clusters i and j, and let $\widehat{\overline{W}}$ be the average weight of all between-cluster edges. Define $\widehat{t}_{ij} = \widehat{p}_{ij} \cdot \widehat{\overline{W}}_{ij}$, $\widehat{t} = \widehat{p} \cdot \widehat{\overline{W}}$, $\widehat{t}_{\max} = \max_{ij} \widehat{t}_{ij}$ and $\widehat{t}_{\mathrm{LB}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\widehat{\mathbf{L}}_k)}{(K-1)\widehat{n}_{\max}}$. For undirected weighted graphs, the first phase of testing the RIM assumption in the AMOS algorithm is identical to undirected unweighted graphs, i.e., the estimated local inter-cluster edge connection probabilities $\widehat{p}_{ij}$'s are used to test the RIM hypothesis. In the second phase, if the clusters pass the homogeneous RIM test (i.e., the estimate of global inter-cluster edge probability $\widehat{p}$ lies in the confidence interval specified in (5.1)), then based on the phase transition results in Theorem 4.9, the clusters pass the homogeneous phase transition test if $\widehat{t} < \widehat{t}_{\mathrm{LB}}$. If the homogeneous RIM test fails, then by Theorem 4.8 the clusters pass the inhomogeneous RIM test if $\widehat{t}_{\max}$ lies in a confidence interval $[0, \psi]$ and $\psi < t^*$. Moreover, since testing $\widehat{t}_{ij} < t^*$ is equivalent to testing $\widehat{p}_{ij} < \frac{t^*}{\widehat{\overline{W}}_{ij}}$, as discussed in Sec. 5.1.3, we can verify $\psi < t^*$ by checking the condition

$$\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij} \left(\frac{\widehat{t}_{\mathrm{LB}}}{\widehat{\overline{W}}_{ij}}, \widehat{p}_{ij} \right) \geq 1 - \alpha', \tag{D.3}$$

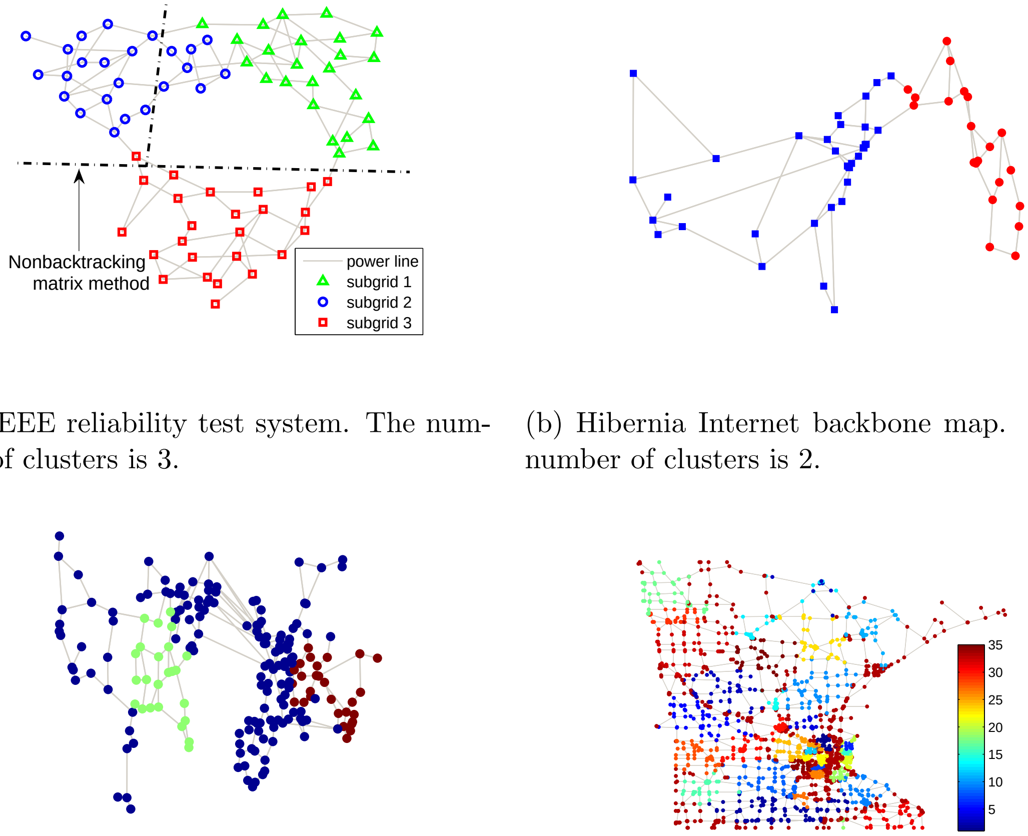
where α' is the precision parameter of the confidence interval.

D.3 Performance of the Louvain method and the nonbacktracking matrix method on real-life network data

Fig. D.1 and Fig. D.2 show the clusters of the datasets in Table 5.1 identified by the the nonbacktracking matrix method [87, 139] and the Louvain method [18], respectively. Comparing the proposed AMOS algorithm with the two method, the clusters identified by AMOS are more consistent with the ground truth meta information provided by the datasets.

The performance of the nonbacktracking matrix method is summarized as follows. For IEEE reliability test system, 8 nodes are clustered incorrectly. For Hibernia Internet backbone map, 3 cities in the north America are clustered with the cities in Europe. For Cogent Internet backbone map, the clusters are inconsistent with the geographic locations. For Minnesota road map, some clusters are not aligned with the geographic separations.

The performance of the Louvain method is summarized as follows. For IEEE reliability test system, the number of clusters is different from the number of actual subgrids. For Hibernia and Cogent Internet backbone maps, although the clusters are consistent with the geographic locations, the Louvain method tends to identify clusters with small sizes. For Minnesota road map, the clusters are inconsistent with the geographic separations.



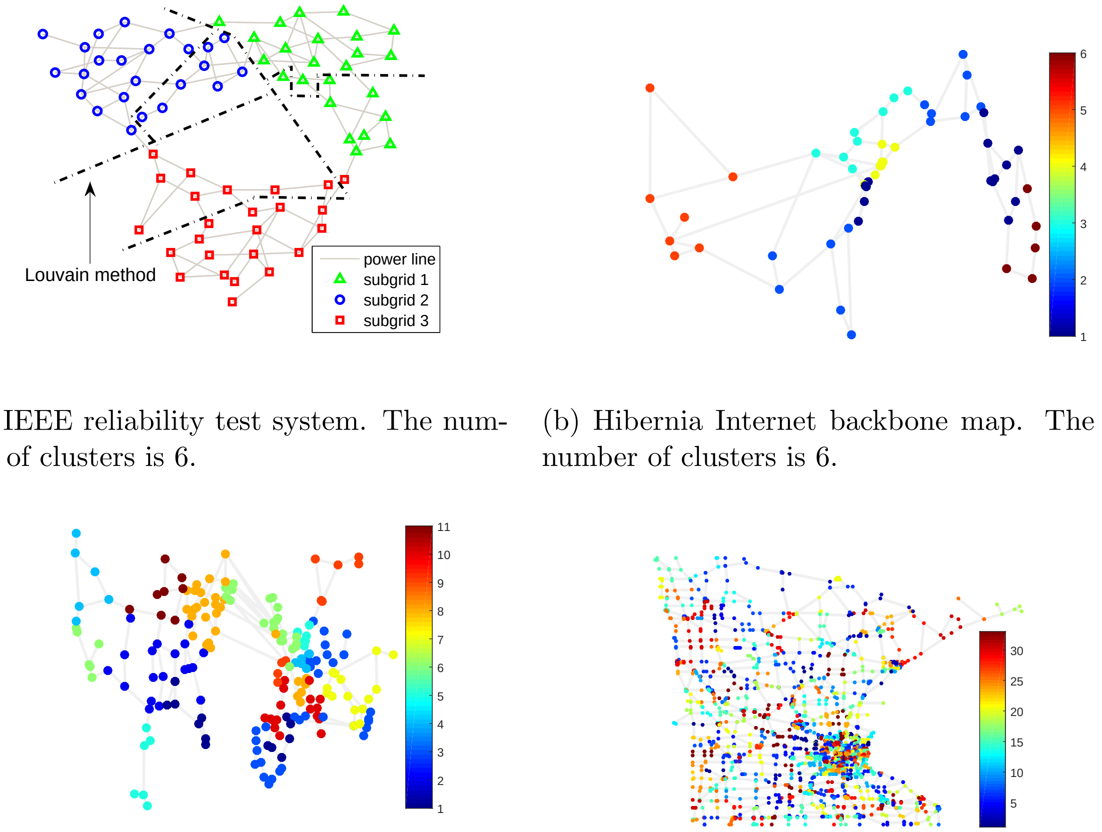
(a) IEEE reliability test system. The number of clusters is 3.

(b) Hibernia Internet backbone map. The number of clusters is 2.

(c) Cogent Internet backbone map. The number of clusters is 3.

(d) Minnesota road map. The number of clusters is 35.

Figure D.1: Clusters found with the nonbacktracking matrix method [87, 139]. For IEEE reliability test system, 8 nodes are clustered incorrectly. For Hibernia Internet backbone map, 3 cities in the north America are clustered with the cities in Europe. For Cogent Internet backbone map, the clusters are inconsistent with the geographic locations. For Minnesota road map, some clusters are not aligned with the geographic separations.



(a) IEEE reliability test system. The number of clusters is 6.

(b) Hibernia Internet backbone map. The number of clusters is 6.

(c) Cogent Internet backbone map. The number of clusters is 11.

(d) Minnesota road map. The number of clusters is 33.

Figure D.2: Clusters found with the Louvain method [18]. For IEEE reliability test system, the number of clusters is different from the number of actual subgrids. For Hibernia and Cogent Internet backbone maps, although the clusters are consistent with the geographic locations, the Louvain method tends to identify clusters with small sizes. For Minnesota road map, the clusters are inconsistent with the geographic separations.

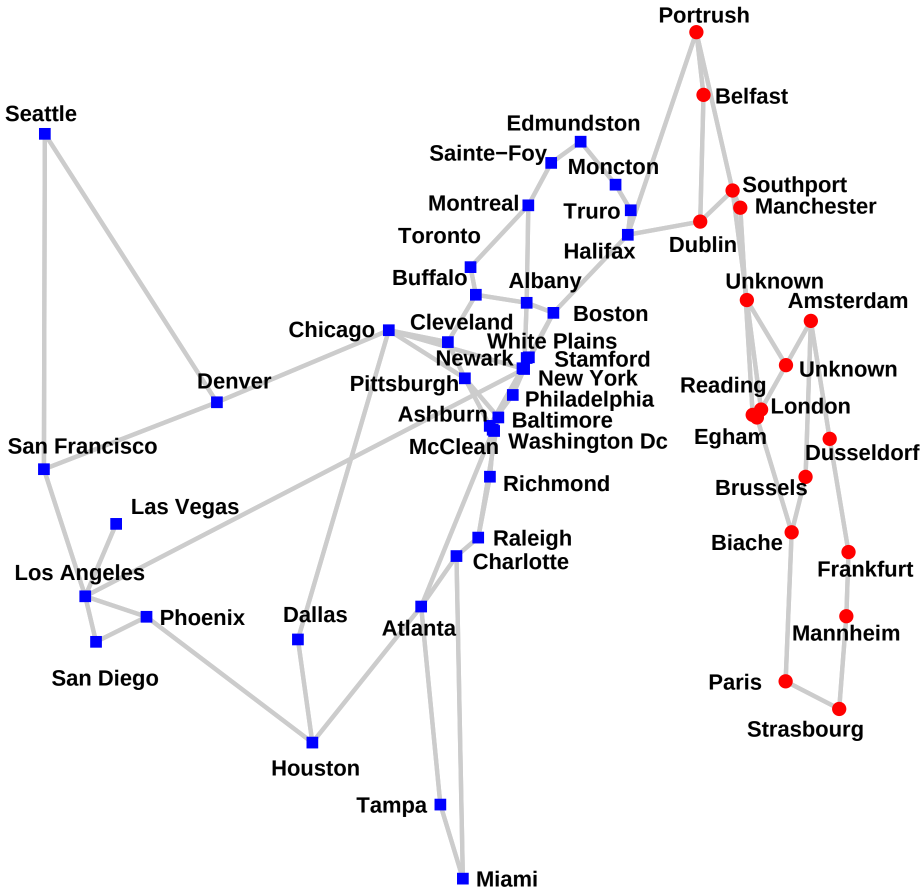


Figure D.3: 2 clusters found with the proposed automated model order selection (AMOS) algorithm for the Hibernia Internet backbone map with city names. The clusters are consistent with the geographic locations in the sense that one cluster contains cities in America and the other cluster contains cities in Europe.

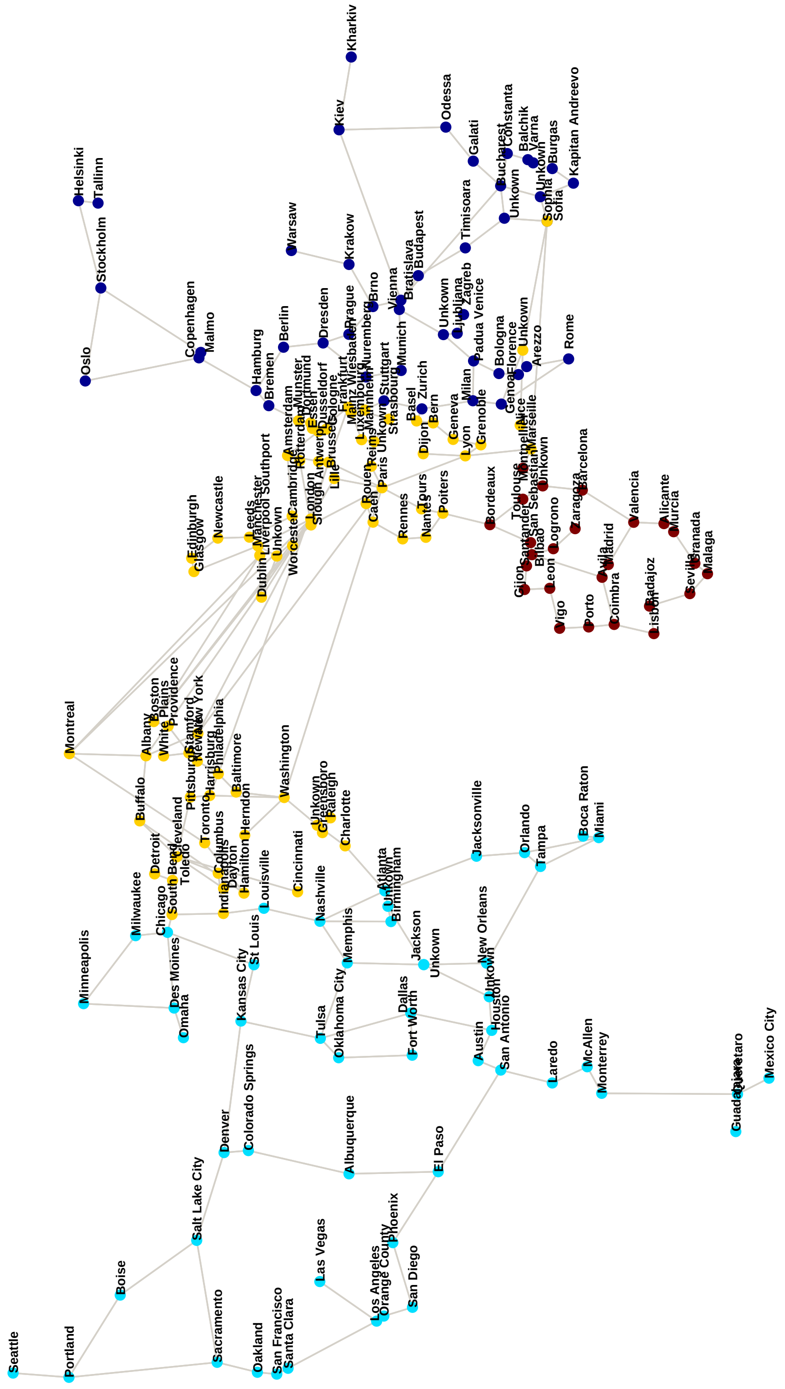


Figure D.4: 4 clusters found with the proposed automated model order selection (AMOS) algorithm for the Cogent Internet backbone map with city names. Clusters are separated by geographic locations except for the cluster containing cities in North Eastern America and West Europe due to many transoceanic connections.

APPENDIX E

Appendix of Chapter VI

E.1 Proof of Theorem 6.1

Given a layer weight vector $\mathbf{w} \in \mathcal{W}_L$, using (6.2) the graph Laplacian matrix $\mathbf{L}^{\mathbf{w}}$ of the graph $G^{\mathbf{w}}$ via convex layer aggregation can be written in the block representation such that its (i,j)-th block of dimension $n_i \times n_j$ satisfies

$$\mathbf{L}^{\mathbf{w}}_{ij} = \begin{cases} \mathbf{L}^{\mathbf{w}}_i + \sum_{z=1,\ z \neq i}^{K} \mathbf{S}^{\mathbf{w}}_{iz}, & \text{if } i = j, \\ -\mathbf{F}^{\mathbf{w}}_{ij}, & \text{if } i \neq j, \end{cases} \tag{E.1}$$

for $1 \leq i,j \leq K$, where $\mathbf{S}^{\mathbf{w}}_{ij} = \text{diag}(\sum_{\ell=1}^{L} w_\ell \mathbf{F}^{(\ell)}_{ij} \mathbf{1}_{n_j})$ is the diagonal nodal strength matrix contributed by the inter-cluster edges between clusters i and j of the graph $G^{\mathbf{w}}$, and $\mathbf{F}^{\mathbf{w}}_{ij} = \sum_{\ell=1}^{L} w_\ell \mathbf{F}^{(\ell)}_{ij}$.

Applying the block representation in (E.1) to the minimization problem in (6.3), let $\boldsymbol{\nu} \in \mathbb{R}^{(K-1)}$ and $\mathbf{U} \in \mathbb{R}^{(K-1)\times(K-1)}$ with $\mathbf{U} = \mathbf{U}^T$ be the Lagrange multiplier of the constraints $\mathbf{X}^T \mathbf{1}_n = \mathbf{0}_{K-1}$ and $\mathbf{X}^T \mathbf{X} = \mathbf{I}_{K-1}$, respectively. The Lagrangian function is

$$\Gamma(\mathbf{X}) = \text{trace}(\mathbf{X}^T\mathbf{L}^{\mathbf{w}}\mathbf{X}) - \boldsymbol{\nu}^T\mathbf{X}^T\mathbf{1}_n - \text{trace}\Big(\mathbf{U}(\mathbf{X}^T\mathbf{X} - \mathbf{I}_{K-1})\Big). \tag{E.2}$$

Let $\mathbf{Y} \in \mathbb{R}^{n\times(K-1)}$ be the solution of (6.3). Differentiating (E.2) with respect to $\mathbf{X}$ and substituting $\mathbf{Y}$ into the equations, we obtain the optimality condition

$$2\mathbf{L}^{\mathbf{w}}\mathbf{Y} - \mathbf{1}_n\boldsymbol{\nu}^T - 2\mathbf{Y}\mathbf{U} = \mathbf{O}, \tag{E.3}$$

where $\mathbf{O}$ is a matrix of zero entries. Left multiplying (E.3) by $\mathbf{1}_n^T$, we obtain

$$\boldsymbol{\nu} = \mathbf{0}_{K-1}. \tag{E.4}$$

Left multiplying (E.3) by $\mathbf{Y}^T$ and using (E.4), we have

$$\mathbf{U} = \mathbf{Y}^T\mathbf{L}\mathbf{Y} = \text{diag}(\lambda_2(\mathbf{L}^{\mathbf{w}}), \lambda_3(\mathbf{L}^{\mathbf{w}}), \ldots, \lambda_K(\mathbf{L}^{\mathbf{w}})), \tag{E.5}$$

which we denote by the diagonal matrix $\boldsymbol{\Lambda}$. Therefore, by (6.3) we have

$$S_{2:K}(\mathbf{L}^{\mathbf{w}}) = \text{trace}(\mathbf{U}). \tag{E.6}$$

Now let $\mathbf{X} = [\mathbf{X}_1^T, \mathbf{X}_2^T, \ldots, \mathbf{X}_K^T]^T$ and $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T$, where $\mathbf{X}_k \in \mathbb{R}^{n_k\times(K-1)}$ and $\mathbf{Y}_k \in \mathbb{R}^{n_k\times(K-1)}$. With (E.5), the Lagrangian function in (E.2) can be written as

$$\begin{aligned}\Gamma(\mathbf{X}) = &\sum_{k=1}^{K}\text{trace}(\mathbf{X}_k^T\mathbf{L}_k^{\mathbf{w}}\mathbf{X}_k) + \sum_{k=1}^{K}\sum_{j=1,j\neq k}^{K}\text{trace}(\mathbf{X}_k^T\mathbf{S}_{kj}^{\mathbf{w}}\mathbf{X}_k)\\ &-\sum_{k=1}^{K}\sum_{j=1,j\neq k}^{K}\text{trace}(\mathbf{X}_k^T\mathbf{F}_{kj}^{\mathbf{w}}\mathbf{X}_j) - \sum_{k=1}^{K}\text{trace}(\mathbf{U}\mathbf{X}_k^T\mathbf{X}_k) + \text{trace}(\mathbf{U}).\end{aligned} \tag{E.7}$$

Differentiating (E.7) with respect to $\mathbf{X}_k$ and substituting $\mathbf{Y}_k$ into the equation, we obtain the optimality condition that for all $k \in \{1, 2, \ldots, K\}$,

$$\mathbf{L}_k^{\mathbf{w}} \mathbf{Y}_k + \sum_{j=1, j \neq k}^{K} \mathbf{S}_{kj}^{\mathbf{w}} \mathbf{Y}_k - \sum_{j=1, j \neq k}^{K} \mathbf{F}_{kj}^{\mathbf{w}} \mathbf{Y}_j - \mathbf{Y}_k \mathbf{U} = \mathbf{O}. \tag{E.8}$$

Using the concentration results for $\mathbf{F}_{ij}^{(\ell)}$ from Appendix E.3 that

$$\frac{\mathbf{F}_{ij}^{(\ell)}}{\sqrt{n_i n_j}} \xrightarrow{\text{a.s.}} t_{ij}^{(\ell)} \mathbf{1}\mathbf{1}^T \tag{E.9}$$

as $n_i, n_j \to \infty$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, where $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence and $\mathbf{1}$ is the constant vector of unit norm, we have

$$\frac{\mathbf{F}_{ij}^{\mathbf{w}}}{\sqrt{n_i n_j}} = \frac{\sum_{\ell}^{L} w_\ell \mathbf{F}_{ij}^{(\ell)}}{\sqrt{n_i n_j}} \xrightarrow{\text{a.s.}} \sum_{\ell}^{L} w_\ell t_{ij}^{(\ell)} \mathbf{1}\mathbf{1}^T \tag{E.10}$$

and

$$\frac{\mathbf{S}_{ij}^{\mathbf{w}}}{n_j} = \frac{\text{diag}(\sum_{\ell}^{L} w_\ell \mathbf{F}_{ij}^{(\ell)} \mathbf{1}_{n_j})}{n_j} \xrightarrow{\text{a.s.}} \sum_{\ell}^{L} w_\ell t_{ij}^{(\ell)} \mathbf{I}. \tag{E.11}$$

Using (E.11) and left multiplying (E.8) by $\frac{\mathbf{1}_{n_k}^T}{n}$ gives

$$\frac{1}{n} \left[\sum_{\ell=1}^{L} \sum_{j=1, j \neq k}^{K} n_j w_\ell t_{kj}^{(\ell)} \mathbf{1}_{n_k}^T \mathbf{Y}_k - \sum_{\ell=1}^{L} \sum_{j=1, j \neq k}^{K} n_k w_\ell t_{kj}^{(\ell)} \mathbf{1}_{n_j}^T \mathbf{Y}_j - \mathbf{1}_{n_k}^T \mathbf{Y}_k \mathbf{U} \right] \xrightarrow{\text{a.s.}} \mathbf{0}_{K-1}^T,$$

$$\forall\, k \in \{1, \ldots, K\}. \tag{E.12}$$

Using the centrality relation $\mathbf{1}_{n_K}^T \mathbf{Y}_{n_K} = -\sum_{j=1}^{K-1} \mathbf{1}_{n_j}^T \mathbf{Y}_{n_j}$ and (E.6), (E.12) can be represented as an asymptotic form of Sylvester's equation

$$\frac{1}{n} \left(\widetilde{\mathbf{W}}^{\mathbf{w}} \mathbf{Z} - \mathbf{Z} \boldsymbol{\Lambda} \right) \xrightarrow{\text{a.s.}} \mathbf{O}, \tag{E.13}$$

where $\mathbf{Z} = [\mathbf{Y}_{n_1}^T \mathbf{1}_{n_1}, \mathbf{Y}_{n_2}^T \mathbf{1}_{n_2}, \ldots, \mathbf{Y}_{n_{K-1}}^T \mathbf{1}_{n_{K-1}}]^T \in \mathbb{R}^{(K-1)\times(K-1)}$ and $\widetilde{\mathbf{W}}^{\mathbf{w}}$ is the matrix defined in Theorem 6.1.

Let $\otimes$ denote the Kronecker product defined in Appendix H.1 and let $\mathbf{vec}(\mathbf{Z})$ denote the vectorization operation of $\mathbf{Z}$ by stacking the columns of $\mathbf{Z}$ into a column vector. Then (E.13) can be represented as

$$\frac{1}{n}(\mathbf{I}_{K-1} \otimes \widetilde{\mathbf{W}}^{\mathbf{w}} - \mathbf{\Lambda} \otimes \mathbf{I}_{K-1})\mathbf{vec}(\mathbf{Z}) \xrightarrow{\text{a.s.}} \mathbf{0}, \tag{E.14}$$

where the matrix $\mathbf{I}_{K-1} \otimes \widetilde{\mathbf{W}}^{\mathbf{w}} - \mathbf{\Lambda} \otimes \mathbf{I}_{K-1}$ is the Kronecker sum, denoted by $\widetilde{\mathbf{W}}^{\mathbf{w}} \oplus -\mathbf{\Lambda}$. Observe that $\mathbf{vec}(\mathbf{Z}) = \mathbf{0}$ is always a trivial solution to (E.14), and if $\widetilde{\mathbf{W}}^{\mathbf{w}} \oplus -\mathbf{\Lambda}$ is non-singular, $\mathbf{vec}(\mathbf{Z}) = \mathbf{0}$ is the unique solution to (E.14). Since $\mathbf{vec}(\mathbf{Z}) = \mathbf{0}$ implies $\mathbf{1}_{n_k}^T \mathbf{Y}_{n_k} = \mathbf{0}_{K-1}^T$ for all $k = 1, 2, \ldots, K$, the centroid $\frac{\mathbf{1}_{n_k}^T \mathbf{Y}_{n_k}}{n_k}$ of each cluster in the eigenspace is a zero vector, the clusters are not separable, and therefore correct clustering is not possible. Therefore, a sufficient condition for multilayer SGC with layer weight vector $\mathbf{w}$ to fail is that the matrix $\mathbf{I}_{K-1} \otimes \widetilde{\mathbf{W}}^{\mathbf{w}} - \mathbf{\Lambda} \otimes \mathbf{I}_{K-1}$ be non-singular. Moreover, using the property of the Kronecker sum that the eigenvalues of $\widetilde{\mathbf{W}}^{\mathbf{w}} \oplus -\mathbf{\Lambda}$ satisfy $\{\lambda_\ell(\widetilde{\mathbf{W}}^{\mathbf{w}} \oplus -\mathbf{\Lambda})\}_{z=1}^{(K-1)^2} = \{\lambda_i(\widetilde{\mathbf{W}}^{\mathbf{w}}) - \lambda_j(\mathbf{\Lambda})\}_{i,j=1}^{K-1}$, the sufficient condition on failure of multilayer SGC is that for every $\mathbf{w} \in \mathcal{W}_L$, $\lambda_i\left(\frac{\widetilde{\mathbf{W}}^{\mathbf{w}}}{n}\right) \neq \lambda_j\left(\frac{\mathbf{L}^{\mathbf{w}}}{n}\right)$ for all $i = 1, 2, \ldots, K-1$ and $j = 2, 3, \ldots, K$.

E.2 Proof of Theorem 6.2

Following the derivations in Appendix E.1, since $\mathbf{1}_{n_k}^T \mathbf{Y}_k = -\sum_{j=1, j\neq k}^{K} \mathbf{1}_{n_j}^T \mathbf{Y}_j$ by the centrality constraint, under the block-wise identical noise model (i.e., $t_{ij}^{(\ell)} = t^{(\ell)}$ for all $\ell = 1, 2, \ldots, L$), the optimality condition in (E.12) can be simplified to

$$\left(t^{\mathbf{w}} \mathbf{I}_{K-1} - \frac{\mathbf{U}}{n}\right) \mathbf{Y}_k^T \mathbf{1}_{n_k} \xrightarrow{\text{a.s.}} \mathbf{0}_{K-1},\ \forall\ k, \tag{E.15}$$

where $t^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell t^{(\ell)}$ is the aggregated noise level given a layer weight vector $\mathbf{w}$. The optimality condition in (E.15) implies that one of the two cases below has to hold:

$$\text{Case 1: } \frac{\mathbf{U}}{n} \xrightarrow{\text{a.s.}} t^{\mathbf{w}} \mathbf{I}_{K-1}; \tag{E.16}$$

$$\text{Case 2: } \mathbf{Y}_k^T \mathbf{1}_{n_k} \xrightarrow{\text{a.s.}} \mathbf{0}_{K-1},\ \forall\ k. \tag{E.17}$$

Note that with (E.6), Case 1 implies

$$\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n} = \frac{\text{trace}(\mathbf{U})}{n} \xrightarrow{\text{a.s.}} (K-1)t^{\mathbf{w}}. \tag{E.18}$$

Furthermore, in Case 1, left multiplying (E.8) by $\frac{\mathbf{Y}_k^T}{n}$ and using (E.9) and (E.11) gives

$$\frac{1}{n}\left[\mathbf{Y}_k^T \mathbf{L}_k^{\mathbf{w}} \mathbf{Y}_k + \sum_{j=1, j\neq k}^{K} n_j t^{\mathbf{w}} \mathbf{Y}_k^T \mathbf{Y}_k - \sum_{j=1, j\neq k}^{K} t^{\mathbf{w}} \mathbf{Y}_k^T \mathbf{1}_{n_k} \mathbf{1}_{n_j}^T \mathbf{Y}_j - \mathbf{Y}_k^T \mathbf{Y}_k \mathbf{U}\right] \xrightarrow{\text{a.s.}} \mathbf{O},\ \forall\ k. \tag{E.19}$$

Since $\mathbf{1}_{n_k}^T \mathbf{Y}_k = -\sum_{j=1, j\neq k}^{K} \mathbf{1}_{n_j}^T \mathbf{Y}_j$, (E.19) can be simplified as

$$\frac{1}{n}\left[\mathbf{Y}_k^T \mathbf{L}_k^{\mathbf{w}} \mathbf{Y}_k + (n-n_k) t^{\mathbf{w}} \mathbf{Y}_k^T \mathbf{Y}_k + t^{\mathbf{w}} \mathbf{Y}_k^T \mathbf{1}_{n_k} \mathbf{1}_{n_k}^T \mathbf{Y}_k - \mathbf{Y}_k^T \mathbf{Y}_k \mathbf{U}\right] \xrightarrow{\text{a.s.}} \mathbf{O},\ \forall\ k. \tag{E.20}$$

Taking the trace of (E.20) and using (E.16), we have

$$\frac{1}{n}\left[\text{trace}(\mathbf{Y}_k^T \mathbf{L}_k^{\mathbf{w}} \mathbf{Y}_k)\right] + \frac{t^{\mathbf{w}}}{n}\left[\text{trace}(\mathbf{Y}_k^T \mathbf{1}_{n_k} \mathbf{1}_{n_k}^T \mathbf{Y}_k) - n_k \text{trace}(\mathbf{Y}_k^T \mathbf{Y}_k)\right] \xrightarrow{\text{a.s.}} 0,\ \forall\ k. \tag{E.21}$$

Since (E.21) has to be satisfied for all values of $t^{\mathbf{w}}$ in Case 1, this implies the following two conditions have to hold simultaneously:

$$\begin{aligned}&\tfrac{1}{n}\left[\mathrm{trace}(\mathbf{Y}_k^T\mathbf{L}_k^{\mathbf{w}}\mathbf{Y}_k)\right]\xrightarrow{\text{a.s.}}0,~\forall~k;\\&\tfrac{1}{n}\left[\mathrm{trace}(\mathbf{Y}_k^T\mathbf{1}_{n_k}\mathbf{1}_{n_k}^T\mathbf{Y}_k)-n_k\mathrm{trace}(\mathbf{Y}_k^T\mathbf{Y}_k)\right]\xrightarrow{\text{a.s.}}0,~\forall~k.\end{aligned}\tag{E.22}$$

Since $\mathbf{L}_k^{\mathbf{w}}=\sum_{\ell=1}^{L}w_\ell\mathbf{L}_k^{(\ell)}$ is a positive semidefinite (PSD) matrix, $\mathbf{L}_k^{\mathbf{w}}\mathbf{1}_{n_k}=\mathbf{0}_{n_k}$, and $\lambda_2(\mathbf{L}_k^{\mathbf{w}})>0$, $\frac{1}{n}\left[\mathrm{trace}(\mathbf{Y}_k^T\mathbf{L}_k^{\mathbf{w}}\mathbf{Y}_k)\right]\xrightarrow{\text{a.s.}}0$ implies that every column of $\mathbf{L}_k^{\mathbf{w}}$ is a constant vector. Therefore, (E.22) implies that in Case 1,

$$\mathbf{Y}_k\xrightarrow{\text{a.s.}}\mathbf{1}_{n_k}\mathbf{1}_{K-1}^T\mathbf{V}_k=\left[v_1^k\mathbf{1}_{n_k},v_2^k\mathbf{1}_{n_k},\ldots,v_{K-1}^k\mathbf{1}_{n_k}\right],\tag{E.23}$$

where $\mathbf{V}=\mathrm{diag}(v_1^k,v_2^k,\ldots,v_{K-1}^k)$ is a diagonal matrix.

To prove the phase transition results in Theorem 6.2 (a), let $\mathcal{S}=\{\mathbf{X}\in\mathbb{R}^{n\times(K-1)}:~\mathbf{X}^T\mathbf{X}=\mathbf{I}_{K-1},~\mathbf{X}^T\mathbf{1}_n=\mathbf{0}_{K-1}\}$. In Case 2, since $\mathbf{Y}_k^T\mathbf{1}_{n_k}\xrightarrow{\text{a.s.}}\mathbf{0}_{K-1}~\forall~k$ from (E.17), we have

$$\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n}\xrightarrow{\text{a.s.}}\min_{\mathbf{X}\in\mathcal{S}}\left\{\frac{1}{n}\left[\sum_{k=1}^{K}\mathrm{trace}(\mathbf{X}_k^T\mathbf{L}_k^{\mathbf{w}}\mathbf{X}_k)+t^{\mathbf{w}}\sum_{k=1}^{K}(n-n_k)\mathrm{trace}(\mathbf{X}_k^T\mathbf{X}_k)\right]\right\}\tag{E.24}$$

$$\geq\min_{\mathbf{X}\in\mathcal{S}}\left\{\frac{1}{n}\sum_{k=1}^{K}\mathrm{trace}(\mathbf{X}_k^T\mathbf{L}_k^{\mathbf{w}}\mathbf{X}_k)\right\}+\min_{\mathbf{X}\in\mathcal{S}}\left\{\frac{t^{\mathbf{w}}}{n}\sum_{k=1}^{K}(n-n_k)\mathrm{trace}(\mathbf{X}_k^T\mathbf{X}_k)\right\}\tag{E.25}$$

$$=\min_{k\in\{1,2,\ldots,K\}}\left\{\frac{S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{n}\right\}+\frac{(K-1)t^{\mathbf{w}}}{n}\min_{k\in\{1,2,\ldots,K\}}(n-n_k)\tag{E.26}$$

$$=\min_{k\in\{1,2,\ldots,K\}}\left\{\frac{S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{n}\right\}+\frac{(K-1)(n-n_{\max})t^{\mathbf{w}}}{n},\tag{E.27}$$

where $n_{\max}=\max_{k\in\{1,2,\ldots,K\}}n_k$.

Similarly, let $\mathcal{S}_k=\{\mathbf{X}\in\mathbb{R}^{n\times(K-1)}:~\mathbf{X}_k^T\mathbf{X}_k=\mathbf{I}_{K-1},~\mathbf{X}_j=\mathbf{O}_{n_j\times(K-1)}~\forall~j\neq$

$k,~ \mathbf{X}^T \mathbf{1}_n = \mathbf{0}_{K-1}\}$. Since $\mathcal{S}_k \subseteq \mathcal{S}$, in Case 2, we have

$$\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n} \xrightarrow{\text{a.s.}} \min_{\mathbf{X} \in \mathcal{S}} \left\{ \frac{1}{n} \left[\sum_{k=1}^{K} \text{trace}(\mathbf{X}_k^T \mathbf{L}_k^{\mathbf{w}} \mathbf{X}_k) + t^{\mathbf{w}} \sum_{k=1}^{K} (n - n_k) \text{trace}(\mathbf{X}_k^T \mathbf{X}_k) \right] \right\} \tag{E.28}$$

$$\leq \min_{k \in \{1,2,\ldots,K\}} \min_{\mathbf{X} \in \mathcal{S}_k} \left\{ \frac{1}{n} \left[\sum_{k=1}^{K} \text{trace}(\mathbf{X}_k^T \mathbf{L}_k^{\mathbf{w}} \mathbf{X}_k) + t^{\mathbf{w}} \sum_{k=1}^{K} (n - n_k) \text{trace}(\mathbf{X}_k^T \mathbf{X}_k) \right] \right\} \tag{E.29}$$

$$= \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{1}{n} \left[S_{2:K}(\mathbf{L}_k^{\mathbf{w}}) + (K-1)(n - n_k) t^{\mathbf{w}} \right] \right\} \tag{E.30}$$

$$\leq \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{1}{n} \left[S_{2:K}(\mathbf{L}_k^{\mathbf{w}}) + (K-1)(n - n_{\min}) t^{\mathbf{w}} \right] \right\} \tag{E.31}$$

$$= \min_{k \in \{1,2,\ldots,K\}} \left\{ \frac{S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{n} \right\} + \frac{(K-1)(n - n_{\min}) t^{\mathbf{w}}}{n}, \tag{E.32}$$

where $n_{\min} = \min_{k \in \{1,2,\ldots,K\}} n_k$. Therefore, we obtain the phase transition results in Theorem 6.2 (a).

Proceeding to Theorem 6.2 (b), we first note that each cluster-wise eigenvector component $\mathbf{Y}_k$ in $\mathbf{Y}$ has to either satisfy the cluster-wise separability in (E.23) or the zero row-sum condition in (E.17). To show the conditions (b-1) to (b-3) in Theorem 6.2 (b), recall the eigenvector matrix $\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_K^T]^T$, where $\mathbf{Y}_k$ is the $n_k \times (K-1)$ matrix with row vectors representing the nodes from cluster k. Since $\mathbf{Y}^T \mathbf{Y} = \sum_{k=1}^{K} \mathbf{Y}_k^T \mathbf{Y}_k = \mathbf{I}_{(K-1)\times(K-1)}$, $\mathbf{Y}^T \mathbf{1}_n = \sum_{k=1}^{K} \mathbf{Y}_k^T \mathbf{1}_{n_k} = \mathbf{0}_{K-1}$, and from (E.23) when $t^{\mathbf{w}} < t^{\mathbf{w}*}$ the matrix $\mathbf{Y}_k \xrightarrow{\text{a.s.}} \mathbf{1}_{n_k} \mathbf{1}_{K-1}^T \mathbf{V}_k = \left[v_1^k \mathbf{1}_{n_k}, v_2^k \mathbf{1}_{n_k}, \ldots, v_{K-1}^k \mathbf{1}_{n_k} \right]$ as $n_k \to \infty ~\forall~ k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, we have

$$\begin{aligned} &\textstyle\sum_{k=1}^{K} n_k \mathbf{v_k} \mathbf{v_k}^T = \mathbf{I}_{(K-1)\times(K-1)}; \\ &\textstyle\sum_{k=1}^{K} n_k \mathbf{v_k} = \mathbf{0}_{K-1}, \end{aligned} \tag{E.33}$$

where $\mathbf{v_k} = \mathbf{V}_k \mathbf{1}_{n_k} = [v_1^k, v_2^k, \ldots, v_{K-1}^k]^T$. (E.33) suggests that some $\mathbf{v_k}$ cannot be a zero vector since $\sum_{k=1}^{K} n_k (v_j^k)^2 = 1$ for all $j \in \{1, 2, \ldots, K-1\}$, and from (E.33) we have

$$\begin{aligned} & \textstyle\sum_{k: v_j^k > 0} n_k v_j^k = - \sum_{k: v_j^k < 0} n_k v_j^k, \\ & \forall\, j \in \{1, 2, \ldots, K-1\}; \\ & \textstyle\sum_{k: v_i^k v_j^k > 0} n_k v_i^k v_j^k = - \sum_{k: v_i^k v_j^k < 0} n_k v_i^k v_j^k, \\ & \forall\, i, j \in \{1, 2, \ldots, K-1\}, i \neq j. \end{aligned} \tag{E.34}$$

As a results, the optimality conditions of $\mathbf{v}_k$ in (E.33) and (E.34) lead to the conditions (b-1) to (b-3) in Theorem 6.2.

Lastly, comparing (E.18) with (E.27) and (E.32), as a function of $t^{\mathbf{w}}$ the slope of $\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n}$ changes at some critical value $t^{\mathbf{w}*}$ that separates Case 1 and Case 2. By the continuity of $\frac{S_{2:K}(\mathbf{L}^{\mathbf{w}})}{n}$, a lower bound on $t^{\mathbf{w}*}$ is

$$t_{\mathrm{LB}}^{\mathbf{w}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{(K-1) n_{\max}}, \tag{E.35}$$

and an upper bound on $t^{\mathbf{w}*}$ is

$$t_{\mathrm{UB}}^{\mathbf{w}} = \frac{\min_{k \in \{1,2,\ldots,K\}} S_{2:K}(\mathbf{L}_k^{\mathbf{w}})}{(K-1) n_{\min}}. \tag{E.36}$$

In particular, if $c = 1$, then $n_{\max} = n_{\min} = \frac{n}{K}$ and hence the expressions in (E.27) and (E.32) are identical, which completes Theorem 6.2 (c).

E.3 Proof of Theorem 6.3

The following lemma provides bounds on the smallest $K-1$ nonzero eigenvalues of $\mathbf{L}^{\mathbf{w}}$ under the block-wise non-identical noise model.

Lemma. *Under the block-wise non-identical noise model in Sec. 6.1.2 with maximum noise level $\{t_{\max}^{(\ell)}\}_{\ell=1}^{L}$ for each layer, given a layer weight vector $\mathbf{w} \in \mathcal{W}_L$, let $t_{\min}^{\mathbf{w}} =$*

$\sum_{\ell=1}^{L} w_\ell \min_{i \neq j} t_{ij}^{(\ell)}$, $t_{\max}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \max_{i \neq j} t_{ij}^{(\ell)}$, and let $t^{\mathbf{w}*}$ be the critical threshold value for the block-wise identical noise model specified by Theorem 6.2. If $t_{\max}^{\mathbf{w}} < t^{\mathbf{w}*}$, the following statement holds almost surely as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$:

$$t_{\min}^{\mathbf{w}} \leq \lambda_j \left(\frac{\mathbf{L}^{\mathbf{w}}}{n} \right) \leq t_{\max}^{\mathbf{w}}, \; \forall \; j = 2, 3, \ldots, K. \tag{E.37}$$

Proof. We first show that when $t_{\max}^{\mathbf{w}} < t^{\mathbf{w}*}$, the second eigenvalue of $\frac{\mathbf{L}^{\mathbf{w}}}{n}$, $\lambda_2(\frac{\mathbf{L}^{\mathbf{w}}}{n})$, lies within the interval $[t_{\min}^{\mathbf{w}}, t_{\max}^{\mathbf{w}}]$ almost surely as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Under the block-wise non-identical noise model in Sec. 6.1.2, by (E.9) with proper scaling the entries of each interconnection matrix $\mathbf{F}_{ij}^{(\ell)}$ converge to $t_{ij}^{(\ell)}$ almost surely as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Let $\mathbf{W}^{\mathbf{w}}(t^{\mathbf{w}})$ be the weight matrix of the aggregated graph $G^{\mathbf{w}}$ under the block-wise identical noise model with aggregated noise level $t^{\mathbf{w}}$. Then the weight matrix $\mathbf{W}^{\mathbf{w}}$ can be written as $\mathbf{W}^{\mathbf{w}} = \mathbf{W}^{\mathbf{w}}(t_{\min}^{\mathbf{w}}) + \mathbf{\Delta W}^{\mathbf{w}}$, and the corresponding graph Laplacian matrix can be written as $\mathbf{L}^{\mathbf{w}} = \mathbf{L}^{\mathbf{w}}(t_{\min}^{\mathbf{w}}) + \mathbf{\Delta L}^{\mathbf{w}}$, where $\mathbf{L}^{\mathbf{w}}(t_{\min}^{\mathbf{w}})$ and $\mathbf{\Delta L}^{\mathbf{w}}$ are associated with $\mathbf{W}^{\mathbf{w}}(t^{\mathbf{w}})$ and $\mathbf{\Delta W}^{\mathbf{w}}$, respectively. Since $t_{\min}^{\mathbf{w}} = \sum_{\ell=1}^{L} w_\ell \min_{i \neq j} t_{ij}^{(\ell)}$, as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, $\frac{\mathbf{\Delta W}^{\mathbf{w}}}{n}$ is a symmetric nonnegative matrix almost surely, and $\frac{\mathbf{\Delta L}^{\mathbf{w}}}{n}$ is a graph Laplacian matrix almost surely. By the PSD property of a graph Laplacian matrix, we obtain $\lambda_2(\frac{\mathbf{L}^{\mathbf{w}}}{n}) \geq t_{\min}^{\mathbf{w}}$ almost surely as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Similarly, following the same procedure we can show that $\lambda_2(\frac{\mathbf{L}^{\mathbf{w}}}{n}) \leq t_{\max}^{\mathbf{w}}$ almost surely as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. Lastly, when $t^{\mathbf{w}} < t^{\mathbf{w}*}$, using the fact from (E.16) that $\lambda_j(\frac{\mathbf{L}^{\mathbf{w}}(t^{\mathbf{w}})}{n}) \xrightarrow{\text{a.s.}} t^{\mathbf{w}}$ for all $j \in \{2, 3, \ldots, K\}$, we obtain

$$t_{\min}^{\mathbf{w}} = \lambda_j \left(\frac{\mathbf{L}(t_{\min}^{\mathbf{w}})}{n} \right) \leq \lambda_j \left(\frac{\mathbf{L}^{\mathbf{w}}}{n} \right) \leq \lambda_j \left(\frac{\mathbf{L}(t_{\max}^{\mathbf{w}})}{n} \right) = t_{\max}^{\mathbf{w}} \tag{E.38}$$

almost surely for all $j \in \{2, 3, \ldots, K\}$ as $n_k \to \infty \; \forall \; k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$. □

Proceeding to proving Theorem 6.3, applying the Davis-Kahan $\sin \theta$ theorem [22] to the eigenvector matrices $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ associated with the graph Laplacian matrices $\frac{\mathbf{L}^{\mathbf{w}}}{n}$ and $\frac{\widetilde{\mathbf{L}}^{\mathbf{w}}}{n}$, respectively, we obtain an upper bound on the distance of column spaces spanned by $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$, which is $\| \sin \mathbf{\Theta}(\mathbf{Y}, \widetilde{\mathbf{Y}}\|_F \leq \frac{\|\mathbf{L}^{\mathbf{w}} - \widetilde{\mathbf{L}}^{\mathbf{w}}\|_F}{n\delta}$, where $\delta = \inf\{|x-y| : x \in \{0\} \cup [\lambda_{K+1}(\frac{\mathbf{L}^{\mathbf{w}}}{n}), \infty),\ y \in [\lambda_2(\frac{\widetilde{\mathbf{L}}^{\mathbf{w}}}{n}), \lambda_K(\frac{\widetilde{\mathbf{L}}^{\mathbf{w}}}{n})]\}$. Under the block-wise identical noise model, if $t^{\mathbf{w}} < t^{\mathbf{w}*}$, using the fact from (E.16) that $\lambda_j(\frac{\widetilde{\mathbf{L}}^{\mathbf{w}}}{n}) \xrightarrow{\text{a.s.}} t^{\mathbf{w}}$ for all $j \in \{2, 3, \ldots, K\}$ as $n_k \to \infty\ \forall\ k$ and $\frac{n_{\min}}{n_{\max}} \to c > 0$, the interval $[\lambda_2(\frac{\widetilde{\mathbf{L}}^{\mathbf{w}}}{n}), \lambda_K(\frac{\widetilde{\mathbf{L}}^{\mathbf{w}}}{n})]$ reduces to a point $t^{\mathbf{w}}$ almost surely. Therefore, δ reduces to $\delta_{t^{\mathbf{w}}}$ as defined in Theorem 6.3. Furthermore, if $t^{\mathbf{w}}_{\max} \leq t^{\mathbf{w}*}$, then (6.6) holds for all $t^{\mathbf{w}} \leq t^{\mathbf{w}}_{\max}$. Taking the minimum over all upper bounds in (6.6) for every $t^{\mathbf{w}} \leq t^{\mathbf{w}}_{\max}$, we obtain (6.7).

E.4 Proof of the condition in (6.12)

First, using the Anscombe transformation on $\{\widehat{p}_{ij}^{(\ell)}\}$ for variance stabilization [8], let $A_{ij}(x) = \sin^{-1}\sqrt{\frac{x + \frac{c'}{\widehat{n}_i \widehat{n}_j}}{1 + \frac{2c'}{\widehat{n}_i \widehat{n}_j}}}$, where $c' = \frac{3}{8}$. By the central limit theorem, $\sqrt{4\widehat{n}_i\widehat{n}_j + 2} \cdot \left(A_{ij}(\widehat{p}_{ij}^{(\ell)}) - A_{ij}(p_{ij}^{(\ell)})\right) \xrightarrow{d} N(0,1)$ for all $p_{ij}^{(\ell)} \in (0,1)$ as $\widehat{n}_i, \widehat{n}_j \to \infty$, where $\xrightarrow{d}$ denotes convergence in distribution and $N(0,1)$ denotes the standard normal distribution [8]. Therefore, under the null hypothesis $H_0^{(\ell)}$, from [23, Theorem 2.1] an asymptotic $100(1-\alpha')\%$ confidence interval for $\widehat{t}^{(\ell)}_{\max}$ is $[0, \psi_\ell]$, where $\psi(\alpha'_\ell, \{\widehat{t}_{ij}^{(\ell)}\})$ is a function of the precision parameter $\alpha'_\ell \in [0,1]$ and $\{\widehat{t}_{ij}^{(\ell)}\}$, which satisfies $\prod_{i=1}^K \prod_{j=i+1}^K \Phi\left(\sqrt{4\widehat{n}_i\widehat{n}_j + 2} \cdot \left(A_{ij}(\psi_\ell) - A_{ij}\left(\frac{\widehat{t}_{ij}^{(\ell)}}{\widehat{\overline{W}}_{ij}^{(\ell)}}\right)\right)\right) = 1 - \alpha'_\ell$, where $\Phi(\cdot)$ is the cdf of the standard normal distribution, and we use the relation $\widehat{t}_{ij}^{(\ell)} = \widehat{p}_{ij}^{(\ell)} \cdot \widehat{\overline{W}}_{ij}^{(\ell)}$.

As a result, if $\psi_\ell < t^{\mathbf{w}}_{\text{LB}}$, then $\widehat{t}^{(\ell)}_{\max} < t^{\mathbf{w}}_{\text{LB}}$ with probability at least $1 - \alpha'_\ell$. Note that verifying $\psi_\ell < t^{\mathbf{w}}_{\text{LB}}$ is equivalent to checking the condition

$$\prod_{i=1}^{K} \prod_{j=i+1}^{K} F_{ij}\left(\frac{t^{\mathbf{w}}_{\text{LB}}}{\overline{W}_{ij}^{(\ell)}}, \widehat{p}_{ij}^{(\ell)}\right) \geq 1 - \alpha'_\ell, \tag{E.39}$$

where $F_{ij}(\frac{t^{\mathbf{w}}_{\mathrm{LB}}}{\overline{W}^{(\ell)}_{ij}}, \widehat{p}^{(\ell)}_{ij}) = \Phi\left(\sqrt{4\widehat{n}_i\widehat{n}_j + 2} \cdot \left(A_{ij}(\frac{t^{\mathbf{w}}_{\mathrm{LB}}}{\overline{W}^{(\ell)}_{ij}}) - A_{ij}(\widehat{p}^{(\ell)}_{ij})\right)\right) \cdot \mathbb{I}_{\{\widehat{p}^{(\ell)}_{ij} \in (0,1)\}}$ $+ \mathbb{I}_{\{\widehat{t}^{(\ell)}_{ij} < t^{\mathbf{w}}_{\mathrm{LB}}\}} \mathbb{I}_{\{\widehat{p}^{(\ell)}_{ij} \in \{0,1\}\}}$. Finally, we replace $t^{\mathbf{w}}_{\mathrm{LB}}$ and $\overline{W}^{(\ell)}_{ij}$ in (E.39) with the empirical estimates $\widehat{t}^{\mathbf{w}}_{\mathrm{LB}}$ and $\widehat{\overline{W}}^{(\ell)}_{ij}$, respectively, which leads to (6.12).

APPENDIX F

Appendix of Chapter VII

F.1 Proof of Theorem 7.1

From (7.1) a graph is connected if and only if the algebraic connectivity is greater than zero. Furthermore, the smallest eigenvalue of the associated graph Laplacian matrix is always 0. Therefore $n-q-r$ is the number of connected components (including the singleton nodes) in $\widetilde{G}$ [38] by the fact that $n-q$ and r are the node size and rank of $\widetilde{\mathbf{L}}$, respectively. Since the definition of a deep community excludes singleton nodes, the first inequality in (7.4) becomes equality if all connected components in $\widetilde{G}$ are non-singleton.

Using a well-known matrix norm inequality [76] that $\|\mathbf{M}\|_* \leq r\|\mathbf{M}\|_2$ for any square matrix $\mathbf{M}$ of rank r, where $\|\mathbf{M}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{M}\mathbf{x}\|_2 = \lambda_n(\mathbf{M})$. We have

$$n-q-r \leq n-q-\frac{\|\widetilde{\mathbf{L}}\|_*}{\lambda_n(\widetilde{\mathbf{L}})} = n-q-\frac{2\widetilde{m}}{\lambda_n(\widetilde{\mathbf{L}})},$$

where $\|\widetilde{\mathbf{L}}\|_* = \mathrm{trace}(\widetilde{\mathbf{L}}) = 2\widetilde{m}$ is the total degree of $\widetilde{G}$.

Next we show that the second inequality in (7.4) becomes an equality if each non-singleton connected graph is a complete subgraph of the same size. Consider a graph consisting of g disjoint complete subgraphs of $n' \geq 2$ nodes and $n'(n'-1)/2$ edges. The largest eigenvalue of each subgraph is n' and $\|\widetilde{\mathbf{L}}\|_* = g \cdot n'(n'-1)$. The upper bound becomes $g \cdot n' - \frac{gn'(n'-1)}{n'} = g$, which is exactly the number of non-singleton connected components in $\widetilde{G}$. These results can be directly applied to edge removals in G by setting $q = 0$ since no nodes are removed.

F.2 Proof of Theorem 7.2

Let r be the rank of $\widetilde{\mathbf{L}}$. We prove that there exists an $n \times (n-r)$ binary matrix $\mathbf{X} = [\mathbf{x}_1 \; \mathbf{x}_2 \ldots \mathbf{x}_{n-r}]$ whose columns $\{\mathbf{x}_i\}_{i=1}^{n-r}$ satisfy: 1) $\|\mathbf{x}_i\|_1$ is the size of the i-th connected component of $\widetilde{G}$; 2) they are orthogonal; 3) they span null($\widetilde{\mathbf{L}}$). Assume $\widetilde{G}$ consists of g connected components. Then there exits a matrix permutation (node relabeling) such that

$$\widetilde{\mathbf{L}} = \begin{bmatrix} \widetilde{\mathbf{L}}_1 & 0 & 0 & 0 \\ 0 & \widetilde{\mathbf{L}}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \widetilde{\mathbf{L}}_g \end{bmatrix}. \tag{F.1}$$

Associated with the i-th block matrix $\widetilde{\mathbf{L}}_i$ we define $\mathbf{x}_i$ as an $n \times 1$ binary vector $\mathbf{x}_i$ in null($\widetilde{\mathbf{L}}$) having the form $\mathbf{x}_i = [0 \ldots 0 \; 1 \ldots 1 \; 0 \ldots 0]^T$, where the locations of the nonzero entries correspond to the indexes of the i-th block matrix. It is obvious that $\|\mathbf{x}_i\|_1 = \sum_{j=1}^n |x_{ij}|$ equals the size of the i-th component and such $\{\mathbf{x}_i\}_{i=1}^{n-r}$ are mutually orthogonal. Furthermore, there exists no other binary matrix which is sparser than $\mathbf{X}$ with column span equal to null($\widetilde{\mathbf{L}}$). If there existed another binary matrix that were sparser than $\mathbf{X}$, then it would contradict the fact that its column vectors have sums equal to the component sizes of $\widetilde{G}$. Therefore the largest non-singleton connected component size of $\widetilde{G}$ is $\psi(\widetilde{G}) = \|\mathbf{X}\|_1 = \max_i \|\mathbf{x}_i\|_1$.

F.3 Proof of Lemma 7.3

By the relation

$$\sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{V}}\mathbf{A}_{ij}(y_i-y_j)^2=\sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{N}_i}(y_i-y_j)^2 \tag{F.2}$$

and $\mathcal{V}=\{\mathcal{V}/\mathcal{R}\}\cup\{\mathcal{R}\}$, we have

$$\begin{aligned} f(\mathcal{R}) &= \sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{N}_i}(y_i-y_j)^2-\frac{1}{2}\sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{V}}\mathbf{A}_{ij}(y_i-y_j)^2+\frac{1}{2}\sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{V}/\mathcal{R}}\mathbf{A}_{ij}(y_i-y_j)^2 \\ &= \frac{1}{2}\sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{N}_i}(y_i-y_j)^2+\frac{1}{2}\sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{V}/\mathcal{R}}\mathbf{A}_{ij}(y_i-y_j)^2 \\ &\geq 0. \end{aligned} \tag{F.3}$$

$f(\varnothing)=0$ follows directly from the definition of $f(\mathcal{R})$ in (7.15).

F.4 Proof of Theorem 7.4

We first prove the monotonic property. Consider two node removal sets $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \mathcal{V}$. Then using Lemma 7.3 and the fact that $\mathcal{R}_1/\mathcal{R}_2 = \varnothing$,

$$\begin{aligned}
&f(\mathcal{R}_2) - f(\mathcal{R}_1) \\
&= \sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{N}_i}(y_i - y_j)^2 - \sum_{i\in\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 - \frac{1}{2}\sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 \\
&= \sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{V}}\mathbf{A}_{ij}(y_i - y_j)^2 - \sum_{i\in\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 - \sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 \\
&\ + \frac{1}{2}\sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 \\
&= \sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\left(\sum_{j\in\mathcal{V}}\mathbf{A}_{ij}(y_i - y_j)^2 - \sum_{j\in\mathcal{R}_2}\mathbf{A}_{ij}(y_i - y_j)^2\right) + \frac{1}{2}\sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 \\
&= \sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{V}/\mathcal{R}_2}\mathbf{A}_{ij}(y_i - y_j)^2 + \frac{1}{2}\sum_{i\in\mathcal{R}_2/\mathcal{R}_1}\sum_{j\in\mathcal{R}_2/\mathcal{R}_1}\mathbf{A}_{ij}(y_i - y_j)^2 \\
&\geq 0. \qquad \text{(F.4)}
\end{aligned}$$

Therefore $f(\mathcal{R})$ is a monotonic increasing set function (i.e., $f(\mathcal{R}_2) \geq f(\mathcal{R}_1)$ for all $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \mathcal{V}$).

Furthermore, $f(\mathcal{R})$ is a submodular set function [59, 103] since for any node $v \in \mathcal{V}, v \notin \mathcal{R}_2$, $\mathcal{R}_1 \subset \mathcal{R}_2 \subset \mathcal{V}$, we have from (7.15) that

$$\begin{aligned}
f(\mathcal{R}_1 \cup v) - f(\mathcal{R}_1) &= \sum_{j\in\mathcal{N}_v}(y_v - y_j)^2 - \sum_{j\in\mathcal{R}_1}\mathbf{A}_{vj}(y_v - y_j)^2 \\
&\geq \sum_{j\in\mathcal{N}_v}(y_v - y_j)^2 - \sum_{j\in\mathcal{R}_2}\mathbf{A}_{vj}(y_v - y_j)^2 \\
&= f(\mathcal{R}_2 \cup v) - f(\mathcal{R}_2). \qquad \text{(F.5)}
\end{aligned}$$

This diminishing returns property of $f(\mathcal{R})$ establishes that f is submodular [86].

F.5 Proof of Theorem 7.5

By submodularity of $f(\mathcal{R})$ in Theorem 7.4, there exists a $v \in \mathcal{R}_{\text{opt}}/\mathcal{R}_k$ [59] such that

$$f(\mathcal{R}_k \cup v) - f(\mathcal{R}_k) \geq \frac{1}{q}\left(f(\mathcal{R}_{\text{opt}}) - f(\mathcal{R}_k)\right). \tag{F.6}$$

After algebraic manipulation, we have

$$f(\mathcal{R}_{\text{opt}}) - f(\mathcal{R}_{k+1}) \leq \left(1 - \frac{1}{q}\right)(f(\mathcal{R}_{\text{opt}}) - f(\mathcal{R}_k)) \tag{F.7}$$

and therefore

$$f(\mathcal{R}_{\text{opt}}) - f(\mathcal{R}_q) \leq \left(1 - \frac{1}{q}\right)^q f(\mathcal{R}_{\text{opt}}) \leq \frac{1}{e} f(\mathcal{R}_{\text{opt}}). \tag{F.8}$$

Applying this result to (7.13), we have

$$\begin{aligned} \lambda_2(\widetilde{\mathbf{L}}(\mathcal{R}_q)) &\leq \lambda_2(\mathbf{L}) - f(\mathcal{R}_q) \\ &\leq \lambda_2(\mathbf{L}) - \left(1 - e^{-1}\right) f(\mathcal{R}_{\text{opt}}). \end{aligned} \tag{F.9}$$

APPENDIX G

Appendix of Chapter VIII

G.1 Details of the collected real-world event propagation traces on Twitter

Ee collected the traces of three recent events on Twitter during a period of two weeks through the Twitter API. These events include URLs and hashtags specified as follows.

- **Obama FB:** we tracked the tweets including the URL "http://Facebook.com/POTUS" from November 9th to November 23rd in 2015. The URL links to U.S. President Obama's personal Facebook page, and was firstly being posted by his personal Twitter account on November 9th 2015.
- **Premier 12:** we tracked the tweets including the hashtag "#premier12" from November 19th to December 3rd in 2015. Premier 12 is a flagship international baseball tournament organized by the World Baseball Softball Confederation (WBSC), featuring the twelve best-ranked national baseball teams in the world.

- **AlphaGo:** we tracked the tweets including the hashtag “#AlphaGo” from January 27th to February 10th in 2016. AlphaGo is a computer program developed by Google DeepMind in London to play the board game Go. On January 27th 2016, the news of AlphaGo defeating a European Go champion was announced along with the algorithm published in Nature [146].

G.2 Derivation of the iterative state equation in (8.1)

Since $\mathbf{A}_t$ accounts for the adjacency matrix of activated follower links for event propagation during the t-th time frame, the i-th entry of the vector $\mathbf{A}_{t+1}^T \mathbf{r}_t$ can be expressed as $[\mathbf{A}_{t+1}^T \mathbf{r}_t]_i = \sum_{j=1}^n [\mathbf{A}_t]_{ij} [\mathbf{r}_t]_j$, which is the number of tweets regarding the event that user i decides to share on Twitter during the $t+1$-th time frame. Therefore, the entry-wise thresholded binary vector $\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$ indicates the status of new users participating in event propagation during the $t+1$-th time frame. Lastly, since $\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$ represents the vector of event propagation increment, $\mathbf{r}_{t+1} = \mathbb{T}(\mathbf{r}_t + \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t))$ accounts for the event propagation status of all users since the beginning to the $t+1$-th time frame.

G.3 Proof of the upper bound in (8.3)

First, observe from (8.1) that the sparsity level $\|\mathbf{r}_t\|_0$ of $\mathbf{r}_t$ is a non-decreasing function in t. Therefore, the condition that $\|\mathbf{r}_F\|_0 \leq s$ implies $\|\mathbf{r}_t\|_0 \leq s$ for all $t \leq F$. Let $\mathbf{1}_n$ denote the n-dimensional column vector of all ones. Then the sparsity level $\|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_0$ of the binary vector $\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$ can be expressed as

$$\|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_0 = \mathbf{1}_n^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t). \tag{G.1}$$

Decomposing the term $\mathbf{1}_n^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$, we have

$$\mathbf{1}_n^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t) = \mathbf{r}_t^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t) + (\mathbf{1}_n - \mathbf{r}_t)^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t). \tag{G.2}$$

Let $\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n [\mathbf{x}]_i^2\right)^{1/2}$ denote the Euclidean norm of a vector $\mathbf{x}$. We can derive an upper bound on the term $\mathbf{r}_t^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$, which is

$$\begin{aligned}
\mathbf{r}_t^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t) &\overset{(a)}{\leq} \mathbf{r}_t^T \mathbf{A}_{t+1}^T \mathbf{r}_t \\
&\overset{(b)}{=} \mathbf{r}_t^T \mathbf{A}_{t+1} \mathbf{r}_t \\
&\overset{(c)}{=} \|\mathbf{r}_t\|_2^2 \cdot \frac{\mathbf{r}_t^T}{\|\mathbf{r}_t\|_2} \mathbf{A}_{t+1} \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|_2} \\
&\overset{(d)}{\leq} \|\mathbf{r}_t\|_2^2 \cdot \max_{\mathbf{x}:\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A}_{t+1} \mathbf{x} \\
&\overset{(e)}{=} \|\mathbf{r}_t\|_2^2 \cdot \lambda_{\max}(\mathbf{A}_{t+1}) \\
&\overset{(f)}{\leq} s \cdot \lambda_{\max}(\mathbf{A}_{t+1}) \\
&\overset{(g)}{\leq} s \cdot \lambda_{\max}(\mathbf{A}),
\end{aligned} \tag{G.3}$$

where (a) is due to the fact that $\mathbb{T}(\cdot)$ is a threshold function and $\mathbf{A}_{t+1}^T \mathbf{r}_t$ is a nonnegative vector, (b) is true since $\mathbf{r}_t^T \mathbf{A}_{t+1} \mathbf{r}_t$ is a real value, (c) is a simple arithmetic operation, (d) is due to the fact that $\frac{\mathbf{r}_t}{\|\mathbf{r}_t\|_2}$ is a vector of unit Euclidean norm, (e) is from the Courant-Fischer theorem [76], (f) uses the fact that $\mathbf{r}_t$ is a binary vector such that $\|\mathbf{r}_t\|_2^2 = \sum_{i=1}^n [\mathbf{r}_t]_i^2 = \sum_{i=1}^n [\mathbf{r}_t]_i = \|\mathbf{r}_t\|_0 \leq s$, and (g) is due to the fact fact all the nonzero entries in $\mathbf{A}_t$ also appear in $\mathbf{A}$, and hence $\lambda_{\max}(\mathbf{A}_{t+1}) \leq \lambda_{\max}(\mathbf{A})$, which can be verified by using the matrix perturbation theorem [76].

Next, using the Cauchy-Schwartz inequality, the term $(\mathbf{1}_n - \mathbf{r}_t)^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$ is upper bounded by

$$\begin{aligned}(\mathbf{1}_n - \mathbf{r}_t)^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t) &\leq \|\mathbf{1}_n - \mathbf{r}_t\|_2 \cdot \|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_2 \\ &\leq \sqrt{n} \cdot \sqrt{s}. \end{aligned} \tag{G.4}$$

$\|\mathbf{1}_n - \mathbf{r}_t\|_2 \leq \|\mathbf{1}_n\|_2 = \sqrt{n}$ is a trivial upper bound since $\mathbf{r}_t$ is a binary vector. $\|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_2 \leq \sqrt{s}$ since $\|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_2^2 = \|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_0 \leq \|\mathbf{r}_{t+1}\|_0 \leq s$ by the iterative state equation in (8.1) and the assumption that $\|\mathbf{r}_F\|_0 \leq s$.

Combining the two established upper bounds on $\mathbf{r}_t^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$ and $(\mathbf{1}_n - \mathbf{r}_t)^T \mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)$, we obtain the upper bound on $\|\mathbb{T}(\mathbf{A}_{t+1}^T \mathbf{r}_t)\|_0$ as in (8.3).

G.4 Proof of the bounds in (8.7) and (8.8)

Given a follower link removal set $\mathcal{E}_{\mathcal{R}}$ with cardinality $|\mathcal{E}_{\mathcal{R}}| = q \geq 1$, the adjacency matrix $\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})$ after removing the follower links in $\mathcal{E}_{\mathcal{R}}$ from the original network can be written as a matrix perturbation to the adjacency matrix $\mathbf{A}$ of the original Twitter follower network, which takes the form

$$\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}}) = \mathbf{A} - \sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} \mathbf{e}_i \mathbf{e}_j^T, \tag{G.5}$$

where $\mathbf{e}_i$ denotes the n-dimensional column vector of zeros except that its i-th entry is 1.

Left and right multiplying the left leading eigenvector $\mathbf{y}$ of $\mathbf{A}$ to the matrix perturbation equation, we obtain

$$\begin{aligned}\mathbf{y}^T \left(\mathbf{A} - \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} \mathbf{e}_i \mathbf{e}_j^T \right) \mathbf{y} &= \lambda_{\max}(\mathbf{A}) - \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i [\mathbf{y}]_j \\ &= \mathbf{y}^T \widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R}) \mathbf{y} \\ &\leq \lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R})), \qquad \text{(G.6)}\end{aligned}$$

where the inequality is from the Courant-Fischer theorem [76] that $\lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R})) = \max_{\mathbf{x}:\|\mathbf{x}\|_2=1} \mathbf{x}^T \widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R})\mathbf{x}$. Therefore, we obtain the lower bound on $\lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R}))$ as in (8.7).

To obtain an upper bound on $\lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R}))$ in terms of $\lambda_{\max}(\mathbf{A})$ and $\sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i [\mathbf{y}]_j$, we first note that $([\mathbf{y}]_i - [\mathbf{y}]_j)^2 = [\mathbf{y}]_i^2 + [\mathbf{y}]_j^2 - 2[\mathbf{y}]_i[\mathbf{y}]_j \geq 0$ for any i and j. Summing this inequality over the set $\mathcal{E}_\mathcal{R}$ gives

$$\begin{aligned}0 &\leq \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i^2 + \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_j^2 - 2 \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i [\mathbf{y}]_j \\ &\overset{(a)}{\leq} 2q - 2 \sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i [\mathbf{y}]_j, \qquad \text{(G.7)}\end{aligned}$$

where (a) is due to the fact that $\mathbf{y}$ has unit Euclidean norm such that $\sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i^2 \leq |\mathcal{E}_\mathcal{R}| \cdot \max_i [\mathbf{y}]_i^2 \leq |\mathcal{E}_\mathcal{R}| \cdot 1 = q$. Therefore, we obtain the inequality

$$\sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i [\mathbf{y}]_j \leq q. \qquad \text{(G.8)}$$

Lastly, assume $\sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\mathbf{y}]_i [\mathbf{y}]_j > 0$ and let $\widetilde{\mathbf{y}}$ denote the left leading eigenvector of $\widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R})$ such that $\sum_{(i,j)\in\mathcal{E}_\mathcal{R}} [\widetilde{\mathbf{y}}]_i [\widetilde{\mathbf{y}}]_j = \epsilon$. Left and right multiplying $\widetilde{\mathbf{y}}$ of $\widetilde{\mathbf{A}}(\mathcal{E}_\mathcal{R})$ to the matrix perturbation equation gives

$$\begin{aligned}\lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}})) &\leq \lambda_{\max}(\mathbf{A}) - \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} [\widetilde{\mathbf{y}}]_i[\widetilde{\mathbf{y}}]_j \\ &= \lambda_{\max}(\mathbf{A}) - \frac{\sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} [\widetilde{\mathbf{y}}]_i[\widetilde{\mathbf{y}}]_j}{\sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} [\mathbf{y}]_i[\mathbf{y}]_j} \cdot \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} [\mathbf{y}]_i[\mathbf{y}]_j \\ &\leq \lambda_{\max}(\mathbf{A}) - \frac{\epsilon}{q} \cdot \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} [\mathbf{y}]_i[\mathbf{y}]_j, \qquad \text{(G.9)}\end{aligned}$$

which leads to the upper bound on $\lambda_{\max}(\widetilde{\mathbf{A}}(\mathcal{E}_{\mathcal{R}}))$ as in (8.8).

G.5 Implementation of follower link score functions

We consider the score function of a follower link (i, j) that takes the form

$$\text{score}(i, j) = [\mathbf{x}]_i \cdot [\widetilde{\mathbf{x}}]_j, \qquad \text{(G.10)}$$

where $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ are nonnegative n-dimensional vectors.

The following reports on the implementation and computation time complexity of returning q follower links of the highest score for different follower link score functions.

- **LES:** $\mathbf{x} = \widetilde{\mathbf{x}} = \mathbf{y}$, where $\mathbf{y}$ is the left leading eigenvector of the adjacency matrix $\mathbf{A}$. The computation time complexity is $O(mq)$, which is analyzed in Sec. 8.2.3.

- **InDeg:** $\mathbf{x} = \widetilde{\mathbf{x}} = \mathbf{d}_{\text{in}}$, where $\mathbf{d}_{\text{in}}$ is the vector of in-degree of each user, and its j-th element $[\mathbf{d}_{\text{in}}]_j = \sum_{i=1}^{n} [\mathbf{A}]_{ij}$ is the number of followers of user j. The computation time complexity is $O(mq)$.

- **NetMelt:** NetMelt [157] is an edge removal algorithm proposed to decrease the largest eigenvalue $\lambda_{\max}(\mathbf{A})$ of the adjacency matrix $\mathbf{A}$, where $\mathbf{x} = \mathbf{y}$ and $\widetilde{\mathbf{x}} = \mathbf{z}$, and $\mathbf{z}$ denotes the right leading eigenvector of $\mathbf{A}$. The computation time complexity is $O(mq + n)$.

- **NoN-LES-Bet (NoN-LES-Wit):** NoN-LES-Bet (NoN-LES-Wit) exploits the NoN structure and evaluates the score function using $\mathbf{x} = \widetilde{\mathbf{x}} = \mathbf{y}^{\text{bet}}$ ($\mathbf{x} = \widetilde{\mathbf{x}} = \mathbf{y}^{\text{wit}}$), where $\mathbf{y}^{\text{bet}}$ ($\mathbf{y}^{\text{bit}}$) denotes the left leading eigenvector of the between-network (within-network) adjacency matrix $\mathbf{A}^{\text{bet}}$ ($\mathbf{A}^{\text{wit}}$). The computation time complexity is $O(mq)$.

- **NoN-InDeg-Bet (NoN-InDeg-Wit):** NoN-InDeg-Bet and NoN-InDeg-Wit are extensions of the InDeg score tailored to the NoN structure. Specifically, for NoN-InDeg-Bet (NoN-InDeg-Wit) we set $\mathbf{x} = \widetilde{\mathbf{x}} = \mathbf{d}_{\text{in}}^{\text{bet}}$ ($\mathbf{x} = \widetilde{\mathbf{x}} = \mathbf{d}_{\text{in}}^{\text{wit}}$), where $\mathbf{d}_{\text{in}}^{\text{bet}}$ ($\mathbf{d}_{\text{in}}^{\text{wit}}$) is the in-degree vector that only accounts for the between-network (within-network) follower links in the Twitter follower network. The computation time complexity is $O(mq)$.

- **NoN-NetMelt-Bet (NoN-NetMelt-Wit):** Non-NetMelt-Bet and NoN-NetMelt-Wit are NetMelt algorithms that incorporate the NoN structure. For NoN-Melt-Bet (NoN-NetMelt-Wit), we set $\mathbf{x} = \mathbf{y}^{\text{bet}}$ and $\widetilde{\mathbf{x}} = \mathbf{z}^{\text{bet}}$ ($\mathbf{x} = \mathbf{y}^{\text{wit}}$ and $\widetilde{\mathbf{x}} = \mathbf{z}^{\text{wit}}$), where $\mathbf{y}^{\text{bet}}$ and $\mathbf{z}^{\text{bet}}$ ($\mathbf{y}^{\text{wit}}$ and $\mathbf{z}^{\text{wit}}$) denote the left and right leading eigenvectors of $\mathbf{A}^{\text{bet}}$ ($\mathbf{A}^{\text{wit}}$). The computation time complexity is $O(mq + n)$.

We also implemented score functions based on the right leading eigenvector of the adjacency matrix. However, its effect on reducing event propagation is not prominent, so we omit the results in the chapter.

APPENDIX H

Appendix of Chapter X

H.1 Kronecker Product

If $\mathbf{X}_1$ is a $r_1 \times \ell_1$ matrix and $\mathbf{X}_2$ is a $r_2 \times \ell_2$ matrix, then the Kronecker product $\mathbf{X}_1 \otimes \mathbf{X}_2$ is a a $r_1 r_2 \times \ell_1 \ell_2$ matrix is defined as

$$\mathbf{X}_1 \otimes \mathbf{X}_2 = \begin{bmatrix} [\mathbf{X}]_{11}\mathbf{X}_2 & [\mathbf{X}]_{12}\mathbf{X}_2 & \dots & [\mathbf{X}]_{1\ell_1}\mathbf{X}_2 \\ [\mathbf{X}]_{21}\mathbf{X}_2 & [\mathbf{X}]_{22}\mathbf{X}_2 & \dots & [\mathbf{X}]_{2\ell_1}\mathbf{X}_2 \\ \vdots & \vdots & \vdots & \vdots \\ [\mathbf{X}]_{r_1 1}\mathbf{X}_2 & [\mathbf{X}]_{r_1 2}\mathbf{X}_2 & \dots & [\mathbf{X}]_{r_1 \ell_1}\mathbf{X}_2 \end{bmatrix}. \tag{H.1}$$

Some useful properties of Kronecker product are

$$(\mathbf{X}_1 \otimes \mathbf{X}_2)^T = \mathbf{X}_1^T \otimes \mathbf{X}_2^T; \tag{H.2}$$

$$\mathbf{X}_1 \otimes (\mathbf{X}_2 + \mathbf{X}_3) = \mathbf{X}_1 \otimes \mathbf{X}_2 + \mathbf{X}_1 \otimes \mathbf{X}_3. \tag{H.3}$$

If $\mathbf{X}_1$ is a $r_1 \times \ell_1$ matrix, $\mathbf{X}_2$ is a $r_2 \times \ell_2$ matrix, $\mathbf{X}_3$ is an $\ell_1 \times \ell_3$ matrix and $\mathbf{X}_4$ is an $\ell_2 \times \ell_4$ matrix, then

$$(\mathbf{X}_1 \otimes \mathbf{X}_2) \cdot (\mathbf{X}_3 \otimes \mathbf{X}_4) = (\mathbf{X}_1 \cdot \mathbf{X}_3) \otimes (\mathbf{X}_2 \cdot \mathbf{X}_4). \tag{H.4}$$

H.2 Proof of (10.3)

Following (10.2),

$$\begin{aligned} \mathbf{r}_{t+1} &= \mathbb{T}\left(\sum_{h=1}^{t+1} \mathbf{w}_h\right) \\ &= \mathbb{T}\left(\sum_{h=1}^{t} \mathbf{w}_h + \mathbf{w}_{t+1}\right) \\ &\equiv \mathbb{T}\left(\mathbb{T}\left(\sum_{h=1}^{t} \mathbf{w}_h\right) + \mathbf{w}_{t+1}\right) \\ &\equiv \mathbb{T}\left(\mathbf{r}_t + \mathbf{B}\mathbf{r}_t\right). \end{aligned} \tag{H.5}$$

H.3 Proof of (10.4)

Following the definition of $\mathbf{w}_1$ we have

$$\begin{aligned}
[\mathbf{w}_1]_j &= \sum_{i=1}^{N} [\mathbf{r}_0]_i [\mathbf{W}]_{ij} \\
&= \sum_{i=1}^{N} [\mathbf{r}_0]_i \sum_{k=1}^{K} [\mathbf{A}_k]_{ij} [\mathbf{P}]_{kj} \\
&= \sum_{k=1}^{K} \sum_{i=1}^{N} [\mathbf{r}_0]_i [\mathbf{A}_k]_{ij} [\mathbf{P}]_{kj} \\
&= \sum_{k=1}^{K} \mathbf{r}_0^T \mathbf{A}_k \mathbf{e}_j^N [\mathbf{P}]_{kj} \\
&= \mathbf{r}_0^T \sum_{k=1}^{K} \left([\mathbf{P}]_{kj} \mathbf{A}_k \right) \mathbf{e}_j^N . \qquad \text{(H.6)}
\end{aligned}$$

Since $\sum_{k=1}^{K} \mathbf{P}_{kj} \mathbf{A}_k = \mathbf{A} \cdot \left[\mathrm{col}_j(\mathbf{P}) \otimes \mathbf{I}_n \right]$, applying it to (H.6) we have

$$\begin{aligned}
[\mathbf{w}_1]_j &= \mathbf{r}_0^T \mathbf{A} \left[\mathrm{col}_j(\mathbf{P}) \otimes \mathbf{I}_n \right] \mathbf{e}_j \qquad \text{(H.7)} \\
&= \mathbf{e}_j^T \left[\mathrm{col}_j(\mathbf{P})^T \otimes \mathbf{I}_n \right] \mathbf{A}^T \mathbf{r}_0 .
\end{aligned}$$

H.4 Proof of (10.5)

Using (H.2) and (H.4) gives

$$\begin{aligned}
\mathbf{w}_1 &= \begin{bmatrix} \mathbf{e}_1^T & \mathbf{0}_N^T & \dots & \mathbf{0}_N^T \\ \mathbf{0}_N^T & \mathbf{e}_2^T & \mathbf{0}_N^T & \vdots \\ \vdots & \vdots & \vdots & \mathbf{0}_N^T \\ \mathbf{0}_N^T & \dots & \mathbf{0}_N^T & \mathbf{e}_N^T \end{bmatrix} \cdot \begin{bmatrix} \mathrm{col}_1(\mathbf{P})^T \otimes \mathbf{I}_n \\ \mathrm{col}_2(\mathbf{P})^T \otimes \mathbf{I}_n \\ \vdots \\ \mathrm{col}_n(\mathbf{P})^T \otimes \mathbf{I}_n \end{bmatrix} \cdot \mathbf{A}^T \mathbf{r}_0 \\
&= \left(\mathbf{I}_n \otimes \mathbf{1}_N^T\right) \cdot \left(\mathbf{P}^T \otimes \mathbf{I}_n\right) \cdot \mathbf{A}^T \mathbf{r}_0 \\
&= \left(\mathbf{I}_n \cdot \mathbf{P}^T\right) \otimes \left(\mathbf{1}_N^T \cdot \mathbf{I}_n\right) \mathbf{A}^T \mathbf{r}_0 \\
&= \left(\mathbf{P}^T \otimes \mathbf{1}_N^T\right) \mathbf{A}^T \mathbf{r}_0 \\
&= \left(\mathbf{P} \otimes \mathbf{1}_N\right)^T \mathbf{A}^T \mathbf{r}_0.
\end{aligned} \tag{H.8}$$

H.5 Proof of (10.8)

Following (10.7),

$$\begin{aligned}
\mathbf{r}_{t+1} &= \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\sum_{h=1}^{t+1} \mathbf{w}_h \right) \right) \\
&= \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\sum_{h=1}^{t} \mathbf{w}_h + \mathbf{w}_{t+1} \right) \right) \\
&\equiv \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\mathbb{T} \left(\sum_{h=1}^{t} \mathbf{w}_h \right) + \mathbf{w}_{t+1} \right) \right) \\
&\equiv \mathbb{H}_{\mathbf{a}} \left(\mathbb{T} \left(\mathbf{r}_t + \left(\mathbf{P} \otimes \mathbf{1}_N\right)^T \mathbf{A}^T \mathbf{r}_t \right) \right).
\end{aligned} \tag{H.9}$$

H.6 Proof of (10.11)

When a subset of edges $\mathcal{E}_{\mathcal{R}} \subset \mathcal{E}$ are removed from G_C, the resulting adjacency matrix of $G_C \setminus \mathcal{E}_{\mathcal{R}}$ is

$$\widetilde{\mathbf{A}}_C\left(\mathcal{E}_{\mathcal{R}}\right) = \mathbf{A}_C - \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \mathbf{e}_i^U \mathbf{e}_j^{N^T}. \tag{H.10}$$

Therefore, the corresponding induced adjacency matrix $\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)$ is

$$\begin{aligned}\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right) &= \mathbf{B} - \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \mathbf{e}_j^N \mathbf{e}_i^{U^T} \mathbf{A}_C - \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \mathbf{A}_C^T \mathbf{e}_i^U \mathbf{e}_j^{N^T} + \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \sum_{(\ell,s)\in\mathcal{E}_{\mathcal{R}}} \mathbf{e}_j^N \mathbf{e}_i^{U^T} \mathbf{e}_\ell^U \mathbf{e}_s^{N^T} \\ &= \mathbf{B} - \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \mathbf{e}_j^N \mathbf{e}_i^{U^T} \mathbf{A}_C - \sum_{(i,j)\in\mathcal{E}_{\mathcal{R}}} \mathbf{A}_C^T \mathbf{e}_i^U \mathbf{e}_j^{N^T} \\ &\quad + \sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\mathcal{E}_{\mathcal{R}}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}_{\mathcal{R}}} \mathbf{e}_j^N \mathbf{e}_s^{N^T}. \end{aligned} \tag{H.11}$$

Recall that $\mathbf{u}$ is the largest eigenvector of $\mathbf{B}$. Left and right multiplying (H.11) by $\mathbf{u}^T$ and $\mathbf{u}$ and using the Courant-Fischer theorem [76] we have

$$\lambda_{\max}\left(\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)\right) \geq \lambda_{\max}(\mathbf{B}) - f(\mathcal{E}_{\mathcal{R}}), \tag{H.12}$$

where $f(\mathcal{E}_{\mathcal{R}})$ is defined in (10.12).

H.7 An equivalent expression of $f(\mathcal{E}_{\mathcal{R}})$

The following lemma provides an equivalent representation of the function $f(\mathcal{E}_{\mathcal{R}})$ in (10.11), which also implies that $f(\mathcal{E}_{\mathcal{R}})$ is nonnegative as it can be represented by a sum of nonnegative terms.

Lemma. *Let $\varnothing$ denote the empty set. Then $f(\varnothing) = 0$ and*

$$f(\mathcal{E}_{\mathcal{R}}) = \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s + 2 \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E} / \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s.$$

Proof. $f(\varnothing) = 0$ is a direct result from the definition of $f(\mathcal{E}_{\mathcal{R}})$. $f(\mathcal{E}_{\mathcal{R}})$ has an equivalent expression that

$$\begin{aligned}
f(\mathcal{E}_{\mathcal{R}}) &= 2 \sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} \mathbf{u}^T \mathbf{A}_C^T \mathbf{e}_i^U [\mathbf{u}]_j - \sum_{i \in \mathcal{V}_{user}} \sum_{\in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s \\
&= 2 \sum_{(i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}_{host}} [\mathbf{A}_C]_{is} [\mathbf{u}]_s [\mathbf{u}]_j - \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s \\
&= 2 \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \left(\sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_{\mathcal{R}}} + \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E} / \mathcal{E}_{\mathcal{R}}} \right) [\mathbf{u}]_j [\mathbf{u}]_s \\
&\quad - \sum_{i \in \mathcal{V}} \sum_{\in \mathcal{V}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}, (i,s) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s \\
&= \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}, (i,s) \in \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s + 2 \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}, (i,j) \in \mathcal{E}_{\mathcal{R}}} \sum_{s \in \mathcal{V}, (i,s) \in \mathcal{E} / \mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j [\mathbf{u}]_s.
\end{aligned} \tag{H.13}$$

The nonnegativity of $\mathbf{u}$ suggests that $f(\mathcal{E}_{\mathcal{R}}) \geq 0$. □

H.8 Proof of Lemma 10.1

For any edge removal set $\mathcal{E}_{\mathcal{R}} \subset \mathcal{E}$ with $|\mathcal{E}_{\mathcal{R}}| = q$, let $\mathbf{v}$ be the largest eigenvector of $\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}})$. Left and right multiplying (H.11) by $\mathbf{v}^T$ and $\mathbf{v}$ gives

$$\begin{aligned}
\lambda_{\max}\left(\widetilde{\mathbf{B}}(\mathcal{E}_{\mathcal{R}})\right) &= \mathbf{v}^T \mathbf{B} \mathbf{v} - g(\mathcal{E}_{\mathcal{R}}) \\
&\leq \lambda_{\max}(\mathbf{B}) - g(\mathcal{E}_{\mathcal{R}})
\end{aligned} \tag{H.14}$$

by the Courant-Fischer theorem [76], where $g(\mathcal{E}_\mathcal{R}) = \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_\mathcal{R}} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_\mathcal{R}}$ $[\mathbf{v}]_j [\mathbf{v}]_s + 2 \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_\mathcal{R}} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}/\mathcal{E}_\mathcal{R}} [\mathbf{v}]_j [\mathbf{v}]_s$ is obtained by following the same derivation procedure as in Lemma H.7.

Next, recall from the Perron-Frobenius theorem [76] that the entries of $\mathbf{u}$ and $\mathbf{v}$ are all nonnegative and bounded. Therefore, there must exist one edge removal set $\mathcal{E}_\mathcal{R}$ with $|\mathcal{E}_\mathcal{R}| = q$ such that $g(\mathcal{E}_\mathcal{R}) > 0$. Otherwise $g(\mathcal{E}_\mathcal{R}) = 0$ for every edge removal set with cardinality $q \geq 1$ implies that $\mathbf{v}$ is a zero vector, which contradicts the fact that $\mathbf{v}$ is an eigenvector. Finally, since $f(\mathcal{E}_\mathcal{R}) > 0$, there exists a constant $c > 0$ such that $g(\mathcal{E}_\mathcal{R}) \geq c \cdot f(\mathcal{E}_\mathcal{R})$. Applying this inequality to (H.14) gives $\lambda_{\max}\left(\widetilde{\mathbf{B}}\left(\mathcal{E}_\mathcal{R}\right)\right) \leq \lambda_{\max}(\mathbf{B}) - c \cdot f(\mathcal{E}_\mathcal{R})$.

H.9 Monotonicity of $f(\mathcal{E}_\mathcal{R})$

Lemma. *$f(\mathcal{E}_\mathcal{R})$ is a monotonic increasing set function.*

Proof. For any two subsets $\mathcal{E}_{\mathcal{R}1}, \mathcal{E}_{\mathcal{R}2} \subset \mathcal{E}$ satisfying $\mathcal{E}_{\mathcal{R}1} \subset \mathcal{E}_{\mathcal{R}2}$, let $\Delta\mathcal{E}_\mathcal{R} = \mathcal{E}_{\mathcal{R}2}/\mathcal{E}_{\mathcal{R}1}$. Using the relation $\mathcal{E}_{\mathcal{R}2} = \mathcal{E}_{\mathcal{R}1} \cup \Delta\mathcal{E}_\mathcal{R}$ and $\mathcal{E}_{\mathcal{R}1} \cap \Delta\mathcal{E}_\mathcal{R} = \varnothing$, from Lemma H.7 $f(\mathcal{E}_{\mathcal{R}2})$ can be represented as

$$\begin{aligned} f(\mathcal{E}_{\mathcal{R}2}) = \sum_{i \in \mathcal{V}_{user}} &\left(\sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}1}} + \sum_{j \in \mathcal{V}_{host}, (i,j) \in \Delta\mathcal{E}_\mathcal{R}} \right) \left(\sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_{\mathcal{R}1}} + \sum_{s \in \mathcal{V}_{host}, (i,s) \in \Delta\mathcal{E}_\mathcal{R}} \right) [\mathbf{u}]_j [\mathbf{u}]_s \\ + 2 \sum_{i \in \mathcal{V}_{user}} &\left(\sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}1}} + \sum_{j \in \mathcal{V}_{host}, (i,j) \in \Delta\mathcal{E}_\mathcal{R}} \right) \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E} \setminus \mathcal{E}_\mathcal{R}} [\mathbf{u}]_j [\mathbf{u}]_s. \quad \text{(H.15)} \end{aligned}$$

Similarly, using the relation $\Delta\mathcal{E}_{\mathcal{R}} = (\mathcal{E} \setminus \mathcal{E}_{\mathcal{R}1}) \setminus (\mathcal{E} \setminus \mathcal{E}_{\mathcal{R}2})$, from Lemma H.7 we have

$$\begin{aligned} f(\mathcal{E}_{\mathcal{R}1}) = &\sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\mathcal{E}_{\mathcal{R}1}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}_{\mathcal{R}1}} [\mathbf{u}]_j[\mathbf{u}]_s \\ &+ 2 \sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\mathcal{E}_{\mathcal{R}1}} \left(\sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}\setminus\mathcal{E}_{\mathcal{R}2}} + \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}\setminus\Delta\mathcal{E}_{\mathcal{R}}} \right) [\mathbf{u}]_j[\mathbf{u}]_s. \end{aligned} \tag{H.16}$$

Therefore,

$$\begin{align} f(\mathcal{E}_{\mathcal{R}2}) - f(\mathcal{E}_{\mathcal{R}1}) = &\sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\Delta\mathcal{E}_{\mathcal{R}}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}_{\mathcal{R}2}} [\mathbf{u}]_j[\mathbf{u}]_s \notag \\ &+ 2 \sum_{i_{user}\in\mathcal{V}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\Delta\mathcal{E}_{\mathcal{R}}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}\setminus\mathcal{E}_{\mathcal{R}2}} [\mathbf{u}]_j[\mathbf{u}]_s \notag \\ &- \sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\mathcal{E}_{\mathcal{R}1}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\Delta\mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j[\mathbf{u}]_s \notag \\ \geq &\sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\Delta\mathcal{E}_{\mathcal{R}}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}_{\mathcal{R}1}} [\mathbf{u}]_j[\mathbf{u}]_s \tag{H.17} \\ &+ 2 \sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\Delta\mathcal{E}_{\mathcal{R}}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}\setminus\mathcal{E}_{\mathcal{R}2}} [\mathbf{u}]_j[\mathbf{u}]_s \notag \\ &- \sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\mathcal{E}_{\mathcal{R}1}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\Delta\mathcal{E}_{\mathcal{R}}} [\mathbf{u}]_j[\mathbf{u}]_s \notag \\ = &\,2 \sum_{i\in\mathcal{V}_{user}} \sum_{j\in\mathcal{V}_{host},(i,j)\in\Delta\mathcal{E}_{\mathcal{R}}} \sum_{s\in\mathcal{V}_{host},(i,s)\in\mathcal{E}\setminus\mathcal{E}_{\mathcal{R}2}} [\mathbf{u}]_j[\mathbf{u}]_s \tag{H.18} \\ \geq &\, 0, \tag{H.19} \end{align}$$

where the inequality in (H.17) uses the Perron-Frobenious theorem [76] that $[\mathbf{u}]_s \geq 0$ and the fact that $\mathcal{E}_{\mathcal{R}1} \subset \mathcal{E}_{\mathcal{R}2}$. The inequality in (H.19) is due to the nonnegativity of the largest eigenvector $\mathbf{u}$. □

H.10 Proof of Theorem 10.2

It has been proved in Lemma H.9 that $f(\mathcal{E}_{\mathcal{R}})$ is a monotone increasing set function. Here we prove that $f(\mathcal{E}_{\mathcal{R}})$ is submodular. For any $\mathcal{E}_{\mathcal{R}1} \subset \mathcal{E}_{\mathcal{R}2} \subset \mathcal{E}$ and $e \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{R}2}$, let $e = (u, v) \in \mathcal{E}$, from (H.17) we have

$$\begin{aligned}
\Delta f(e|\mathcal{E}_{\mathcal{R}2}) &= f(\mathcal{E}_{\mathcal{R}2} \cup e) - f(\mathcal{E}_{\mathcal{R}2}) \\
&= \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{user}, (i,j)=e} \left(\sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E}_{\mathcal{R}2}} + \sum_{s \in \mathcal{V}_{host}, (i,s)=e} \right) [\mathbf{u}]_j [\mathbf{u}]_s \\
&\quad + 2 \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j)=e} \sum_{s \in \mathcal{V}_{host}, (i,s) \in \mathcal{E} \setminus (\mathcal{E}_{\mathcal{R}2} \cup e)} [\mathbf{u}]_j [\mathbf{u}]_s \\
&\quad - \sum_{i \in \mathcal{V}_{user}} \sum_{j \in \mathcal{V}_{host}, (i,j) \in \mathcal{E}_{\mathcal{R}2}} \sum_{s \in \mathcal{V}_{host}, (i,s)=e} [\mathbf{u}]_j [\mathbf{u}]_s \\
&= [\mathbf{u}]_u [\mathbf{u}]_v + 2 \sum_{s \in \mathcal{V}_{host}, (u,s) \in \mathcal{E} \setminus (\mathcal{E}_{\mathcal{R}2} \cup e)} [\mathbf{u}]_u [\mathbf{u}]_s \\
&\leq [\mathbf{u}]_u [\mathbf{u}]_v + 2 \sum_{s \in \mathcal{V}_{host}, (u,s) \in \mathcal{E} \setminus (\mathcal{E}_{\mathcal{R}1} \cup e)} [\mathbf{u}]_u [\mathbf{u}]_s && \text{(H.20)} \\
&= \Delta f(e|\mathcal{E}_{\mathcal{R}1}), && \text{(H.21)}
\end{aligned}$$

where the inequality in (H.21) holds due to the fact that $\mathcal{E} \setminus (\mathcal{E}_{\mathcal{R}2} \cup e) \subset \mathcal{E} \setminus (\mathcal{E}_{\mathcal{R}1} \cup e)$ and the entries of $\mathbf{u}$ are nonnegative from the Perron-Frobenious theorem [76]. Therefore, $f(\mathcal{E}_{\mathcal{R}})$ is a monotone submodular set function.

H.11 Proof of Theorem 10.3

Let $\mathcal{E}_{\mathcal{R}}^s$ with $|\mathcal{E}_{\mathcal{R}}^s| = s$ be the greedy edge removal set obtained from Algorithm 10.4. By submodularity of $f(\mathcal{E}_{\mathcal{R}})$ from Theorem 10.2, for every $s < q$ there exists an edge $e \in \mathcal{E}_{\mathcal{R}}^{opt} / \mathcal{E}_{\mathcal{R}}^s$ such that

$$f(\mathcal{E}_{\mathcal{R}}^s \cup e) - f(\mathcal{E}_{\mathcal{R}}^s) \geq \frac{1}{q} \left(f(\mathcal{E}_{\mathcal{R}}^{opt}) - f(\mathcal{E}_{\mathcal{R}}^s) \right). \tag{H.22}$$

After algebraic manipulation, we have

$$f(\mathcal{E}_{\mathcal{R}}^{opt}) - f(\mathcal{E}_{\mathcal{R}}^{s+1}) \leq \left(1 - \frac{1}{q}\right)\left(f(\mathcal{E}_{\mathcal{R}}^{opt}) - f(\mathcal{E}_{\mathcal{R}}^{s})\right) \tag{H.23}$$

and therefore by telescoping (H.23) we have

$$f(\mathcal{E}_{\mathcal{R}}^{opt}) - f(\mathcal{E}_{\mathcal{R}}^{q}) \leq \left(1 - \frac{1}{q}\right)^{q} f(\mathcal{E}_{\mathcal{R}}^{opt}) \leq \frac{1}{\mathsf{e}} f(\mathcal{E}_{\mathcal{R}}^{opt}). \tag{H.24}$$

Applying (H.24) and the fact that $0 < f(\mathcal{E}_{\mathcal{R}}^{q}) \leq f(\mathcal{E}_{\mathcal{R}}^{opt})$ to (10.11), there exists some constant $c > 0$ such that

$$\lambda_{\max}(\mathbf{B}) - c\left(1 - \mathsf{e}^{-1}\right) \cdot f(\mathcal{E}_{\mathcal{R}}^{opt}) \geq \lambda_{\max}\left(\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}^{q}\right)\right) \geq \lambda_{\max}(\mathbf{B}) - f(\mathcal{E}_{\mathcal{R}}^{opt}). \tag{H.25}$$

The proof is complete by setting $c' = c\left(1 - \mathsf{e}^{-1}\right)$.

H.12 Proof of Corollary 10.4

This corollary is a direct result of Lemma 10.1 and Theorem 10.3 by replacing $\mathbf{B}$ with $\widetilde{\mathbf{B}}\left(\mathcal{E}_{\mathcal{R}}\right)$ and setting $q = 1$.

H.13 Proof of Theorem 10.5

We use the fact from the Perron-Frobenius theorem that if a square matrix $\mathbf{X}$ is irreducible and nonnegative, then $\lambda_{\max}(\mathbf{X}) \leq \max_s \sum_t [\mathbf{X}]_{st}$. A square nonnegative matrix $\mathbf{X}$ is irreducible means that for every pair of indices s and t, there exists a natural number z such that $[\mathbf{X}^z]_{st} > 0$. Since $\widetilde{\mathbf{B}}(i,j)$ is a matrix of nonnegative entries, if $\widetilde{\mathbf{B}}(i,j)$ is irreducible, from (10.9) we have

$$\begin{aligned}\lambda_{\max}\left(\widetilde{\mathbf{B}}(i,j)\right) &\leq \max_{s\in\{1,2,\ldots,N\}}\left[\widetilde{\mathbf{B}}(i,j)\mathbf{1}_N\right]_s\\ &= \max_{s\in\{1,2,\ldots,N\}}\left[\mathbf{B}\mathbf{1}_N-\mathbf{A}_C^T\mathbf{e}_i^U-[\mathbf{d}^U]_i\mathbf{e}_j^N+\mathbf{e}_j^N\right]_s\\ &\leq d_{\max}^{user}\cdot d_{\max}^{host}-\max_{s\in\{1,2,\ldots,N\}}\left[\left([\mathbf{d}^U]_i-1\right)\mathbf{e}_j^N-\mathbf{A}_C^T\mathbf{e}_i^U\right]_s, \qquad \text{(H.26)}\end{aligned}$$

where (H.26) uses the fact that for all $t\in\{1,2,\ldots,N\}$,

$$[\mathbf{B}\mathbf{1}_N]_t=[\mathbf{A}_C^T\mathbf{A}_C\mathbf{1}_N]_t=[\mathbf{A}_C^T\mathbf{d}_U]_t\leq d_{\max}^{user}\cdot d_{\max}^{host}. \qquad \text{(H.27)}$$

Remark 8.1. If $\widetilde{\mathbf{B}}(i,j)$ is reducible, one can obtain a similar upper bound as in Theorem 10.5 since the largest eigenvalue of $\widetilde{\mathbf{B}}(i,j)$ is the maximum value of the largest eigenvalue of block-wise irreducible nonnegative submatrices of $\widetilde{\mathbf{B}}(i,j)$.

H.14 Proof of (10.14)

By the Courant-Fischer theorem [76], (H.2) and (H.3) we have

$$\begin{aligned}\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right) &\geq \mathbf{y}^T\widetilde{\mathbf{J}}(\mathcal{H})\mathbf{y}\\ &= \mathbf{y}^T(\widetilde{\mathbf{P}}_{\mathcal{H}}\otimes\mathbf{1}_N)^T\mathbf{A}^T\mathbf{y}\\ &= \lambda_{\max}\left(\mathbf{J}\right)-\mathbf{y}^T\Delta\mathbf{J}_{\mathcal{H}}\mathbf{y}, \qquad \text{(H.28)}\end{aligned}$$

where

$$\Delta\mathbf{J}_{\mathcal{H}}=\left[\left(\sum_{(k,j)\in\mathcal{H}}\left([\mathbf{P}]_{kj}-\epsilon_{kj}\right)\mathbf{e}_k^K\mathbf{e}_j^{N^T}\right)\otimes\mathbf{1}_N\right]^T\mathbf{A}^T. \qquad \text{(H.29)}$$

H.15 Monotonicity of $\phi(\mathcal{H})$

Lemma 8.2. *$\phi(\varnothing) = 0$ and $\phi(\mathcal{H})$ is a monotonic increasing set function.*

Proof. By definition $\phi(\varnothing) = 0$ since $\Delta \mathbf{J}_{\varnothing}$ is a zero matrix. For any two sets $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{V}_{app} \times \mathcal{V}_{mac}$,

$$\begin{aligned} \phi(\mathcal{H}_2) - \phi(\mathcal{H}_1) &= \mathbf{y}^T \left(\Delta \mathbf{J}_{\mathcal{H}_2} - \Delta \mathbf{J}_{\mathcal{H}_1}\right) \mathbf{y} \\ &= \mathbf{y}^T \left(\Delta \mathbf{J}_{\mathcal{H}_2 \setminus \mathcal{H}_1}\right) \mathbf{y} \\ &\geq 0 \end{aligned} \tag{H.30}$$

since $\Delta \mathbf{J}_{\mathcal{H}_2 \setminus \mathcal{H}_1}$ is a nonnegative matrix and $\mathbf{y}$ is a nonnegative vector by the Perron-Frobenious theorem [76]. Therefore, $\phi(\mathcal{H})$ is a monotonic increasing set function. □

H.16 Efficient update of step 5 in Algorithm 10.6 with score recalculation

Using the notations in Algorithm 10.6, when hardening the edge (k^*, j^*) the entry $[\mathbf{P}^\eta]_{k^* j^*}$ changes to $\epsilon_{k^* j^*}$. Let the original value of $[\mathbf{P}^\eta]_{k^* j^*}$ before hardening be ψ. Then the update in step 5 is equivalent to

$$\mathbf{J}^{old} = \mathbf{J}^{old} - \mathbf{H}^T \mathbf{A}^T, \tag{H.31}$$

where $\mathbf{H} = [\mathbf{0}, \ldots, \mathbf{h}, \ldots, \mathbf{0}]$ is a matrix of zeros except that the $[(k^* - 1) \cdot N + 1]$-th to $(k^* \cdot N)$-th entry of $\mathbf{H}$'s j^*-th column $\mathbf{h}$ is $\psi - \epsilon_{k^* j^*}$.

H.17 Proof of Theorem 10.6

For any two hardening sets $\mathcal{H}_1$ and $\mathcal{H}_2$ satisfying $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{V}_{app} \times \mathcal{V}_{host}$, using (H.29) and (H.30) we have the additivity for the score function $\phi(\mathcal{H})$ as

$$\phi(\mathcal{H}_2) = \phi(\mathcal{H}_1) + \phi(\mathcal{H}_2 \setminus \mathcal{H}_1). \tag{H.32}$$

For any hardening set $\mathcal{H}$ of cardinality $|\mathcal{H}| = \eta \geq 1$, let $\mathcal{H} = \{H_s\}_{s=1}^{\eta}$, where H_s is the s-th element in $\mathcal{H}$, and let $\mathcal{H}^{\eta} = \{H_s^{\eta}\}_{s=1}^{\eta}$. Then with (H.32) we have

$$\begin{aligned} \phi(\mathcal{H}) &= \sum_{s=1}^{\eta} \phi(H_s) \\ &\leq \sum_{s=1}^{\eta} \phi(H_s^{\eta}) \\ &= \phi(\mathcal{H}^{\eta}), \end{aligned} \tag{H.33}$$

where the maximum of $\phi(\mathcal{H})$ is attained when $\mathcal{H}$ contains η edges of highest hardening scores. Therefore, $\mathcal{H}^{\eta}$ is a maximizer of $\phi(\mathcal{H})$.

H.18 Proof of Corollary 10.8

This corollary is a direct result of Theorem 10.7 by replacing $\mathbf{J}$ with $\widetilde{\mathbf{J}}(\mathcal{H})$ and setting $\eta = 1$.

H.19 Proof of Theorem 10.7

We first show the relation that $\lambda_{\max}(\mathbf{J}) \geq \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right)$. For any hardening set $\mathcal{H}$, let $\widetilde{\mathbf{y}}$ be the largest eigenvector of $\widetilde{\mathbf{J}}(\mathcal{H})$. With (H.29) we have

$$\begin{aligned}
\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right) &= \widetilde{\mathbf{y}}^T \widetilde{\mathbf{J}}\left(\mathcal{H}\right)\widetilde{\mathbf{y}} \\
&= \widetilde{\mathbf{y}}^T \mathbf{J}\widetilde{\mathbf{y}} - \widetilde{\mathbf{y}}^T \Delta\mathbf{J}_{\mathcal{H}}\widetilde{\mathbf{y}} \\
&\leq \lambda_{\max}\left(\mathbf{J}\right) - \widetilde{\mathbf{y}}^T \Delta\mathbf{J}_{\mathcal{H}}\widetilde{\mathbf{y}} \\
&\leq \lambda_{\max}\left(\mathbf{J}\right).
\end{aligned} \tag{H.34}$$

The fact that $\widetilde{\mathbf{y}}^T \mathbf{J}\widetilde{\mathbf{y}} \leq \lambda_{\max}\left(\mathbf{J}\right)$ is from the Courant-Fischer theorem [76], and the last inequality uses the fact that $\widetilde{\mathbf{y}}^T \Delta\mathbf{J}_{\mathcal{H}}\widetilde{\mathbf{y}} \geq 0$ from the Perron-Frobenious theorem [76] due to the fact that all entries in $\Delta\mathbf{J}_{\mathcal{H}}$ and $\widetilde{\mathbf{y}}$ are nonnegative.

If $\lambda_{\max}(\mathbf{J}) > 0$, then by (H.28) and (H.34) we have $\phi(\mathcal{H}^{opt}) > 0$. Otherwise $\phi(\mathcal{H}^{opt}) = 0$ implies that $\mathbf{y}$ is a zero vector, which contradicts the fact that $\mathbf{y}$ is the largest eigenvector of $\mathbf{J}$. Therefore, if $\lambda_{\max}(\mathbf{J}) > 0$ we have $\lambda_{\max}(\mathbf{J}) > \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{opt})\right)$. When $|\mathcal{H}| = \eta$, since $\mathcal{H}^{opt}$ is the minimizer of $\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H})\right)$ and $\mathcal{H}^{\eta}$ is the maximizer of $\phi(\mathcal{H})$, we have

$$\begin{aligned}
\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{\eta})\right) &\geq \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{opt})\right) \\
&\geq \lambda_{\max}(\mathbf{J}) - \phi(\mathcal{H}^{opt}) \\
&\geq \lambda_{\max}(\mathbf{J}) - \phi(\mathcal{H}^{\eta}).
\end{aligned} \tag{H.35}$$

By the facts that $\lambda_{\max}(\mathbf{J}) > \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{opt})\right)$ and $\lambda_{\max}(\mathbf{J}) \geq \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{\eta})\right)$, if $\phi(\mathcal{H}^{\eta}) >$

0, with (H.35) there exists some constant $c'' > 0$ such that

$$\begin{aligned} &\lambda_{\max}(\mathbf{J}) - c'' \cdot \phi(\mathcal{H}^{\eta}) \geq \lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{opt})\right); \\ &\lambda_{\max}\left(\widetilde{\mathbf{J}}(\mathcal{H}^{opt})\right) \geq \lambda_{\max}(\mathbf{J}) - \phi(\mathcal{H}^{\eta}). \end{aligned} \tag{H.36}$$

BIBLIOGRAPHY

[1] Abbe, E., and C. Sandon (2015), Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms, *arXiv preprint arXiv:1503.00609*.

[2] Abbe, E., A. S. Bandeira, and G. Hall (2014), Exact recovery in the stochastic block model, *arXiv preprint arXiv:1405.3267*.

[3] Adamic, L. A., and N. Glance (2005), The political blogosphere and the 2004 U.S. election: Divided they blog, in *ACM Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD)*, pp. 36–43.

[4] Airoldi, E. M., D. M. Blei, S. E. Fienberg, and E. P. Xing (2008), Mixed membership stochastic blockmodels, *J. Mach. Learn. Res.*, *9*, 1981–2014.

[5] Alamgir, M., and U. von Luxburg (2011), Phase transition in the family of p-resistances, in *Advances in Neural Information Processing Systems (NIPS)*, pp. 379–387.

[6] Albert, R., H. Jeong, and A.-L. Barabási (2000), Error and attack tolerance of complex networks, *Nature*, *406*(6794), 378–382.

[7] Alon, N., M. Krivelevich, and B. Sudakov (1998), Finding a large hidden clique in a random graph, *Random Structures and Algorithms*, *13*(3-4), 457–466.

[8] Anscombe, F. J. (1948), The transformation of poisson, binomial and negative-binomial data, *Biometrika*, *35*(3/4), 246–254.

[9] Balakrishnan, S., M. Xu, A. Krishnamurthy, and A. Singh (2011), Noise thresholds for spectral clustering, pp. 954–962.

[10] Barbillon, P., S. Donnet, E. Lazega, and A. Bar-Hen (2016), Stochastic block models for multiplex networks: an application to a multilevel network of researchers, *Journal of the Royal Statistical Society: Series A (Statistics in Society)*.

[11] Basu, S., A. Banerjee, and R. J. Mooney (2004), Active semi-supervision for pairwise constrained clustering, in *SIAM International Conference on Data Mining (SDM)*, vol. 4, pp. 333–344.

[12] Belkin, M., and P. Niyogi (2003), Laplacian eigenmaps for dimensionality reduction and data representation, *Neural computation*, *15*(6), 1373–1396.

[13] Benaych-Georges, F., and R. R. Nadakuditi (2012), The singular values and vectors of low rank perturbations of large rectangular random matrices, *Journal of Multivariate Analysis*, *111*(0), 120–135.

[14] Benjamini, Y., and Y. Hochberg (1995), Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289–300.

[15] Benzi, K., B. Ricaud, and P. Vandergheynst (2016), Principal patterns on graphs: Discovering coherent structures in datasets, *IEEE Trans. Signal Inf. Process. Netw.*, *2*(2), 160–173.

[16] Bertrand, A., and M. Moonen (2013), Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks, *Signal Processing*, *93*(5), 1106–1117.

[17] Bickel, P. J., and A. Chen (2009), A nonparametric view of network models and newmangirvan and other modularities, *Proceedings of the National Academy of Sciences*, *106*(50), 21,068–21,073.

[18] Blondel, V. D., J.-L. Guillaume, R. Lambiotte, and E. Lefebvre (2008), Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment*, (10).

[19] Buldyrev, S. V., R. Parshani, G. Paul, H. E. Stanley, and S. Havlin (2010), Catastrophic cascade of failures in interdependent networks, *Nature*, *464*(7291), 1025–1028.

[20] Cai, D., Z. Shao, X. He, X. Yan, and J. Han (2005), Community mining from multi-relational networks, in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 445–452, Springer.

[21] Cantador, I., P. Brusilovsky, and T. Kuflik (2011), 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec), in *ACM conference on Recommender systems*.

[22] Chandler Davis, W. M. K. (1970), The rotation of eigenvectors by a perturbation. iii, *SIAM Journal on Numerical Analysis*, *7*(1), 1–46.

[23] Chang, Y.-P., and W.-T. Huang (2000), Generalized confidence intervals for the largest value of some functions of parameters under normality, *Statistica Sinica*, pp. 1369–1383.

[24] Chen, P.-Y., and A. Hero (2015), Deep community detection, *IEEE Trans. Signal Process.*, *63*(21), 5706–5719.

[25] Chen, P.-Y., and A. Hero (2015), Phase transitions in spectral community detection, *IEEE Trans. Signal Process.*, *63*(16), 4339–4347.

[26] Chen, P.-Y., and A. O. Hero (2013), Node removal vulnerability of the largest component of a network, in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*.

[27] Chen, P.-Y., and A. O. Hero (2014), Assessing and safeguarding network resilience to nodal attacks, *IEEE Commun. Mag.*, *52*(11), 138–143.

[28] Chen, P.-Y., and A. O. Hero (2014), Local Fiedler vector centrality for detection of deep and overlapping communities in networks, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1120–1124.

[29] Chen, P.-Y., and A. O. Hero (2015), Universal phase transition in community detectability under a stochastic block model, *Phys. Rev. E*, *91*, 032,804.

[30] Chen, P.-Y., and A. O. Hero (2015), Phase transtions in spectral community detection in large noisy networks, pp. 3402–3406.

[31] Chen, P.-Y., and A. O. Hero (2016), Phase transitions and a model order selection criterion for spectral graph clustering, *arXiv preprint arXiv:1604.03159*.

[32] Chen, P.-Y., S.-M. Cheng, and K.-C. Chen (2012), Smart attacks in smart grid communication networks, *IEEE Commun. Mag.*, *50*(8), 24–29.

[33] Chen, P.-Y., S.-M. Cheng, and K.-C. Chen (2014), Information fusion to defend intentional attack in Internet of things, *IEEE Internet Things J.*, *1*(4), 337–348.

[34] Chen, P.-Y., S. Choudhury, and A. O. Hero (2016), Multi-centrality graph spectral decompositions and their application to cyber intrusion detection, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4553–4557.

[35] Chen, P.-Y., C.-C. Tu, P.-S. Ting, Y.-Y. Lo, D. Koutra, and A. O. Hero (2016), Identifying influential links for event propagation on twitter: A network of networks approach, *arXiv preprint arXiv:1609.05378*.

[36] Chen, P.-Y., B. Zhang, M. A. Hasan, and A. O. Hero (2016), Incremental method for spectral clustering of increasing orders, in *ACM International Conference on Knowledge Discovery and Data Mining (KDD) Workshop on Mining and Learning with Graphs*, arXiv preprint arXiv:1512.07349.

[37] Chen, Y., G. Paul, S. Havlin, F. Liljeros, and H. E. Stanley (2008), Finding a better immunization strategy, *Phys. Rev. Lett.*, *101*, 058,701.

[38] Chung, F. R. K. (1997), *Spectral Graph Theory*, American Mathematical Society.

[39] Cohen, R., S. Havlin, and D. ben Avraham (2003), Efficient immunization strategies for computer networks and populations, *Phys. Rev. Lett.*, *91*, 247,901.

[40] de Abreu, N. M. M. (2007), Old and new results on algebraic connectivity of graphs, *Linear Algebra and its Applications*, *423*, 53–73.

[41] de Arruda, G. F., A. L. Barbieri, P. M. Rodríguez, F. A. Rodrigues, Y. Moreno, and L. d. F. Costa (2014), Role of centrality for the identification of influential spreaders in complex networks, *Phys. Rev. E*, *90*, 032,812.

[42] De Domenico, M., A. Lima, P. Mougel, and M. Musolesi (2013), The anatomy of a scientific rumor, *Scientific reports*, *3*.

[43] De Domenico, M., A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas (2013), Mathematical formulation of multilayer networks, *Phys. Rev. X*, *3*, 041,022.

[44] De Domenico, M., A. Solé-Ribalta, S. Gómez, and A. Arenas (2014), Navigability of interconnected networks under random failures, *Proceedings of the National Academy of Sciences (PNAS)*, *111*(23), 8351–8356.

[45] De Domenico, M., A. Lancichinetti, A. Arenas, and M. Rosvall (2015), Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems, *Phys. Rev. X*, *5*, 011,027.

[46] De Domenico, M., V. Nicosia, A. Arenas, and V. Latora (2015), Structural reducibility of multilayer networks, *Nature Communications*, *6*.

[47] Decelle, A., F. Krzakala, C. Moore, and L. Zdeborová (2011), Inference and phase transitions in the detection of modules in sparse networks, *Phys. Rev. Lett.*, *107*, 065,701.

[48] Decelle, A., F. Krzakala, C. Moore, and L. Zdeborová (2011), Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications, *Phys. Rev. E*, *84*, 066,106.

[49] Del Vicario, M., A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi (2016), The spreading of misinformation online, *Proceedings of the National Academy of Sciences*, *113*(3), 554–559.

[50] Dhanjal, C., R. Gaudel, and S. Clémençon (2014), Efficient eigen-updating for spectral graph clustering, *Neurocomputing*, *131*, 440–452.

[51] Dong, X., P. Frossard, P. Vandergheynst, and N. Nefedov (2012), Clustering with multi-layer graphs: A spectral perspective, *IEEE Trans. Signal Process.*, *60*(11), 5820–5831.

[52] Dong, X., P. Frossard, P. Vandergheynst, and N. Nefedov (2014), Clustering on multi-layer graphs via subspace analysis on grassmann manifolds, *IEEE Trans. Signal Process.*, *62*(4), 905–918.

[53] Easley, D., and J. Kleinberg (2010), *Networks, Crowds, and Markets: Reasoning About A Highly Connected World*, Cambridge University Press.

[54] Everett, M., and S. P. Borgatti (2005), Ego network betweenness, *Social Networks*, *27*(1), 31–38.

[55] Fiedler, M. (1973), Algebraic connectivity of graphs, *Czechoslovak Mathematical Journal*, *23*(98), 298–305.

[56] Fortunato, S. (2010), Community detection in graphs, *Physics Reports*, *486*(3-5), 75–174.

[57] Fortunato, S., and M. Barthelemy (2007), Resolution limit in community detection, *Proc. National Academy of Sciences*, *104*(1), 36–41.

[58] Freeman, L. (1977), A set of measures of centrality based on betweenness, *Sociometry*, *40*, 35–41.

[59] Fujishige, S. (1990), *Submodular Functions and Optimization*, Annals of Discrete Math., North Holland.

[60] Gao, C., and J. Liu (2013), Modeling and restraining mobile virus propagation, *IEEE Trans. Mobile Comput.*, *12*(3), 529–541.

[61] Gao, J., S. V. Buldyrev, S. Havlin, and H. E. Stanley (2011), Robustness of a network of networks, *Phys. Rev. Lett.*, *107*, 195,701.

[62] Ghosh, A., and S. Boyd (2006), Growing well-connected graphs, in *IEEE Conference on Decision and Control*, pp. 6605–6611.

[63] Girvan, M., and M. E. J. Newman (2002), Community structure in social and biological networks, *Proc. National Academy of Sciences*, *99*(12), 7821–7826.

[64] Gleich, D. (2008), MatlabBGL: A matlab graph library, *https://www.cs.purdue.edu/homes/dgleich*.

[65] Goldenberg, A., A. X. Zheng, S. E. Fienberg, and E. M. Airoldi (2010), A survey of statistical network models, *Foundations and Trends® in Machine Learning*, *2*(2), 129–233.

[66] Goldman, H., R. McQuaid, and J. Picciotto (2011), Cyber resilience for mission assurance, in *IEEE International Conference on Technologies for Homeland Security (HST)*, pp. 236–241.

[67] Greene, D., and P. Cunningham (2013), Producing a unified graph representation from multiple social network views, in *ACM Web Science Conference*, pp. 118–121.

[68] Grigg, C., et al. (1999), The IEEE reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee, *IEEE Trans. Power Syst.*, *14*(3), 1010–1020.

[69] Guo, J.-M. (2005), A new upper bound for the laplacian spectral radius of graphs, *Linear Algebra and its Applications*, *400*, 61–66.

[70] Hajek, B., Y. Wu, and J. Xu (2015), Achieving exact cluster recovery threshold via semidefinite programming, in *IEEE International Symposium on Information Theory (ISIT)*, pp. 1442–1446.

[71] Han, Q., K. Xu, and E. Airoldi (2015), Consistent estimation of dynamic and multi-layer block models, in *International Conference on Machine Learning*, pp. 1511–1520.

[72] Hartigan, J. A., and M. A. Wong (1979), Algorithm AS 136: A k-means clusterin algorithm, *Applied statistics*, pp. 100–108.

[73] Hastings, M. B. (2006), Community detection as an inference problem, *Phys. Rev. E*, *74*, 035,102.

[74] Holland, P. W., K. B. Laskey, and S. Leinhardt (1983), Stochastic blockmodels: First steps, *Social Networks*, *5*(2), 109–137.

[75] Holme, P., B. J. Kim, C. N. Yoon, and S. K. Han (2002), Attack vulnerability of complex networks, *Phys. Rev. E*, *65*, 056,109.

[76] Horn, R. A., and C. R. Johnson (1990), *Matrix Analysis*, Cambridge University Press.

[77] Iacovacci, J., Z. Wu, and G. Bianconi (2015), Mesoscopic structures reveal the network between the layers of multiplex data sets, *Phys. Rev. E*, *92*, 042,806.

[78] Jennings, A., and J. J. McKeown (1992), *Matrix computation*, John Wiley & Sons Inc.

[79] Jia, P., J. Yin, X. Huang, and D. Hu (2009), Incremental laplacian eigenmaps by preserving adjacent information between data points, *Pattern Recognition Letters*, *30*(16), 1457–1463.

[80] Karrer, B., and M. E. J. Newman (2011), Stochastic blockmodels and community structure in networks, *Phys. Rev. E*, *83*, 016,107.

[81] Kim, J., and J.-G. Lee (2015), Community detection in multi-layer graphs: A survey, *ACM SIGMOD Record*, *44*(3), 37–48.

[82] Kimura, M., K. Saito, and H. Motoda (2009), Blocking links to minimize contamination spread in a social network, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *3*(2), 9.

[83] Kitsak, M., L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse (2010), Identification of influential spreaders in complex networks, *Nature Physics*, *6*(11), 888–893.

[84] Kivelä, M., A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter (2014), Multilayer networks, *Journal of complex networks*, *2*(3), 203–271.

[85] Knight, S., H. Nguyen, N. Falkner, R. Bowden, and M. Roughan (2011), The Internet topology zoo, *IEEE J. Sel. Areas Commun.*, *29*(9), 1765–1775.

[86] Krause, A., and D. Golovin (2012), Submodular function maximization, *Tractability: Practical Approaches to Hard Problems*, *3*, 19.

[87] Krzakala, F., C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborova, and P. Zhang (2013), Spectral redemption in clustering sparse networks, *Proc. National Academy of Sciences*, *110*, 20,935–20,940.

[88] Kuczynski, J., and H. Wozniakowski (1992), Estimating the largest eigenvalue by the power and lanczos algorithms with a random start, *SIAM journal on matrix analysis and applications*, *13*(4), 1094–1122.

[89] Kuncheva, Z., and G. Montana (2015), Community detection in multiplex networks using locally adaptive random walks, in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1308–1315, ACM.

[90] Lanczos, C. (1950), 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*, *45*(4).

[91] Latala, R. (2005), Some estimates of norms of random matrices., *Proc. Am. Math. Soc.*, *133*(5), 1273–1282.

[92] Le, C. M., and R. Vershynin (2015), Concentration and regularization of random graphs, *arXiv preprint arXiv:1506.00669*.

[93] Le, C. M., E. Levina, and R. Vershynin (2015), Sparse random graphs: regularization and concentration of the laplacian, *arXiv preprint arXiv:1502.03049*.

[94] Lehoucq, R. B., D. C. Sorensen, and C. Yang (1998), *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, vol. 6, Siam.

[95] Leonardi, N., and D. Van De Ville (2013), Tight wavelet frames on multislice graphs, *IEEE Trans. Signal Process.*, *61*(13), 3357–3367.

[96] Lusseau, D., K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson (2003), The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, *Behavioral Ecology and Sociobiology*, *54*(4), 396–405.

[97] Luxburg, U. (2007), A tutorial on spectral clustering, *Statistics and Computing*, *17*(4), 395–416.

[98] Merris, R. (1994), Laplacian matrices of graphs: a survey, *Linear Algebra and its Applications*, *197-198*, 143–176.

[99] Miller, B., N. Bliss, and P. Wolfe (2010), Toward signal processing theory for graphs and non-Euclidean data, in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 5414–5417.

[100] Miller, B., N. Bliss, and P. J. Wolfe (2010), Subgraph detection using eigenvector L1 norms, pp. 1633–1641.

[101] Mucha, P. J., T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela (2010), Community structure in time-dependent, multiscale, and multiplex networks, *Science*, *328*(5980), 876–878.

[102] Nadakuditi, R. R., and M. E. J. Newman (2012), Graph spectra and the detectability of community structure in networks, *Phys. Rev. Lett.*, *108*, 188,701.

[103] Nemhauser, G. L., L. A. Wolsey, and M. L. Fisher (1978), An analysis of approximations for maximizing submodular set functions I, *Mathematical Programming*, *14*, 265–294.

[104] Newman, M. E. J. (2004), Fast algorithm for detecting community structure in networks, *Phys. Rev. E*, *69*, 066,133.

[105] Newman, M. E. J. (2006), Modularity and community structure in networks, *Proc. National Academy of Sciences*, *103*(23), 8577–8582.

[106] Newman, M. E. J. (2006), Finding community structure in networks using the eigenvectors of matrices, *Phys. Rev. E*, *74*, 036,104.

[107] Newman, M. E. J. (2010), *Networks: An Introduction*, Oxford University Press, Inc.

[108] Ng, A. Y., M. I. Jordan, and Y. Weiss (2002), On spectral clustering: Analysis and an algorithm, in *Advances in neural information processing systems (NIPS)*, pp. 849–856.

[109] Ni, J., H. Tong, W. Fan, and X. Zhang (2014), Inside the atoms: ranking on a network of networks, in *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1356–1365.

[110] Ni, J., H. Tong, W. Fan, and X. Zhang (2015), Flexible and robust multi-network clustering, in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 835–844, ACM.

[111] Ning, H., W. Xu, Y. Chi, Y. Gong, and T. S. Huang (2007), Incremental spectral clustering with application to monitoring of evolving blog communities., in *SIAM International Conference on Data Mining (SDM)*, pp. 261–272.

[112] Ning, H., W. Xu, Y. Chi, Y. Gong, and T. S. Huang (2010), Incremental spectral clustering by efficiently updating the eigen-system, *Pattern Recognition*, *43*(1), 113–127.

[113] Nowzari, C., V. M. Preciado, and G. J. Pappas (2016), Analysis and control of epidemics: A survey of spreading processes on complex networks, *IEEE Control Syst. Mag.*, *36*(1), 26–46.

[114] Olfati-Saber, R., J. Fax, and R. Murray (2007), Consensus and cooperation in networked multi-agent systems, *Proc. IEEE*, *95*(1), 215–233.

[115] Oselio, B., A. Kulesza, and A. O. Hero (2014), Multi-layer graph analysis for dynamic social networks, *IEEE J. Sel. Topics Signal Process.*, *8*(4), 514–523.

[116] Oselio, B., A. Kulesza, and A. Hero (2015), Information extraction from large multi-layer social networks, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5451–5455.

[117] Papalexakis, E. E., L. Akoglu, and D. Ience (2013), Do more views of a graph help? community detection and clustering in multi-graphs, in *International Conference on Information Fusion*, pp. 899–905, IEEE.

[118] Parlett, B. N. (1980), *The symmetric eigenvalue problem*, vol. 7, SIAM.

[119] Parshani, R., S. V. Buldyrev, and S. Havlin (2010), Interdependent networks: Reducing the coupling strength leads to a change from a first to second order percolation transition, *Phys. Rev. Lett.*, *105*, 048,701.

[120] Pastor-Satorras, R., C. Castellano, P. Van Mieghem, and A. Vespignani (2015), Epidemic processes in complex networks, *Rev. Mod. Phys.*, *87*, 925–979.

[121] Paul, S., and Y. Chen (2015), Community detection in multi-relational data with restricted multi-layer stochastic blockmodel, *arXiv preprint arXiv:1506.02699*.

[122] Peixoto, T. P. (2013), Eigenvalue spectra of modular networks, *Phys. Rev. Lett.*, *111*, 098,701.

[123] Pentland, A., N. Eagle, and D. Lazer (2009), Inferring social network structure using mobile phone data, *Proceedings of the National Academy of Sciences (PNAS)*, *106*(36), 15,274–15,278.

[124] Polito, M., and P. Perona (2001), Grouping and dimensionality reduction by locally linear embedding, in *Advances in neural information processing systems (NIPS)*.

[125] Pons, P., and M. Latapy (2006), Computing communities in large networks using random walks., *J. Graph Algorithms Appl.*, *10*(2), 191–218.

[126] Poon, L. K. M., A. H. Liu, T. Liu, and N. L. Zhang (2012), A model-based approach to rounding in spectral clustering, in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 68–694.

[127] Pothen, A., H. D. Simon, and K.-P. Liou (1990), Partitioning sparse matrices with eigenvectors of graphs, *SIAM J. Matrix Anal. Appl.*, *11*(3), 430–452.

[128] Pothen, A., H. D. Simon, and K.-P. Liou (1990), Partitioning sparse matrices with eigenvectors of graphs, *SIAM journal on matrix analysis and applications*, *11*(3), 430–452.

[129] Potthoff, R. F., and M. Whittinghill (1966), Testing for homogeneity: I. the binomial and multinomial distributions, *Biometrika*, *53*(1-2), 167–182.

[130] Prakash, B. A., D. Chakrabarti, N. C. Valler, M. Faloutsos, and C. Faloutsos (2012), Threshold conditions for arbitrary cascade models on arbitrary networks, *Knowledge and Information Systems*, *33*(3), 549–575.

[131] Radicchi, F. (2013), Detectability of communities in heterogeneous networks, *Phys. Rev. E*, *88*, 010,801.

[132] Radicchi, F. (2014), A paradox in community detection, *EPL (Europhysics Letters)*, *106*(3), 38,001.

[133] Radicchi, F., and A. Arenas (2013), Abrupt transition in the structural formation of interconnected networks, *Nature Physics*, *9*(11), 717–720.

[134] Radicchi, F., and C. Castellano (2016), Leveraging percolation theory to single out influential spreaders in networks, *Phys. Rev. E*, *93*, 062,314.

[135] Raghavan, U. N., R. Albert, and S. Kumara (2007), Near linear time algorithm to detect community structures in large-scale networks, *Phys. Rev. E*, *76*, 036,106.

[136] Ranjan, G., Z.-L. Zhang, and D. Boley (2014), Incremental computation of pseudo-inverse of laplacian, in *Combinatorial Optimization and Applications*, pp. 729–749, Springer.

[137] Resnick, S. (2013), *A Probability Path*, Birkhäuser Boston.

[138] Ronhovde, P., and Z. Nussinov (2009), Multiresolution community detection for megascale networks by information-based replica correlations, *Phys. Rev. E*, *80*, 016,109.

[139] Saade, A., F. Krzakala, M. Lelarge, and L. Zdeborova (2015), Spectral detection in the censored block model, *arXiv:1502.00163*.

[140] Sabidussi, G. (1966), The centrality index of a graph, *Psychometrika*, *31*(4), 581–603.

[141] Saha, T. K., B. Zhang, and M. Al Hasan (2015), Name disambiguation from link data in a collaboration graph using temporal and topological features, *Social Network Analysis and Mining*, *5*, 1–14.

[142] Saumell-Mendiola, A., M. A. Serrano, and M. Boguñá (2012), Epidemic spreading on interconnected networks, *Phys. Rev. E*, *86*, 026,106.

[143] Schaeffer, S. E. (2007), Graph clustering, *Computer Science Review*, *1*(1), 27–64.

[144] Shi, J., and J. Malik (2000), Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, *22*(8), 888–905.

[145] Shuman, D., S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst (2013), The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, *IEEE Signal Process. Mag.*, *30*(3), 83–98.

[146] Silver, D., et al. (2016), Mastering the game of go with deep neural networks and tree search, *Nature*, *529*(7587), 484–489.

[147] Simes, R. J. (1986), An improved bonferroni procedure for multiple tests of significance, *Biometrika*, *73*(3), 751–754.

[148] Spielman, D. A., and S.-H. Teng (2007), Spectral partitioning works: Planar graphs and finite element meshes, *Linear Algebra and its Applications*, *421*(2-3), 284–305.

[149] Stanley, N., S. Shai, D. Taylor, and P. J. Mucha (2016), Clustering network layers with the strata multilayer stochastic block model, *IEEE Transactions on Network Science and Engineering*, *3*(2), 95–105.

[150] Talagrand, M. (1995), Concentration of measure and isoperimetric inequalities in product spaces, *Publications Mathmatiques de l'Institut des Hautes tudes Scientifiques*, *81*(1), 73–205.

[151] Tang, J., J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su (2008), Arnetminer: extraction and mining of academic social networks, in *ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998.

[152] Tang, J., T. Lou, and J. Kleinberg (2012), Inferring social ties across heterogenous networks, in *ACM International Conference on Web Search and Data Mining*, pp. 743–752.

[153] Tang, L., X. Wang, and H. Liu (2009), Uncoverning groups via heterogeneous interaction analysis, in *IEEE International Conference on Data Mining*, pp. 503–512, IEEE.

[154] Tang, L., X. Wang, and H. Liu (2012), Community detection via heterogeneous interaction analysis, *Data Mining and Knowledge Discovery*, *25*(1), 1–33.

[155] Tang, W., Z. Lu, and I. S. Dhillon (2009), Clustering with multiple graphs, in *IEEE International Conference on Data Mining*, pp. 1016–1021, IEEE.

[156] Taylor, D., S. Shai, N. Stanley, and P. J. Mucha (2016), Enhanced detectability of community structure in multilayer networks through layer aggregation, *Phys. Rev. Lett.*, *116*, 228,301.

[157] Tong, H., B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos (2012), Gelling, and melting, large graphs by edge manipulation, in *ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 245–254.

[158] Tong, H., B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos (2012), Gelling, and melting, large graphs by edge manipulation, in *ACM CIKM*, pp. 245–254.

[159] Vallès-Català, T., F. A. Massucci, R. Guimerà, and M. Sales-Pardo (2016), Multilayer stochastic block models reveal the multilayer structure of complex networks, *Phys. Rev. X*, *6*, 011,036.

[160] Van Dongen, S. M. (2000), Graph clustering by flow simulation, Ph.D. thesis, University of Utrecht.

[161] Van Mieghem, P. (2010), *Graph Spectra for Complex Networks*, Cambridge University Press.

[162] Vickers, M., and S. Chan (1981), Representing classroom social structure, *Victoria Institute of Secondary Education, Melbourne.*

[163] Watts, D. J., and S. H. Strogatz (1998), Collective dynamics of 'small-world' networks, *Nature*, *393*(6684), 440–442.

[164] Wedin, P.-A. (1972), Perturbation bounds in connection with singular value decomposition, *BIT Numerical Mathematics*, *12*(1), 99–111.

[165] Wen, H., E. A. Leicht, and R. M. D'Souza (2011), Improving community detection in networks by targeted node removal, *Phys. Rev. E*, *83*, 016,114.

[166] White, S., and P. Smyth (2005), A spectral clustering approach to finding communities in graph., in *SIAM International Conference on Data Mining (SDM)*, vol. 5, pp. 76–84.

[167] Wilks, S. S. (1938), The large-sample distribution of the likelihood ratio for testing composite hypotheses, *The Annals of Mathematical Statistics*, *9*(1), 60–62.

[168] Wu, Z., Z. Bu, J. Cao, and Y. Zhuang (2015), Discovering communities in multi-relational networks, in *User Community Discovery*, pp. 75–95, Springer.

[169] Xiao, S., G. Xiao, and T. H. Cheng (2008), Tolerance of intentional attacks in complex communication networks, *IEEE Commun. Mag.*, *45*(1), 146–152.

[170] Xiaowen, D. (2014), Multi-view signal processing and learning on graphs, Ph.D. thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE.

[171] Xu, K. S., and A. O. Hero (2014), Dynamic stochastic blockmodels for time-evolving social networks, *IEEE J. Sel. Topics Signal Process.*, *8*(4), 552–562.

[172] Yang, J., and S. Counts (2010), Predicting the speed, scale, and range of information diffusion in twitter, *International Conference on Web and Social Media*, *10*, 355–358.

[173] Zachary, W. W. (1977), An information flow model for conflict and fission in small groups, *Journal of Anthropological Research*, *33*(4), 452–473.

[174] Zaki, M. J., and W. M. Jr (2014), *Data Mining and Analysis: Fundamental Concepts and Algorithms*, Cambridge University Press.

[175] Zaki, M. J., and W. Meira Jr (2014), *Data mining and analysis: fundamental concepts and algorithms*, Cambridge University Press.

[176] Zelnik-Manor, L., and P. Perona (2004), Self-tuning spectral clustering, in *Advances in neural information processing systems (NIPS)*, pp. 1601–1608.

[177] Zhang, B., T. K. Saha, and M. Al Hasan (2014), Name disambiguation from link data in a collaboration graph, in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 81–84.

[178] Zhang, B., T. K. Saha, and M. A. Hasan (2014), Name disambiguation from link data in a collaboration graph, in *ASONAM*, pp. 81–84.

[179] Zhang, P., and C. Moore (2014), Scalable detection of statistically significant communities and hierarchies, using message passing for modularity, *Proceedings of the National Academy of Sciences*, *111*(51), 18,144–18,149.

[180] Zhang, P., F. Krzakala, J. Reichardt, and L. Zdeborov (2012), Comparative study for inference of hidden classes in stochastic block models, *Journal of Statistical Mechanics: Theory and Experiment*, p. P12021.

[181] Zhang, P., F. Krzakala, J. Reichardt, and L. Zdeborov (2012), Comparative study for inference of hidden classes in stochastic block models, *Journal of Statistical Mechanics: Theory and Experiment*, p. P12021.

[182] Zou, C. C., D. Towsley, and W. Gong (2007), Modeling and simulation study of the propagation and defense of internet e-mail worms, *IEEE Trans. Depend. Sec. Comput.*, *4*(2), 105–118.