Semidefinite Programming Approaches to Network Clustering and Smoothing

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Zhifei Yan, M.S.
Graduate Program in Statistics

The Ohio State University

2017

Dissertation Committee:

Vincent Q. Vu, Advisor
Yoonkyung Lee
Yunzhang Zhu
Abstract

Community detection and link prediction are two important problems in network analysis. In this dissertation, we propose semidefinite programming approaches to network community detection and edge probabilities estimation. Interestingly, despite different motivations, it turns out that the proposed approaches to the two problems are based on the same semidefinite program (SDP) with appropriately chosen input matrices. Our SDP relies on a semidefinite relaxation of the set of normalized clustering matrices. We derive optimality conditions of the SDP from a geometric point of view and deal with a convex body, called the Fantope. This enables us to analyze the SDP by making use of the properties of the Fantope. We design an efficient alternating direction method of multipliers algorithm to solve the SDP, which only requires computation of a small number of leading eigenvectors of symmetric matrices.

For community detection, the SDP is derived from the partition criterion of maximizing the sum of average intra-cluster similarities over all clusters. The feasible set of our SDP is contained in the Fantope, which enables us to connect our SDP to sparse PCA and spectral clustering. Unlike previously proposed SDPs for community detection which use the adjacency matrix as the input matrix, we also consider using the symmetric normalized graph Laplacian matrix and the squared adjacency matrix as the input matrix to the SDP. The former choice of the input matrix is motivated
by the connection between our SDP and spectral clustering and the latter builds an interesting connection between network clustering and multivariate data clustering. Using optimality conditions, we analyze the population consistency of the SDP using different input matrices under various network models. We show the exact recovery property of our SDP under strongly assortative stochastic block models when the input matrix is chosen to be the adjacency matrix. For network edge probability estimation, the SDP is derived from a least squares criterion. Since the solution to our SDP is a symmetric doubly stochastic matrix, it can be used as a smoother matrix to smooth the adjacency matrix. The estimated edge probability matrix is the smoothed adjacency matrix. We also connect our SDP-based method of estimating edge probabilities to a recently proposed neighborhood smoothing method. Our numerical experiments show that the proposed SDP has good empirical performances for community detection, edge probability estimation, and link prediction and it outperforms other state-of-the-art methods on Facebook collegiate networks for both community detection and link prediction.
Dedicated to my family
Acknowledgments

I owe my deepest gratitude to my advisor, Dr. Vincent Vu, for his mentorship over the past several years. I thank him for trusting me to do work independently, challenging me to sharpen my statistical thinking skills, instilling his wisdom into me, and financially supporting me for my research work. Without his endless support and training, I would not have achieved statistical maturity as I have today. My gratitude also goes to my dissertation committee members, Dr. Yoonkyung Lee and Dr. Yunzhang Zhu. Thank you for your time devoted to reading and providing constructive comments on my research work.

I would like to give my special thanks to Dr. Peter Craigmile, Dr. Mario Peruggia, and Dr. Trisha Van Zandt for providing me with great research opportunities and financial support. I want to further thank Dr. Peter Craigmile for kindly allowing me to sit in his classes and patiently answering my numerous questions.

Finally, I want to express my sincere love to my parents and my fiancée. I thank my father, Zongshu Yan, and my mother, Jie Wang, for offering me a happy growing up experience and encouraging me to chase my dreams. Lastly, to my fiancéé, Ranran Dong, thank you for taking care of me when I am stressful, making me laugh when I am not even in the mood to smile, and brightening up my days in graduate study.

This dissertation research is supported in part by the National Science Foundation under grant DMS-1513621.
Vita

2012 ................................. B.S. Mathematics, Fudan University

2015 ................................. M.S. Statistics, The Ohio State University

2012-present ......................... Graduate Research and Teaching Associate, The Ohio State University

Fields of Study

Major Field: Statistics
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Algorithms</td>
<td>xii</td>
</tr>
<tr>
<td>1. Introduction and Literature Review</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Multivariate Data Clustering</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Model-based Methods</td>
<td>3</td>
</tr>
<tr>
<td>1.1.2 Partitioning Methods</td>
<td>4</td>
</tr>
<tr>
<td>1.1.3 Spectral Clustering</td>
<td>6</td>
</tr>
<tr>
<td>1.2 Network Community Detection</td>
<td>7</td>
</tr>
<tr>
<td>1.2.1 Model-based Methods</td>
<td>8</td>
</tr>
<tr>
<td>1.2.2 Partitioning Methods</td>
<td>14</td>
</tr>
<tr>
<td>1.2.3 Spectral Clustering</td>
<td>15</td>
</tr>
<tr>
<td>1.3 Estimating Network Edge Probabilities</td>
<td>16</td>
</tr>
<tr>
<td>2. The Semidefinite Program and its General Analysis Framework</td>
<td>20</td>
</tr>
<tr>
<td>2.1 Problem Setup and A Semidefinite Program</td>
<td>21</td>
</tr>
<tr>
<td>2.2 Optimality Conditions</td>
<td>24</td>
</tr>
<tr>
<td>2.3 A Sufficient Condition for Fisher Consistency of the SDP</td>
<td>29</td>
</tr>
<tr>
<td>2.4 An Efficient Algorithm for Solving the SDP</td>
<td>31</td>
</tr>
</tbody>
</table>

vii
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>36</td>
</tr>
<tr>
<td>3.1</td>
<td>46</td>
</tr>
<tr>
<td>3.2</td>
<td>57</td>
</tr>
<tr>
<td>3.3</td>
<td>58</td>
</tr>
<tr>
<td>3.4</td>
<td>60</td>
</tr>
<tr>
<td>3.5</td>
<td>61</td>
</tr>
</tbody>
</table>
3.6 Permuted adjacency matrices of Caltech network according to estimated labels using different methods. Blue pixels in the visualized adjacency matrices stand for entries of one and white pixels stand for entries of zero.

3.7 Permuted adjacency matrices of Simmons College network according to estimated labels using different methods. Blue pixels in the visualized adjacency matrices stand for entries of one and white pixels stand for entries of zero.

4.1 Left panel: true edge probability matrix generated from graphon 1. Middle panel: estimated probability matrix from SDP-BF. Right panel: estimated probability matrix from NBS. Blue represents large values and white represents small values.

4.2 Average mean squared errors and the corresponding one standard deviation error bars for estimating edge probability matrices generated from graphon 1 with different numbers of nodes. The number of replicates is 20.

4.3 Left panel: true edge probability matrix generated from graphon 2. Middle panel: estimated probability matrix from SDP-BF. Right panel: estimated probability matrix from NBS. Blue represents large values and white represents small values.

4.4 Average mean squared errors and the corresponding one standard deviation error bars for estimating edge probability matrices generated from graphon 2 with different numbers of nodes. The number of replicates is 20.

4.5 Left panel: true edge probability matrix generated from graphon 3. Middle panel: estimated probability matrix from SDP-BF. Right panel: estimated probability matrix from NBS. Blue represents large values and white represents small values.

4.6 Average mean squared errors and the corresponding one standard deviation error bars for estimating edge probability matrices generated from graphon 3 with different numbers of nodes. The number of replicates is 20.
4.7 Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ under graphon 1. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 10$ chosen from cross-validation and NBS with different values of $C$ under graphon 1.

4.8 Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ under graphon 2. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 10$ chosen from cross-validation and NBS with different values of $C$ under graphon 2.

4.9 Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ under graphon 3. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 10$ chosen from cross-validation and NBS with different values of $C$ under graphon 3.

4.10 Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ on the Caltech network. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 20$ chosen from cross-validation and NBS with different values of $C$ on the Caltech network.
# List of Algorithms

1. ADMM algorithm for solving SDP-BF ................................. 32
2. Frank-Wolfe algorithm for solving the subproblem (2.13) in ADMM . 35
Chapter 1: Introduction and Literature Review

Network data describe interactions between individuals. In a network, an individual is represented by a node and the interaction between two individuals is represented by an edge. Examples of real networks include social networks, collaboration networks, road networks, and so on. In a social network, each user is a node and an edge exists between two users if they are friends. In a collaboration network, each author is considered as a node and an edge exists between two authors if they have co-authorship in a paper. In a road network, each intersection or endpoint represents a node and an edge exists between two nodes if there is a road connecting them.

An important task of network analysis is network clustering, also called community detection, where a cluster in a network is called a community. There is no unique way of defining a network community. A widely used notion of community is that a community is a group of nodes with many links between themselves and fewer links to the rest of the network. For example, communities can be formed by institutions or hobbies in a social network, affiliations or research interests in a collaboration network, and districts or cities in a road network. The goal of community detection is to cluster nodes into communities in a network. Another interesting problem in network analysis is to estimate edge probabilities based on a network. A practical application of this is predicting links between nodes in a network, i.e., link prediction. In social networks
such as Facebook network, link prediction can be used to recommend potential friends to users.

This chapter reviews several different approaches to network community detection and edge probabilities estimation. Since there are close connections between multivariate data clustering and network clustering, we start with reviewing methods for multivariate data clustering in Section 1.1. In Section 1.2, we introduce various approaches to network community detection. Section 1.3 provides a summary of ways of estimating network edge probabilities.

1.1 Multivariate Data Clustering

Over the years, researchers have proposed numerous methods and algorithms for multivariate data clustering including model-based methods (Fraley and Raftery, 2002), partitioning methods (Hagen and Kahng, 1992; MacQueen, 1967; Shi and Malik, 2000), semidefinite programming (SDP) methods (Awasthi et al., 2015; Iguchi et al., 2015; Mixon et al., 2016; Peng and Xia, 2005), and spectral clustering (Hagen and Kahng, 1992; Shi and Malik, 2000). Model-based approaches rely on fitting a probabilistic model for the data. In multivariate data clustering, finite mixture models such as the Gaussian mixture model (GMM) where each cluster corresponds to a mixture component have been extensively employed. Partitioning methods partition objects into clusters based on the optimization of global criteria. The well-known $k$-means clustering formulation (MacQueen, 1967) belongs to this class. Unfortunately, the $k$-means problem is NP hard (Aloise et al., 2009) and the commonly used two-step iterative algorithm has no guarantee to find the global optimal solution. Recently, semidefinite relaxations of $k$-means problem have appeared in the literature.
SDP approaches are attractive since they solve global optimization problems and theoretical properties of solutions can be understood. Finally, spectral methods project objects onto a low dimensional subspace where they are easily separable and perform clustering in the low dimensional subspace (Hagen and Kahng, 1992; Shi and Malik, 2000).

1.1.1 Model-based Methods

For multivariate data clustering, perhaps the best known probabilistic generative model is the Gaussian mixture model. In the GMM, for each observation, its cluster label is generated from a categorical distribution based on mixture weights. Conditional on its cluster label, an observation is then generated from the cluster-specific Gaussian distribution. More formally, let \( \pi = (\pi_1, \cdots, \pi_K) \) denote the mixture weights, \( \mu_1, \cdots, \mu_K \in \mathbb{R}^p \) denote the component mean vectors, and \( \Sigma_1, \cdots, \Sigma_K \succ 0 \) denote the component covariance matrices. A GMM can be represented as the form

\[
x_i | g_i \sim N_p(\mu_{g_i}, \Sigma_{g_i}), i = 1, \cdots, n,
\]

\[
P(g_i = k) = \pi_k, k = 1, \cdots, K,
\]

where the random variable \( g_i \) represents the latent cluster membership for \( i \)th observation \( x_i \). Write \( \theta = (\pi, \mu_1, \cdots, \mu_K, \Sigma_1, \cdots, \Sigma_K) \). Given independent observations of \( x_1, \cdots, x_n \), the inferential task is to estimate the model parameters \( \theta \). A common approach is to use the expectation-maximization (EM) algorithm to find the maximum likelihood estimates of model parameters (Dempster et al., 1977). As a model-based clustering approach, once the parameter estimates are computed, one can choose the cluster label that maximizes the conditional probabilities \( P(g_i | x_i) \) as the label estimate for \( i \)th observation. Unfortunately, since the log-likelihood is a non-concave
function and typically has many local maxima, the performance of EM algorithm to fit the GMM is sensitive to the initial value.

1.1.2 Partitioning Methods

Partitioning methods partition the objects into clusters based on the optimization of global partition criteria. One example is the $k$-means clustering formulation (MacQueen, 1967). Given a set of multivariate observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and the number of clusters $K$, the $k$-means clustering aims to find the partition of $n$ observations into $K$ clusters $C_1, \cdots, C_K$ such that the sum of squared distance between each observation and its cluster center is minimized. Let $n_k = |C_k|$ be the size of $k$th cluster. More formally, the $k$-means clustering objective function is

$$\min_{C_1, \cdots, C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \frac{1}{n_k} \sum_{j \in C_k} x_j\|^2,$$

(1.1)

where $\| \cdot \|$ denotes the Euclidean norm of a vector. Unfortunately, the $k$-means problem is NP hard (Aloise et al., 2009) and the commonly used two-step iterative algorithm has no guarantee to find the global optimal solution. Over the past decade, semidefinite relaxation methods have been proposed for solving the $k$-means problem.

To introduce semidefinite relaxations of the $k$-means problem, we define the notion of normalized clustering matrices. Given a partition of $n$ objects into $K$ clusters of sizes $n_1, \cdots, n_K$, we define the corresponding normalized clustering matrix as an $n \times n$ matrix $X$ where $X_{ij} = 1/n_k$ if objects $i$ and $j$ both belong to the $k$th cluster, and $X_{ij} = 0$ if they belong to different clusters. The set of all normalized clustering matrices corresponding to all partitions of $n$ objects into $K$ clusters is

$$\mathcal{X}_{\text{norm}} := \{ P \text{ diag} \{ E_{n_1}/n_1, \cdots, E_{n_K}/n_K \} P^T : P \text{ is a permutation matrix}, \sum_{i=1}^{K} n_i = n \},$$
where $E_n$ is the $n \times n$ matrix of all ones. Peng and Xia (2005) introduced the following semidefinite relaxation of (1.1):

$$
\begin{align*}
\text{minimize} & \quad \langle D, X \rangle \\
\text{subject to} & \quad X \succeq 0, X \preceq 0, X \mathbf{1}_n = \mathbf{1}_n, \text{tr}(X) = K,
\end{align*}
$$

(1.2)

where $D$ is the Euclidean distance matrix whose $(i, j)$th entry is defined as $||x_i - x_j||^2$, and $\mathbf{1}_n$ denotes the $n \times 1$ vector of all ones. It relies on a semidefinite relaxation of $\mathcal{X}$. Recently, analyses of performance guarantees of (1.2) for multivariate data clustering under probability models have appeared. Awasthi et al. (2015) and Iguchi et al. (2015) analyzed the exact recovery guarantees of (1.2) under a particular probability model called the stochastic ball model where observations from the same cluster fall within a unit ball from the cluster center. They showed that under the stochastic ball model, the solution to (1.2) can perfectly recover the cluster membership of all observations with high probability. Recently, Mixon et al. (2016) have showed that under the subgaussian mixture model, the Frobenius norm between the solution to (1.2) and the perfect clustering solution can be controlled to be arbitrarily small with high probability if the minimal distance between cluster centers is sufficiently greater than the maximal within cluster variation.

Besides partition criteria directly motivated from the distances between observations, the multivariate data can also be represented in the form of a similarity graph based on some notion of similarity between pairs of observations. In this case, each observation $x_i$ is viewed as a node in a graph, and a pair of nodes $i$ and $j$ is connected by an edge with nonnegative weight $w_{ij}$ defined to be the similarity between the observations $x_i$ and $x_j$. One example is the Gaussian similarity measure $w_{ij} := \exp \left( \frac{||x_i - x_j||^2}{2\sigma^2} \right)$. Through this way, multivariate observations $x_1, \ldots, x_n$
are transformed into a similarity graph with symmetric weighted adjacency matrix $W = (w_{ij})_{1 \leq i,j \leq n}$. Then the clustering problem becomes a graph partitioning problem and graph cut objective functions can be used as global partition criteria. The idea is to find a partition of nodes into clusters such that edges connecting nodes in the same cluster have high weights and edges connecting nodes from different clusters have low weights. Two common graph cut criteria are to minimize the ratio cut of Hagen and Kahng (1992) defined by

$$\text{RatioCut}(C_1, \cdots, C_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{\text{Cut}(C_i, \overline{C_i})}{|C_i|}$$  \hspace{1cm} (1.3)$$

and to minimize the normalized cut of Shi and Malik (2000) defined by

$$\text{NCut}(C_1, \cdots, C_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{\text{Cut}(C_i, \overline{C_i})}{\text{Vol}(C_i)},$$  \hspace{1cm} (1.4)$$

where $|C_i|$ denotes the number of nodes in $C_i$, $\text{Vol}(C_i)$ is the sum of degrees of nodes in $C_i$, $\overline{C_i}$ is the complement of $C_i$, and $\text{Cut}(C_i, \overline{C_i}) = \sum_{a \in C_i, b \in \overline{C_i}} w_{ab}$. Unfortunately, the two graph cut problems both require a computationally intractable combinatorial search and are in general NP hard.

### 1.1.3 Spectral Clustering

In the field of multivariate data clustering, after we transform multivariate observations into a similarity graph by a similarity measure, spectral clustering projects observations onto a low dimensional subspace by taking the spectral decomposition of some form of the graph Laplacian matrix and performs clustering in the low dimensional subspace. Based on the symmetric weighted adjacency matrix $W$ of a similarity graph, there are different forms of graph Laplacians. Let $D$ denote the diagonal matrix with diagonal entries being the degrees $d_1, \cdots, d_n$ of nodes
1, · · · , n. Following Luxburg (2007), we define the unnormalized graph Laplacian matrix by \( L = D - W \), the symmetric normalized graph Laplacian matrix by \( L_{\text{sym}} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \), and the random walk normalized graph Laplacian matrix by \( L_{\text{rw}} = D^{-1}L = I - D^{-1}W \). We introduce unnormalized spectral clustering as an example. Given the weighted adjacency matrix \( W \), we form the unnormalized graph Laplacian matrix \( L \). Then we compute the matrix \( U \in \mathbb{R}^{n \times K} \) whose columns are the \( K \) eigenvectors corresponding to the \( K \) smallest eigenvalues of \( L \). Finally, \( k \)-means clustering is performed to the rows of \( U \) to obtain cluster label estimates. In a similar fashion, normalized spectral clustering algorithms of Shi and Malik (2000) and Ng et al. (2001) work with the random walk normalized graph Laplacian \( L_{\text{rw}} \) and the symmetric normalized graph Laplacian \( L_{\text{sym}} \), respectively. Moreover, it can be shown that the ratio cut problem (1.3) can be relaxed to the unnormalized spectral clustering and the normalized cut problem (1.4) can be relaxed to the normalized spectral clustering by ignoring the discreteness constraint in optimization problems (Luxburg, 2007).

1.2 Network Community Detection

Similar to multivariate data clustering, community detection approaches include model-based methods (Amini, Chen, et al., 2013; Bickel and Chen, 2009; Bickel, Choi, et al., 2013; Decelle et al., 2011; Karrer and Newman, 2011), partitioning methods (Lancichinetti and Fortunato, 2011; Newman, 2006), semidefinite programming methods (Amini and Levina, 2014; Cai and Li, 2015; Chen, Li, et al., 2016; Guédon and Vershynin, 2016), and spectral clustering (Amini, Chen, et al., 2013; Lei and Rinaldo, 2015; Rohe et al., 2011). Model-based approaches try to fit a probabilistic
model that can generate networks with communities based on the observed network. In network community detection, two widely used models are the stochastic block model (SBM) (Holland et al., 1983) and the degree corrected stochastic block model (DCSBM) (Karrer and Newman, 2011). On the other hand, partitioning methods partition nodes into communities based on optimization of global partition criteria. One example of this is the modularity maximization method (Newman, 2006). Unfortunately, likelihood-based methods and optimization problems derived from global partition criteria are often computationally intractable. Recently, semidefinite programming approaches have been proposed for solving computationally intractable likelihood optimization problems (Amini and Levina, 2014; Cai and Li, 2013; Guédon and Vershynin, 2016) and partitioning problems (Chen, Li, et al., 2016). They are attractive since they solve global optimization problems and are expected to provide exact or approximate solutions to original problems. Finally, spectral methods extract community information of nodes from a matrix formed by the first few leading eigenvectors of the adjacency matrix or the graph Laplacian (Lei and Rinaldo, 2015; Rohe et al., 2011).

1.2.1 Model-based Methods

From the statistical modeling point of view, perhaps the best known model with well-defined communities is the stochastic block model (Holland et al., 1983). In the past decade, many generalizations of the SBM have been proposed to account for different features of a real network such as degree-corrected stochastic block model (Karrer and Newman, 2011), mixed membership stochastic block model (Airoldi et al., 2008).
al., 2008), overlapping community model (Zhang et al., 2014), and so on. Generally speaking, model-based methods try to fit a generative model for networks with communities based on its likelihood. We will focus on the SBM and the DCSBM.

**Stochastic Block Model**

Under the SBM, an undirected graph is generated based on a community-wise edge probability matrix where within-community edge probabilities are assumed to be greater than between-community edge probabilities, and the edges of the graph are generated independently. More formally, consider an undirected graph with \( n \) nodes, and each of these nodes belongs to one and only one of the \( K \) disjoint communities. Let \( C_k \) be the set of node indices of the \( k \)th community and \( n_k = |C_k| \) be the size of \( k \)th community. We let \( A \) denote the symmetric binary adjacency matrix of this graph. Let \( g_i \in \{1, \cdots, K\} \) be the community label of node \( i \). We define a symmetric community-wise edge probability matrix \( B \in [0,1]^{K \times K} \), where \( B_{kl} \) is the edge probability between a pair of nodes from communities \( k \) and \( l \). Let \( P = (p_{ij})_{1 \leq i,j \leq n} \in [0,1]^{n \times n} \) denote the probability matrix of graph edges where \( p_{ij} \) is the edge probability between nodes \( i \) and \( j \). In the SBM, \( A_{ij} \)'s are assumed to be independent Bernoulli random variables with connecting probabilities \( p_{ij} = B_{g_i,g_j} \) for all pairs of \((i,j)\). We exclude self loops, so \( p_{ii} = 0 \) and \( A_{ii} = 0 \). To create community structure, the notions of weak and strong assortativity are commonly assumed in the SBM literature (Abbe et al., 2016; Amini and Levina, 2014; Cai and Li, 2015).

**Definition 1.1.** Let \( p_k := B_{kk}, q_k := \max_{l \neq k} B_{kl} \). Write \( p_{\min} := \min_k p_k, q_{\max} := \max_k q_k \). A stochastic block model is called weakly assortative if \( p_k > q_k, \forall k = 1, \cdots, K \). A stochastic block model is called strongly assortative if \( p_{\min} > q_{\max} \).
We call \( Z = (Z_{ij})_{1 \leq i \leq n, 1 \leq j \leq K} \in \{0, 1\}^{n \times K} \) the membership matrix if the \( i \)th row of \( Z \) is 1 at column \( g_i \) and 0 elsewhere. Using the membership matrix, we can express the edge probability matrix \( P = ZBZ^T \) except for the diagonal entries. We define the clustering matrix \( X := ZZ^T \), where \( X_{ij} = 1 \) if nodes \( i \) and \( j \) belong to the same community, and \( X_{ij} = 0 \) otherwise. For community detection in the SBM, the goal is to recover the membership matrix \( Z \) up to column permutations, or equivalently, recover \( X \).

Under the SBM, likelihood-based methods include profile likelihood maximization (Bickel and Chen, 2009), variational likelihood method (Bickel, Choi, et al., 2013), Bayesian fitting methods with Gibbs sampling and belief propagation (Decelle et al., 2011), and pseudo-likelihood method (Amini, Chen, et al., 2013). However, likelihood-based methods are usually computationally difficult for large-scale networks, and their theoretical properties are only justified for relatively dense networks. Next, we focus on SDP-based methods that are derived from maximizing the likelihood under the SBM. Let \( \mathcal{X} \) denote the set of all clustering matrices, i.e.,

\[
\mathcal{X} := \{ P \text{ diag}\{E_{n_1}, \ldots, E_{n_K}\} P^T : P \text{ is a permutation matrix, } \sum_{i=1}^K n_i = n \},
\]

where \( E_n \) is the \( n \times n \) matrix of all ones. To motivate the SDP relaxations of the likelihood maximization problem, we consider a special case of the SBM where within-community edge probabilities are all equal to \( p \) and between-community edge probabilities are all equal to \( q \), with \( p > q \). This model is also known as the planted partition model. Given \( X \) and \( B \), in matrix formulation, the likelihood maximization is equivalent to (see Amini and Levina, 2014; Cai and Li, 2015, for detail)

\[
\begin{align*}
\text{maximize} & \quad \langle A - \lambda E_n, X \rangle \\
\text{subject to} & \quad X \in \mathcal{X},
\end{align*}
\]
where $\lambda := \frac{\log(1-q) - \log(1-p)}{\log p - \log q + \log(1-q) - \log(1-p)} > 0$. In practice, values of $p$ and $q$ are unknown and the parameter $\lambda$ is viewed as a tuning parameter. Unfortunately, finding a global solution to (1.5) requires a computationally intractable combinatorial search. However, the feasible set $\mathcal{X}$ can be relaxed to the intersection of the cone of positive semidefinite matrices with an affine set and the problem is relaxed to an SDP. Let us introduce the following SDP relaxation of (1.5), which is essentially the SDP proposed in Cai and Li (2015):

$$\max \langle A - \lambda E_n, X \rangle$$

subject to $X \succeq 0, 0 \leq X \leq 1$. We call this SDP method SDP-CL. In Cai and Li (2015), the authors further modified the objective function by penalizing the trace to control the possible outliers. However, this will introduce an additional tuning parameter and seems only necessary for theoretical analysis (see Remark 2.2 in Cai and Li, 2015).

A related SDP relaxation proposed by Guédon and Vershynin (2016) is

$$\max \langle A, X \rangle$$

subject to $X \succeq 0, X \preceq 0, \text{diag}(X) = 1_n, \sum_{i,j=1}^n X_{ij} = \tilde{\lambda}$,

$$\tilde{\lambda} = \sum_{i=1}^K n_i^2.$$  

We refer to this method as SDP-GV. We notice that there is a close relation between SDP-CL and SDP-GV. If we view $\lambda \geq 0$ in (1.6) and $\tilde{\lambda} \geq 0$ in (1.7) as tuning parameters and relax the equality constraints $\text{diag}(X) = 1_n, \sum_{i,j=1}^n X_{ij} = \lambda$ in (1.7) to be inequality constraints $\text{diag}(X) \leq 1_n, \sum_{i,j=1}^n X_{ij} \leq \tilde{\lambda}$, then by Lagrange duality, there is a one-to-one correspondence between the parameters $\lambda$ in (1.6) and $\tilde{\lambda}$ in (1.7). In this sense, SDP-CL and SDP-GV are essentially equivalent.

With the appearance of various SDP relaxations of likelihood maximization problem under the SBM, consistency of SDP-based methods has been analyzed. Generally
speaking, theoretical results can be divided into two types. The first type is strong consistency, also called exact recovery. Let $X_0$ and $\hat{X}$ denote the true clustering matrix and the SDP solution matrix, respectively. We say that the SDP is strongly consistent if $X_0$ is the unique solution to the SDP with high probability. In Cai and Li (2015), the authors have shown that if the expected degrees grow no slower than $\log n$, then $\hat{X} = X_0$ with high probability under strongly assortative SBMs if the gap between within-community edge probabilities and between-community edge probabilities is sufficiently large. Similar exact recovery results have been established in Amini and Levina (2014) for an SDP under weakly assortative SBMs with equal sized communities. In fact, it is now well known that the condition that the expected degrees grow no slower than $\log n$ is a necessary condition for exact recovery. And this type of networks are called relatively dense networks (Guédon and Vershynin, 2016). The second type of consistency results is called weak consistency. According to Guédon and Vershynin (2016), if the expected degrees grow slower than $\log n$, then with high probability a positive fraction of the nodes will be isolated. So it is impossible to achieve strong consistency. A particular interesting regime is the sparse regime where the expected degrees are bounded. Recently, Guédon and Vershynin (2016) have shown that SDP-GV can achieve arbitrary relative accuracy with high probability for sparse networks with bounded expected degrees under a generalized version of strongly assortative SBMs. In fact, their analysis holds for any SDP whose feasible set is a relaxation of the set of clustering matrices and is a subset of $\{X \succeq 0, \text{diag}(X) \leq 1_n\}$. Hence, many existing SDPs for network community detection including those proposed in Amini and Levina (2014), Cai and Li (2015), and
Chen and Xu (2016) are able to achieve arbitrary relative accuracy with high probability for sparse networks with bounded expected degrees under strongly assortative SBMs.

**Degree-corrected Stochastic Block Model**

The stochastic block model is attractive due to its simplicity. However, it fails to capture the degree inhomogeneity of real networks (Karrer and Newman, 2011). In an SBM, nodes belonging to the same community have the same degree distribution, while node degrees are often highly inhomogeneous in real networks. Some nodes will have many more connections than others, leading to extreme high degrees, while others may have very few connections to the rest of the network, leading to extreme small degrees. Degree-corrected SBM (Karrer and Newman, 2011) was proposed to capture such degree inhomogeneity. In a DCSBM, we introduce $n$ additional nonnegative degree parameters $\theta_1, \cdots, \theta_n$, one for each node. The edge probability between nodes $i$ and $j$ is defined to be $\theta_i \theta_j B_{g_i g_j}$. Except for self-connection probabilities of nodes which are set to be zero, we can express the edge probability matrix $P = \Theta Z B Z^T \Theta$, where $\Theta := \text{diag}\{\theta_1, \cdots, \theta_n\}$. To have identifiability, we assume that $\max_{1 \leq i \leq n} \theta_i = 1$, which is the constraint used in Chen, Li, et al. (2016).

Under the DCSBM, likelihood-based methods include profile likelihood maximization (Karrer and Newman, 2011) and pseudo-likelihood method (Amini, Chen, et al., 2013). As the case for the SBM, these likelihood-based methods are computationally difficult for large networks and their effectiveness highly depends on the initial value of the algorithm.
1.2.2 Partitioning Methods

Similar to multivariate data clustering, we can design meaningful partition criteria for network community detection. The graph cut criteria mentioned previously can also be applied to network community detection. Another well-known partitioning method is the modularity maximization proposed by Newman (2006). Given a partition of nodes into communities, the idea of modularity is to compare the number of edges between two nodes to the expected number of edges between them assuming edges are placed at random for all pairs of nodes belonging to the same community. Formally, let \( d_i = \sum_{j=1}^{n} A_{ij} \) be the degree of node \( i \). The total number of edges is \( m = \frac{1}{2} \sum_i d_i \). For nodes \( i \) and \( j \) that belong to the same community, if edges are placed at random, then for each edge that connects to node \( i \), the probability that it connects to node \( j \) is \( d_j / 2m \). Since there are \( d_i \) edges connecting to node \( i \), the expected number of edges between nodes \( i \) and \( j \) is \( d_i d_j / 2m \), and the difference between the number of edges and the expected number of edges if edges are placed at random is \( A_{ij} - d_i d_j / 2m \). The modularity of a partition determined by a clustering matrix \( X \) is defined as

\[
Q(X) = \sum_{1 \leq i, j \leq n} (A_{ij} - \frac{d_i d_j}{2m})X_{ij}.
\]

Intuitively, a good partition tends to have large modularity. Thus, the modularity maximization method is to find the clustering matrix \( X \) such that the modularity is maximized, i.e.,

\[
\text{maximize} \sum_{1 \leq i, j \leq n} (A_{ij} - \frac{d_i d_j}{2m})X_{ij} \quad \text{subject to} \quad X \in \mathcal{X}.
\] (1.8)

However, Fortunato and Barthelemy (2007) have pointed out that (1.8) suffers a resolution limit in that it fails to detect small communities and Lancichinetti and
Fortunato (2011) have suggested to solve the generalized modularity maximization

\[
\begin{align*}
\text{maximize} & \quad \sum_{1 \leq i,j \leq n} (A_{ij} - \lambda d_i d_j)X_{ij} \\
\text{subject to} & \quad X \in \mathcal{X}.
\end{align*}
\]  

(1.9)

Solving (1.9) is computationally intractable. Relaxing the set of all clustering matrices \( \mathcal{X} \) leads to the following SDP relaxation of the generalized modularity maximization proposed by Chen, Li, et al. (2016):

\[
\begin{align*}
\text{maximize} & \quad \langle A - \lambda dd^T, X \rangle \\
\text{subject to} & \quad X \succeq 0, 0 \leq X \leq 1, \text{diag}(X) = 1_n.
\end{align*}
\]  

(1.10)

Hereafter, we call this SDP SDP-MM, since it is based on the generalized modularity maximization. Chen, Li, et al. (2016) have showed that under density gap conditions of the DCSBM, the solution to (1.10) can recover the true clustering matrix.

### 1.2.3 Spectral Clustering

Spectral methods for network community detection are similar to spectral clustering algorithms for multivariate data. Given the spectral decomposition of the adjacency matrix of a network or its graph Laplacian, the community information is contained in the first few leading eigenvectors and a clustering heuristic such as \( k \)-means can be applied to the rows of the matrix whose columns are those eigenvectors.

Rohe et al. (2011) considered the spectral clustering to the normalized graph Laplacian defined as \( L = D^{-1/2}AD^{-1/2} \). The algorithm obtains the label estimates based on the leading \( K \) eigenvectors corresponding to the \( K \) largest absolute eigenvalues of \( L \). The rationale is that the eigenvectors of a population version of \( L \) corresponding to the \( K \) largest absolute eigenvalues can infer the true labels under the SBM with \( K \) communities. Lei and Rinaldo (2015) considered the spectral clustering to the adjacency matrix \( A \). The algorithm obtains the label estimates based on the leading
$K$ eigenvectors corresponding to the $K$ largest absolute eigenvalues of $A$. Similar to Rohe et al. (2011), the rationale of the algorithm is that the eigenvectors of the expected adjacency matrix corresponding to the $K$ largest absolute eigenvalues can infer the true labels under the SBM with $K$ communities. Lei and Rinaldo (2015) have shown the consistency of this version of spectral clustering for relatively dense networks under the SBM and the DCSBM where the maximum expected degrees grow no slower than $\log n$. For sparse networks, it has been observed that regular spectral clustering proposed by Rohe et al. (2011) fails to provide correct label estimates due to the presence of extreme low degree nodes (Amini, Chen, et al., 2013). Amini, Chen, et al. (2013) have proposed a regularized spectral clustering algorithm by adding a small constant to each entry of the adjacency matrix. Let $A_\lambda := A + (\lambda/n)1_n1_n^T$, where $\lambda$ is chosen to be the expected degree. The regularized spectral clustering algorithm computes the degree matrix and forms the symmetric normalized graph Laplacian based on regularized adjacency matrix $A_\lambda$, and performs spectral clustering as before mentioned spectral clustering that is based on the normalized graph Laplacian. Recent work of Le et al. (2017) has shown that the above regularization scheme makes the graph Laplacian concentrate even for sparse networks where the maximum expected degree is bounded, and thus the regularized spectral clustering works in this case.

1.3 Estimating Network Edge Probabilities

When a network is generated from an SBM, we can effectively estimate the community-wise edge probability matrix $B$ after we obtain the label estimates of nodes. Let $\hat{g}_1, \cdots, \hat{g}_n$ denote the label estimates of $n$ nodes. Then for any $1 \leq k, l \leq \frac{n}{2}$
K, we can estimate $B_{kl}$ by

$$\hat{B}_{kl} = \frac{\sum_{i<j} A_{ij} \mathbb{I}(\hat{g}_i = k, \hat{g}_j = l)}{\sum_{i<j} \mathbb{I}(\hat{g}_i = k, \hat{g}_j = l)}.$$ 

That is, we estimate the edge probability between a node from community $k$ and a node from community $l$ by the empirical edge probability based on the estimated community structure (Olhede and Wolfe, 2014; Wolfe and Olhede, 2013). In this case, the effectiveness of estimating edge probabilities purely depends on the quality of label estimates. The SBM is certainly a restrictive network model for the task of estimating network edge probabilities. A far more general network model for unlabeled networks can be determined by a symmetric function $f(x, y) : [0, 1]^2 \mapsto [0, 1]$ called a graphon. Same as in community detection, we observe an undirected network containing $n$ nodes. Let $A \in \{0, 1\}^{n \times n}$ denote the symmetric binary adjacency matrix of the network, where $A_{ij} = 1$ if there is an edge between nodes $i$ and $j$ and zero otherwise. Let $P = (p_{ij})_{1 \leq i, j \leq n} \in [0, 1]^{n \times n}$ denote the probability matrix of network edges where $p_{ij}$ is the edge probability between nodes $i$ and $j$. Let us consider the following generating mechanism of a network. Let $f$ be a symmetric measurable function mapping from $[0, 1]^2$ to $[0, 1]$, which is called a graphon. We first generate $\xi_1, \cdots, \xi_n$ independently from Uniform$[0, 1]$. Conditional on $\xi_1, \cdots, \xi_n$, we then generate $A_{ij} \sim \text{Bernoulli}(p_{ij})$ independently, where $p_{ij} = f(\xi_i, \xi_j)$. It has been pointed out by Diaconis and Janson (2007) that graphon $f$ in the above random graph model is not identifiable, since for any measure preserving transformation $\phi : [0, 1] \mapsto [0, 1]$, the graphon $f'(x, y) := f(\phi(x), \phi(y))$ defines the same random graph model as $f$. Given a single realization of the network generated from a smooth graphon function, how can we estimate the matrix of edge probabilities? One way of estimating edge probability matrix $P$ is to estimate graphon function $f$. Wolfe and Olhede (2013)
proposed to approximate the graphon \( f \) by fitting an SBM using profile likelihood methods and estimate graphon \( f \) up to a measure preserving transformation using the corresponding step function approximation. Similarly, Olhede and Wolfe (2014) proposed to estimate graphon \( f \) up to a measure preserving transformation by fitting an SBM with equal-sized communities. Another line of work of estimating \( f \) is to impose assumptions on a graphon such that it is identifiable. According to Chan and Airoldi (2014), strict monotonicity of degrees is the necessary and sufficient condition for a graphon to be identifiable. Specifically, this requires that \( g(u) := \int_0^1 f(u, v)dv \) is strictly increasing or decreasing. Under this assumption, Chan and Airoldi (2014) proposed a sorting-and-smoothing algorithm to estimate a graphon, which first sorts the observed graph according to its empirical degrees and then smooths the sorted graph to obtain an estimate of the graphon. More recently, Zhang et al. (2015) proposed the neighborhood smoothing method to estimate network edge probabilities directly without estimating the graphon \( f \) or \( \xi_i \). For each node \( i \), let \( N_i \) denote the set of its neighbors. Zhang et al. (2013) define a good neighbor node \( i' \) of node \( i \) to have \( f(\xi_{i'}, \cdot) \approx f(\xi_i, \cdot) \), and thus \( P_{i'} \approx P_i \), where \( P_i \) is the \( i \)th row of \( P \). Following this idea, they derived a distance measure between nodes \( i \) and \( i' \):

\[
d^2(i, i') = \max_{k \neq i, i'} |\langle A_i - A_{i'}, A_k \rangle|/n.
\]

Let \( q_i(h) \) denote the \( h \) sample quantile of the set \( \{d(i, i') : i' \neq i\} \). The set of neighbors of node \( i \) is specified to be

\[
N_i = \{i' \neq i : d(i, i') \leq q_i(h)\},
\]

18
where $h$ is a tuning parameter. After the set of neighbors is formed for each of the nodes, the neighborhood smoothing estimates edge probabilities by a local averaging:

$$
\hat{P}_{ij} = \frac{1}{2\left(\frac{\sum_{i' \in N_i} A_{i'j}}{|N_i|} + \frac{\sum_{j' \in N_j} A_{ij'}}{|N_j|}\right)}.
$$

(1.11)

For a family of piecewise Lipschitz graphon functions, Zhang et al. (2015) have shown that the neighborhood smoothing estimator $\hat{P}$ can achieve a mean squared error rate $O_P(\sqrt{\frac{\log n}{n}})$, where the mean squared error is defined as $\frac{1}{n^2}||\hat{P} - P||_F^2$. Hereafter, we use NBS as the abbreviation for the neighborhood smoothing method. As for the theoretical development, Gao et al. (2015) have proved that the minimax rate of estimating the matrix of edge probabilities generated from graphons in a Hölder class with smoothness parameter $\alpha \geq 1$ is $O_P(\frac{\log n}{n})$. This rate can be achieved by estimating edge probabilities using block averages of entries in the adjacency matrix with the optimal block partition given by a least squares criterion. However, solving the least squares problem requires a computationally intractable combinatorial search over all possible cluster assignments. It is still unknown whether the minimax rate can be achieved by a polynomial time algorithm.
Chapter 2: The Semidefinite Program and its General Analysis Framework

In this chapter, we present our semidefinite program for clustering and its general analysis framework, together with an efficient algorithm to solve the SDP. In Section 2.1, we present a unified criterion of clustering based on an input similarity matrix. Relaxing the partitioning problem leads to our SDP. In Section 2.2, we derive optimality conditions of the SDP from a geometric point of view. Our optimality conditions deal with a convex body, called the Fantope. This enables us to analyze the SDP by making use of the properties of the Fantope. Inspired by optimality conditions, we point out a connection between our SDP and spectral clustering. We derive a sufficient condition for consistency of the SDP based on the optimality conditions, which is shown in Section 2.3. Section 2.4 provides an efficient algorithm to solve the proposed SDP.

The following notations will be used hereafter. Let \( \mathcal{S}^n \) denote the set of \( n \times n \) symmetric matrices, \( \mathcal{S}_+^n \) denote the set of \( n \times n \) positive semidefinite matrices, and \( \mathcal{S}_{++}^n \) denote the set of \( n \times n \) positive definite matrices. \( I_n \) denotes the \( n \times n \) identity matrix. \( \mathbf{1}_n \) denotes the \( n \times 1 \) vector of all ones. \( E_n \) and \( E_{m,n} \) denote the \( n \times n \) and the \( m \times n \) matrix of all ones, respectively. We use \( \|x\| \) and \( \|x\|_1 \) to denote the Euclidean norm and the \( \ell_1 \) norm of a vector \( x \), respectively. We use \( \|X\|_F \) and \( \|X\|_2 \)
to denote the Frobenius norm and the spectral norm of a matrix $X$, respectively. For any square matrix $M$, $\text{diag}(M)$ denotes the diagonal matrix obtained by setting all off-diagonal entries of $M$ to 0. For square matrices $M_1, M_2, \cdots, M_K$, we use $\text{diag}\{M_1, M_2, \cdots, M_K\}$ to denote the block diagonal matrix with $k$th diagonal block being $M_k$. If $M$ is a symmetric matrix, $\lambda_j(M)$ denotes the $j$th largest eigenvalue of $M$. For a convex set $C$, $\mathbb{1}_C(x)$ is the indicator function of $C$, which takes value of 0 if $x \in C$ and $\infty$ otherwise.

2.1 Problem Setup and A Semidefinite Program

We consider $n$ objects, and each of these objects belongs to one and only one of the $K$ disjoint clusters. Let $C_k$ be the indices set of objects in the $k$th cluster and $n_k = |C_k|$ be the size of the $k$th cluster. Let $g_i \in \{1, \cdots, K\}$ be the cluster label of object $i$. We call $Z = (Z_{ij})_{1 \leq i \leq n, 1 \leq j \leq K} \in \{0, 1\}^{n \times K}$ the membership matrix if the $i$th row of $Z$ is 1 at column $g_i$ and 0 elsewhere. The goal of clustering is to recover the cluster membership of the objects.

Given a symmetric matrix of pairwise similarities $S = (S_{ij})_{1 \leq i,j \leq n}$ between the different objects, we perform clustering based on the partition criterion of maximizing the sum of average intra-cluster similarities over all clusters, i.e.,

$$\text{maximize } \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{j=1}^{n} Z_{ik}Z_{jk}S_{ij}/n_k,$$

subject to $Z \in \text{set of all membership matrices}$.

In matrix form, this problem is equivalent to

$$\text{maximize } \langle S, X \rangle$$

subject to $X \in \mathcal{X}_{\text{norm}}$. 

21
where $X_{\text{norm}}$ is the set of normalized clustering matrices. By Theorem 2.40 in Galantai (2004), $X_{\text{norm}}$ is equivalent to the set of rank-$K$ doubly stochastic projection matrices.

Thus, the optimization problem is equivalent to

$$\text{maximize} \quad \langle S, X \rangle$$

subject to \quad $X \geq 0$, $X1_n = 1_n$, $X = X^T$, $X^2 = X$, $\text{tr}(X) = K$.

This problem is NP-hard in general. To relax the feasible set to a convex set, we note that the convex hull of the set of rank-$K$ projection matrices is

$$\mathcal{F}^K := \{X : 0 \preceq X \preceq I_n, \text{tr}(X) = K\}$$

called the trace-$K$ Fantope (Dattorro, 2005, Section 2.3.2). Let $B_{\text{sym}}$ denote the set of $n \times n$ symmetric doubly stochastic matrices, i.e.,

$$B_{\text{sym}} := \{X \in S^n : X \geq 0, X1_n = 1_n\}.$$

Then a convex relaxation of the feasible set in (2.1) is $\mathcal{F}^K \cap B_{\text{sym}}$. Thus, we arrive at our semidefinite relaxation of (2.1):

$$\text{maximize} \quad \langle S, X \rangle$$

subject to \quad $X \in \mathcal{F}^K \cap B_{\text{sym}}$.

We call it SDP-BF, since its feasible set is the intersection of the symmetric Birkhoff polytope and the trace-$K$ Fantope. From now on, without loss of generality, we assume that the objects are sorted according to their cluster labels from the first cluster to the last cluster. Then the true clustering matrix in the set of normalized clustering matrices is $X_0 := \text{diag}\{E_{n_1}/n_1, \cdots, E_{n_K}/n_K\}$. The goal of clustering now becomes estimating $X_0$ using a solution to SDP-BF, denoted by $\hat{X}$.

To use SDP-BF to perform clustering in various scenarios, the key is to choose an appropriate pairwise similarity matrix $S$. In Chapter 3, we will consider clustering
under different models using SDP-BF with different input similarity matrices. We first state a proposition showing that using an inner product kernel matrix is equivalent to using its corresponding negative Euclidean distance matrix as an input matrix to SDP-BF.

**Proposition 2.1.** Let $K$ be an $n \times n$ inner product kernel matrix whose $(i, j)$th entry is $K(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle$, where $\phi(\cdot)$ is the feature map corresponding to the kernel $K$. Let $S$ be the negative Euclidean distance matrix whose $(i, j)$th entry is $-||\phi(x_i) - \phi(x_j)||^2$. We have that solving

$$\begin{align*}
\text{maximize} & \quad \langle K, X \rangle \\
\text{subject to} & \quad X \in F^K \cap B_{\text{sym}}
\end{align*}$$

is equivalent to solving

$$\begin{align*}
\text{maximize} & \quad \langle S, X \rangle \\
\text{subject to} & \quad X \in F^K \cap B_{\text{sym}}.
\end{align*}$$

**Proof.** Since $S_{ij} = -||\phi(x_i)||^2 + 2\langle \phi(x_i), \phi(x_j) \rangle - ||\phi(x_j)||^2$, $S$ can be expressed as $2K + \alpha 1_n^T + 1_n \alpha^T$, where $\alpha = -(||\phi(x_1)||^2, \ldots, ||\phi(x_n)||^2)^T$. We notice that adding a $1_n \alpha^T + \alpha 1_n^T$ term to the input matrix has no effect on the solution to SDP-BF due to the constraint $X 1_n = 1_n$. Thus, using the negative Euclidean distance matrix $S$ as an input matrix is equivalent to using the inner product kernel matrix $K$ as an input matrix to SDP-BF.

**Connection to Sparse PCA**

Let $||X||_1 := \sum_{i,j} |X_{ij}|$ denote the $\ell_1$ norm of $X$. We can further relax the constraint $X 1_n = 1_n$ in SDP-BF to be $\langle E_n, X \rangle \leq n$. Since $X \succeq 0$, we have $\langle E_n, X \rangle = ||X||_1$. Then we can further relax the feasible set of SDP-BF to $\{X : X \in F^K, X \succeq 0, ||X||_1 \leq n\}$. By Lagrangian duality, there exists a $\rho \geq 0$ that makes this further
relaxed SDP equivalent to

\[
\text{maximize} \quad \langle S, X \rangle - \rho \|X\|_1 \\
\text{subject to} \quad X \in \mathcal{F}^K, X \geq 0.
\]

Removing the nonnegativity constraint in the above optimization problem leads to the SDP relaxation of sparse PCA based on the Fantope proposed in Vu et al. (2013).

2.2 Optimality Conditions

Before we derive optimality conditions of SDP-BF, we first introduce several well-known definitions and theorems in convex analysis which will be used in the derivation.

**Definition 2.2.** (Section 5.4 in Bertsekas, 2009). Let \( f: \mathbb{R}^n \to (-\infty, \infty] \) be a proper convex function. We say that \( y \in \mathbb{R}^n \) is a subgradient of \( f \) at \( x \in \text{dom}(f) \) if

\[
f(z) \geq f(x) + \langle y, z - x \rangle, \forall z \in \mathbb{R}^n.
\]

The set of all subgradients of \( f \) at \( x \) is called the subdifferential of \( f \) at \( x \) and is denoted by \( \partial f(x) \).

**Definition 2.3.** (Section 5.4 in Bertsekas, 2009). Let \( C \) be a nonempty convex set. For \( x \in C \), the normal cone to \( C \) at \( x \) is defined to be

\[
N_C(x) = \{ s : \langle s, y - x \rangle \leq 0, \forall y \in C \}.
\]

**Lemma 2.4.** (Section 5.4 in Bertsekas, 2009). Let \( \mathbb{I}_C(x) \) be the indicator function of a convex set \( C \), which takes value of 0 if \( x \in C \) and \( \infty \) otherwise. For \( x \in C \), we have that

\[
\partial \mathbb{I}_C(x) = N_C(x).
\]
Lemma 2.5. (Section 5.4 in Bertsekas, 2009) Let $f : \mathbb{R}^n \rightarrow (-\infty, \infty]$ be a proper convex function. By the definition of subdifferential, $\hat{x}$ minimizes $f$ over $\mathbb{R}^n$ if and only if $0 \in \partial f(\hat{x})$.

The next lemma regarding a basic property of the Fantope will be used in the analysis. Its proof can be found in Overton and Womersley (1992).

Lemma 2.6. (Lemma 1 in Lei and Vu, 2015) Let $S$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. Then $\max_{H \in F^K} \langle S, H \rangle = \sum_{i=1}^{k} \lambda_i$ and the maximum is achieved by the projector of a $k$-dimensional principal subspace of $S$. Moreover, the $k$-dimensional principal subspace of $S$ is unique and thus the maximizer is unique if and only if $\lambda_k > \lambda_{k+1}$.

SDP-BF can be written as

$$\min_X \mathbb{1}_{F^K \cap B_{\text{sym}}} (X) - \langle S, X \rangle. \quad (2.3)$$

By Lemmas 2.4 and 2.5, $\hat{X}$ is a solution to (2.3) if and only if

$$0 \in \partial \left( \mathbb{1}_{F^K \cap B_{\text{sym}}} - \langle S, \cdot \rangle \right) (\hat{X})$$

$$= N_{F^K \cap B_{\text{sym}}} (\hat{X}) - S. \quad (2.4)$$

To concretize the normal cone to $F^K \cap B_{\text{sym}}$ in (2.4), we derive the following lemma.

Lemma 2.7. In the ambient space of symmetric matrices,

1. Let $X \in F^K \cap B_{\text{sym}}$. If $K > 1$, then

$$N_{F^K \cap B_{\text{sym}}} (X) = N_{F^K} (X) + N_{B_{\text{sym}}} (X).$$

2. Let $X \in B_{\text{sym}}$. Then $N_{B_{\text{sym}}} (X)$ consists of symmetric matrices of the form

$$\alpha 1^n_T + 1_n \alpha^T - Y,$$
where $\alpha \in \mathbb{R}^n$ and $Y$ is a symmetric matrix with nonnegative entries that equal zero wherever $X$ is nonzero.

3. Let $X \in \mathcal{F}^K$. Then

$$N_{\mathcal{F}^K}(X) = \{ T \in \mathbb{S}^n : X \in \arg \max_{H \in \mathcal{F}^K} \langle T, H \rangle \}.$$  

Moreover, if $T \in N_{\mathcal{F}^K}(X)$ is a symmetric matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_K > \lambda_{K+1} \geq \cdots \geq \lambda_n$, then the right-hand side indicates that $X$ must be the projector (orthogonal projection matrix) of the $K$-dimensional principal subspace of $T$.

Proof. (1) Let $\mathcal{C}^K$ denote the set of trace $K$ matrices. We have

$$N_{\mathcal{F}^K \cap \mathcal{B}_{\text{sym}}}(X) \supseteq N_{\mathcal{F}^K}(X) + N_{\mathcal{B}_{\text{sym}}}(X) \supseteq N_{\mathbb{S}^n_+}(X) + N_{\mathcal{C}^K}(X) + N_{\mathcal{B}_{\text{sym}}}(X).$$ \hfill (2.5)

Note that when $1 < K \leq n, n > 1, \frac{K-1}{n-1} I_n + \frac{n-K}{n(n-1)} E_n \in \mathbb{S}^n_+ \cap \mathcal{C}^K \cap \mathcal{B}_{\text{sym}}$. Since $\text{relint}(\mathbb{S}^n_+) = \mathbb{S}^n_+ \cap \mathcal{C}^K$, and $\text{relint}(\mathcal{B}_{\text{sym}}) = \mathcal{B}_{\text{sym}}$, we have $\text{relint}(\mathbb{S}^n_+) \cap \text{relint}(\mathcal{C}^K) \cap \text{relint}(\mathcal{B}_{\text{sym}}) \neq \emptyset$ and thus

$$N_{\mathbb{S}^n_+}(X) + N_{\mathcal{C}^K}(X) + N_{\mathcal{B}_{\text{sym}}}(X) = N_{\mathbb{S}^n_+ \cap \mathcal{C}^K \cap \mathcal{B}_{\text{sym}}}(X).$$ \hfill (2.6)

The claimed result follows by combining (2.5), (2.6), and the fact that $\mathcal{F}^K \cap \mathcal{B}_{\text{sym}} = \mathbb{S}^n_+ \cap \mathcal{C}^K \cap \mathcal{B}_{\text{sym}}$.

(2) We consider the sets

$$C_0 := \{ X \in \mathbb{S}^n : X \geq 0 \},$$

$$C_1 := \{ X \in \mathbb{S}^n : X 1_n = 1_n \},$$

26
then $B_{\text{sym}} = C_0 \cap C_1$. Note that $\text{relint}(C_0) = C_0$ and $\text{relint}(C_1) = C_1$. Since $\text{relint}(C_0) \cap \text{relint}(C_1) \neq \emptyset$, we have

$$N_{B_{\text{sym}}}(X) = N_{C_0}(X) + N_{C_1}(X)$$

for all $X \in B_{\text{sym}}$. Let us first consider $N_{C_0}(X)$ for a fixed $X \in C_0$. We want to find $Y$ such that $\langle Y, \tilde{X} \rangle \leq \langle Y, X \rangle$ for all $\tilde{X} \in C_0$. First note that we must have $Y \leq 0$, otherwise we can always find $\tilde{X} \geq 0$ such that $\langle Y, \tilde{X} \rangle > \langle Y, X \rangle$. Now we have $\langle Y, X \rangle \leq 0$. If $\langle Y, X \rangle < 0$, we can always choose $\tilde{X} = 0$ which leads to $\langle Y, \tilde{X} \rangle > \langle Y, X \rangle$. Hence, we must have $\langle Y, X \rangle = 0$ which is equivalent to $Y_{ij}X_{ij} = 0, \forall i, j$. Thus

$$N_{C_0}(X) = \{Y \in \mathbb{S}^n : Y \leq 0, Y_{ij}X_{ij} = 0, \forall i, j\}.$$

Next, let us consider $N_{C_1}(X)$ for a fixed $X \in C_1$. We want to find $Z$ such that $\langle Z, \tilde{X} - X \rangle \leq 0$ for all $\tilde{X} \in C_1$. First, note that

$$C_1 = \{PSP + \mathbf{1}_n \mathbf{1}_n^T/n : S \in \mathbb{S}^n\},$$

where $P := I_n - \mathbf{1}_n \mathbf{1}_n^T/n$. To show this fact, the right-hand side is clearly contained in $C_1$. Conversely, for any $X \in C_1$, it can be expressed as $PXP + \mathbf{1}_n \mathbf{1}_n^T/n$ by expanding $(P + P^\perp)X(P + P^\perp)$, where $P^\perp := I_n - P$. Now, our problem reduces to finding $Z$ such that $\langle Z, PSP \rangle \leq 0, \forall S \in \mathbb{S}^n$. If for some $S \in \mathbb{S}^n$, we have $\langle Z, PSP \rangle < 0$, then $\langle Z, P(-S)P \rangle > 0$ and thus $Z \notin N_{C_1}(X)$. Hence, $Z$ must satisfy $\langle Z, PSP \rangle = 0, \forall S \in \mathbb{S}^n$. This implies that the set of $Z$ is the orthogonal complement of $\{PSP : S \in \mathbb{S}^n\}$ in the ambient symmetric matrix space, which is characterized by $\{\alpha \mathbf{1}_n^T + \mathbf{1}_n \alpha^T : \alpha \in \mathbb{R}^n\}$ (see Dattorro, 2005, Section E.7.2). Thus

$$N_{C_1}(X) = \{\alpha \mathbf{1}_n^T + \mathbf{1}_n \alpha^T : \alpha \in \mathbb{R}^n\}.$$
The result follows by combining the previous arguments.

(3) follows by the definition of the normal cone and Lemma 2.6.

Combining (2.4) and Lemma 2.7, we arrive at the optimality condition of SDP-BF which we summarize in Theorem 2.8.

**Theorem 2.8.** Let $\hat{X} \in F^K \cap B_{\text{sym}}$. Then $\hat{X}$ is a solution to SDP-BF if and only if

$$S \in N_{F^K}(\hat{X}) + N_{B_{\text{sym}}}(\hat{X}),$$

where

$$N_{B_{\text{sym}}}(\hat{X}) = \{\alpha 1_n^T + 1_n \alpha^T - Y : \alpha \in \mathbb{R}^n, Y \in S^n, Y \succeq 0, Y_{ij} \hat{X}_{ij} = 0, \forall i, j\},$$

$$N_{F^K}(\hat{X}) = \{T \in S^n : \hat{X} \in \arg\max_{H \in F^K} \langle T, H \rangle\}.$$ 

In other words, $\hat{X} \in F^K \cap B_{\text{sym}}$ is a solution to SDP-BF if and only if there exists a pair $(\hat{\alpha}, \hat{Y})$ satisfying the conditions of $N_{B_{\text{sym}}}(\hat{X})$ such that

$$\hat{X} \in \arg\max_{H \in F^K} \{S - (\hat{\alpha}1_n^T + 1_n \hat{\alpha}^T - \hat{Y}), H\}. \quad (2.7)$$

From the primal-dual optimality point of view, the pair $(\hat{\alpha}, \hat{Y})$ is a solution to the dual problem

$$\text{minimize} \quad \sum_{i=1}^K \lambda_i (S - \alpha 1_n^T - 1_n \alpha^T + Y) + 2 \alpha^T 1_n \quad (2.8)$$

subject to $Y \succeq 0$,

where $\alpha \in \mathbb{R}^n$ and $Y \in S^n$. The discussion leads to the following corollary.
Corollary 2.9. \( \hat{X} \) and \( (\hat{\alpha}, \hat{Y}) \) is a pair of primal optimal solution to SDP-BF and dual optimal solution to (2.8) if and only if

\[
\hat{X} \in \mathcal{F}^K \cap \mathcal{B}_{sym}, \tag{2.9}
\]
\[
\hat{Y} \in \mathbb{S}^n, \hat{Y} \geq 0, \tag{2.10}
\]
\[
\hat{X} \in \arg \max_{H \in \mathcal{F}^K} \{ S - (\hat{\alpha}1_n^T + 1_n\hat{\alpha}^T) + \hat{Y}, H \}, \tag{2.11}
\]
\[
\hat{Y}_{ij}\hat{X}_{ij} = 0, \forall i, j. \tag{2.12}
\]

Corollary 2.9 implies that given a dual optimal solution \( (\hat{\alpha}, \hat{Y}) \), we can recover all primal optimal solutions to SDP-BF through (2.9)–(2.12).

**Connection to Spectral Clustering**

(2.7) leads to a connection between SDP-BF and spectral clustering. Let \( \tilde{S} := S - (\hat{\alpha}1_n^T + 1_n\hat{\alpha}^T - \hat{Y}) \). By Lemma 2.6, given a dual optimal solution \( (\hat{\alpha}, \hat{Y}) \) to (2.8), if \( \lambda_K(\tilde{S}) > \lambda_{K+1}(\tilde{S}) \), then (2.7) indicates that the solution to SDP-BF is the projector of the \( K \)-dimensional principal subspace of \( \tilde{S} \) and is unique. This implies that SDP-BF can be considered as spectral clustering operating on an adjusted input matrix and its solution may have a regularization effect provided by the symmetric doubly stochastic constraint compared to the solution of spectral clustering operating on the input matrix.

### 2.3 A Sufficient Condition for Fisher Consistency of the SDP

We say that SDP-BF is Fisher consistent if \( X_0 \) is the unique solution to SDP-BF when \( \mathbb{E}S \) is used as the input matrix. Using optimality conditions derived in Section 2.2, we are able to find sufficient conditions for Fisher consistency of SDP-BF under various input matrices. The following notations will be used hereafter. For
an $n \times n$ matrix $M$, we let $M_{C_kC_l}$ denote the submatrix of $M$ induced on indices $C_k \times C_l, \forall k \neq l$ and $M_{C_k}$ denote the submatrix of $M$ induced on indices $C_k \times C_k, \forall k$.

Recall that we sort objects according to their cluster labels from the first cluster to the last cluster without loss of generality. Thus $M_{C_k}$ and $M_{C_kC_l}$ are the $k$th diagonal block and the $(k,l)$th off-diagonal block of partitioned $M$ according to the cluster sizes $n_1, \cdots, n_K$. We have the following theorem for consistency of SDP-BF, which will be used frequently in the derivation of Fisher consistency under various models.

**Theorem 2.10.** Consider an input matrix $S \in S^n$ where $S_{C_k} = a_k E_{n_k} + c_k I_{n_k}$. Let $b_k \geq S_{ij}, \forall i \in C_k, j \in C_l, k \neq l$. Write $c_{\text{max}} = \max_k c_k$. If

\[
n_k(a_k - b_k) > c_{\text{max}} - c_k, \forall k = 1, \cdots, K,
\]

then $X_0$ is the unique solution to SDP-BF.

**Proof.** We choose $\alpha = (\alpha_1 1_{n_1}^T, \cdots, \alpha_K 1_{n_K}^T)^T$ where $\alpha_k := b_k/2$. Then the $(k,l)$th off-diagonal block of $S - 1_n \alpha^T - \alpha 1_n^T$ is $S_{C_kC_l} - \frac{b_k+b_l}{2} E_{n_k,n_l}$. By the choice of $b_k$ and $b_l$, we have $S_{C_kC_l} - \frac{b_k+b_l}{2} E_{n_k,n_l} \leq 0$. Thus, we can choose $Y_{C_kC_l} \geq 0$ to make $(k,l)$th off-diagonal block of $S - 1_n \alpha^T - \alpha 1_n^T + Y$ be a zero matrix. The above argument holds for any $k \neq l$. Note that $Y_{C_k} = 0, \forall k$ by the definition of $X_0$ and (2.12).

The $k$th diagonal block of $S - 1_n \alpha^T - \alpha 1_n^T + Y$ becomes $(a_k - b_k) E_{n_k} + c_k I_{n_k}$. Its eigenvalues are $(a_k - b_k) n_k + c_k$ with multiplicity 1 and $c_k$ with multiplicity $n_k - 1$.

Write $\tilde{S} = S - 1_n \alpha^T - \alpha 1_n^T + Y$. Since $\min_k ((a_k - b_k) n_k + c_k) > c_{\text{max}}$, the leading $K$ eigenvalues of $\tilde{S}$ are $(a_k - b_k) n_k + c_k, k = 1, \cdots, K$, and $\lambda_K(\tilde{S}) > \lambda_{K+1}(\tilde{S})$. Moreover, the projector of the $K$-dimensional principal subspace of $\tilde{S}$ is $X_0$. By (2.11), $X_0$ is the unique solution to SDP-BF. \qed
Remark 2.11. In the statement of Theorem 2.10, we require that $b_k$ is an upper bound of $S_{ij}, \forall i \in C_k, j \in C_l, k \neq l$. In practice, to satisfy the sufficient condition (2.13) and to make the spectral gap $\lambda_K(\tilde{S}) - \lambda_{K+1}(\tilde{S})$ large, we wish to choose $b_k$ as small as possible.

2.4 An Efficient Algorithm for Solving the SDP

In this section, we derive an efficient algorithm for solving SDP-BF. The main part of the algorithm is an alternating direction method of multipliers (ADMM) algorithm. ADMM is now widely used in many sparse and low-rank optimization problems (e.g., Amini and Levina, 2014; Cai and Li, 2015; Chen, Li, et al., 2016; Vu et al., 2013). In current studies of SDP approaches for cluster analysis, ADMM algorithms for solving SDPs require a full spectral decomposition of a symmetric matrix in each iteration of ADMM (Amini and Levina, 2014; Cai and Li, 2015; Chen, Li, et al., 2016). In contrast, our ADMM algorithm only requires to compute the leading $K - 1$ eigenvectors of symmetric matrices in each iteration of ADMM. When the number of clusters $K$ is small, our algorithm is more scalable to the case where the number of objects $n$ is large compared to other state-of-the-art algorithms.

2.4.1 The ADMM Algorithm

Let $C := \{Y : Y \in \mathcal{F}^K, Y 1_n = 1_n\}$. We write SDP-BF as an equivalent equality constrained problem:

$$\begin{align*}
\text{minimize} & \quad -\langle S, X \rangle + \mathbb{I}_{X \succeq 0}(X) + \mathbb{I}_C(Y) \\
\text{subject to} & \quad X = Y.
\end{align*}$$

The augmented Lagrangian associated with (2.14) has the form

$$L_\rho(X, Y, U) = -\langle S, X \rangle + \mathbb{I}_{X \succeq 0}(X) + \mathbb{I}_C(Y) + \frac{\rho}{2} \|X - Y + U\|_F^2 - \frac{\rho}{2} \|U\|_F^2,$$

31
where \( U = \frac{1}{\rho} \Lambda \) is the scaled ADMM dual variable and \( \rho > 0 \) is the ADMM penalty parameter (see Boyd et al., 2010, Section 3.1). We first minimize \( L_\rho \) with respect to \( X \), which is equivalent to computing

\[
\arg\min_{X \geq 0} \|X - (Y - U + \frac{1}{\rho}S)\|_F^2 = \max(Y - U + \frac{1}{\rho}S, 0).
\]

Next, we minimize \( L_\rho \) with respect to \( Y \). This is equivalent to computing

\[
P_C(X + U) = \arg\min_{Y \in C} \frac{1}{2}\|Y - (X + U)\|_F^2,
\]

where \( P_C(\cdot) \) is the Euclidean projection onto the set \( C \). As for the dual variable update, in each ADMM iteration, \( U \) is updated to \( U + X - Y \). The complete ADMM algorithm is summarized in Algorithm 1.

**Algorithm 1** ADMM algorithm for solving SDP-BF

- **Initialization:** \( Y_0 = 0, U_0 = 0, \rho = 10 \)
- repeat \( t = 0, 1, 2, \ldots \)
  1. \( X_{t+1} = \max(Y_t - U_t + \frac{1}{\rho}S, 0) \)
  2. \( Y_{t+1} = P_C(X_{t+1} + U_t) \)
  3. \( U_{t+1} = U_t + X_{t+1} - Y_{t+1} \)
- until \( \max(||X_t - Y_t||_F, ||Y_t - Y_{t-1}||_F) < \epsilon \)
- return \( Y_t \)

### 2.4.2 Frank-Wolfe Algorithm for Solving the Subproblem

In each ADMM iteration, we need to solve a subproblem (2.15) given the current updates of \( X \) and \( U \). We use Frank-Wolfe algorithm (Frank and Wolfe, 1956) to solve it. Roughly speaking, Frank-Wolfe algorithm solves a constrained convex minimization problem by sequentially moving towards the minimizer of the linear approximation of the objective function at the current update. A revisit of Frank-Wolfe
algorithm for sparse convex optimization can be found in Jaggi (2013). To apply
Frank-Wolfe algorithm to solving (2.15), we first derive the following lemma.

**Lemma 2.12.** Let \(D := \{Z : Z \in \mathcal{F}^{K^{-1}}, Z1_n = 0\}\). Then
\[
C = \frac{E_n}{n} + D.
\]

**Proof.** For any \(Y \in C\), since \((1, 1_n)\) is an eigenvalue-eigenvector pair of \(Y\), the spectral
decomposition of \(Y\) can be written as \(Y = \frac{1_n 1_n^T}{n} + Z\). The eigenvalue-eigenvector pairs
of \(Z\) are the same as those of \(Y\) except that \((1, 1_n)\) eigenvalue-eigenvector pair of \(Y\)
becomes \((0, 1_n)\) eigenvalue-eigenvector pair of \(Z\). Since \(0 \preceq Y \preceq I\) and \(\text{tr}(Y) = K\),
we have that \(0 \preceq Z \preceq I\), \(\text{tr}(Z) = K - 1\), and \(Z1_n = 0\), i.e., \(Z \in D\). The argument
can be reversed to show that for any \(Z \in D\), \(Z + \frac{1_n 1_n^T}{n} \in C\). 

By Lemma 2.12, to solve (2.15), it suffices to solve
\[
\min_{Z \in D} f(Z), \tag{2.16}
\]
where \(f(Z) := \frac{1}{2} \|Z - (X + U - \frac{E_n}{n})\|_F^2\). We apply Frank-Wolfe algorithm to solving
(2.16) iteratively. At current update \(Z_k \in D\), we find the updating direction
\[
S_k = \arg \min_{S \in D} \langle S, \nabla f(Z_k) \rangle,
\]
which is equivalent to solving
\[
S_k = \arg \max_{S \in D} \langle S, X + U - \frac{E_n}{n} - Z_k \rangle. \tag{2.17}
\]
Then we choose the best step size using line search. That is, we choose the step size
\[
\alpha_k := \min_{\alpha \in [0, 1]} f(Z_k + \alpha (S_k - Z_k))
\]
\[
= \min(\max(0, \frac{\langle S_k - Z_k, X + U - \frac{E_n}{n} - Z_k \rangle}{\|S_k - Z_k\|_F^2}), 1). \tag{2.18}
\]
Finally, we update $Z_{k+1} = Z_k + \alpha_k(S_k - Z_k)$. To solve (2.17), we need the following lemma.

**Lemma 2.13.** Let $W \in \mathbb{R}^{n \times (n-1)}$ be a matrix whose columns form an orthonormal set of basis of $R(1_n)^\perp$, i.e., $WW^T = I_n - \frac{E_n}{n}, W^T W = I_{n-1}$. Then the following two sets are equivalent:

$$D := \{ Z : Z \in \mathcal{F}^{K-1}, Z 1_n = 0 \},$$

$$\bar{D} := \{ W \tilde{Z} W^T : \tilde{Z} \in S^{n-1}, \tilde{Z} \in \mathcal{F}^{K-1} \}.$$

**Proof.** For any $W \tilde{Z} W^T$, by definition of $W$, we have that $W \tilde{Z} W^T 1_n = 0$. Moreover, by definition of $W$, $W \tilde{Z} W^T$ has the same eigenvalues as $\tilde{Z}$ with an additional eigenvalue of 0. Since $\tilde{Z} \in \mathcal{F}^{K-1}$, we have that $W \tilde{Z} W^T \in \mathcal{F}^{K-1}$. Thus $W \tilde{Z} W^T \in D$. This shows that $\bar{D} \subset D$. On the other hand, for any $Z \in D$, since $Z 1_n = 0$, $(0, 1_n)$ is an eigenvalue-eigenvector pair of $Z$. Discarding the $(0, 1_n)$ eigenvalue-eigenvector pair of $Z$, we can express the spectral decomposition of $Z$ as $U \Lambda U^T$, where the columns of $U \in \mathbb{R}^{n \times (n-1)}$ forms the remaining $n - 1$ eigenvectors of $Z$ and the diagonal entries of the diagonal matrix $\Lambda \in \mathbb{R}^{(n-1) \times (n-1)}$ are the corresponding eigenvalues. Since $U^T U = I_{n-1}$ and $U^T 1_n = 0$, the columns of $U$ form an orthonormal set of basis of $R(1_n)^\perp$. Thus, there exists an orthonormal matrix $O \in \mathbb{R}^{(n-1) \times (n-1)}$ such that $U = W O$. Let $\tilde{Z} = O \Lambda O^T$, then $Z = W \tilde{Z} W^T$. Since $Z \in \mathcal{F}^{K-1}$, we have that $\Lambda \in \mathcal{F}^{K-1}$, and thus $\tilde{Z} \in \mathcal{F}^{K-1}$. This shows that $D \subset \bar{D}$. 

By Lemmas 2.6 and 2.13, the solution to (2.17) is

$$S_k = W \arg \max_{\tilde{S} \in \mathcal{F}^{K-1}} \langle \tilde{S}, W^T (X + U - \frac{E_n}{n} - Z_k) W \rangle W^T$$

$$= WV_k V_k^T W^T, \quad (2.19)$$

34
where the columns of \( V_k \in \mathbb{R}^{(n-1) \times (K-1)} \) are the leading \( K-1 \) eigenvectors of \( W^T(X + U - \frac{E_n}{n} - Z_k)W \). The complete Frank-Wolfe algorithm for solving (2.15) is summarized in Algorithm 2.

**Algorithm 2** Frank-Wolfe algorithm for solving the subproblem (2.15) in ADMM

**Input:** \( W \in \mathbb{R}^{n \times (n-1)} \) whose columns form an arbitrary orthonormal basis of \( R(1_n)^\perp \), \( \text{niter} = 10 \)

**Initialize:** \( Z_1 \in D \)

for \( k = 1 \) to \( \text{niter} \) do

- Compute \( V_k \) whose columns are the leading \( K-1 \) eigenvectors of \( W^T(X + U - \frac{E_n}{n} - Z_k)W \).
- Let \( S_k = WV_kV_k^TW \).
- Let \( \alpha_k = \min(\max(0, \frac{(S_k-Z_k)(X+U-\frac{E_n}{n}-Z_k)}{||S_k-Z_k||_F^2}), 1) \).
- Set \( Z_{k+1} = Z_k + \alpha_k(S_k - Z_k) \).

**return** \( Z_k + \frac{E_n}{n} \)

### 2.4.3 A Toy Example

In this section, we provide an example of community detection using SDP-BF. Consider a network with 60 nodes and 2 communities. The size of the first community is 20 and the size of the second community is 40. The network is generated from an SBM with community-wise edge probability matrix \( B = \begin{pmatrix} 0.4 & 0.1 \\ 0.1 & 0.6 \end{pmatrix} \). The observed adjacency matrix is shown in Figure 2.1a. We cannot see the community structure hidden in the network from the adjacency matrix under a random ordering of nodes. Using the adjacency matrix as the input matrix to SDP-BF, we solve the SDP-BF using the proposed algorithm. The solution matrix to SDP-BF is shown in Figure 2.1b. We see that there are roughly two types of rows in the solution matrix corresponding to two communities. We then apply a \( k \)-means clustering algorithm to rows of the solution matrix produced by SDP-BF to obtain explicit label estimates.
of nodes. Finally we order nodes according to their estimated labels from the first community to the second community and permute the adjacency matrix according to this ordering. The resulting permuted adjacency matrix is shown in Figure 2.1c. We see that our algorithm successfully detects the community structure in the observed network.

![Graphs showing observed adjacency matrix, solution matrix of SDP, and permuted adjacency matrix.](image)

Figure 2.1: Left panel: observed adjacency matrix. Middle panel: solution matrix of SDP-BF. Right panel: permuted adjacency matrix according to estimated labels obtained from the solution matrix.
Chapter 3: Network Community Detection and its Connection to Multivariate Data Clustering

Derived from a unified clustering criterion, SDP-BF works for both multivariate data clustering and network community detection with appropriately chosen input similarity matrices. It turns out that when the input matrix $S$ is chosen to be the Gram matrix for multivariate observations, SDP-BF is equivalent to an SDP relaxation of the $k$-means problem (Awasthi et al., 2015; Iguchi et al., 2015; Mixon et al., 2016; Peng and Xia, 2005). On the other hand, the majority of existing SDP methods for network community detection are motivated by the likelihood maximization problem under the SBM and rely on SDP relaxations of the set of clustering matrices (Amini and Levina, 2014; Cai and Li, 2015; Guédon and Vershynin, 2016). Since SDP-BF relies on an SDP relaxation of the set of normalized clustering matrices, this naturally raises the question about the performance of SDP-BF applied to network community detection compared to existing SDPs. Are there any differences in performance of network community detection between SDP-BF and other SDPs under various scenarios? Furthermore, all existing SDPs for community detection use the adjacency matrix as the input matrix (Amini and Levina, 2014; Cai and Li, 2015; Chen, Li, et al., 2016; Guédon and Vershynin, 2016). However, the adjacency matrix is not the only choice of the similarity matrix for nodes in networks. Motivated from...
the connection between SDP-BF and spectral clustering, we consider using the symmetric normalized graph Laplacian as the input matrix to SDP-BF. We also consider using the squared adjacency matrix $A^2$ as the input matrix to SDP-BF. This leads to a connection between network community detection and multivariate data clustering. The rest of the chapter is organized as follows. In Section 3.1, we show Fisher consistency of SDP-BF using different input matrices under various network models. A connection between network community detection and multivariate data clustering is discussed in Section 3.2. In Section 3.3, we show the exact recovery property of SDP-BF under strongly assortative SBMs when the input matrix is chosen to be the adjacency matrix. In Section 3.4, we conduct simulation studies under various model setups and compare SDP-BF with other state-of-the-art SDPs and spectral methods for community detection. Finally, we test all methods on two real Facebook collegiate networks and results show that SDP-BF with the symmetric normalized graph Laplacian as the input matrix achieves the lowest misclassification rates.

3.1 Population Analysis

In this section, we study Fisher consistency of SDP-BF using different input matrices under various network models. In Section 3.1.1, we show Fisher consistency of SDP-BF when the adjacency matrix is used as the input matrix under assortative SBMs. In Section 3.1.2, we show Fisher consistency of SDP-BF when the symmetric normalized graph Laplacian is used as the input matrix under balanced assortative planted partition models. In Section 3.1.3, we introduce a $k$-means type model for network data where there are $K$ distinct rows in the edge probability matrix corresponding to $K$ communities. It includes the classic SBM as a special case. We show
that, when the minimum distance among $K$ distinct rows of the edge probability matrix is sufficiently large, SDP-BF is Fisher consistent when the squared adjacency matrix is used as the input matrix. We then provide an example of SBM that generates mixed networks with both assortative and disassortative communities. We show that for this example, SDP-BF with the squared adjacency matrix as the input is guaranteed to be Fisher consistent, while SDP-BF with the adjacency matrix as the input fails to produce the true clustering solution.

### 3.1.1 Adjacency Matrix

We consider a slightly generalized version of the classic SBM where the edge probability matrix $P$ is block-wise constant for within-community edge probabilities and can take arbitrary probabilities for between-community edge probabilities. Specifically, we assume

$$p_{ij} = p_k, \forall i, j \in C_k,$$

and

$$q_k := \max_{i \in C_k, j \in C_l, l \neq k} p_{ij}.$$  

(3.1)

Write $p_{\min} = \min_k p_k, q_{\max} = \max_k q_k$. Under the SBM (3.1), let us consider using the adjacency matrix $A$ as the input matrix to SDP-BF. Under the population setting, $\bar{A} := \mathbb{E} A = P - \text{diag}(P)$ is used as the input matrix to SDP-BF.

**Strongly Assortative SBM**

Under a strongly assortative SBM, we assume that $p_{\min} > q_{\max}$. We have the following proposition for Fisher consistency of SDP-BF.

**Proposition 3.1.** Under the SBM (3.1) with the strong assortativity condition $p_{\min} > q_{\max}$, let $\bar{A}$ be the input matrix to SDP-BF. Then $X_0$ is the unique optimal solution to the SDP.
Proof. Since $\bar{A} = P - \text{diag}(P)$, we have that
\[
\bar{A}_{ij} = p_{ij}, \forall i \in C_k, j \in C_l, k \neq l,
\]
\[
\bar{A}_{ij} = p_{k}, \forall i, j \in C_k, i \neq j,
\] (3.2)
\[
\bar{A}_{ii} = 0, \forall i.
\]
Under Theorem 2.10, this corresponds to $a_k = p_{k}, c_k = -p_{k}$. We choose $b_k = q_{\text{max}}, \forall k$. By Theorem 2.10, a sufficient condition for consistency is that $n_k(p_k - q_{\text{max}}) > p_k - p_{\text{min}}, \forall k = 1, \cdots, K$. This in turn can be implied by the strong assortativity condition $p_{\text{min}} > q_{\text{max}}$. □

Weakly Assortative SBM

Under a weakly assortative SBM, we assume that $p_k > q_k, \forall k = 1, \cdots, K$. The following proposition shows that for weakly assortative SBMs, SDP-BF is Fisher consistent when community sizes are sufficiently large.

**Proposition 3.2.** Under the SBM (3.1) with the weak assortativity condition $p_k > q_k, \forall k$, let $\bar{A}$ be the input matrix to SDP-BF. If
\[
n_k(p_k - q_k) > p_k - p_{\text{min}}, \forall k = 1, \cdots, K,
\]
then $X_0$ is the unique optimal solution to the SDP.

Proof. Since $\bar{A}$ takes the forms of (3.2), under Theorem 2.10, this corresponds to $a_k = p_k, c_k = -p_k$. We choose $b_k = q_k, \forall k$. The claimed sufficient condition follows from Theorem 2.10. We note that by weak assortativity condition, we have $p_k > q_k, \forall k$. Thus, for fixed edge probabilities, when the community sizes are large enough, the claimed sufficient condition always holds. □
### 3.1.2 Symmetric Normalized Graph Laplacian

By (2.7), SDP-BF can be considered as spectral clustering operating on an adjusted input matrix. When the adjacency matrix is used as the input matrix, SDP-BF relates to the spectral clustering algorithm operating on the adjacency matrix. Recall that spectral clustering can also operate on the symmetric normalized graph Laplacian \( L := D^{-1/2}AD^{-1/2} \). This version of spectral clustering together with the connection between SDP-BF and spectral clustering motivates us to consider using the symmetric normalized graph Laplacian as the input matrix to SDP-BF. We have that \( L_{ii} = 0, \forall i \). For all \( i \neq j \), \( L_{ij} = 0 \), if \( A_{ij} = 0 \) and \( L_{ij} = 1/(\sqrt{d_i} \sqrt{d_j}) \), if \( A_{ij} = 1 \).

Under the population setting, \( L := E L \) is used as the input matrix to SDP-BF. We have that for all \( i \neq j \),

\[
L_{ij} = E(E(L_{ij}|A_{ij}))
\]

\[
= E(L_{ij}|A_{ij} = 0) P(A_{ij} = 0) + E(L_{ij}|A_{ij} = 1) P(A_{ij} = 1)
\]

\[
= E(L_{ij}|A_{ij} = 0) P(A_{ij} = 0) + E(L_{ij}|A_{ij} = 1) P(A_{ij} = 1)
\]

\[
= E\left(\frac{1}{\sqrt{1 + \sum_{k \neq j} A_{ik}} \sqrt{1 + \sum_{k \neq i} A_{jk}}}\right) P(A_{ij} = 1)
\]

\[
= E\left(\frac{1}{\sqrt{1 + \sum_{k \neq j} A_{ik}}} \right) E\left(\frac{1}{\sqrt{1 + \sum_{k \neq i} A_{jk}}}\right) P(A_{ij} = 1).
\]

The next proposition shows Fisher consistency of SDP-BF when the symmetric normalized graph Laplacian is used as the input matrix under balanced assortative planted partition models.

**Proposition 3.3.** Consider a balanced assortative planted partition model where within-community edge probabilities are \( p \) and between-community edge probabilities are \( q \), with \( p > q \). Let \( \bar{L} \) be the input matrix to SDP-BF. Then \( X_0 \) is the unique optimal solution to the SDP.
Proof. It is clear that $L$ is a block constant matrix with diagonal entries being zero. Furthermore, the entries in the diagonal blocks are the same, and the entries in the off-diagonal blocks are the same. By Theorem 2.10 and the expression of $L_{ij}$, it suffices to show that

$$
\mathbb{E}(1 + \sum_{k \neq j} A_{ik})^{-1/2} > \mathbb{E}(1 + \sum_{k \neq j'} A_{i'k})^{-1/2},
$$

(3.3)

where nodes $i$ and $j$ belong to the same community and nodes $i'$ and $j'$ belong to different communities. Let $U := \sum_{k \neq j} A_{ik}$ and $V := \sum_{k \neq j'} A_{i'k}$. Denote the community size by $r$. Note that $U \overset{d}{=} X + Y$ and $V \overset{d}{=} X + Z$, where $X \sim \text{Binomial}(r - 2, p) + \text{Binomial}(n - r - 1, q)$, $Y \sim \text{Bernoulli}(q)$ and $Z \sim \text{Bernoulli}(p)$. Since

$$
\mathbb{P}(U = u) = \mathbb{P}(X = u - 1) \mathbb{P}(Y = 1) + \mathbb{P}(X = u) \mathbb{P}(Y = 0) = p \mathbb{P}(X = u - 1) + (1 - p) \mathbb{P}(X = u),
$$

for all $u = 0, \ldots, n - 2$, we have that

$$
\mathbb{E}(1 + U)^{-1/2} = \sum_{u=0}^{n-2} (1 + u)^{-1/2}(q \mathbb{P}(X = u - 1) + (1 - q) \mathbb{P}(X = u))
$$

$$
= q \sum_{u=0}^{n-2} (1 + u)^{-1/2} \mathbb{P}(X = u - 1) + (1 - q) \sum_{u=0}^{n-2} (1 + u)^{-1/2} \mathbb{P}(X = u)
$$

$$
= qa + (1 - q)b,
$$

where $a := \sum_{u=0}^{n-2}(1 + u)^{-1/2} \mathbb{P}(X = u - 1)$ and $b := \sum_{u=0}^{n-2}(1 + u)^{-1/2} \mathbb{P}(X = u)$. Similarly,

$$
\mathbb{E}(1 + V)^{-1/2} = p \sum_{v=0}^{n-2} (1 + v)^{-1/2} \mathbb{P}(X = v - 1) + (1 - p) \sum_{v=0}^{n-2} (1 + v)^{-1/2} \mathbb{P}(X = v)
$$

$$
= pa + (1 - p)b.
$$

Since $p > q$, (3.3) is equivalent to $a < b$. Note that

$$
a = \sum_{u=0}^{n-3}(u + 2)^{-1/2} \mathbb{P}(X = u),
$$

42
and
\[ b = \sum_{u=0}^{n-3} (u + 1)^{-1/2} \mathbb{P}(X = u). \]

It is clear that \( a < b \). This completes the proof.

### 3.1.3 Squared Adjacency Matrix

For network data, the adjacency matrix and the graph Laplacian are two commonly used matrix representations. However, if we think of the \( i \)th row of the adjacency matrix as a feature vector of the \( i \)th node, we can treat a network dataset as a multivariate dataset where a node is an observation and whether or not nodes are connected to a given node is a variable. If we view the adjacency matrix in this way, it is natural to use the Gram matrix \( AA^T \) induced by the adjacency matrix as a similarity matrix, which is equal to \( A^2 \) for undirected graphs. Recall that by Proposition 2.1, this choice of the input matrix to SDP-BF is equivalent to using the negative Euclidean distance matrix.

Under what model and conditions can we guarantee Fisher consistency of the SDP with \( A^2 \) as the input matrix? Let us consider a \( k \)-means type model for network data. We assume that there are \( K \) distinct rows \( P_1, \cdots, P_K \) in the edge probability matrix \( P \) corresponding to \( K \) communities, where \( P_k \) is the vector of edge probabilities for a node in community \( k \). We notice that the classic SBM satisfies this assumption. For notational convenience, we do not exclude self-loops under this model. Let \( \overline{A^2} := \mathbb{E}(A^2) \). Then we have that
\[
\overline{A^2}_{ij} = \langle P_k, P_l \rangle, i \in C_k, j \in C_l, k \neq l,
\]
\[
\overline{A^2}_{ij} = ||P_k||^2, i, j \in C_k, i \neq j,
\]
\[
\overline{A^2}_{ii} = ||P_k||_1, i \in C_k.
\]
We have the following sufficient condition for Fisher consistency.

**Proposition 3.4.** Under the above defined network model, let $\overline{A}^2$ be the input matrix to SDP-BF. Define $d_k := \|P_k\|_1 - \|P_k\|^2$, $\forall k = 1, \cdots, K$. Write $d_{\text{max}} = \max_k d_k$. If

$$\frac{1}{2} \min_{l \neq m} \|P_l - P_m\|^2 > \frac{d_{\text{max}} - d_k}{n_k}, \forall k = 1, \cdots, K,$$

then $X_0$ is the unique optimal solution to the SDP.

**Proof.** Due to the constraint $X1_n = 1_n$, using $\overline{A}^2$ as input matrix is equivalent to using $\widetilde{A}^2$, where

$$\widetilde{A}^2_{ij} = -\frac{1}{2} \|P_k - P_l\|^2, i \in C_k, j \in C_l, k \neq l,$$

$$\widetilde{A}^2_{ij} = 0, i, j \in C_k, i \neq j,$$

$$\widetilde{A}^2_{ii} = d_k, i \in C_k.$$

Under Theorem 2.10, this corresponds to $a_k = 0, c_k = d_k$. We choose $b_k = -\frac{1}{2} \min_{l \neq m} \|P_l - P_m\|^2, \forall k = 1, \cdots, K$. The claimed sufficient condition follows from Theorem 2.10. ~\Box

**Example 3.5.** Let us consider a balanced SBM with community-wise edge probability matrix $B \in [0,1]^{K \times K}$. Each community has equal size $m$. We denote the $k$th row of $B$ by $B_k$. In this case, the sufficient condition (3.4) becomes

$$\frac{m}{2} \min_{l \neq m} \|B_l - B_m\|^2 > \max_k (\|B_k\|_1 - \|B_k\|^2) - \min_k (\|B_k\|_1 - \|B_k\|^2).$$

(3.5)

As an example, consider a balanced SBM with three communities, each of size 20. The community-wise edge probability matrix

$$B = \begin{pmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.05 & 0.2 \\ 0.1 & 0.2 & 0.3 \end{pmatrix}$$

44
We note that this model will create mixed networks with the first and the third communities to be assortative and the second community to be disassortative. Let $\overline{A}$ and $\overline{A}^2$ denote $\mathbb{E}(A)$ and $\mathbb{E}(A^2)$, respectively. One can check that the sufficient condition (3.5) holds, and thus $X_0$ is the unique optimal solution to SDP-BF when $\overline{A}^2$ is used as the input matrix. Figures 3.1a and 3.1b show the solution matrix of SDP-BF when $\overline{A}$ and $\overline{A}^2$ are used as the input matrix, respectively. We see that when $\overline{A}$ is used as the input matrix, the solution fails to identify the second and the third community, while using $\overline{A}^2$ yields the true clustering solution. Since using the adjacency matrix as the input matrix is not consistent under the population setting, it has no hope to yield good solution under the sample setting. On the other hand, we expect that using squared adjacency matrix as the input matrix in this case can yield good clustering solution. In Section 3.4 we compare the performance of various methods under this setup.

### 3.2 Connection to Multivariate Data Clustering

Using squared adjacency matrix as the input matrix to SDP-BF leads to a connection between network community detection and multivariate data clustering. Let us consider the following $k$-means probability model for a set of $p$-dimensional data vectors, $x_i, i = 1, \ldots, n$, with $K$ clusters:

$$x_i = \mu_{g_i} + \varepsilon_i, i = 1, \ldots, n,$$

where $g_i \in \{1, \cdots, K\}$ is the cluster label of $i$th observation, $\mu_k$ is the $k$th cluster center, and $\varepsilon_i$ measures the variation of $i$th observation from its cluster center. We assume that $\varepsilon_i$’s are independent with zero means and the cluster centers $\mu_1, \cdots, \mu_K$ are different. We further assume that for observations belonging to the same cluster,
Figure 3.1: Left panel: the solution matrix of the SDP when the expected adjacency matrix is used as the input matrix. Right panel: the solution matrix of the SDP when the expected squared adjacency matrix is used as the input matrix.

their variations from the cluster center follow the same distribution, i.e., $\varepsilon_i \sim F_{g_i}$, $i = 1, \cdots, n$.

We choose the input similarity matrix $S$ to be negative Euclidean distance matrix, i.e., the $(i, j)$th entry $S_{ij} = -||x_i - x_j||^2$. By Proposition 2.1, this choice is equivalent to using the Gram matrix as the input matrix. Under the population setting, we let $\mathbb{S} := \mathbb{E} S$ be the input matrix to SDP-BF. We have that

$$ \mathbb{S}_{ij} = -||(\mu_k - \mu_l)||^2 - d_k - d_l, i \in C_k, j \in C_l, $$

$$ \mathbb{S}_{ij} = -2d_k, i, j \in C_k, i \neq j, $$

$$ \mathbb{S}_{ii} = 0, \forall i, $$

where $d_k := \mathbb{E} ||\varepsilon||^2, \varepsilon \sim F_k$. The next proportion shows Fisher consistency of the SDP in this case.
Proposition 3.6. Under the $k$-means model (3.6), let $S$ be the input matrix to SDP-BF. Write $d_{\text{max}} = \max_k d_k$. If

\[
\frac{1}{2} \min_{i \neq m} ||\mu_i - \mu_m||^2 > \frac{d_{\text{max}} - d_k}{n_k}, \forall k = 1, \cdots, K, \tag{3.7}
\]

then $X_0$ is the unique optimal solution to the SDP.

Proof. Due to the constraint $X1_n = 1_n$, using $\tilde{S}$ as input matrix is equivalent to using $\bar{S}$, where

\[
\tilde{S}_{ij} = -||\mu_k - \mu_l||^2, i \in C_k, j \in C_l, k \neq l,
\]

\[
\tilde{S}_{ij} = 0, i, j \in C_k, i \neq j,
\]

\[
\bar{S}_{ii} = 2d_k, i \in C_k.
\]

Under Theorem 2.10, this corresponds to $a_k = 0, c_k = 2d_k$. We choose $b_k = -\min_{l \neq m} ||\mu_i - \mu_m||^2, \forall k = 1, \cdots, K$. The claimed sufficient condition follows from Theorem 2.10.

The sufficient condition (3.7) essentially requires that the minimal cluster center separation is sufficiently larger than variations within clusters. Comparing sufficient conditions (3.4) and (3.7), we see that they take the same form. Specifically, in network case, $d_k$ can be written as $\sum_{j=1}^n \text{Var}(A_{ij})$ for a node $i$ belonging to $C_k$, and in multivariate data case, $d_k$ can be written as $\sum_{j=1}^p \text{Var}(x_{ij})$ for an observation $i$ belonging to $C_k$. Thus, despite different data types, performing network community detection using SDP-BF with squared adjacency matrix as the input matrix is conceptually equivalent to performing $k$-means clustering to the rows of the adjacency matrix by treating nodes as multivariate observations.
3.3 Exact Recovery under the SBM

In this section, we show exact recovery property of SDP-BF under strongly assortative SBMs. Throughout this section, we consider the case when the adjacency matrix $A$ is used as the input matrix to the SDP. To show exact recovery, we adopt the commonly used technique, namely primal-dual witness construction (see, for example, Amini and Levina, 2014; Awasthi et al., 2015; Cai and Li, 2015; Lei and Vu, 2015). The idea is to construct candidate dual optimal solutions such that the true clustering solution $X_0$ together with the constructed solution to the dual problem satisfy the optimal conditions (2.9)–(2.12). Recall that $C_k$ denotes the indices set of nodes in the $k$th community and its size is $n_k = |C_k|$. Let $1_{C_k} \in \mathbb{R}^n$ denote the indicator vector of $C_k$, whose entry is equal to one on $C_k$ and zero elsewhere. Then the true clustering solution $X_0$ can be expressed as $\sum_{k=1}^{K} 1_{C_k}1_{C_k}^T/n_k$. We first consider the planted partition model where within-community edge probabilities are all equal to $p$ and between-community edge probabilities are all equal to $q$, with $p > q$. Later, we generalize the result to general strongly assortative SBMs with $p_{\min} > q_{\max}$. The next lemma is useful for primal-dual witness construction.

Lemma 3.7. (Section E.7.2 in Dattorro, 2005). Let $S^n_c := \{Y \in S^n : Y1_n = 0\}$. In the ambient symmetric matrix space, the orthogonal complement space of $S^n_c$ is $S_n^{\perp} = \{u1_n^T + 1_n u^T : u \in \mathbb{R}^n\}$. Furthermore, for any $X \in S^n$, the corresponding unique decomposition is given by

$$X = P_{1_n^{\perp}}X P_{1_n^{\perp}} + 1_n ((1_n - \frac{1}{2n} 1_n 1_n^T \frac{X1_n}{n})^T + ((1_n - \frac{1}{2n} 1_n 1_n^T \frac{X1_n}{n})1_n^T),$$

where $P_{1_n^{\perp}}$ is the orthogonal projection matrix onto $R(1_n)^\perp$. 

48
We want to construct dual variables \((\alpha, Y)\) such that the true clustering solution \(X_0\) together with \((\alpha, Y)\) satisfies the optimality conditions of SDP-BF in Corollary 2.9 with the input similarity matrix \(S\) chosen to be the adjacency matrix \(A\). We have the following sufficient condition for exact recovery.

**Lemma 3.8.** Assume that \(Y\) is dual feasible (i.e., \(Y \in S^n, Y \geq 0\)), and there exist \(\alpha \in \mathbb{R}^n\) and \(a \in \mathbb{R}\) such that

\[
(C1) \quad A - (1_n \alpha^T + \alpha 1_n^T) + Y = -\Lambda + aI_n.
\]

\[
(C2) \quad \Lambda 1_{C_k} = 0, k = 1, \cdots, K, \text{ and } u^T \Lambda u > 0, \forall u \in \text{span}\{1_{C_k}\}^\perp.
\]

\[
(C3) \quad Y_{C_k} = 0, k = 1, \cdots, K.
\]

Then \(X_0\) is the unique optimal solution to SDP-BF, and \((\alpha, Y)\) is dual optimal.

**Proof.** It suffices to check \(X_0\) together with \((\alpha, Y)\) satisfies the optimality conditions (2.11) and (2.12) in Corollary 2.9. By the form of \(X_0\), since \((X_0)_{C_k} > 0\), to satisfy complementary slackness condition (2.12), we require \(Y_{C_k} = 0, k = 1, \cdots, K\). This is condition (C3). By condition (C1) and Lagrangian optimality condition (2.11), we require

\[
X_0 \in \arg \max_{H \in \mathcal{F}^K} \langle -\Lambda + aI_n, H \rangle = \arg \min_{H \in \mathcal{F}^K} \langle \Lambda, H \rangle
\]

Note that condition (C2) states that the \(K\) smallest eigenvalues of \(\Lambda\) are zero with the corresponding eigenspace being \(\text{span}\{1_{C_k}\}\), and the \((K+1)\)th smallest eigenvalue is greater than zero. By Lemma 2.6, \(X_0\) is the unique solution to (3.8). \(\blacksquare\)
We wish to construct \((\alpha, Y, a, \Lambda)\) such that the conditions in Lemma 3.8 are satisfied. According to conditions (C1) and (C3), for diagonal blocks, we have that

\[
A_{C_k} - (1_{n_k} \alpha_{C_k}^T + \alpha_{C_k} 1_{n_k}^T) - aI_{n_k} + \Lambda_{C_k} = 0, \quad k = 1, \ldots, K. \tag{3.9}
\]

For off-diagonal blocks, we have that

\[
A_{C_l C_k} - (1_{n_l} \alpha_{C_k}^T + \alpha_{C_l} 1_{n_k}^T) + Y_{C_l C_k} + \Lambda_{C_l C_k} = 0, \quad k \neq l. \tag{3.10}
\]

Note that \(\Lambda 1_{C_k} = 0, \forall k\) in condition (C2) is equivalent to

\[
\Lambda_{C_l C_k} 1_{n_k} = 0, \forall l, k.
\]

For diagonal blocks of \(\Lambda\), since \(\Lambda_{C_k} 1_{n_k} = 0\) and \(\Lambda_{C_k}\) is symmetric, it has to have the form

\[
\Lambda_{C_k} = P_{1_{n_k}^\perp} X_{n_k} P_{1_{n_k}^\perp}, \text{ for some } X_{n_k} \in \mathbb{S}^{n_k}, \tag{3.11}
\]

where \(P_{1_{n_k}^\perp}\) is the orthogonal projection matrix onto \(R(1_{n_k})^\perp\). For off-diagonal blocks of \(\Lambda\), since \(\Lambda_{C_l C_k} 1_{n_k} = 0, \forall k \neq l\), and \(\Lambda_{C_l C_k} = \Lambda_{C_k C_l}^T\), it has to have the form

\[
\Lambda_{C_l C_k} = P_{1_{n_l}^\perp} X_{C_l C_k} P_{1_{n_k}^\perp}, \text{ for some } X_{C_l C_k} = X_{C_k C_l}^T. \tag{3.12}
\]

Let us first choose \(\alpha_{C_k}, k = 1, \ldots, K\) based on (3.9) and (3.11). Rearrange (3.9), we have that

\[
A_{C_k} - aI_{n_k} = (1_{n_k} \alpha_{C_k}^T + \alpha_{C_k} 1_{n_k}^T) - \Lambda_{C_k}, \quad k = 1, \ldots, K.
\]

By Lemma 3.7 and (3.11), the decomposition on the right hand side is unique and we have that for all \(k = 1, \ldots, K\),

\[
\Lambda_{C_k} = -P_{1_{n_k}^\perp} (A_{C_k} - aI_{n_k}) P_{1_{n_k}^\perp}, \quad \alpha_{C_k} = (I_{n_k} - \frac{1}{2n_k} 1_{n_k} 1_{n_k}^T) (A_{C_k} - aI_{n_k}) 1_{n_k}. \tag{3.11}
\]
Next, we choose $\Lambda_{C_lC_k}$ and $Y_{C_lC_k}, \forall l \neq k$ such that they satisfy (3.10) and (3.12). We choose

$$
\Lambda_{C_lC_k} := -P_{\frac{1}{n_l}A_{C_lC_k}P_{\frac{1}{n_k}}}, \forall l \neq k,
$$

which satisfies (3.12). Then, by (3.10), we have that

$$
Y_{C_lC_k} = \mathbf{1}_{n_l}\alpha_{C_k}^T + \alpha_{C_l}\mathbf{1}_{n_k}^T + P_{\frac{1}{n_l}A_{C_lC_k}P_{\frac{1}{n_k}}} - A_{C_lC_k}, \forall l \neq k.
$$

Note that this choice of $Y$ is symmetric.

Now, the two remaining conditions in Lemma 3.8 that need to be satisfied are $u^T\Lambda u > 0, \forall u \in \text{span}\{\mathbf{1}_{C_k}\}^\perp$, and $Y_{C_lC_k} \geq 0, \forall l \neq k$. To satisfy these two conditions, we impose constraints on the free variable $a$. The next lemma provides the conditions on $a$ under which the two remaining conditions in Lemma 3.8 are satisfied. Thus, it provides a deterministic sufficient condition for exact recovery using SDP-BF.

**Lemma 3.9.** Consider a planted partition model where within-community edge probabilities are equal to $p$. Given the previous choices of $\alpha, Y$, and $\Lambda$, we have that

1. $Y \geq 0$, if $\forall l \neq k, i \in C_l, j \in C_k$,

$$
a \leq \frac{1}{n_l + n_k} (2n_kd_i(C_l) + 2n_l d_j(C_k) - n_l \frac{\mathbf{1}_{n_k}^T A_{C_k}\mathbf{1}_{n_k}}{n_k} - n_k \frac{\mathbf{1}_{n_l}^T A_{C_l}\mathbf{1}_{n_l}}{n_l} - 2n_k d_j(C_l) - 2n_l d_i(C_k) + 2\mathbf{1}_{n_l}^T A_{C_lC_k}\mathbf{1}_{n_k}),
$$

(3.13)

where $d(C_k) := A_{C_k}\mathbf{1}_{n_k} \in \mathbb{R}^n$ and $d_i(C_k)$ is the $i$-th component of $d(C_k)$.

2. $u^T\Lambda u > 0, \forall u \in \text{span}\{\mathbf{1}_{C_k}\}^\perp$ if

$$
a > -p + \|A - E\|_2.
$$

(3.14)
Thus, if \( \forall l \neq k, i \in C_l, j \in C_k \),

\[
\|A - E\|_2 - \eta < \frac{1}{n_l + n_k}(2n_kd_i(C_i) + 2n_id_j(C_k) - n_l - \frac{1^T_n A_C k 1_{n_k}}{n_k} - n_k - \frac{1^T_n A_C i 1_{n_l}}{n_l}) - 2n_kd_j(C_i) - 2n_id_i(C_k) + 21^T_n A_C i C_k 1_{n_k},
\]

(3.15)

then \( X_0 \) is the unique solution to SDP-BF.

**Proof.** Let \( e_i \) be the \( i \)th unit vector. To show claim (1), for all \( l \neq k, i \in C_l, j \in C_k \), we have that

\[
(Y_{C_l C_k})_{ij} = e_i^T Y_{C_l C_k} e_j
\]

\[
= (\alpha_{C_k})_j + (\alpha_{C_l})_i - \frac{d_j(C_i)}{n_l} - \frac{d_i(C_k)}{n_k} + \frac{1^T_n A_{C_l C_k} 1_{n_k}}{n_l n_k}.
\]

By definition of \( \alpha_{C_l} \), we have that

\[
(\alpha_{C_k})_j = (e_j - \frac{1}{2n_k} 1_{n_k})^T (A_{C_k} - aI_{n_k}) 1_{n_k}
\]

\[
= \frac{d_j(C_k)}{n_k} - \frac{a}{2n_k} - \frac{1}{2n_k^2} 1^T_{n_k} A_{C_k} 1_{n_k}.
\]

Similarly,

\[
(\alpha_{C_l})_i = \frac{d_i(C_l)}{n_l} - \frac{a}{2n_l} - \frac{1}{2n_l^2} 1^T_{n_l} A_{C_l} 1_{n_l}.
\]

Thus, we have that

\[
(Y_{C_l C_k})_{ij} = \frac{d_j(C_k)}{n_k} - \frac{a}{2n_k} - \frac{1}{2n_k^2} 1^T_{n_k} A_{C_k} 1_{n_k} + \frac{d_i(C_l)}{n_l} - \frac{a}{2n_l} - \frac{1}{2n_l^2} 1^T_{n_l} A_{C_l} 1_{n_l} - \frac{d_j(C_i)}{n_l} - \frac{d_i(C_k)}{n_k} + \frac{1^T_{n_l} A_{C_l C_k} 1_{n_k}}{n_l n_k}.
\]

Setting \((Y_{C_l C_k})_{ij} \geq 0\) and rearranging terms, we obtain (3.13). To show claim (2), for any \( u \in \text{span}\{1_{C_k}\}^\perp \), it can be represented as \( u = (u_1^T, \ldots, u_K^T)^T \), where \( u_k \in \mathbb{R}^{n_k} \), \( u_k^T 1_{n_k} = 0 \), \( \forall k \). Expanding \( u^T \Lambda u \), we have that

\[
u^T \Lambda u = \sum_k u_k^T A_{C_k} u_k + \sum_{l \neq k} u_l^T A_{C_l C_k} u_k
\]

\[
= -\sum_k u_k^T (A_{C_k} - aI_{C_k}) u_k - \sum_{l \neq k} u_l^T A_{C_l C_k} u_k
\]

\[
= -u^T A u + a ||u||^2
\]

\[
= -u^T (A - E A) u + p ||u||^2 + a ||u||^2.
\]

52
Setting $u^T A u > 0$, we require that for all $u \in \text{span}\{1_{c_k}\}$,

$$a > -p + \frac{u^T (A - E A) u}{||u||^2}.$$  

This is implied by

$$a > -p + \max_{||x||=1} x^T (A - E A) x,$$

which is in turn implied by

$$a > -p + ||A - E A||_2.$$

To show the probabilistic conditions for exact recovery, we proceed with the probabilistic arguments similar to those in Amini and Levina [2014]. The idea is to use the concentration inequality of the adjacency matrix for relatively dense networks to obtain an upper bound of $||A - E A||_2$, and use Bernstein’s inequality and union bound argument to obtain an lower bound of the right hand side of (3.15). We have the following theorem for exact recovery of SDP-BF under planted partition models.

The detailed proof is provided in Appendix A.

**Theorem 3.10.** Let $A \in \{0, 1\}^{n \times n}$ be drawn from a planted partition model with $n$ nodes and $K$ communities of sizes $n_1, \ldots, n_K$. The within-community edge probability is denoted by $p$ and the between-community edge probability is denoted by $q$. Let $n^+ := \max_k n_k, n^- := \min_k n_k$. Assume that $n^- > e$. For any $c_1 > 1$ and $c_2 > 2$, assume that $p \geq \max\{C' \frac{\log n^-}{n^-}, \frac{4c_1 \log n}{9(n^- - 1)}\}$. Then, if

$$n^- (p - q) > C \sqrt{pn^+} + C \sqrt{qn^-} + 3 \sqrt{4c_1 p(n^- - 1) \log n} + 3 \sqrt{4c_2 qn^- \log n},$$

$X_0$ is the unique solution to SDP-BF with probability at least $1 - cK(n^-)^{-r} - cn^{-r} - 2n^{-(c_1 - 1)} - 2n^{-(c_2 - 2)}$.  

53
Consider an adjacency matrix generated from a general SBM parameterized by a community-wise edge probability matrix $B$. Let $p_{\min} := \min_k B_{kk}$ and $q_{\max} := \max_{k \neq l} B_{kl}$. By Corollary 4.2 in Amini and Levina (2014), exact recovery under this SBM has higher probability than exact recovery under the planted partition model where within-community edge probabilities are equal to $p_{\min}$ and between-community edge probabilities are equal to $q_{\max}$. Thus, Theorem 3.10 can be generalized to general SBMs with $p$ and $q$ replaced by $p_{\min}$ and $q_{\max}$.

3.4 Simulation Studies and Real Data Applications

In this section, we conduct numerical experiments to compare SDP-BF with other state-of-the-art SDP-based methods and spectral clustering algorithms for network community detection. We consider three different versions of SDP-BF using three different input matrices. SDP-BF-adj uses the adjacency matrix $A$ as the input matrix to SDP-BF, SDP-BF-adjsq uses the squared adjacency matrix $A^2$ as the input matrix to SDP-BF, and SDP-BF-lap uses the symmetric normalized graph Laplacian $D^{-1/2}AD^{-1/2}$ as the input matrix to SDP-BF. We choose SDP-CL (1.6) proposed by Cai and Li (2015), SDP-GV (1.7) proposed by Guédon and Vershynin (2016), and SDP-MM (1.10) proposed by Chen, Li, et al. (2016) as competing SPD-based methods. The former two SDPs were motivated from likelihood maximization under the SBM and the latter SDP was designed to work for networks generated from the DCSBM. For SDP-CL, as suggested by Cai and Li (2015), the tuning parameter $\lambda$ is chosen to be the mean connectivity density of the subgraph induced by nodes whose degrees are between the 20th and 80th percentile of degrees of all nodes. For SDP-GV, no data-based choice of $\lambda$ was suggested by Guédon and Vershynin (2016). We choose
̂\lambda to be the sum of squared community sizes, \( \sum_k n_k^2 \), which is the best choice for ̂\lambda.

We take a note that in practice, it is impossible to know true community sizes and a data-based choice of ̂\lambda must be designed for SDP-GV to make it have practical usage. For SDP-MM, \( \lambda \) is chosen to be \( (A, E_n)^{-1} \) as suggested by Chen, Li, et al. (2016). For all SDP-based methods, after we obtain the solution matrix from the SDP, we perform \( k \)-means clustering algorithm with \( k = K \) to rows of the solution matrix to obtain explicit label estimates of nodes. For spectral clustering algorithms, we consider two different versions. The first version is the one based on eigenvectors of the adjacency matrix \( A \) analyzed in Lei and Rinaldo (2015). Specifically, it first computes the matrix whose columns are the leading \( K \) eigenvectors corresponding to the \( K \) largest absolute eigenvalues of \( A \). Then it performs \( k \)-means clustering algorithm with \( k = K \) to the rows of the matrix obtained from the first step to obtain explicit label estimates of nodes. Hereafter we refer to this version of spectral clustering as ADJSC. The second version is the regularized spectral clustering proposed by Amini, Chen, et al. (2013). It starts with forming a regularized adjacency matrix \( A_\lambda := A + c_n 1_n 1_n^T \), where \( \lambda \) is the average degree of a graph. The symmetric normalized graph Laplacian is computed by \( L_\lambda = D_\lambda^{-1/2} A_\lambda D_\lambda^{1/2} \), where \( D_\lambda \) is the diagonal matrix of node degrees computed from \( A_\lambda \). It then computes the matrix whose columns are the leading \( K \) eigenvectors corresponding to the \( K \) largest absolute eigenvalues of \( L_\lambda \) and performs \( k \)-means clustering algorithm with \( k = K \) to the rows of the matrix to obtain explicit label estimates of nodes. Following Amini, Chen, et al. (2013), we set \( c = 0.25 \) in numerical experiments. Hereafter this regularized spectral clustering is referred to as RSC. We use the misclassification rate to evaluate the performances of methods, which is the proportion of misclassified nodes. Since the community memberships of
nodes are unique only up to relabeling of the communities, to compute the number of misclassified nodes, we always apply the best permutation of community labels to label estimates of nodes output by the $k$-means algorithm and compare it to true node labels. For each of the simulation scenarios, we generate 50 random graphs. For each method, we compute the average misclassification rate over 50 replicates as the performance evaluation criterion.

3.4.1 Simulation Studies

Sparse Networks

We first consider the case of sparse networks whose expected degrees remain bounded in the number of nodes $n$. Consider a strongly assortative planted partition model with three equal-sized communities where within-community edge probabilities $p = 10/n$ and between-community edge probabilities $q = 1/n$. Regardless of the number of nodes, expected degrees of random graphs generated from it are bounded by 10. As $n$ becomes larger, networks become sparser. We set $n$ to be 60, 180, 300, 420, 540, 660. Figure 3.2 shows average misclassification rates of different methods over 50 randomly generated graphs for different values of $n$. We see that as graphs become sparser, misclassification rates of SDP-BF-adj, SDP-BF-adjsq, and ADJSC increase, which indicates that these three methods may not be suitable for community detection in sparse networks. On the other hand, misclassification rates of all other SDP-based methods and the regularized spectral clustering remain stable at low values as the number of nodes increases. For SDP-BF, we notice that by using the symmetric normalized graph Laplacian as the input matrix, it achieves comparable performance to other SDP-based methods and RSC for community detection in sparse networks.
Figure 3.2: Average misclassification rates for sparse graphs versus the number of nodes. Sparse graphs are generated from a planted partition model with three equal-sized communities where within-community edge probabilities \( p = \frac{10}{n} \) and between-community edge probabilities \( q = \frac{1}{n} \). The results are averaged over 50 replicates.

**Weakly Assortative SBM**

We consider a weakly assortative SBM with three equal-sized communities where the community-wise edge probability matrix is

\[
B = \begin{pmatrix}
0.25 & 0.05 & 0.2 \\
0.05 & 0.1 & 0.08 \\
0.2 & 0.08 & 0.25
\end{pmatrix}
\]  (3.16)

We notice that under this configuration, the second community is a weakly assortative community. We set the number of nodes \( n \) to be 60, 180, 300, 420, 540, 660. The results are presented in Figure 3.3. We see that in general SDP-BF-adj provides low misclassification rates over all values of \( n \) compared to other methods. SDP-BF-adjsq
Figure 3.3: Average misclassification rates for random graphs generated from the weakly assortative SBM specified by the community-wise edge probability matrix $B$ shown in (3.16). The results are averaged over 50 replicates.

also shows relatively low misclassification rates for all values of $n$. SDP-BF-lap and SDP-MM perform the best when the number of nodes $n$ becomes large. On the other hand, SDP-CL, SDP-GV, ADJSC, and RSC have high misclassification rates for all values of $n$, indicating that strong assortativity may be a necessary condition for them to work.

Mixed Networks with Both Assortative and Disassortative Communities

We consider the SBM that generates mixed networks with both assortative and disassortative communities introduced in Example 3.5. The community-wise edge
probability matrix is provided by
\[
B = \begin{pmatrix}
0.3 & 0.1 & 0.1 \\
0.1 & 0.05 & 0.2 \\
0.1 & 0.2 & 0.3 \\
\end{pmatrix}
\] (3.17)

It will create mixed networks with the first and the third communities to be assortative and the second community to be disassortative. Let us consider three equal-sized communities. We set \( n \) to be \( 60, 180, 300, 420, 540, 660 \). In Example 3.5, we have shown that SDP-BF-adjsq is guaranteed to achieve population consistency in this scenario, while SDP-BF-adj fails to recover communities empirically under the population setup. Thus, we expect that SDP-BF-adjsq has a good chance of recovering node labels for random graphs generated from this scenario and that SDP-BF-adj may fail to do so. The results are presented in Figure 3.4. It is clear that SDP-BF-adjsq performs much better than all other methods, and it almost recovers node labels perfectly for large values of \( n \). On the other hand, all other SDP-based methods remain at large values of misclassification rates when \( n \) increases. For spectral clustering algorithms ADJSC and RSC, although their misclassification rates decrease as \( n \) increases, their performances are not as good as SDP-BF-adjsq.

**Degree Corrected SBM**

Finally, we consider random graphs generated from the DCSBM. Consider a DCSBM with \( n = 300 \) nodes and \( K = 3 \) equal-sized communities. Recall that in a DCSBM, we introduce \( n \) nonnegative degree parameters \( \theta_1, \cdots, \theta_n \), one for each node. For two nodes \( i \) and \( j \), we set their edge probability to be \( p\theta_i\theta_j \) if they belong to the same community and \( q\theta_i\theta_j \) if they belong to different communities. We set \( p = 0.1 \) and \( q = 0.03 \). The degree parameters \( \theta_i \) are independently and identically sampled from a beta distribution \( \text{Beta}(\alpha, 1) \), where we choose \( \alpha \) to be \( 1, 2, 3, 4, 5, 6 \).
Figure 3.4: Average misclassification rates for random graphs generated from the mixed SBM specified by the community-wise edge probability matrix $B$ shown in (3.17). The results are averaged over 50 replicates.

We note that as $\alpha$ increases, degree parameters $\theta_i$ become more and more concentrated around one, and thus the DCSBM becomes closer to a SBM. The results are shown in Figure 3.5. We see that in general, as $n$ increases and the DCSBM becomes closer to a SBM, misclassification rates decrease for all methods. Since SDP-MM is designed to work under the DCSBM, it is natural to use it as a benchmark. We see that SDP-MM and SDP-CL have similar performance and perform the best among all methods. Except SDP-BF-adjsq, other SDP-based methods have comparable performance to SDP-MM and SDP-CL.
Figure 3.5: Average misclassification rates for random graphs generated from a DCSBM. The edge probability between nodes $i$ and $j$ is $0.1 \theta_i \theta_j$ if $i$ and $j$ belong to the same community and $0.03 \theta_i \theta_j$ if they belong to different communities, where $\theta_i \sim \text{Beta}(\alpha, 1)$. The results are averaged over 50 replicates.

### 3.4.2 Facebook Collegiate Networks

In this section, we compare the performance of different methods using Facebook networks from Traud et al. (2011). The dataset contains Facebook friendship networks from each of the 100 American colleges and universities recorded on a single day in September 2005. Each node represents a Facebook user and each edge between two users represents the friendship relation between the two users. Besides networks, the dataset also includes gender, class year, and data fields that represent (using numerical identifiers) high school, major, and dormitory residence of each user if available.
Caltech Network

Let us first investigate the Caltech network. It has been observed in Traud et al. (2011) that the community structure at Caltech is related to dormitory residence. Thus, we focus on the subgraph induced by nodes whose dormitory residences are known and consider the largest connected component of the subgraph. The resulting network consists of 590 nodes and 12822 edges. We use dormitory residence as true labels of nodes. Since there are 8 values of dormitory residence, we set the true number of communities $K = 8$ in the analysis. For SDP-GV, since there is no data-based choice of the tuning parameter, we set the value of its tuning parameter to be the sum of squared community sizes based on the true labels of dormitory residence. Results show that misclassification rates of SDP-BF-adj, SDP-BF-adjsq, SDP-BF-lap, SDP-CL, SDP-GV, SDP-MM, ADJSC, and RSC are 0.331, 0.371, 0.181, 0.398, 0.266, 0.220, 0.379, and 0.347, respectively. This shows that SDP-BF with the symmetric normalized graph Laplacian as the input matrix performs the best among all methods, and it misclassifies 107 nodes. Figure 3.6 shows permuted adjacency matrices of Caltech network according to estimated labels using different methods. Blue pixels in the visualized adjacency matrices stand for entries of one and white pixels stand for entries of zero. We see that except SDP-BF-lap and SDP-MM, other methods tend to incorrectly classify low degree nodes as a community.

Simmons College Network

Next we investigate the Simmons College network. It has been observed in Traud et al. (2011) that the community structure at Simmons College is related to class
year. Thus, we focus on the subgraph induced by nodes whose class years are known and consider the largest connected component of the subgraph. The resulting network consists of 1158 nodes and 24449 edges. We use class year as true labels of nodes. Since there are 4 values of class year, we set the true number of communities $K = 4$ in the analysis. Again for SDP-GV, we set the value of its tuning parameter to be the sum of squared community sizes based on the true labels of class year. Results show that misclassification rates of SDP-BF-adj, SDP-BF-adjsq, SDP-BF-lap, SDP-CL, SDP-GV, SDP-MM, ADJSC, and RSC are 0.395, 0.378, 0.116, 0.290, 0.282, 0.116, 0.387, and 0.277, respectively. This shows that SDP-BF with the symmetric normalized graph Laplacian as the input matrix and SDP-MM perform the best among all methods,
and they both misclassify 135 nodes. Figure 3.7 shows permuted adjacency matrices of Simmons College network according to estimated labels using different methods. Blue pixels in the visualized adjacency matrices stand for entries of one and white pixels stand for entries of zero. Again, we see that except SDP-BF-lap and SDP-MM, other methods tend to incorrectly classify low degree nodes as a community.

Figure 3.7: Permuted adjacency matrices of Simmons College network according to estimated labels using different methods. Blue pixels in the visualized adjacency matrices stand for entries of one and white pixels stand for entries of zero.
Chapter 4: Estimating Network Edge Probabilities by Semidefinite Programming

In this chapter, we propose a semidefinite programming approach to estimating network edge probabilities generated from smooth graphon functions based on a single realization of the network. Interestingly, despite different motivations, it turns out that the proposed SDP approach is based on the same SDP as that is used for community detection. In Section 4.1, we introduce the motivation and the method. In Section 4.2, we conduct simulation studies for three graphon functions considered in Zhang et al. (2015), and compare the performance of the proposed method with the neighborhood smoothing method. The application of the method to the link prediction problem is presented in Section 4.3.

4.1 SDP for Network Edge Probability Matrix Estimation

We consider undirected graphs. Let $A$ denote the symmetric binary adjacency matrix of a network and $P$ denote the edge probability matrix. We assume that for all $i \leq j$, $A_{ij}$ is generated from Bernoulli($P_{ij}$) independently. Unlike the study of community detection, we extend the network edge probability matrix $P$ from block models to general probability matrices generated from smooth graphon functions. The goal is to estimate $P$ from a single realization of $A$. Following the idea of neighborhood
smoothing of Zhang et al. (2015), for a node $i$, a node $j$ is its neighbor if $P_j \approx P_i$, where $P_i$ is the $i$th row of $P$. If the set of neighbors of node $i$ is found, we can estimate $P_i$ by averaging the rows of $A$ corresponding to node $i$’s neighbors. Let us consider a normalized clustering matrix $X$ that partitions nodes into $K$ clusters. The idea is to use $X$ to smooth the adjacency matrix $A$ such that the smoothed adjacency matrix is as close to $A$ as possible. We use the smoothed adjacency matrix as the estimated edge probability matrix. To guarantee the symmetry of the estimated edge probability matrix, we smooth the rows of $A$ and the columns of $A$ separately and take the average of the two smoothed matrices. Formally, we wish to solve

$$\begin{align*}
\text{minimize} & \quad \| \frac{1}{2} (AX +XA) - A \|^2_F \\
\text{subject to} & \quad X \geq 0, X\mathbf{1}_n = \mathbf{1}_n, X = X^T, X^2 = X, \text{tr}(X) = K.
\end{align*}$$

(4.1)

Since $X$ is a projection matrix, we have that for any $X$,

$$\begin{align*}
\| \frac{1}{2} (AX +XA) - A \|^2_F &= \| \frac{1}{2} (AX - A) + \frac{1}{2} (XA - A) \|^2_F \\
&= \frac{1}{2} \| AX - A \|^2_F + \frac{1}{2} \langle AX - A, XA - A \rangle \\
&= \| AX - A \|^2_F - \frac{1}{2} \| XA(I_n - X) \|^2_F.
\end{align*}$$

Thus, for any $X$ belonging to the feasible set, we have that

$$\| AX - A \|^2_F \geq \| \frac{1}{2} (AX +XA) - A \|^2_F.$$

As a surrogate to the original problem (4.1), we solve

$$\begin{align*}
\text{minimize} & \quad \| AX - A \|^2_F \\
\text{subject to} & \quad X \geq 0, X\mathbf{1}_n = \mathbf{1}_n, X = X^T, X^2 = X, \text{tr}(X) = K.
\end{align*}$$

(4.2)
Since $X$ is a projection matrix, solving (4.2) is equivalent to solving $\max_X \langle A^2, X \rangle$.

Relaxing the feasible set to be $\mathcal{F}^K \cap \mathcal{B}_{\text{sym}}$, we arrive at the semidefinite program

$$
\begin{align*}
\text{maximize} & \quad \langle A^2, X \rangle \\
\text{subject to} & \quad X \in \mathcal{F}^K \cap \mathcal{B}_{\text{sym}}.
\end{align*}
$$

(4.3)

Let $\hat{X}$ denote a solution to (4.3). The estimated edge probability matrix is given by $\hat{P} = \frac{1}{2}(A\hat{X} + \hat{X}A)$. In this chapter, we call this method of estimating the edge probability matrix SDP-BF.

### 4.1.1 Connection to Community Detection

When the graphon function is a piecewise constant function, it will generate block-wise constant edge probability matrices after appropriate permutations of node labels. In other words, the generated edge probability matrix forms an SBM. Given an adjacency matrix generated from a block model, by the discussion in Section 3.1.3, we expect that the solution $\hat{X}$ to (4.3) is close to the normalized clustering matrix according to the true block structure. Smoothing the adjacency matrix by $\frac{1}{2}(A\hat{X} + \hat{X}A)$ has the effect of averaging rows and columns of $A$ in a block-wise fashion, and thus the smoothed adjacency matrix is close to the true probability matrix $P$. Furthermore, by results in Chapter 3, performing community detection using SDP-BF with the adjacency matrix as the input matrix requires that the block model is assortative.

On the other hand, using squared adjacency matrix as the input matrix has a good chance of recovering the true community structure even for mixed networks with both assortative and disassortative communities. Since piecewise constant graphon functions cannot guarantee the assortativity for generated edge probability matrices, this indicates the necessity of using squared adjacency matrix $A^2$ instead of $A$ to the SDP when estimating edge probabilities.
4.1.2 Connection to Neighborhood Smoothing

For general smooth graphon functions, no block structure exists in generated edge probability matrices. Since the solution $\hat{X}$ to the SDP (4.3) is a symmetric doubly stochastic matrix, we can view $\hat{X}$ as a smoother matrix and treat $K$ as a tuning parameter controlling the amount of smoothing. Based on the expression of $\hat{P}$, we have that

$$\hat{P}_{ij} = \frac{1}{2}(\sum_{k=1}^{n} A_{ik}\hat{X}_{kj} + \sum_{k=1}^{n} \hat{X}_{ik}A_{kj}).$$

Since $\hat{X}$ is a symmetric doubly stochastic matrix, $\sum_{k=1}^{n} A_{ik}\hat{X}_{kj}$ is a weighted average of entries in $i$th row of $A$ with weights given by the “affinity” between the corresponding nodes and node $j$ and $\sum_{k=1}^{n} \hat{X}_{ik}A_{kj}$ is a weighted average of entries in $j$th columns of $A$ with weights given by the “affinity” between the corresponding nodes and node $i$. Recall that the SDP (4.3) can be viewed as a relaxation of $k$-means clustering to the rows of $A$. Thus the “affinity” between two nodes depends on the Euclidean distance between the corresponding rows of $A$. This matches the idea of neighborhood smoothing of Zhang et al. (2015) in that a node’s neighborhood consists of nodes with similar rows in the adjacency matrix. Compared to (1.11), unlike neighborhood smoothing which assigns equal weights to nodes in a node’s neighborhood and zero weights to the other nodes, our method uses a soft assignment of weights provided by the solution to the SDP. So our method follows the same idea as neighborhood smoothing while providing a more principled and flexible way of determining the neighborhood.
4.2 Simulation Studies

In this section, we conduct simulation studies to compare SDP-BF with NBS regarding the performance of estimating network edge probabilities. We compare SDP-BF with NBS since NBS is reported to outperform several benchmark methods including universal singular value thresholding, sort and smooth methods, and step function approximations based on fitting an SBM. For the purpose of comparison, we consider three graphon functions that are not piecewise constant considered in Zhang et al. (2015). Specifically, we consider

- **graphon 1**: \( f(u, v) = \sin(5\pi(u + v - 1) + 1)/2 + 0.5 \),
- **graphon 2**: \( f(u, v) = 1 - (1 + \exp(15(0.8|u - v|^{1/5} - 0.1)))^{-1} \),
- **graphon 3**: \( f(u, v) = (u^2 + v^2)/3 \cos(1/(u^2 + v^2)) + 0.15 \).

For a given graphon function \( f \) and number of nodes \( n \), we generate a network edge probability matrix \( P \) from it. Then we generate adjacency matrices using \( P \). Based on a single adjacency matrix, we obtain an estimated edge probability matrix \( \hat{P} \) and evaluate it using mean squared error \( \frac{1}{n^2} \| \hat{P} - P \|_F^2 \). As for the choice of tuning parameter, in this study, we choose the best tuning parameter value over a sequence of candidate values for both methods in terms of minimizing the mean squared error. For SDP-BF, we consider values of \( K \) to be integers varying from 2 to 20. For NBS, it uses \( h \) sample quantile to threshold sets of distances between nodes and determines the set of neighbors of each node, where \( h \) is the tuning parameter. We consider values of \( h \) to be 0.01, 0.02, \ldots, 0.2. For a generated probability matrix \( P \) from a graphon \( f \), we repeat the experiment for 20 randomly generated adjacency matrices.
Figure 4.1 shows the true edge probability matrix generated from graphon 1 when the number of nodes is 500 together with estimated probability matrices from SDP-BF and NBS for a single replicate. We see that both methods successfully estimate the true probability matrix. Figure 4.2 shows average mean squared errors and the corresponding one standard deviation error bars for both methods over 20 replicates when the number of nodes is chosen to be 100, 200, 300, 400, 500, respectively. We see that overall SDP-BF achieves smaller mean squared errors.

Figure 4.3 shows the true edge probability matrix generated from graphon 2 when the number of nodes is 500 together with estimated probability matrices from SDP-BF and NBS for a single replicate. We see that SDP-BF estimates the concentration of small values around the diagonal line better than NBS. This can be further verified by looking at average mean squared errors and the corresponding one standard deviation error bars for both methods over 20 replicates, which is shown in Figure 4.4. We see that SDP-BF has much smaller mean squared errors compared to NBS.

Figure 4.5 shows the true edge probability matrix generated from graphon 3 when the number of nodes is 500 together with estimated probability matrices from SDP-BF and NBS for a single replicate. In general, both methods recover the pattern in the lower right corner, although NBS yields a smoother estimator. This is partly due to the fact that the tuning parameter is chosen based on the criterion of minimizing the mean squared error. In addition, both methods fail to capture the details in the upper left corner of the true probability matrix. If we look at average mean squared errors and the corresponding one standard deviation error bars for both methods over 20 replicates shown in Figure 4.6, we see that SDP-BF achieves smaller mean squared errors compared to NBS.
Figure 4.1: Left panel: true edge probability matrix generated from graphon 1. Middle panel: estimated probability matrix from SDP-BF. Right panel: estimated probability matrix from NBS. Blue represents large values and white represents small values.

Figure 4.2: Average mean squared errors and the corresponding one standard deviation error bars for estimating edge probability matrices generated from graphon 1 with different numbers of nodes. The number of replicates is 20.
Figure 4.3: Left panel: true edge probability matrix generated from graphon 2. Middle panel: estimated probability matrix from SDP-BF. Right panel: estimated probability matrix from NBS. Blue represents large values and white represents small values.

Figure 4.4: Average mean squared errors and the corresponding one standard deviation error bars for estimating edge probability matrices generated from graphon 2 with different numbers of nodes. The number of replicates is 20.
Figure 4.5: Left panel: true edge probability matrix generated from graphon 3. Middle panel: estimated probability matrix from SDP-BF. Right panel: estimated probability matrix from NBS. Blue represents large values and white represents small values.

Figure 4.6: Average mean squared errors and the corresponding one standard deviation error bars for estimating edge probability matrices generated from graphon 3 with different numbers of nodes. The number of replicates is 20.
4.3 Application to Link Prediction

In this section, we apply SDP-BF to the link prediction problem and compare its performance with that of NBS. We follow the framework of Zhang et al. (2015). We manually create missing node pairs by generating independent $O_{ij} \sim \text{Bernoulli}(1-p)$, where $p$ is the missing rate. If $O_{ij} = 1$, then the connectivity between node pair $(i, j)$ is observed, and $O_{ij} = 0$ if the connectivity is missing. Given the true adjacency matrix $A$, what we observe is

$$A_{\text{obs}} := A \circ O,$$

where $\circ$ denotes the element-wise matrix multiplication. Through this way, we can estimate the edge probability matrix $\hat{P}$ based on $A_{\text{obs}}$, predict links for those missing node pairs based on $\hat{P}$, and test the performance of link prediction by comparing with the true connectivity for missing node pairs. Let

$$n_P := |\{(i, j) : A_{ij} = 1, O_{ij} = 0\}|$$

be the total number of missing connected node pairs and

$$n_N := |\{(i, j) : A_{ij} = 0, O_{ij} = 0\}|$$

be the total number of missing disconnected node pairs. For a threshold $0 < t < 1$, a missing node pair $(i, j)$ is predicted to be connected if $\hat{P}_{ij} > t$ and disconnected otherwise. Let

$$n_{TP}(t) := |\{(i, j) : \hat{P}_{ij} > t, A_{ij} = 1, O_{ij} = 0\}|$$

be the number of missing connected node pairs that are predicted to be connected and

$$n_{FP}(t) := |\{(i, j) : \hat{P}_{ij} > t, A_{ij} = 0, O_{ij} = 0\}|$$
be the number of missing disconnected node pairs that are predicted to be connected. Then the false positive rate \( r_{FP}(t) \) is defined to be the proportion of missing disconnected node pairs that are predicted to be connected, i.e., \( r_{FP}(t) = \frac{n_{FP}(t)}{n_N} \), and the true positive rate \( r_{TP}(t) \) is defined to be the proportion of missing connected node pairs that are predicted to be connected, i.e., \( r_{TP}(t) = \frac{n_{TP}(t)}{n_P} \). It is common to use receiver operating characteristic (ROC) curve to evaluate the performance of link prediction (Zhang et al., 2015; Zhao et al., 2017). In our numerical experiments, we vary \( t \) from 0.01 to 0.99 with a step size 0.01, and compare ROC curves produced by SDP-BF with those produced by NBS. For the quantile threshold parameter \( h \) of NBS, Zhang et al. (2013) have shown that \( h = C \sqrt{\frac{\log n}{n}} \) for any \( C \in (0, 1] \) works in theory and suggested varying \( C \) in the range \( 5^{-4}, 5^{-3.5}, \ldots, 1 \). In our experiments, we try all nine choices of \( C \) for NBS. For the tuning parameter \( K \) in SDP-BF, we propose a cross-validation approach to choose it.

### 4.3.1 Cross-validation for Choosing the Tuning Parameter

We now introduce our cross-validation approach to choosing the tuning parameter \( K \) for SDP-BF. We adopt the node pair splitting technique proposed in Hoff (2008), where the node pairs, instead of the nodes, are split into a training set and a test set. For a \( V \)-fold cross-validation, we randomly split the set of node pairs into \( V \) roughly equal sized parts. For each \( 1 \leq v \leq V \), let \( C^{(v)} \) denote the set of node pairs in \( v \)th part. Let \( O^{(-v)} \) be the indicator matrix of all node pairs except those belonging to the \( v \)th part, i.e.,

\[
O_{ij}^{(-v)} = \begin{cases} 
1 & \text{if } (i, j) \notin C^{(v)}, \\
0 & \text{if } (i, j) \in C^{(v)}. 
\end{cases}
\]
Let us consider a specific fold \( v \) and a candidate value \( k \) for \( K \). We use

\[
A_{\text{obs}}^{(-v)} := A_{\text{obs}} \circ O^{(-v)}
\]

as the training adjacency matrix and test the performance of link prediction on node pairs in \( C^{(v)} \) using corresponding entries in \( A_{\text{obs}} \) as the true connectivity pattern. Specifically, let \( \hat{X}^{(k,v)} \) denote the solution to the SDP (4.3) when using \( A_{\text{obs}}^{(-v)} \) for \( A \) and \( k \) for \( K \). Let

\[
\hat{P}^{(k,v)} := \frac{1}{2} (A_{\text{obs}}^{(-v)} \hat{X}^{(k,v)} + \hat{X}^{(k,v)} A_{\text{obs}}^{(-v)})
\]

denote the corresponding estimated edge probability matrix. We then evaluate the performance of using \( \hat{P}^{(k,v)} \) to predict the links between all node pairs in \( C^{(v)} \) by treating the corresponding entries in \( A_{\text{obs}} \) as the truth. Let

\[
N_{TP}^{(v)} := |\{(i, j) : (A_{\text{obs}})_{ij} = 1, O_{ij}^{(-v)} = 0\}|
\]

be the total number of connected node pairs in the \( v \)th part of \( A_{\text{obs}} \) and

\[
N_{FP}^{(v)} := |\{(i, j) : (A_{\text{obs}})_{ij} = 0, O_{ij}^{(-v)} = 0\}|
\]

be the total number of disconnected node pairs in the \( v \)th part of \( A_{\text{obs}} \). For a threshold \( 0 < t < 1 \), let

\[
N_{TP}^{(k,v)}(t) := |\{(i, j) : \hat{P}_{ij}^{(k,v)} > t, (A_{\text{obs}})_{ij} = 1, O_{ij}^{(-v)} = 0\}|
\]

be the number of true positives and

\[
N_{FP}^{(k,v)}(t) := |\{(i, j) : \hat{P}_{ij}^{(k,v)} > t, (A_{\text{obs}})_{ij} = 0, O_{ij}^{(-v)} = 0\}|
\]

be the number of false positives. For each candidate value \( k \) for \( K \), we repeat the above procedure for each \( v = 1, \cdots, V \). We define the cross-validated false positive
rate and the cross-validated true positive rate for a candidate value \( k \) and a threshold \( t \) as

\[
    r_{FP}^{(k)}(t) := \frac{\sum_{v=1}^{V} n_{FP}^{(k,v)}(t)}{\sum_{v=1}^{V} n_{N}^{(v)}}
\]

and

\[
    r_{TP}^{(k)}(t) := \frac{\sum_{v=1}^{V} n_{TP}^{(k,v)}(t)}{\sum_{v=1}^{V} n_{P}^{(v)}},
\]

respectively. We vary \( t \) from 0.01 to 0.99 with a step size 0.01 and obtain the ROC curve for a candidate \( k \). Finally, we compare ROC curves obtained from the above procedure using different values of \( k \), and choose the \( k \) that produces the best ROC curve as the value of \( K \) to be used in the link prediction task. In our numerical experiments, we choose the number of folds \( V \) to be 5 and try candidate values of \( K \) to be 10, 20, 30, 40, 50.

### 4.3.2 Synthetic Networks

Let us first consider networks generated from the three graphon functions listed in Section 4.2. We set the number of nodes \( n = 500 \) and the missing rate of links \( p = 0.3 \). Top panel of Figure 4.7 shows ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of \( K \) under graphon 1. We see that \( K = 10 \) is slightly better than the other values of \( K \) and we choose \( K = 10 \) for SDP-BF in link prediction. Bottom panel of Figure 4.7 shows ROC curves for link prediction using SDP-BF with \( K = 10 \) and NBS with different values of \( C \) under graphon 1. We see that the performance of SDP-BF with \( K = 10 \) is similar to the best performance of NBS, which occurs when \( C = 1 \). The results under graphon 2 are shown in Figure 4.8. The result of 5-fold cross-validation shown in the top panel suggests the choice of \( K = 10 \) for SDP-BF. From the ROC curves for link
prediction shown in the bottom panel, we see that SDP-BF with $K = 10$ performs much better than NBS under all values of $C$. Figure 4.9 presents the results under graphon 3. Again, 5-fold cross-validation result shown in the top panel suggests the choice of $K = 10$ for SDP-BF. The link prediction result shown in the bottom panel demonstrates that the performance of SDP-BF with $K = 10$ is similar to the best performance of NBS, which occurs when $C = 1$.

4.3.3 Real Data Application

Finally, we test the performance of SDP-BF on the Caltech network introduced in Section 3.4.2, and compare its performance with that of NBS. We set the missing rate of links $p = 0.1$. Top panel of Figure 4.10 shows ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$. We see that SDP-BF with $K = 20, 30, 40, 50$ show similar performance, and we choose $K = 20$, which yields smaller degrees of freedom for SDP-BF compared to the other choices. Bottom panel of Figure 4.10 shows ROC curves for link prediction using SDP-BF with $K = 20$ and NBS with different values of $C$ on the Caltech network. We see that SDP-BF with $K = 20$ outperforms NBS under all values of $C$. 

78
Figure 4.7: Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ under graphon 1. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 10$ chosen from cross-validation and NBS with different values of $C$ under graphon 1.
Figure 4.8: Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ under graphon 2. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 10$ chosen from cross-validation and NBS with different values of $C$ under graphon 2.
Figure 4.9: Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ under graphon 3. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 10$ chosen from cross-validation and NBS with different values of $C$ under graphon 3.
Figure 4.10: Top panel: ROC curves for link prediction from 5-fold cross-validation using SDP-BF with different values of $K$ on the Caltech network. Bottom panel: ROC curves for link prediction using SDP-BF with $K = 20$ chosen from cross-validation and NBS with different values of $C$ on the Caltech network.
Chapter 5: Summary and Future Work

5.1 Summary

In this dissertation, we propose semidefinite programming approaches to network community detection and edge probability estimation. Interestingly, despite different motivations, it turns out that the proposed approaches to the two problems are based on the same SDP with appropriately chosen input matrices. Our SDP relies on a semidefinite relaxation of the set of normalized clustering matrices. We derive optimality conditions of the SDP from a geometric point of view and deal with the Fantope as a whole. This enables us to analyze the SDP by making use of the properties of the Fantope. We design an efficient ADMM algorithm for solving the SDP. By using the Frank-Wolfe algorithm to solve the subproblem, we avoid full spectral decompositions of symmetric matrices and our algorithm only requires to compute a small number of leading eigenvectors of symmetric matrices. This makes our algorithm more scalable than other SDP-based methods for community detection.

For community detection, the SDP is derived from the partition criterion of maximizing the sum of average intra-cluster similarities over all clusters. The feasible set of our SDP is contained in the Fantope, which enables us to connect our SDP to sparse PCA and spectral clustering. Unlike previously proposed SDPs for community
detection which use the adjacency matrix as the input matrix, we also consider using other input matrices to the SDP for community detection. Motivated by the connection between our SDP and spectral clustering, we consider using the symmetric normalized graph Laplacian as the input matrix to the SDP. Inspired by the fact that our SDP is equivalent to an SDP relaxation of the $k$-means problem when the input matrix is chosen to be the Gram matrix for multivariate observations, we consider using the squared adjacency matrix as the input matrix to the SDP. This builds an interesting connection between network clustering and multivariate data clustering. Using optimality conditions, we analyze the population consistency of the SDP using different input matrices under various network models. We show the exact recovery property of our SDP under strongly assortative stochastic block models when the input matrix is chosen to be the adjacency matrix. These results are complementary to existing results about the performance guarantee of the SDP for multivariate data clustering. Simulation studies and real data applications show that the proposed SDP has good empirical performance for community detection. First, our SDP with squared adjacency matrix as the input matrix is the only method that succeeds in recovering communities for mixed networks with both assortative and disassortative communities. Second, our SDP with the adjacency matrix, the squared adjacency matrix, or the symmetric normalized graph Laplacian matrix as the input matrix works well for networks generated from weakly assortative SBMs, while the majority of other considered methods fail. Third, our SDP with the symmetric normalized graph Laplacian matrix as the input matrix achieves the best misclassification error rates for community detection on real Facebook collegiate networks.
For network edge probability estimation, the SDP is derived from a least squares criterion. Since the solution to our SDP is a symmetric doubly stochastic matrix, it can be used as a smoother matrix to smooth the adjacency matrix. The estimated edge probability matrix is the smoothed adjacency matrix. Due to the fact that our SDP with the squared adjacency matrix as the input matrix is a relaxed version of the $k$-means problem applied to the rows of the adjacency matrix, we connect our SDP-based method of estimating edge probabilities to the neighborhood smoothing method. Numerical experiments show that the proposed SDP approach to estimating network edge probabilities outperforms the neighborhood smoothing method on simulated networks. We also apply the proposed method to the link prediction problem, and empirically show that it outperforms the neighborhood smoothing method on both simulated networks and a real Facebook network.

5.2 Future Directions

There are several interesting and important future directions on this research. First, it is still unknown if the proposed SDP with the adjacency matrix as the input matrix can achieve weak consistency for sparse networks with bounded expected degrees. The technique based on the Grothendieck’s inequality used in Guédon and Vershynin (2016) is too loose for our SDP. Recently, Yan and Sarkar (2016) considered a tighter SDP than our SDP, which adds the constraint $X \leq 1/n_{\text{min}}$, where $n_{\text{min}}$ is the minimal community size. This additional constraint enables them to show weak consistency of the SDP for sparse networks following the similar arguments to those in Guédon and Vershynin (2016). However, this constraint is artificial and it is hard to know the minimal community size in practice. Second, we have shown
that our SDP with the symmetric normalized graph Laplacian matrix as the input matrix achieves the best misclassification error rates for community detection on real networks. It would be interesting to analyze the performance of the SDP using the symmetric normalized graph Laplacian matrix as the input matrix and to understand why it has superior performances to other methods for community detection. Third, in practice, we often observe multiple networks for a set of nodes from different perspectives. For example, for a set of Twitter users, we can create a follower network, a mention network, and a retweet network (Greene and Cunningham, 2013). An interesting question would be how to improve the performance of community detection by combining different networks compared to using a single network. A natural solution is to use a single weighted matrix combining adjacency matrices of different networks as the input matrix to the SDP. The key to this approach is to design a principled way to choosing the weights for different networks. A related problem is community detection with covariates information for nodes. In practice, besides a network, we often have additional covariates information for each of the nodes. How can we make use of covariates information for community detection? Recall that the SDP works for multivariate data clustering if the input matrix is chosen to be the Gram matrix or other appropriate kernel inner product matrices. Hence, a natural solution is to use a single weighted input matrix combining the adjacency matrix of the network and the Gram matrix formed by covariates information of nodes (Yan and Sarkar, 2016). Again, the key to this approach is to choose the weights. Finally, it would be very interesting to analyze theoretical properties of our method of edge probability matrix estimation. It is still unknown whether the minimax rate of $O_P(\frac{\log n}{n})$ established by Gao et al. (2013) can be achieved by a polynomial time
algorithm. Given the superior empirical performance of our SDP-based method, it would be very interesting to establish its theoretical error rate.
Appendix A: Proof of Theorem 3.10

We provide a proof of Theorem 3.10. For relatively dense networks whose expected degrees grow no slower than \( \log n \), we have the following concentration inequality.

**Lemma A.1.** (Proposition 4.1 in Amini and Levina, 2014). Let \( A = (A_{ij}) \) be an \( n \times n \) symmetric binary matrix with independent lower triangle and zero diagonal. There are universal positive constants \( (C, C', c, r) \) such that if

\[
\max_{ij} \text{Var}(A_{ij}) \leq \sigma^2, \text{ for } n\sigma^2 \geq C' \log n
\]

then with probability at least \( 1 - cn^{-r} \),

\[
\|A - \mathbb{E} A\|_2 \leq C\sigma \sqrt{n}.
\]

Using Lemma A.1, we can derive the following probabilistic upper bound for \( \|A - \mathbb{E} A\|_2 \) under planted partition models.

**Lemma A.2.** Let \( A \in \{0,1\}^{n \times n} \) be drawn from a planted partition model with \( n \) nodes and \( K \) communities of sizes \( n_1, \ldots, n_K \). The within-community edge probability is denoted by \( p \) and the between-community edge probability is denoted by \( q \). Let \( n^+ := \max_k n_k, n^- := \min_k n_k \). Assume that \( n^- > e \). If \( p \geq (C' \log n^-)/n^- \) and \( q \geq (C' \log n)/n \), then with probability at least \( 1 - cK(n^-)^{r} - cn^{-r} \),

\[
\|A - \mathbb{E} A\|_2 \leq C(\sqrt{pn^+} + \sqrt{qn^-}). \tag{A.1}
\]
Proof. The proof is a direct generalization of the proof of Corollary F.1 in Amini and Levina (2014) to unequal sized models.

To obtain a uniform probabilistic lower bound for the right hand side of (3.15), we derive the following lemma.

**Lemma A.3.** Let $A \in \{0, 1\}^{n \times n}$ be drawn from a planted partition model with $n$ nodes and $K$ communities of sizes $n_1, \ldots, n_K$. The within-community edge probability is denoted by $p$ and the between-community edge probability is denoted by $q$. Let $n^+ := \max_k n_k, n^- := \min_k n_k$. For any $c_1 > 1$ and $c_2 > 2$, define $t := \sqrt{\frac{4c_1 \log n}{p(n^+ - 1)}}$ and $r := \sqrt{\frac{4c_2 \log n}{qn^+}}$. If $t, r \leq 3$, then,

$$|d_i(C_k) - (n_k - 1)p| \leq (n_k - 1)pt$$

holds uniformly for all $i \in C_k, \forall k$ with probability at least $1 - 2n^{-(c_1 - 1)}$, and

$$|d_i(C_k) - n_kq| \leq n_kqr$$

holds uniformly for all $i \in C_l, \forall l \neq k$ with probability at least $1 - 2n^{-(c_2 - 2)}$.

Proof. The proof is a direct generalization of the proof of Lemma F.1 in Amini and Levina (2014) to unequal sized models.

**Proof of Theorem 3.10.** By Lemma A.3, we have that with probability at least $1 - 2n^{-(c_1 - 1)} - 2n^{-(c_2 - 2)}$, the right hand side of (3.15) is bounded below by

$$\min_{l \neq k} \frac{1}{n_k + n_l} (2n_k(n_l - 1)p(1 - t) + 2n_l(n_k - 1)p(1 - t) - n_l(n_k - 1)p(t + 1) - n_k(n_l - 1)p(t + 1) - 2n_kn_lq(r + 1) - 2n_ln_kq(r + 1) + 2n_kn_lq(1 - r))$$

$$\geq n^{-(p - 3pt - q - 3qr)} + 3pt - p$$

$$= n^{-(p - q)} - 3\sqrt{4c_1p(n^- - 1)\log n} - 3\sqrt{4c_2qn^-\log n} - p,$$

(A.2)
where $n^− := \min_k n_k$. By (3.15), (A.1) and (A.2), if

$$C(\sqrt{mn^+} + \sqrt{qn^−}) < n^−(p − q) − 3\sqrt{4c_1p(n^− − 1)\log n} − 3\sqrt{4c_2qn^− \log n},$$

then with probability at least $1 − cK(n^−)^{−r} − cn^{−r} − 2n^{−(c_1 − 1)} − 2n^{−(c_2 − 2)}$, the sufficient condition (3.15) is satisfied, and thus $X_0$ is the unique solution to SDP-BF.
Bibliography


