Recovery Guarantees for Graph Clustering Problems

by

Jimit Majmudar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Quentin Berthet Research Scientist, Google Research Brain Team, Paris
Supervisor(s):	Stephen Vavasis Professor, Combinatorics and Optimization University of Waterloo
Internal Member:	Chaitanya Swamy Professor, Combinatorics and Optimization University of Waterloo
Internal Member:	Henry Wolkowicz Professor, Combinatorics and Optimization University of Waterloo
Internal-External Member:	Gautam Kamath Assistant Professor, Computer Science University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Graph clustering is widely-studied unsupervised learning problem in which the task is to group similar entities together based on observed pairwise entity interactions. This problem has applications in diverse domains such as social network analysis and computational biology. There are multiple ways to formalize a graph clustering problem. In this thesis, using tools from convex optimization, we develop algorithms for two specific graph clustering formulations – *Overlapping Community Detection* and *Correlation Clustering*. We study these formulations using the provable recovery paradigm which requires establishing theoretical guarantees for recovery of a certain ground truth clustering as posited by a chosen generative model.

In the Overlapping Community Detection problem, we expect clusters in the input graph to potentially overlap, i.e. share some common nodes. For this problem, often a *pure nodes* assumption is made in literature which requires each cluster to have a node that belongs exclusively to that cluster. This assumption, however, may not be satisfied in practice. We propose a linear-programming-based algorithm to provably recover overlapping communities in weighted graphs without explicitly making the pure nodes assumption. We demonstrate the success of our algorithm on synthetic and real-world datasets. In the Correlation Clustering problem, we wish to determine non-overlapping clusters in the input graph without any prior knowledge of the number of clusters. We introduce a new graph generative model based on generating feature vectors/embeddings for the nodes in the graph which are interpreted as latent variables in the model, and propose a tuning-parameter-free semidefinite-programming-based algorithm to recover nodes with sufficiently strong cluster membership. We make progress towards showing that the proposed algorithm is provably robust.

Acknowledgements

I would like to thank my advisor, Steve Vavasis, for his constant support and guidance. I am grateful to Steve for always being present to patiently provide advice, especially on research. I would also like to thank the thesis examining committee members, Quentin Berthet, Gautam Kamath, Chaitanya Swamy, and Henry Wolkowicz, for their valuable feedback.

Thanks to my friends, Hemant, Stefan, Anirudh, Sharat, Priya, Abhinav, Cedric, Dhinakaran, Akshay, Retnika, and Archana, for all the laughs and technical discussions.

I am thankful to my parents, Heena Majmudar and Rankesh Majmudar, and my sister, Chaitasi Majmudar, for their unwavering understanding, patience, support, and encouragement throughout all my academic pursuits, especially the PhD.

Dedication

Dedicated to my mother, Heena Majmudar.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Suppose we are given a collection of text documents and we wish to categorize the documents based on the topic(s) they cover. Or suppose we are given a social network comprising social agents and their pairwise interaction frequencies, and we wish to determine social circles within the network. One abstraction to represent the problem data for such problems is graphs. In particular, we construct a graph in which the nodes represent the entities, and the edges encode some measure of pairwise relationships. The aforementioned problems can then be thought of as determining groups of nodes such that the nodes within the same group are more similar than dissimilar and nodes in different groups are more dissimilar than similar. Due to their ubiquitous nature, graph problems of this style have received attention from diverse research communities such as computer science [14, 75], mathematics [61, 77], statistics [70, 80], physics [63, 73], and biology [58, 66, 67], traditionally, and more contemporary ones such as network science [26, 71] and machine learning [12, 53]. As a result of this diverse interest, sometimes the same idea or concept may have different names. For instance, *graph* is synonymous with *network*, *cluster* is synonymous with *community*, and *graph clustering* is often synonymous with *community detection* or *graph partitioning*; in this thesis, we use the terms *cluster* and *community*, and the phrases *graph clustering* and *community detection* interchangeably. Another consequence of this diverse interest is the variety in the goals achieved by different works. Some works develop heuristic algorithms that are customized for a specific application domain or even a specific problem. On the other hand, some works propose general-purpose algorithms and develop theoretical guarantees for their performance. In this thesis, we focus on two types of clustering problems – one in which we expect the clusters in the graph to have overlaps in terms of shared nodes, and other in which we do not have a priori knowledge about the number of clusters in the graph. We consider these problems in their general form,

i.e. not restricted to a particular application domain, and we provide theoretical analyses regarding the performance of our proposed clustering algorithms.

1.1 Thesis Outline and Contributions

Chapter 2 introduces the mathematical tools used in the rest of the thesis. In Chapter 3, we provide an in-depth discussion on clustering problems which is relevant to this thesis. Chapter 4 contains our work on overlapping community detection, i.e. a clustering problem in which we expect clusters in the input graph to potentially overlap in terms of shared nodes. Our contributions include a simple provable algorithm for recovering the community memberships of each node in weighted graphs. Unlike most existing provable methods, our algorithm: (1) does not explicitly require each community to have a node which belongs exclusively to that community, (2) is relatively easy to implement in practice as it does not involve multiple tuning parameters, and (3) is rooted in linear programming, on which a rich body of literature already exists. Chapter 5 contains our work on Correlation Clustering, i.e. a clustering problem in which we assume no prior knowledge about the number of clusters in the input graph. Our contributions include a simple algorithm for determining clusters in weighted graphs. Using a combination of theoretical analyses and computational experiments, we make progress towards establishing robustness of the proposed algorithm in the presence of asymmetric noise in the input graph. Our algorithm: (1) is relatively easy to implement in practice as it is entirely parameter-free, i.e. involves no tuning parameter, and (2) is rooted in semidefinite programming, on which also a rich body of literature already exists. Lastly, we finish with some conclusions and directions for future work in Chapter 6.

1.2 Notation

For any natural number $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$, $\mathbb{R}^n$ and $\mathbb{R}^n_+$ denote the vector spaces of n-dimensional real-valued vectors and n-dimensional real-valued non-negative vectors respectively, and $\mathbb{S}^n$ denotes the vector space of $n \times n$ symmetric, real-valued matrices. For any $m, n \in [n]$, $\mathbb{R}^{m\times n}$ and $\mathbb{R}^{m\times n}_+$ denote the vector spaces of $m \times n$ real-valued matrices and $m \times n$ real-valued non-negative matrices. Let M be any matrix. We use $\mathbf{m}_i$ and $\mathbf{m}^i$ to denote its column i and the transpose of its row i respectively, and M_{ij} or $M(i, j)$ to denote its entry ij; for any set $\mathcal{R} \subseteq \mathbb{N}$, $M(\mathcal{R}, :)$ (resp. $M(:, \mathcal{R})$) denotes the submatrix of M containing all columns (resp. rows) but only the rows (resp. columns)

indexed by $\mathcal{R}$. For any two sets $\mathcal{R}, \mathcal{S} \subseteq \mathbb{N}$, $M(\mathcal{R}, \mathcal{S})$ denotes the submatrix of M containing the rows indexed by $\mathcal{R}$ and the columns indexed by $\mathcal{S}$. We use $\max(M)$ to denote its largest value, $\|M\|_{\max}$ to denote its largest absolute value, and M_+ and M_- to denote the projections onto the cone of non-negative and non-positive matrices respectively. To show that M is a symmetric positive semidefinite (resp. positive definite) matrix, we use the notation $M \succeq 0$ (resp. $M \succ 0$). If M is an $n \times n$ square matrix, $diag(M)$ denotes a vector in $\mathbb{R}^n$ whose entry i is M_{ii} for each $i \in [n]$, and $Diag(M)$ denotes an $n \times n$ matrix whose diagonal is equal $diag(M)$ and whose each off-diagonal entry is zero. Let $\mathbf{v}$ be any vector. We use v_i or $v(i)$ to denote its entry i; for any set $\mathcal{R} \subseteq \mathbb{N}$, $\mathbf{v}(\mathcal{R})$ denotes the subvector of $\mathbf{v}$ containing entries indexed by $\mathcal{R}$. We use $\max(\mathbf{v})$ to denote its largest value, $\|\mathbf{v}\|_\infty$ to denote its largest absolute value, and $\mathbf{v}_+$ and $\mathbf{v}_-$ to denote the projections onto the cone of non-negative and non-positive vectors respectively. If $\mathbf{v} \in \mathbb{R}^n$, $Diag(\mathbf{v})$ denotes an $n \times n$ matrix whose diagonal is equal to $\mathbf{v}$ and whose each off-diagonal entry is zero. If $\mathbf{v}$ is an entry-wise non-negative vector, then for any $p \in \mathbb{R}$, we denote by $\mathbf{v}^{\circ p}$ the vector obtained by exponentiating each entry of $\mathbf{v}$ with p.

For any two vectors $\mathbf{u}, \mathbf{v}$ of identical dimensions, $\langle \mathbf{u}, \mathbf{v} \rangle$ denotes the Euclidean inner product $\mathbf{u}^T\mathbf{v}$, and for any two matrices X, Y of identical dimensions, $\langle X, Y \rangle$ denotes the trace inner product $trace(X^TY)$. If $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ such that for each $i \in [n]$, $u_i \leq v_i$, then we use $[\mathbf{u}, \mathbf{v}]$ to denote the set $\{\mathbf{x} \in \mathbb{R}^n : u_i \leq x_i \leq v_i \ \forall i \in [n]\}$.

We use $\|\cdot\|$ to denote the ℓ_2-norm for vectors and the spectral norm (largest singular value) for matrices. We use $\|\cdot\|_F$ and $\|\cdot\|_\infty$ to denote the Frobenius norm and the largest row ℓ_1-norm of a matrix respectively. I and E denote the identity matrix and the matrix with each entry set to one respectively whose dimensions will be clear from the context. For any positive integer i, $\mathbf{e}^i$ and $\mathbf{e}_i$ denote row i and column i of the identity matrix respectively and $\mathbf{e}$ denotes the vector with each entry set to one; the dimension of these vectors will be clear from context.

For any graph $G = (V, W)$, i.e. graph with node set V and weighted adjacency matrix W, $L(G)$ denotes the graph Laplacian matrix defined to be $Diag(W\mathbf{e}) - W$. For any subset $V' \subseteq V$ of nodes, $G[V']$ denotes the subgraph of G induced by nodes in V'.

Chapter 2

Background

In this chapter, we discuss the mathematical tools that are relevant to our contributions discussed in the subsequent chapters.

2.1 Convex Optimization

We begin by defining the notion of convexity for sets and functions. Let $\mathcal{V}$ be a Euclidean vector space of n dimensions.

Definition 2.1 (Convex set)**.** *A set $C \subseteq \mathcal{V}$ is said to be convex if for any two points $x, y \in C$ and for any $\lambda \in [0, 1]$, we have that $\lambda x + (1 - \lambda)y \in C$.*

Definition 2.2 (Convex function)**.** *Let $C \subseteq \mathcal{V}$ be a convex set. A function $f : C \to \mathbb{R}$ is said to be convex if for any two points $x, y \in C$ and for any $\lambda \in [0, 1]$, we have that $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$.*

Visually, the line segment connecting any two points in a convex set must lie entirely within the set, and the line segment connecting any two points on the plot of a convex function must lie above the plot. Now we define notions of differentiability and gradient.

Definition 2.3 (Differentiability and Gradient)**.** *Let $f : \mathcal{V} \to \mathbb{R}$ and let $x \in \mathcal{V}$. If there exists a unique $g \in \mathcal{V}$ such that*

$$\lim_{\|h\| \to 0} \frac{f(x + h) - f(x) - \langle g, h \rangle}{\|h\|} = 0$$

then f is said to be differentiable at x, and g is called the gradient of f at x.

The standard notation to represent the gradient of f at x is $\nabla f(x)$. The following fact provides a representation of gradient for certain cases that is relatively easier to deal with.

Fact 2.1 ([57]). *If a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at x, then*

$$\nabla f(x) = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix}$$

where for each $i \in [n]$, $\partial f/\partial x_i$ is the partial derivative of f with respect to x_i.

The domain of convex optimization can be divided into *unconstrained convex optimization* versus *constrained convex optimization*. In the unconstrained setting, typically we have a convex function $f : \mathcal{V} \to \mathbb{R}$ and we are interested in solving

$$\min_{x \in \mathcal{V}} f(x).$$

That is, we wish to determine the smallest possible value the function f can take, and also the point(s) in $\mathcal{V}$ at which the smallest function value is attained. The convexity of f establishes the following fact which is beneficial for solving the above unconstrained convex optimization problem.

Fact 2.2 ([59]). *If $f : \mathcal{V} \to \mathbb{R}$ is a differentiable convex function, then*

$$\{x \in \mathcal{V} : \nabla f(x) = \mathbf{0}\} = \arg\min_{x \in \mathcal{V}} f(x).$$

In the constrained setting, we have a convex function $f : \mathcal{V} \to \mathbb{R}$ and a convex set $C \subseteq \mathcal{V}$ and we are interested in solving

$$\min_{x \in C} f(x).$$

That is, we wish to determine the smallest possible value the function f can take on a *feasible set* C, and also the point(s) in C at which the smallest function value is attained. Note that, conventionally, if a minimization (resp. maximization) problem is infeasible, then its optimal value is set to $+\infty$ (resp. $-\infty$). The optimality conditions for the constrained case are not as easily stated, in full generality, as those for the unconstrained case given in Fact 2.2. In this thesis, we are interested in two special cases of constrained convex optimization, *linear programming* and *semidefinite programming*, which are covered in Sections 2.1.1 and 2.1.2 respectively.

2.1.1 Linear Programming

Let $\mathbf{c} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, then the *primal linear program* in standard equality form is defined as the constrained optimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{P-LP}$$

where the last inequality above implies entry-wise non-negativity requirement. Note that the objective function and the feasible set in the above problem are both convex, and therefore linear programming is a special case of convex optimization. For (P-LP), we define the *Lagrangian dual program*, referred to as just dual program henceforth, as

$$\begin{aligned} \max_{\mathbf{y}} \quad & \mathbf{b}^T \mathbf{y} \\ \text{s.t.} \quad & A^T \mathbf{y} \leq \mathbf{c}. \end{aligned} \tag{D-LP}$$

Let p^* and d^* be the optimal values for the primal program (P-LP) and the dual program (D-LP) respectively. The following facts elucidate the relationship between the two programs.

Fact 2.3 (Weak Duality [59])**.** *$d^* \leq p^*$.*

Fact 2.4 (Strong Duality [59])**.** *If p^* is finite, then $p^* = d^*$. Moreover, p^* is attained at a primal feasible solution $\mathbf{x}^*$, and d^* is attained at a dual feasible solution $\mathbf{y}^*$ such that for each $i \in [n]$, $x_i^*(\mathbf{a}_i^T \mathbf{y}^* - c_i) = 0$.*

Fact 2.4 provides a necessary condition for primal and dual optimal solutions. In fact, thanks to convexity, the same condition also acts as a sufficient condition for optimality, as shown in the following fact.

Fact 2.5 (Karush-Kuhn-Tucker (KKT) Optimality Conditions [59])**.** *Suppose p^* is finite. Then $\mathbf{x}^* \in \mathbb{R}^n$ and $\mathbf{y}^* \in \mathbb{R}^m$ are primal and dual optimal solutions respectively if and only if they satisfy:*

- *Primal feasibility:* $A\mathbf{x}^* = \mathbf{b}$, $\mathbf{x}^* \geq \mathbf{0}$.
- *Dual feasibility:* $A^T \mathbf{y}^* \leq \mathbf{c}$.
- *Complementary slackness:* $x_i^*(\mathbf{a}_i^T \mathbf{y}^* - c_i) = 0$, $\forall i \in [n]$.

2.1.2 Semidefinite Programming

We first introduce notions of positive semidefiniteness and positive definiteness.

Definition 2.4 (Positive Semidefinite Matrix)**.** *A symmetric matrix $M \in \mathbb{S}^n$ is said to be positive semidefinite if for any $\mathbf{x} \in \mathbb{R}^n$, we have $\mathbf{x}^T M \mathbf{x} \geq 0$.*

Definition 2.5 (Positive Definite Matrix)**.** *A symmetric matrix $M \in \mathbb{S}^n$ is said to be positive definite if for any $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, we have $\mathbf{x}^T M \mathbf{x} > 0$.*

It is standard notation to denote the positive semidefiniteness (resp. positive definiteness) of matrix M using $M \succeq 0$ (resp. $M \succ 0$). Now we highlight some sufficient conditions for positive semidefiniteness that are useful in the context of this thesis.

Definition 2.6 (Schur Complement)**.** *Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{m \times n}$, $D \in \mathbb{R}^{m \times m}$, and*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

If A is invertible, then the Schur complement of block A of matrix M is the matrix $D - CA^{-1}B$. Similarly, if D is invertible, then the Schur complement of block D of matrix M is the matrix $A - BD^{-1}C$.

Fact 2.6 (Schur Complement Condition [76])**.** *Let $A \in \mathbb{S}^n$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{S}^m$, and*

$$M = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}.$$

If A is positive definite, then M is positive semidefinite if and only if the Schur complement $C - B^T A^{-1} B$ is positive semidefinite. Similarly, if C is positive definite, then M is positive semidefinite if and only if the Schur complement $A - BC^{-1}B^T$ is positive semidefinite.

Fact 2.7 (Diagonally Dominant Condition [76])**.** *If $M \in \mathbb{S}^n$ such that for each $i \in [n]$*

$$M_{ii} \geq \sum_{j \in [n] \setminus \{i\}} |M_{ij}|$$

then M is positive semidefinite.

Let $C, A_1, \ldots, A_m \in \mathbb{S}^n$ for some $m \in \mathbb{N}$ and $\mathbf{b} \in \mathbb{R}^m$, then the *primal semidefinite program* is defined as the constrained optimization problem

$$\begin{aligned} \min_X \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \langle A_i, X \rangle = b_i \ \forall i \in [m] \\ & X \succeq 0. \end{aligned} \tag{P-SDP}$$

Note that the objective function and the feasible set in the above problem are both convex, and therefore semidefinite programming is a special case of convex optimization. For (P-SDP), we define the *Lagrangian dual program*, referred to as just dual program henceforth, as

$$\begin{aligned} \max_{\mathbf{y}} \quad & \mathbf{b}^T \mathbf{y} \\ \text{s.t.} \quad & C - \sum_{i \in [m]} y_i A_i \succeq 0 \end{aligned} \tag{D-SDP}$$

Let p^* and d^* be the optimal values for the primal program (P-SDP) and the dual program (D-SDP) respectively. The following facts elucidate the relationship between the two programs.

Fact 2.8 (Weak Duality [76])**.** $d^* \leq p^*$.

Fact 2.9 (Strong Duality [76])**.** *If p^* is finite and if (P-SDP) has a feasible solution which is positive definite, then $p^* = d^*$ and d^* is attained at a dual feasible solution $\mathbf{y}^*$. Moreover, if p^* is attained at a primal feasible solution X^*, then*

$$\left\langle X^*, C - \sum_{i \in [m]} y_i^* A_i \right\rangle = 0.$$

Fact 2.9 provides a necessary condition for primal and dual optimal solutions. In fact, thanks to convexity, the same condition also acts as a sufficient condition for optimality, as shown in the following fact.

Fact 2.10 (Karush-Kuhn-Tucker (KKT) Optimality Conditions [76])**.** *Suppose (P-SDP) has a feasible solution which is positive definite and (D-SDP) has a feasible solution $\mathbf{y}$ such that $C - \sum_{i \in [m]} y_i A_i$ is positive definite. Then $X^* \in \mathbb{S}^n$ and $\mathbf{y}^* \in \mathbb{R}^m$ are primal and dual optimal solutions respectively if and only if they satisfy:*

- *Primal feasibility:* $\langle A_i, X^* \rangle = b_i \ \forall i \in [m]$, $X^* \succeq 0$.

- *Dual feasibility:* $C - \sum_{i \in [m]} y_i^* A_i \succeq 0$.
- *Complementary slackness:* $\left\langle X^*, C - \sum_{i \in [m]} y_i^* A_i \right\rangle = 0$.

2.2 Dirichlet Distribution

The Dirichlet distribution is a continuous probability distribution over the unit simplex, because of which it is sometimes used in statistics as a distribution over discrete probabilities. The distribution is parametrized using $k \in \mathbb{N}$, the number of categories, and a positive vector $\boldsymbol{\alpha} \in \mathbb{R}^k$ which, intuitively, represents the affinity towards each of the k categories. For any $\mathbf{x} \in \mathbb{R}^k$ in the unit simplex, i.e. $\mathbf{e}^T\mathbf{x} = 1$ and $\mathbf{x} \geq \mathbf{0}$, the probability density function for Dirichlet distribution is defined as

$$f(\mathbf{x}, \boldsymbol{\alpha}, k) = \frac{\Gamma(\mathbf{e}^T\boldsymbol{\alpha})}{\prod_{i \in [k]} \Gamma(\alpha_i)} \prod_{i \in [k]} x_i^{\alpha_i - 1}$$

where Γ denotes the gamma function, defined as

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$$

for any complex number z with positive real part. The gamma function is an extension of the factorial function to complex numbers in the sense that for any natural number n, $\Gamma(n) = (n-1)!$.

Fact 2.11 ([43])**.** *Define* $\alpha_0 := \sum_{i \in [k]} \alpha_i$*. For each* $i \in [k]$*,*

$$\mathbb{E}[x_i] = \frac{\alpha_i}{\alpha_0}$$
$$Var(x_i) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}.$$

Using Fact 2.11, we can intuitively understand the nature of Dirichlet distribution for some special parameter settings. For instance, if all values in vector $\boldsymbol{\alpha}$ are equal, i.e. $\boldsymbol{\alpha} = t\mathbf{e}$ for some scalar $t > 0$, then the Dirichlet vectors are spread symmetrically around the simplex center $\mathbf{e}/k$. Moreover the magnitude of t determines the spread around the simplex center, i.e. as t increases, the points tend to get closer to the simplex center. Figure 2.1 demonstrates this observation.

(a) $t = 0.3$ (b) $t = 5$ (c) $t = 10$

Figure 2.1: The unit simplex in $\mathbb{R}^k$ with 60 points sampled according to the Dirichlet distribution with $k = 3$ and $\boldsymbol{\alpha} = t\mathbf{e}$ with increasing t.

2.3 Concentration Inequalities

As the name suggests, concentration inequalities provide information about how a random variable concentrates around its expected value. Typically, a concentration inequality gives an upper bound on the probability with which a random variable is sufficiently far from a fixed value such as its expected value. Such inequalities are useful in non-asymptotic analysis of random generative models. Note that in asymptotic analysis, we are usually interested in only the limiting behavior of certain probabilities as some measure of the size of the problem, called n, tends to infinity, whereas in non-asymptotic analysis we are interested in those probabilities even for finite values of n. Perhaps the most simple example of a concentration inequality is Markov's inequality.

Fact 2.12 (Markov's Inequality [78])**.** *Let X be a non-negative random variable. For any scalar $t > 0$,*

$$\Pr(X \geq t) \leq \frac{\mathbb{E}[X]}{t}.$$

The concentration inequality most extensively used in this thesis is Hoeffding's inequality.

Fact 2.13 (Hoeffding's Inequality [78])**.** *Let $X_1, \ldots, X_n$ be bounded random variables such that $X_i \in [l_i, u_i]$ almost surely for each $i \in [n]$. Define $X := X_1 + \cdots + X_n$. For any scalar $t > 0$,*

$$\Pr(|X - \mathbb{E}[X]| \geq t) \leq 2 \exp\left(\frac{-2t^2}{\sum_{i \in [n]} (u_i - l_i)^2}\right).$$

Chapter 3

Clustering

3.1 Unsupervised Machine Learning

Broadly speaking, *machine learning* refers to the task of developing algorithms which learn to solve the problem at hand by taking, as input, data also relevant to the problem. A more formal, often-cited definition, due to Tom Mitchell, is

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

One way to classify machine learning algorithms is *supervised learning* versus *unsupervised learning*. (More recently, *semi-supervised learning*, which combines elements from both supervised and unsupervised learning has also been introduced.) In supervised learning, typically the data available at the learning stage has *features* and *labels* using which the algorithm learns the relationship between the two. Mathematically, the algorithm learns the parameters of a function which maps the features to their corresponding labels. Subsequently, the learned algorithm is able to determine the labels corresponding to any set of features. Some canonical examples of supervised learning problems are linear regression and classification. Unsupervised learning, on the other hand, refers to techniques aimed at obtaining a concise and interpretable summary of the data at hand. This is beneficial because a compressed representation of the data may:

- lead to memory savings and make certain downstream operations using the data possibly more computationally efficient,

- provide structural insights into the data which can be leveraged for other downstream tasks.

The above discussion is better elucidated by the following concrete example.

Example 3.1. *Suppose we have a database of n images which is represented by a non-negative matrix M such that each column contains the pixel values of a distinct image in a vectorized manner. Suppose each image contains m pixels, i.e. $M \in \mathbb{R}_+^{m \times n}$. Then we may posit that M is a low-rank matrix and seek matrices $X \in \mathbb{R}^{m \times k}$ and $Y \in \mathbb{R}^{k \times n}$ such that $M \approx XY$ and $k << n$. By successfully doing so, firstly we are able to store our original image database containing mn entries using only $(m+n)k$ entries at the cost of the approximation error between M and XY. Moreover, we also discover the following structural insight: the columns of X represent k latent images whose different linear combinations yield approximations to images in our database. Another contemporary method to summarize our image database would be using parametric density estimation methods. In such methods, we determine a latent distribution which approximates the distribution underlying our database. Such methods provide knowledge about our data distribution which is particularly useful if we wish to, for instance, generate new synthetic images belonging to the same distribution as that of our database.*

3.2 Clustering and Some of its Combinatorial Formulations

Clustering is an unsupervised learning technique in which the goal is to determine groups of objects from a given collection of objects such that objects within the same group are similar and objects in different groups are dissimilar. The task of clustering, due to its fundamental nature, arises in more than one domains. Some examples are determining social circles in a social network [29, 56, 13], identifying functional modules in biological networks such as protein-protein interaction networks [58], and finding groups of webpages on the World Wide Web that have content on similar topics [28]. The information about the objects may be given to us:

- either as embeddings for each object in a Euclidean space,
- or as similarity scores between each pair of objects.

While it is possible, in certain cases, to transform one type of information to another, in this thesis, we restrict our attention specifically to the second scenario. In other words, we may think of the problem input as a graph such that the nodes of the graph represent the objects and the edges represent pairwise object relationships. Moreover, the two types of information are also related in the sense that it is reasonable to posit that the graph data is induced by some latent structure, one example of which is latent node embeddings. It is important to note that we have not yet pinned down the definition of a *cluster* in a graph and therefore we still do not have a well-defined problem formulation in the name of clustering. However, there isn't a universal notion of cluster that works for all application domains, and this makes it difficult, if not impossible, to consider a single generalized clustering problem. In the following, we discuss a small subset of the different ways in which one can possibly formulate a graph clustering problem, and discuss the challenges within each approach.

Given an unweighted, undirected graph, one way to define a cluster is as a maximal clique. Then the problem of clustering essentially reduces to the MINIMUM CLIQUE COVER problem. Note that a clique cover is a partition of the vertex set such that the subgraph induced by each set in the partition is a clique, and in the MINIMUM CLIQUE COVER problem, we wish to determine a clique cover containing the smallest number of sets. This problem is NP-hard.

If we have reasons to believe that the input graph has only two clusters of roughly equal size, then we may possibly formulate the clustering problem as follows. We seek a balanced partition of vertices into two sets such that there are least number of edges between the two sets. In other words, given an unweighted, undirected graph $G = (V, E)$, the goal is partition V into clusters V_1 and V_2 such that $|V_1|, |V_2| \leq \lceil |V|/2 \rceil$ and the number of the edges between sets V_1 and V_2 is minimized over all partitions. In fact, this is the MINIMUM BISECTION problem which is also known to be NP-hard.

We may even generalize the above approach to weighted, undirected graphs containing any general k number of balanced clusters. That is, we wish to partition V into k sets $V_1, \ldots, V_k$ such that the number of edges between V_i and V_j for any distinct $i, j \in [k]$ are small and the sets $V_1, \ldots, V_k$ are balanced. For this, we define the following two quantities.

$$\text{RatioCut} := \sum_{i \in [k]} \frac{\text{cut}(V_i, V_i^c)}{|V_i|}$$

$$\text{Ncut} := \sum_{i \in [k]} \frac{\text{cut}(V_i, V_i^c)}{|E(G[V_i])|}$$

where V_i^c denotes the complement set of V_i, $\text{cut}(V_i, V_i^c)$ denotes half the sum of edge weights of the edges between V_i and V_i^c, $G[V_i]$ denotes the subgraph induced by vertex set V_i, and $E(G[V_i])$ denotes the edges in $G[V_i]$. Note that RatioCut and Ncut measure the size of clusters using the number of nodes and the number of edges respectively. Therefore to determine balanced clusters, we may consider separate optimization problems minimizing either RatioCut or Ncut. However, these optimization problems are also NP-hard.

Besides the hardness issue, the above combinatorial formulations do not entirely capture the intuitive idea of clustering. Indeed, in the clique cover approach, there is nothing preventing a high number of edges between a pair of clusters, and in the remaining approaches, there is nothing preventing a small number of edges within a cluster. Moreover, to make some of the above formulations work in practice, we require some a priori knowledge about the number of clusters in the graph. The following formulation, called *Correlation Clustering*, completely avoids this requirement.

Let $G = (V, W)$ denote the input graph which has signed edges such that a positive edge weight denotes similarity between its adjacent nodes and a negative edge weight denotes dissimilarity between its adjacent nodes; the magnitude of the edge weight denotes the strength of similarity/dissimilarity. We define the optimization problem of *minimizing disagreements* as:

$$\begin{aligned} \min_{k, V_1, \ldots, V_k, X} \quad & \sum_{i,j \in [n]} -X_{ij} \cdot \min(W_{ij}, 0) + (1 - X_{ij}) \cdot \max(W_{ij}, 0) \\ \text{s.t.} \quad & k \in \mathbb{N} \\ & V_1, \ldots, V_k \text{ is a partition of } V \\ & X \text{ is the cluster matrix for } V_1, \ldots, V_k \end{aligned} \tag{MIN-D}$$

where the cluster matrix for a partition $V_1, \ldots, V_k$ is defined as

$$X_{ij} := \begin{cases} 1 & \text{if } i, j \text{ belong to the same partition set} \\ 0 & \text{otherwise.} \end{cases}$$

Each feasible solution in the above problem specifies a partition of the nodes into clusters, and for a given partition, we consider disagreements to be:

- node pairs whose nodes are placed in the same cluster and which share a negative edge between them, or
- node pairs whose nodes are placed in different clusters and which share a positive edge between them.

Thus we seek a partition which minimizes the disagreements. Note that the number of clusters appears as variable k in the optimization problem and is therefore automatically chosen, i.e. no a priori estimate for the number of clusters is needed. The optimization problem of minimizing disagreements formulated above is equivalent to the optimization problem of *maximizing agreements*, defined as:

$$\begin{aligned} \max_{k,V_1,\ldots,V_k,X} \quad & \sum_{i,j\in[n]} X_{ij} \cdot \max(W_{ij},0) - (1-X_{ij}) \cdot \min(W_{ij},0) \\ \text{s.t.} \quad & k \in \mathbb{N} \\ & V_1,\ldots,V_k \text{ is a partition of } V \\ & X \text{ is the cluster matrix for } V_1,\ldots,V_k. \end{aligned} \tag{MAX-A}$$

Here for a given partition, we consider agreements to be:

- node pairs whose nodes are placed in the same cluster and which share a positive edge between them, or
- node pairs whose nodes are placed in different clusters and which share a negative edge between them.

The optimization problems (MIN-D) and (MAX-A) are equivalent in the sense that both problems seek a clustering such that the corresponding cluster matrix, X^*, maximizes the function $X \mapsto \langle W, X\rangle$ up to different constant additive terms. Moreover, note that the optimization problem (MIN-D) (and therefore (MAX-A)) is NP-hard. A critical issue with all formulations discussed so far is that they implicitly assume that the clusters are disjoint, i.e do not share any vertices. While this assumption may simplify theoretical analyses, it is restrictive from an application point of view. There are many real-world problems where it is more realistic to model the clusters as potentially overlapping. To our best knowledge, there do not exist standard, well-studied combinatorial optimization formulations for overlapping clustering.

3.3 Average Case Analysis/Provable Recovery

To circumvent worst-case hardness difficulties in clustering formulations, the classical approach has been to design *approximation algorithms*. These are algorithms which return a feasible solution to the optimization problem of interest whose objective value is provably close to the optimal value. More recently, a different alternative to approximation

algorithms has become popular; there are two ideas that emerged in separate fields but share similar philosophy: average case analysis in theoretical computer science and provable recovery in statistics/applied mathematics. In these approaches, a precise structure is imposed over the input instances with the intention of simplifying our search for the optimal solution. The imposed structure over the inputs is either in the form of a generative model (i.e. an exact probability distribution) or deterministic structural assumptions. The following examples demonstrate applications of these ideas.

Example 3.2. *In the MAX CLIQUE problem, the goal is to find the largest clique in a given unweighted, undirected graph. This problem is known to be* NP*-hard. This motivates a shift to average case analysis wherein we consider a planted clique model, i.e. the input instances in this model are Erdős-Rényi graphs containing a fictitiously planted clique. For such instances, as shown in [4], we are able to find, with high probability with respect to randomness in the input graph, the largest clique provided its size is* $\Omega(\sqrt{n})$*, in time polynomial in* n*, where* n *is the number of nodes in the input graph.*

Example 3.3. *In the sparse regression problem, we are required to solve the linear system* $A\mathbf{x} = \mathbf{b}$ *where* $A \in \mathbb{R}^{m \times n}$ *with* $m \ll n$ *and* $\mathbf{b} \in Ran(A)$*. Note that this, by itself, is an ill-posed problem since the system has infinitely many solutions. Therefore we wish to find the sparsest solution, i.e. the one with the least number of non-zero entries. This problem is* NP*-hard. Consequently, we consider the problem through the lens of provable recovery. It is shown in [21] that if the matrix* A *satisfies the so-called Restricted Isometry Property (RIP), then the system* $A\mathbf{x} = \mathbf{b}$ *has a unique sparsest solution which can also be recovered in time polynomial in the size of the problem data* $A, \mathbf{b}$*. More interestingly, it is also shown that random matrices whose entries are independent and identically distributed Gaussian random variables satisfy RIP with high probability.*

To study clustering in the average case analysis/provable recovery paradigm, a widely-used generative model is the *Stochastic Block Model (SBM).*

Definition 3.1 (Stochastic Block Model (SBM))**.** *Let* n *and* k *be positive integers denoting the number of nodes and the number of clusters respectively. Let the nodes and the clusters be labelled using the sets* $[n]$ *and* $[k]$ *respectively. Let* B *be a* $k \times k$ *symmetric matrix with each entry in* $[0, 1]$ *representing the cluster-cluster interaction probabilities. For each node* $i \in [n]$*, draw a vector* $\boldsymbol{\theta}^i$ *independently and uniformly at random from the standard basis vectors in* $\mathbb{R}^k$ *representing the membership of node* i*. Generate a graph* G *on* n *nodes such that for each pair of distinct nodes* $i, i' \in [n]$*,* $ii' \in E(G)$ *with probability* $\boldsymbol{\theta}^{i^T} B \boldsymbol{\theta}^{i'}$*. That is,* G *is a random graph with edge probabilities given by the entries of* B*.*

In the special case in which the diagonal and off-diagonal entries of B are all equal to p and q respectively for some $p, q \in [0, 1]$ satisfying $p > q$, we obtain the *planted partition model*. An in-depth survey of SBM can be found in [1]. Typically when working with SBM, the goal is to recover the true cluster labels for each node in the input instance; the SBM literature has a vast number of recovery guarantees, for example [46, 47, 70]. The advantage of SBM is that is provides a neat theoretical framework to study clustering which justifies its popularity. However, an obvious shortcoming of SBM is that it allows the nodes to belong to exactly one cluster. In practice, such an assumption is not always satisfied. For example, in social network analysis, it is expected that some agents belong to multiple social circles or interest groups. Similarly, in the problem of clustering webpages, it is plausible that some webpages span multiple topics. In [2] an extension of SBM, called the *Mixed Membership Stochastic Blockmodel (MMSB)*, was proposed in which nodes are allowed to have memberships in multiple clusters.

Definition 3.2 (Mixed Membership Stochastic Blockmodel (MMSB))**.** *Let n and k be positive integers denoting the number of nodes and the number of clusters respectively. Let the nodes and the clusters be labelled using the sets $[n]$ and $[k]$ respectively. Let B be a $k \times k$ symmetric matrix with each entry in $[0, 1]$ representing the cluster-cluster interaction probabilities. For each node $i \in [n]$, draw a vector $\boldsymbol{\theta}^i \in \mathbb{R}^k$ independently from Dirichlet($\boldsymbol{\alpha}$) (i.e. Dirichlet distribution with parameter $\boldsymbol{\alpha} \in \mathbb{R}^k$) representing the fractional memberships of node i in the k clusters. Generate a graph G on n nodes such that for each pair of distinct nodes $i, i' \in [n]$, $ii' \in E(G)$ with probability $\boldsymbol{\theta}^{iT} B \boldsymbol{\theta}^{i'}$. That is, G is a random graph with edge probabilities given by the entries of $\Theta B \Theta^T$, where Θ is the $n \times k$ matrix whose row i the transpose of $\boldsymbol{\theta}^i$ for each $i \in [n]$.*

Note that the MMSB generalizes the SBM in the sense that the set of membership vectors is generalized from that of the standard basis vectors to the unit simplex. When working with the MMSB, the goal is to recover the vectors $\boldsymbol{\theta}^i$ capturing the fractional membership information for each node $i \in [n]$. The research direction on theoretical analyses of MMSB is relatively new and the literature on this is much sparse compared to the SBM.

To study Correlation Clustering in the provable recovery paradigm, we require a generative model for generating signed graph instances in which a positive (resp. negative) edge weight denotes a similarity (resp. dissimilarity) measure between the adjacent nodes. However, unlike the SBM, there does not exist a single popular generative model for such instances, to our best knowledge. In [40] and [50], the authors propose a generalization of the planted partition model and a semi-random model, i.e. one in which there is a probabilistic component and a deterministic adversarial component, respectively, for generating

random graphs with signed edges.

3.4 Existing Cluster Recovery Techniques

There are many algorithms in literature for provably recovering the cluster ground truth as posited by a generative model. We do not attempt to provide an exhaustive survey of these methods, and instead discuss prevalent high-level algorithmic techniques. Note that we exclude heuristic methods, for a comprehensive survey on which the reader may refer to the survey paper by Fortunato [30].

3.4.1 Spectral Methods

Spectral methods typically make use of either the Laplacian matrix or one of its normalized variants. Recall that for a weighted graph $G = (V, W)$, the Laplacian matrix is defined as $L(G) := D - W$ where $D := Diag(W\mathbf{e})$, i.e. in the Laplacian, each diagonal entry is equal to the degree of the corresponding node and each off-diagonal entry is negative of the corresponding edge weight. One way to normalize the Laplacian is by multiplying $L(G)$ on left and right with $D^{-1/2}$, i.e. consider the matrix $D^{-1/2}(D-W)D^{-1/2} = I-D^{-1/2}WD^{-1/2}$. For the purpose of this discussion, we use $L(G)$. Let USU^T be the spectral decomposition of $L(G)$. Note that because $L(G)$ is real and symmetric, each of its eigenvalues is real. Assume, without loss of generality, that the eigenvalues on the diagonal of S are sorted in non-increasing order, and let k denote a prior estimate of the number of clusters in the graph. The ordering of eigenvalues in S implies that the columns of $U(:, [k])$ contain the eigenvectors of $L(G)$ corresponding to k largest eigenvalues of $L(G)$. Then interpreting the rows of $U(:, [k])$ as points in $\mathbb{R}^k$, we perform k-means or k-medians clustering to obtain a partition for them. Lastly, the nodes in G are partitioned corresponding to the partition obtained for the rows of $U(:, [k])$. In [70], a theoretical analysis of the recovery performance of a spectral algorithm for the SBM is presented. In [85], the authors analyze a spectral algorithm in the context of a generative model for overlapping clusters. For a detailed survey on spectral clustering, the reader may refer to the tutorial paper by Von Luxburg [79].

3.4.2 Convex Relaxation Methods

In the convex relaxation approach, we seek a convex optimization formulation which captures the clustering problem we are interested in. We may begin with a certain optimiza-

tion formulation for the clustering problem of interest but such formulations are inevitably combinatorial (as seen in Section 3.2). To circumvent this challenge, we then *relax* the intractable components of the optimization problem to obtain a convex optimization formulation which is a good approximation of the original optimization problem. This approach is beneficial because of the vast body of existing theoretical literature on convex optimization. For instance, after obtaining a convex optimization formulation, we can use the Karush-Kuhn-Tucker (KKT) optimality conditions discussed in Section 2.1 of Chapter 2 to determine an optimal solution. There is not a fixed recipe for applying this approach as there are multiple choices to be made requiring mathematical ingenuity such as how to relax the original difficult formulation to obtain a convex formulation, and how to construct variables satisfying the KKT conditions. Apart from clustering, convex relaxations have shown success in other machine learning problems as well such as sparse regression [21], low-rank matrix completion [20], and dictionary learning [74]. Moreover, for certain problems, convex relaxation methods are even shown to be optimal. For instance, in [17], a semidefinite programming (SDP)-relaxation approach is presented for the sparse principal component analysis problem; additionally, it is shown that the proposed SDP is optimal in the sense that there does not exist another algorithm with both better statistical and computational performance guarantees, under the assumption that certain instances of the planted clique problem cannot be solved in randomized polynomial time. (For a certain parameter regime for the planted clique problem, it is conjectured that no polynomial time solution exists.)

In the following, we show one derivation of a convex relaxation for clustering. Let $G = (V, W)$ be a graph generated using the planted partition model, defined in Section 3.3. Note that G is an unweighted graph whose adjacency matrix is denoted by W. Determining a clustering of G which maximizes the log-likelihood of the observed graph G yields the discrete optimization problem

$$\begin{aligned} \max_{k, V_1, \ldots, V_k, X} \quad & \langle X, W - \lambda E \rangle \\ \text{s.t.} \quad & k \in \mathbb{N} \\ & V_1, \ldots, V_k \text{ is a partition of } V \\ & X \text{ is the cluster matrix for } V_1, \ldots, V_k \end{aligned} \tag{MAX-LL}$$

where

$$\lambda := \frac{\log(1-q) - \log(1-p)}{\log p - \log q + \log(1-q) - \log(1-p)}.$$

A convex relaxation of (MAX-LL) is

$$\begin{aligned} \max_{X} \quad & \langle X, W - \lambda E \rangle \\ \text{s.t.} \quad & X_{ii} = 1 \ \forall i \in [n] \\ & X \geq 0, X \succeq 0. \end{aligned} \tag{MAX-LL-CONV}$$

In [47], using a construction of variables satisfying the KKT conditions for (MAX-LL-CONV), the authors present conditions under which (MAX-LL-CONV) can provably recover the ground truth clustering in the planted partition model. There are multiple convex relaxations in literature for clustering a graph into disjoint clusters, for example [22, 23, 24, 39, 42], but that is not the case for overlapping clustering.

3.4.3 Combinatorial Methods

As the name suggests, combinatorial methods typically perform discrete operations directly on the input graph to recover the clusters. In [8], the authors make both deterministic and probabilistic structural assumptions on the input graph, and define clusters as dense subgraphs satisfying a given set of properties. They propose recovery algorithms in which clusters are built in a bottom-up fashion by first sampling nodes and then determining cliques in the neighborhoods of the sampled nodes. Similar model and algorithmic ideas are independently developed in [9]. Alternatively, Ray et al. [68] propose a model in which first each node is assigned to a subset of k clusters, for some k denoting the total number of clusters, and then the observed graph is generated by placing, for each pair of nodes i and i', edge ii' with probability proportional to the number of clusters shared by the two nodes. They develop a three-step recovery algorithm: first determine k node sets, called *pure node sets*, such that for each $i \in [k]$, the i^{th} set belongs exclusively to the i^{th} cluster. Then estimate the between- and within-cluster edge density parameters. Lastly, using those parameters apply degree thresholding to assign clusters to the remaining nodes. This is mainly a combinatorial algorithm but it uses convex optimization as a subroutine. Indeed the subroutine to determine clustered pure node sets uses an existent convex relaxation method to detect non-overlapping clusters; note that the subgraph induced by the k pure node sets comprises of k non-overlapping clusters.

Chapter 4

Provable Overlapping Community Detection in Weighted Graphs[1]

4.1 Problem Introduction

As mentioned in Chapter 1, we use the terms *cluster* and *community*, and the phrases *graph clustering* and *community detection* interchangeably. As discussed in Chapter 3, the Mixed Membership Stochastic Blockmodel (MMSB) generalizes the traditional Stochastic Block Model (SBM) by positing that each node may have fractional memberships in the different communities. If n and k denote the number of nodes and the number of communities respectively, matrix $\Theta \in [0,1]^{n\times k}$, called the *node-community distribution matrix*, is generated such that each of its rows is drawn from the Dirichlet distribution with parameters $\boldsymbol{\alpha} \in \mathbb{R}^k$. Then the $n \times n$ *probability matrix* is given as

$$P = \Theta B \Theta^T \tag{4.1}$$

where B is a $k \times k$ *community interaction matrix*. Lastly, a random graph according to MMSB is generated on n nodes by placing an edge between nodes i and j with probability P_{ij}, and based on observing this graph, we are interested in recovering the matrix Θ. MMSB has been shown to be effective in many real-world settings, but the recovery guarantees regarding it are very limited compared to the SBM. In this chapter, we provide a provable linear-programming-based algorithm to recover Θ using P that is relatively easy to implement and that does not require an assumption on the input graph which is made oftentimes but is not realistic.

[1] This chapter is based on the work [49] with the same title.

For a theoretical analysis of the MMSB, it is usually assumed that the user has access to only an unweighted random graph generated according to the model. While this assumption may be necessary in some settings, it makes the analysis difficult without much advantage. Indeed in many settings of practical interest, the user does have access to a similarity measure between node pairs, and this motivates us to work with weighted graphs generated according to the MMSB. For example, in the context of social network analysis, one may define a *communication graph* as an unweighted graph in which edge ij exists if and only if agents i and j exchanged messages in a certain fixed time window. Then the weighted adjacency matrix for the social network may be obtained by averaging the adjacency matrices of multiple observed communication graphs. On the other hand, we make the problem difficult in a more realistic manner; we remove a common assumption in literature which is quite unrealistic if not mathematically problematic. This assumption requires each community in the input graph to contain a node which belongs exclusively to that community. Such nodes are called *pure nodes* in the literature. The notion of pure nodes in community detection is related to that of *separability* in nonnegative matrix factorization in the sense that they both induce a simplicial structure on the data. Although we do not make the pure nodes assumption, the Dirichlet distribution naturally generates increasingly better approximations to pure nodes as n, the number of nodes, gets large, and we use this fact in our analysis. As far as we know, this exact setup has not been studied before.

Among existing provable methods, [85] propose the so-called *Overlapping Continuous Community Assignment Model (OCCAM)* which only slightly differs from MMSB; in OCCAM each row of Θ has unit ℓ_2-norm as opposed to unit ℓ_1-norm in MMSB. They provide a provable algorithm for learning the OCCAM parameters in which one performs k-medians clustering on the rows of the $n \times k$ matrix corresponding to k largest eigenvectors of the adjacency matrix corresponding to the observed unweighted random graph. However, their assumptions may be difficult to verify in practice. Indeed for their k-medians clustering to succeed, they assume that the ground-truth community structure provides the unique global optimal solution of their chosen k-medians loss function, which is also required to satisfy a special curvature condition around this minimum.

A moment-based tensor spectral approach to recover the MMSB parameters Θ and B from an unweighted random graph generated according to the model was shown by [6]. Their approach, however, is not very straightforward to implement and involves multiple tuning parameters. Indeed one of the tuning parameters must be close to the sum of the k Dirichlet parameters, which are not known in advance.

In a series of works, [51, 52, 53] have also tackled the problem of learning the parameters in MMSB from a single random graph generated by the model. However, they require the

pure node assumption. Additionally, they cast the MMSB recovery problem as problems that are nonconvex. Consequently, to get around the nonconvexity, more assumptions on the model parameters are required. For instance, in [51], the MMSB recovery problem is formulated as a *Symmetric Nonnegative Matrix Factorization (SNMF)* problem, which is both nonconvex and NP-hard. Then to ensure the uniqueness of the global optimal solution for the SNMF problem, they require B to be a diagonal matrix. In contrast, not only does our approach directly tackle the factorization in (4.1) to recover Θ, we also do so using linear programming.

Recently [36] have also proposed a linear-programming-based algorithm for recovery in MMSB. However, the connection between their proposed linear programs and ours is unclear, and they require the pure nodes assumption for their method to provably recover the communities.

4.2 Problem Formulation

Recall the notation that for each $i \in [n]$ and $j \in [k]$, $\boldsymbol{\theta}^i$ denotes the transpose of row i and $\boldsymbol{\theta}^j$ denotes column j of matrix Θ. We ask the following question for the MMSB described by (4.1):

Given P, how can we efficiently obtain a matrix $\hat{\Theta} \in [0,1]^{n\times k}$ such that $\hat{\Theta} \approx \Theta$?

Typically, one imposes the pure nodes assumption on Θ which greatly simplifies the above posed problem. That is, one assumes that for each $j \in [k]$, there exists $i \in [n]$ such that $\boldsymbol{\theta}^i = \mathbf{e}_j$, i.e. node i belongs exclusively to community j. In other words, the rows of Θ contain all corners of unit simplex in $\mathbb{R}^k$. However, such an assumption is mathematically problematic and/or practically unrealistic. Indeed if the rows of Θ are sampled from the Dirichlet distribution, then the probability of sampling even one pure node is zero. Moreover, even from a practical standpoint such an assumption may not always be satisfied since in real-world networks, such as protein-protein interaction networks, one encounters communities with no pure nodes. Lastly, note that we are interested in recovering only Θ and not B since the former contains the community membership information of each node which is usually what a user of such methods is interested in.

We provide an answer to posed question without making an explicit assumption regarding the presence of pure nodes. To that effect, we propose a novel simple and efficient convex-optimization-based method to approximate Θ entrywise under a very natural condition that just requires n to be sufficiently large. Such a condition is often satisfied in

practice since real-world graphs in application settings such as social network analysis are usually large-scale.

4.3 SP+LP Recovery Algorithm

We may think of our recovery procedure, *Successive Projection followed by Linear Programming* (SP+LP), as divided into two stages. First, via a preprocessing step, called Successive Projection, we obtain a set $\mathcal{J} \subseteq [n]$ of cardinality k such that $\Theta(\mathcal{J}, :)$ is entrywise close to I up to a permutation of the rows. Intuitively, due to a simplicial structure in the columns of matrix P, such a set $\mathcal{J}$ may be determined by (1) extracting a column of P, called $\mathbf{v}$, with the largest ℓ_2-norm, (2) replacing each column of P with its projection on the orthogonal complement of $\mathbf{v}$, and repeating steps (1) and (2). We may think of the nodes in $\mathcal{J}$ as being *almost pure*, which we then we use to recover approximations to the k columns of Θ, the *community characteristic vectors*, using exactly k linear programs (LPs). The form of the LP in SP+LP can be motivated as follows. Intuitively, the presence of almost pure nodes ensures that the column range of Θ coincides with the range of P; consequently recovering Θ given P may be interpreted as obtaining a certain basis for the range of P. These desired basis vectors, i.e. the columns of Θ, are nonnegative and somewhat sparse in the sense that they contain potentially many entries which are close to zero. Thus we seek nonnegative vectors in the range of P with the smallest ℓ_1-norm (which, for a nonnegative vector, is equal to the sum of its entries) and introduce a non-homogeneous constraint to rule out the trivial solution of the zero vector. Similar optimization formulation techniques have been shown to work for some planted/generative models for the problem for sparse dictionary learning [65, 74]. Note that SP+LP has no tuning parameters other than the number of communities, which is also a parameter for most other community detection algorithms.

Algorithm 1 SP+LP

Input: Matrix P generated according to MMSB, number of communities k
Output: Estimated characteristic vectors $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_k \in [0,1]^n$

1: $\mathcal{J} = \text{SuccessiveProjection}(P)$
2: **for** $i \in [k]$ **do**
3: $\quad (\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y})} \mathbf{e}^T\mathbf{x}$ s.t. $\mathbf{x} \geq \mathbf{0}, x_{\mathcal{J}(i)} \geq 1, \mathbf{x} = P\mathbf{y}$
4: $\quad \hat{\boldsymbol{\theta}}_i = \mathbf{x}^*/\|\mathbf{x}^*\|_\infty$
5: **end for**

Algorithm 2 SuccessiveProjection

Input: Matrix P generated according to MMSB, number of communities k
Output: Estimated set of almost pure nodes $\mathcal{J} \subseteq [n]$

$\mathcal{J} = \{\}, R = P, j = 1$
while $R \neq 0$ and $j \in [k]$ **do**
  $s' = \arg\max_{s \in [n]} \|\mathbf{r}_s\|^2$
  $R = \left(I - \frac{\mathbf{r}_{s'}\mathbf{r}_{s'}^T}{\|\mathbf{r}_{s'}\|^2}\right) R$
  $\mathcal{J} = \mathcal{J} \cup \{s'\}$
  $j = j + 1$
end while

4.4 Theoretical Guarantees

Let Z be a $k \times k$ submatrix of Θ such that for each $j \in [k]$, there exists $i \in [k]$ satisfying

$$\|\mathbf{z}^i - \mathbf{e}_j\|_\infty \leq \|\boldsymbol{\theta}^p - \mathbf{e}_j\|_\infty \tag{4.2}$$

for any $p \in [n]$. The rows of Z do not exactly correspond to the corners of the unit simplex; they are, however, the best entrywise approximations of the corners that can be obtained among the rows of Θ. Note that without loss of generality, through appropriate relabelling of the nodes, we may assume that indices i and j in (4.2) are identical. Define the $k \times k$ matrix $\Delta := Z - I$.

Define $\mathbf{c} := \Theta^T \mathbf{e}$, and let $c_{\min}$ and $c_{\max}$ denote the smallest and largest entries in $\mathbf{c}$ respectively. Let κ and κ_0 denote the condition numbers of B and ΘB respectively, associated with the ℓ_2-norm. Recall that this condition number is the ratio of the largest and smallest singular values of the matrix.

Now we state our main result, which provides complete theoretical justification for the success of `SP+LP` in approximately recovering the k community vectors.

Theorem 4.4.1. *Suppose $k \geq 2$, B is full-rank, and all k parameters of the Dirichlet distribution are equal to $\alpha \in \mathbb{R}$. Let $w := 8\kappa\sqrt{\alpha k + 1}$ and define*

$$\epsilon_1 := \min\left(\frac{1}{\sqrt{k-1}}, \frac{1}{2}\right) \frac{1}{2\sqrt{2}w(1 + 80w^2)}$$
$$\epsilon_2 := \frac{7}{3520\sqrt{2}kw^2}.$$

If $n > \dfrac{\log(p/k)}{\log I_{1-\epsilon}(\alpha,(k-1)\alpha)}$ *for some* $p \in (0,1)$ *and* $\epsilon \in (0, \min\{\epsilon_1, \epsilon_2\})$*, then there exists a permutation* π *of the set* $[k]$ *such that vectors* $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_k$ *returned by* ***SP+LP*** *satisfy*

$$\max_{j\in[k]} \|\hat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_{\pi(j)}\|_\infty = \mathcal{O}(\alpha k^2 \kappa^2 \epsilon) \tag{4.3}$$

with probability at least $1 - p - c_1 e^{-c_2 n}$ *where* c_1, c_2 *are constants that depend on* α, k, κ*. (Here* $I_x(y,z)$ *denotes the regularized incomplete beta function, defined as*

$$I_x(y,z) = \frac{\int_0^x t^{y-1}(1-t)^{z-1}dt}{\int_0^1 t^{y-1}(1-t)^{z-1}dt}$$

for complex numbers y, z *with positive real parts.)*

We note that even though our main result is stated for an equal parameter Dirichlet distribution, our proof techniques extend, in principle, to a setting in which the Dirichlet parameters are different but not too far from each other. Doing so, however, adds only incremental value but makes the analysis significantly tedious.

In line with our algorithm description, we divide the theoretical analysis also in two parts: one for analysis of the preprocessing Successive Projection subroutine, and another for analysis of the LPs in the main algorithm.

Successive Projection Algorithm was first studied by [32] in far more generality than what is used here. Adopting their main recovery theorem to our setup yields the following theorem.

Theorem 4.4.2 ([32])**.** *Suppose that*

$$\|\Delta\|_{\max} < \min\left(\frac{1}{\sqrt{k-1}}, \frac{1}{2}\right) \frac{1}{2\sqrt{2}\kappa_0(1+80\kappa_0^2)} \tag{4.4}$$

and let $\mathcal{J}$ *be the index set of cardinality* k *extracted by Algorithm 2. Then there exists a* $k \times k$ *permutation matrix* Π *such that*

$$\|\Pi\Theta(\mathcal{J},:) - I\|_{\max} \leq 40\sqrt{2}\kappa_0^2 \|\Delta\|_{\max}. \tag{4.5}$$

Theorem 4.4.2 provides theoretical justification for the success of the subroutine highlighted in Algorithm 2. To this end, our contribution is to show that the condition in (4.4) is satisfied in MMSB with high probability provided the number of nodes in the graph is sufficiently large. This involves deriving concentration bounds for the smallest and largest singular values of Θ and ΘB. The following result provides theoretical guarantee for the performance of the LP in Algorithm 1.

Theorem 4.4.3. *Assume $k \geq 2$, B is full-rank, and $c_{\min}/c_{\max} > 1/2$. Suppose for each $s \in [k]$, there exists $p \in [n]$ such that $\|\boldsymbol{\theta}^p - \mathbf{e}_s\|_\infty \leq \eta$ for some $0 \leq \eta < (c_{\min}/c_{\max} - 1/2)/4k$. Let $i \in [n]$ such that $\|\boldsymbol{\theta}^i - \mathbf{e}_j\|_\infty \leq \eta$ for some $j \in [k]$. Then the LP*

$$\begin{aligned} \min \quad & \mathbf{e}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \geq \mathbf{0} \\ & x_i \geq 1 \\ & \mathbf{x} = P\mathbf{y} \end{aligned} \tag{P}$$

has an optimal solution, and if $\mathbf{x}^*$ is an optimal solution then

$$\left\| \frac{\mathbf{x}^*}{\|\mathbf{x}^*\|_\infty} - \boldsymbol{\theta}_j \right\|_\infty \leq 4\eta(2\sqrt{2}k + 1). \tag{4.6}$$

Moreover, the time complexity of solving (P) to obtain $\mathbf{x}^*$ is $\mathcal{O}(n^2)$.

Combining Theorems 4.4.2 and 4.4.3 yields Theorem 4.4.1, which provides entrywise error bounds for the k community characteristic vectors returned by `SP+LP`. We also conclude that the time complexity of `SP+LP` is $\mathcal{O}(n^2)$ since the time complexity of both SuccessiveProjection and solving (P) is $\mathcal{O}(n^2)$. To our best knowledge, there does not exist a competing provable algorithm whose time complexity is under $\mathcal{O}(n^2)$. Note that all time complexity expressions mentioned in this chapter hide dependence on k, since we assume k to be fixed with respect to n.

Using Theorem 4.4.3, we also make a note about identifiability. For the MMSB, we say that the model is *identifiable* if no two distinct pairs of Θ and B yield the same matrix P. It is shown in [53] that, unless an assumption about the entries of B is made, the MMSB is identifiable if and only if each community has a pure node. Since we do not assume the presence of pure nodes for each community, we present a result regarding the *near identifiability* of the MMSB.

Corollary 4.4.4. *Let Θ and $\bar{\Theta}$ be $n \times k$ node-community distribution matrices satisfying the conditions of Theorem 4.4.3 for some η and $\bar{\eta}$ respectively. Let B and $\bar{B}$ be $k \times k$ full-rank community interaction matrices such that $\Theta B \Theta^T = \bar{\Theta} \bar{B} \bar{\Theta}^T$. Then there exists a permutation π of the set $[k]$ such that*

$$\max_{j \in [k]} \|\bar{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_{\pi(j)}\|_\infty \leq 4(\bar{\eta} + \eta)(2\sqrt{2}k + 1). \tag{4.7}$$

4.5 Proofs

In this section, we build the necessary tools using which we ultimately provide a proof of Theorem 4.4.1 and Corollary 4.4.4.

4.5.1 LP Analysis

We begin by developing a proof of Theorem 4.4.3. Let $\eta \in (0,1)$ and assume for now that for each $j \in [k]$, there exists $i \in [n]$ such that $\|\boldsymbol{\theta}^i - \mathbf{e}_j\|_\infty \leq \eta$. Moreover, assume, without loss of generality, that for each $i \in [k]$

$$\|\boldsymbol{\theta}^i - \mathbf{e}_i\|_\infty \leq \eta. \tag{4.8}$$

Indeed such a property can always be satisfied with appropriate relabelling of the nodes. Define $I' := \Theta([k], :)$.

Lemma 4.5.1. *Suppose M is a $k \times k$ matrix whose rows belong to the unit simplex. If*

$$\|\mathbf{m}^i - \mathbf{e}_i\|_\infty \leq \delta \tag{4.9}$$

for each $i \in [k]$ and for some $\delta \in \left[0, \frac{1}{2\sqrt{2k}}\right]$, then

$$\|M^{-T} - I\|_\infty \leq 2\sqrt{2}\delta k. \tag{4.10}$$

Proof. Since each row of M belongs to the unit simplex and satisfies (4.9), we note that ℓ_2-norm of each row of $M - I$ is bounded above by $\delta\sqrt{2}$. This implies that

$$\|M - I\| \leq \delta\sqrt{2k}. \tag{4.11}$$

Moreover

$$\begin{aligned} |\|M^{-1}\| - 1| &\leq \|M^{-1} - I\| \qquad \text{(using reverse triangle inequality)} \\ &= \|(M - I)M^{-1}\| \\ &\leq \|M - I\|\|M^{-1}\| \end{aligned}$$

which implies that

$$\|M^{-1}\| \leq \frac{1}{1 - \|M - I\|}. \tag{4.12}$$

Then, we have

$$\begin{aligned}
\|M^{-T} - I\|_\infty &\leq \sqrt{k}\|M^{-T} - I\| \\
&= \sqrt{k}\|M^{-1} - I\| \\
&\leq \sqrt{k}\|M - I\|\|M^{-1}\| \\
&\leq \frac{\sqrt{k}\|M - I\|}{1 - \|M - I\|} && \text{(using (4.12))} \\
&\leq \frac{\sqrt{2}\delta k}{1 - \delta\sqrt{2k}} \\
&\leq 2\sqrt{2}\delta k. && \text{(by assumption on } \delta)
\end{aligned}$$

□

For any $i \in [k]$, consider the LP

$$\begin{aligned}
\min \quad & \mathbf{c}^T\mathbf{y} \\
\text{s.t.} \quad & \Theta\mathbf{y} \geq \mathbf{0} \\
& \mathbf{y}^T\boldsymbol{\theta}^i \geq 1.
\end{aligned} \tag{Pi}$$

and its dual

$$\begin{aligned}
\max \quad & \beta \\
\text{s.t.} \quad & \beta\boldsymbol{\theta}^i + \Theta^T\mathbf{u} = \mathbf{c} \\
& \beta, \mathbf{u} \geq 0.
\end{aligned} \tag{Di}$$

Note that both (Pi) and (Di) are feasible optimization problems. Thus let $\mathbf{y}^*$ and $(\beta^*, \mathbf{u}^*)$ be a (Pi)-(Di) optimal solution pair.

Lemma 4.5.2. *Suppose* $\eta \leq \dfrac{1}{2\sqrt{2}k}\dfrac{c_{\min}}{c_{\max}}$.

Then

$$c_i - 2\sqrt{2}\eta k c_{\max} \leq \beta^* \leq \frac{c_i}{1-\eta}. \tag{4.13}$$

Proof. The upper bound follows from observing that $\beta^* = \mathbf{c}^T\mathbf{y}^*$ due to Strong Duality and that $\mathbf{e}_i/\theta_{ii}$ is a feasible solution for (Pi), combined with the fact that $\theta_{ii} \geq 1 - \eta$.

For the lower bound we construct a feasible solution for (Di). Define $\mathbf{z}$ as the solution of the system $I'^T\mathbf{z} = \mathbf{c}$. Note that the rows of I' belong to the unit simplex and for any $i \in [k]$, we have

$$\begin{aligned}\|I'(i,:) - \mathbf{e}^i\|_\infty &\leq \eta \\ &\leq \frac{1}{2\sqrt{2k}}. \qquad \text{(by assumption on } \eta)\end{aligned}$$

Therefore using Lemma 4.5.1, we conclude that $\|I'^{-T} - I\|_\infty \leq 2\sqrt{2}\eta k$.

Then for any $s \in [k]$, we have

$$\begin{aligned}|z_s - c_s| &\leq \|\mathbf{z} - \mathbf{c}\|_\infty \\ &\leq \|I'^{-T} - I\|_\infty c_{\max} \\ &\leq 2\sqrt{2}\eta k c_{\max}.\end{aligned}$$

Moreover since $\eta \leq \frac{1}{2\sqrt{2}k}\frac{c_{\min}}{c_{\max}}$, we conclude that $\mathbf{z} \geq \mathbf{0}$. Now define the point $(\beta', \mathbf{u}')$ such that

$$\beta' := z_i$$

and

$$u'_s := \begin{cases} z_s, & \text{if } s \in [k] \setminus \{i\} \\ 0, & \text{otherwise.} \end{cases}$$

Note that $(\beta', \mathbf{u}')$ is feasible for (Di) with objective value

$$\beta' \geq c_i - 2\sqrt{2}\eta k c_{\max}.$$

□

Define the vector $\mathbf{r} := \Theta^T\mathbf{u}^*/2$. We shall prove some bounds on the entries of $\mathbf{r}$ which will be used for subsequent proofs.

Lemma 4.5.3. *Suppose* $\eta \leq \frac{1}{2\sqrt{2}k}\frac{c_{\min}}{c_{\max}}$. *Then we have the following inequalities.*

1. $0 \leq r_i \leq 2k\eta c_{\max}$.
2. *For any* $s \in [k] \setminus \{i\}$,
$$c_{\min} - \frac{\eta}{1-\eta}c_{\max} \leq r_s \leq \frac{c_{\max}}{2}.$$

Proof. First note that $\mathbf{r} \geq \mathbf{0}$ by definition and therefore the lower bound on r_i follows. From the feasibility of $(\beta^*, \mathbf{u}^*)$ for (Di), we have for any $s \in [k]$

$$r_s = \frac{c_s - \beta^* \theta_{is}}{2}. \tag{4.14}$$

The upper bound on r_i follows from (4.14), and using the lower bound on β^* from Lemma 4.5.2 and the fact that $\theta_{ii} \geq 1 - \eta$. Indeed, we have

$$\begin{aligned}
r_i &= \frac{c_i - \beta^* \theta_{ii}}{2} \\
&\leq \frac{c_i - [(c_i - 2\sqrt{2}\eta k c_{\max})(1-\eta)]}{2} \\
&= \frac{\eta c_i + 2\sqrt{2}\eta(1-\eta)k c_{\max}}{2} \\
&\leq \frac{\eta c_{\max}[1 + 2\sqrt{2}(1-\eta)k]}{2} \\
&\leq \eta c_{\max}\left(\frac{1+3k}{2}\right) \\
&\leq 2k\eta c_{\max}. && (\because k \geq 2)
\end{aligned}$$

For any $s \in [k] \setminus \{i\}$, the upper bound on r_s follows from (4.14), and noting that β^* and θ_{is} are nonnegative and $c_s \leq c_{\max}$.

For any $s \in [k] \setminus \{i\}$, the lower bound on r_s follows from (4.14), and using the upper bound on β^* from Lemma 4.5.2, the fact that $c_s \geq c_{\min}$ and the fact that $\theta_{is} \leq \eta$. □

Lemma 4.5.4. *Suppose* $\eta \leq \frac{1}{3k}\frac{c_{\min}}{c_{\max}}$. *Then* $\|\mathbf{r}\|_\infty \leq \frac{c_{\max}}{2}$.

Proof. We prove this statement by proving that $\|\mathbf{r}\|_\infty$ is attained at some index in $[k] \setminus \{i\}$. It suffices to show that $r_i \leq r_s$ for any $s \in [k] \setminus \{i\}$. Note that by assumption $\eta \leq \frac{1}{3k}\frac{c_{\min}}{c_{\max}} \leq \frac{1}{2\sqrt{2}k}\frac{c_{\min}}{c_{\max}}$, the entries of $\mathbf{r}$ are bounded according to Lemma 4.5.3.

We have

$$\begin{aligned} c_{\min} &\geq 2\eta c_{\max}\frac{3k}{2} && \text{(by assumption on } \eta) \\ &\geq 2\eta c_{\max}(k+1) && (\because k \geq 2) \\ &= 2\eta c_{\max} + 2k\eta c_{\max} \\ &\geq \frac{\eta}{1-\eta}c_{\max} + 2k\eta c_{\max} && (\because \eta \leq 1/2) \end{aligned}$$

which is equivalent to

$$2k\eta c_{\max} \leq c_{\min} - \frac{\eta}{1-\eta}c_{\max}.$$

Therefore using Lemma 4.5.3, we conclude that $r_i \leq r_s$ for any $s \in [k] \setminus \{i\}$. □

Lemma 4.5.5. *Suppose* $\frac{c_{\min}}{c_{\max}} > \frac{1}{2}$ *and* $\eta < \frac{1}{3k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right)$. *Then for any* $s \in [k] \setminus \{i\}$, *if* y_s^* *is positive, we have*

$$y_s^* < 2\sqrt{2}\eta k. \tag{4.15}$$

Proof. Pick any $s \in [k] \setminus \{i\}$ such that $y_s^* > 0$. Consider the auxiliary LP

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{y} \\ \text{s.t.} \quad & \Theta\mathbf{y} \geq \mathbf{0} \\ & \mathbf{y}^T\boldsymbol{\theta}^i \geq 1 \\ & y_s \geq 2\sqrt{2}\eta k \end{aligned} \tag{Pi-aux}$$

and its dual

$$\begin{aligned} \max \quad & \beta + (2\sqrt{2}\eta k)\gamma \\ \text{s.t.} \quad & \beta\boldsymbol{\theta}^i + \gamma\mathbf{e_s} + \Theta^T\mathbf{u} = \mathbf{c} \\ & \beta, \gamma, \mathbf{u} \geq 0. \end{aligned} \tag{Di-aux}$$

If we show that $\mathbf{y}^*$ is not an optimal solution to (Pi-aux), then we can conclude that $y_s^* < 2\sqrt{2}\eta k$. Therefore our goal is to show that the optimal value of (Pi-aux) is greater than $\mathbf{c}^T\mathbf{y}^*$. Equivalently, we may also show that the optimal value of (Di-aux) is greater than β^*. We do so by constructing a feasible solution for (Di-aux) at which the objective value is greater than β^*.

Now define $\bar{I}$ to be identical to I' except the s^{th} row which is set to be $\mathbf{e_s}^T$. Let $\mathbf{z}^*$ be the solution to the system

$$\bar{I}^T\mathbf{z} = \mathbf{r} \tag{4.16}$$

where recall that $\mathbf{r} = \Theta^T \mathbf{u}^*/2$.

Note that the rows of $\bar{I}$ belong to the unit simplex and for any $i \in [k]$, we have

$$\begin{aligned} \|\bar{I}(i,:) - \mathbf{e}^i\|_\infty &\leq \eta \\ &\leq \frac{1}{2\sqrt{2k}}. \qquad \text{(by assumption on } \eta) \end{aligned}$$

Therefore using Lemma 4.5.1, we conclude that

$$\|\bar{I}^{-T} - I\|_\infty \leq 2\sqrt{2}\eta k. \tag{4.17}$$

Define the point

$$\begin{bmatrix} \bar{\beta} \\ \bar{\gamma} \\ \bar{\mathbf{u}} \end{bmatrix} := \begin{bmatrix} \beta^* \\ 0 \\ \mathbf{u}^*/2 \end{bmatrix} + \begin{bmatrix} \beta' \\ \gamma' \\ \mathbf{u}' \end{bmatrix} \tag{4.18}$$

where $\beta' := z_i^*$, $\gamma' := z_s^*$ and

$$u'_p := \begin{cases} z_p^* & \text{if } p \in [k] \setminus \{i, s\} \\ 0 & \text{otherwise.} \end{cases}$$

First we argue that $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$ is feasible for (Di-aux). From (4.18), we have

$$\begin{aligned} \bar{\beta}\boldsymbol{\theta}^i + \bar{\gamma}\mathbf{e_s} + \Theta^T\bar{\mathbf{u}} &= \beta^*\boldsymbol{\theta}^i + \Theta^T\mathbf{u}^*/2 + \beta'\boldsymbol{\theta}^i + \gamma'\mathbf{e_s} + \Theta^T\mathbf{u}' \\ &= \mathbf{c} - \mathbf{r} + \beta'\boldsymbol{\theta}^i + \gamma'\mathbf{e_s} + \Theta^T\mathbf{u}' \\ &\qquad (\because (\beta^*, \mathbf{u}^*) \text{ is feasible for (Di)}) \\ &= \mathbf{c} - \mathbf{r} + \bar{I}^T\mathbf{z}^* \\ &\qquad \text{(using the definition of } (\beta', \gamma', \mathbf{u}')) \\ &= \mathbf{c}. \\ &\qquad \text{(using (4.16))} \end{aligned}$$

To argue about the nonnegativity of $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$, it suffices to argue that

1. $z_i^* + \beta^* \geq 0$
2. $\mathbf{z}^*([k] \setminus \{i\}) \geq \mathbf{0}$.

Note that our assumption on η implies $\eta < \frac{1}{2\sqrt{2}k}\frac{c_{\min}}{c_{\max}}$ and therefore Lemmas 4.5.2 and 4.5.3 apply.

We have

$$\begin{aligned} z_i^* &= \bar{I}^{-T}(i,i)r_i + \sum_{p\in[k]\setminus\{i\}} \bar{I}^{-T}(i,p)r_p \\ &\geq 0 + \sum_{p\in[k]\setminus\{i\}} \bar{I}^{-T}(i,p)r_p && (\because \bar{I}^{-T}(i,i) \geq 0, r_i \geq 0) \\ &\geq -2\sqrt{2}\eta k\frac{c_{\max}}{2}. && (\text{using (4.17) and Lemma 4.5.3}) \end{aligned} \tag{4.19}$$

Combining the lower bound on z_i^* with the lower bound on β^* in Lemma 4.5.2 we get

$$\begin{aligned} z_i^* + \beta^* &\geq c_i - 3\sqrt{2}\eta k c_{\max} \\ &\geq c_{\min} - 3\sqrt{2}\eta k c_{\max} \\ &> 0. \end{aligned}$$

The last inequality above follows from our assumption on η. Indeed, we have

$$\begin{aligned} \eta &< \frac{1}{3k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right) \\ &< \frac{1}{3\sqrt{2}k}\frac{c_{\min}}{c_{\max}}. && \left(\because \frac{c_{\min}}{c_{\max}} \leq 1\right) \end{aligned}$$

Similarly, for any $t \in [k] \setminus \{i\}$ we have

$$\begin{aligned} z_t^* &\geq r_t - \|\bar{I}^{-T} - I\|_\infty \|\mathbf{r}\|_\infty && (\text{using (4.16)}) \\ &\geq r_t - 2\sqrt{2}\eta k\frac{c_{\max}}{2} && (\text{using (4.17) and Lemma 4.5.4}) \\ &\geq c_{\min} - \frac{\eta}{1-\eta}c_{\max} - 2\sqrt{2}\eta k\frac{c_{\max}}{2}. && (\text{using Lemma 4.5.3}) \end{aligned} \tag{4.20}$$

Our assumption on η yields a positive lower bound on the above expression. Indeed,

we have

$$\begin{aligned} c_{\min} &> \frac{c_{\max}}{2} + 3k\eta c_{\max} && \text{(by assumption on } \eta) \\ &\geq \frac{c_{\max}}{2} + 2(k+1)\eta c_{\max} && (\because k \geq 2) \\ &= \frac{c_{\max}}{2} + 2\eta c_{\max} + 2\eta k c_{\max} \\ &\geq \frac{c_{\max}}{2} + \frac{\eta}{1-\eta} c_{\max} + \sqrt{2}\eta k c_{\max} && (\because \eta \leq 1/2) \end{aligned}$$

Using the above in (4.20), we get

$$z_t^* > c_{\max}/2. \tag{4.21}$$

Therefore $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$ is feasible for (Di-aux).

Now we argue that the objective value of (Di-aux) at $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$ is greater than β^*. Indeed note that

$$\begin{aligned} \beta' + (2\sqrt{2}\eta k)\gamma' &= z_i^* + (2\sqrt{2}\eta k) z_s^* \\ &> -\sqrt{2}\eta k c_{\max} + 2\sqrt{2}\eta k \frac{c_{\max}}{2} && \text{(using (4.19) and (4.21))} \\ &= 0. \end{aligned}$$

That is, $\beta' + (2\sqrt{2}\eta k)\gamma' > 0$ or equivalently, $\bar{\beta} + (2\sqrt{2}\eta k)\bar{\gamma} > \beta^*$ thereby concluding the proof. □

Lemma 4.5.6. *Suppose* $\frac{c_{\min}}{c_{\max}} > \frac{1}{2}$ *and* $\eta < \frac{1}{4k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right)$. *Then for any* $s \in [k] \setminus \{i\}$, *if* y_s^* *is negative, we have*

$$y_s^* > -4\sqrt{2}\eta k. \tag{4.22}$$

Proof. Pick any $s \in [k] \setminus \{i\}$ such that $y_s^* < 0$. Consider the auxiliary LP

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{y} \\ \text{s.t.} \quad & \Theta \mathbf{y} \geq \mathbf{0} \\ & \mathbf{y}^T \boldsymbol{\theta}^i \geq 1 \\ & y_s \leq -4\sqrt{2}\eta k \end{aligned} \tag{Pi-aux}$$

and its dual

$$\begin{aligned} \max \quad & \beta + (4\sqrt{2}\eta k)\gamma \\ \text{s.t.} \quad & \beta\boldsymbol{\theta}^i - \gamma\mathbf{e_s} + \Theta^T\mathbf{u} = \mathbf{c} \\ & \beta, \gamma, \mathbf{u} \geq 0. \end{aligned} \tag{Di-aux}$$

If we show that $\mathbf{y}^*$ is not an optimal solution to (Pi-aux), then we can conclude that $y_s^* > -4\sqrt{2}\eta k$. Therefore our goal is to show that the optimal value of (Pi-aux) is greater than $\mathbf{c}^T\mathbf{y}^*$. Equivalently, we may also show that the optimal value of (Di-aux) is greater than β^*. We do so by constructing a feasible solution for (Di-aux) at which the objective value is greater than β^*.

Let $\mathbf{z}^*$ be the solution to the system

$$I'^T\mathbf{z} = \mathbf{r} + \frac{c_{\max}}{2}\mathbf{e_s} \tag{4.23}$$

where recall that $\mathbf{r} = \Theta^T\mathbf{u}^*/2$.

Note that the rows of I' belong to the unit simplex and for any $i \in [k]$, we have

$$\begin{aligned} \|I'(i,:) - \mathbf{e}^i\|_\infty &\leq \eta \\ &\leq \frac{1}{2\sqrt{2k}}. \qquad \text{(by assumption on } \eta) \end{aligned}$$

Therefore using Lemma 4.5.1, we conclude that

$$\|I'^{-T} - I\|_\infty \leq 2\sqrt{2}\eta k. \tag{4.24}$$

Define the point

$$\begin{bmatrix} \bar{\beta} \\ \bar{\gamma} \\ \bar{\mathbf{u}} \end{bmatrix} := \begin{bmatrix} \beta^* \\ 0 \\ \mathbf{u}^*/2 \end{bmatrix} + \begin{bmatrix} \beta' \\ c_{\max}/2 \\ \mathbf{u}' \end{bmatrix} \tag{4.25}$$

where $\beta' := z_i^*$ and

$$u'_p := \begin{cases} z_p^* & \text{if } p \in [k] \setminus \{i\} \\ 0 & \text{otherwise.} \end{cases}$$

First we argue that $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$ is feasible for (Di-aux). From (4.25), we have

$$\begin{aligned}
\bar{\beta}\boldsymbol{\theta}^i - \bar{\gamma}\mathbf{e_s} + \Theta^T\bar{\mathbf{u}} &= \beta^*\boldsymbol{\theta}^i + \Theta^T\mathbf{u}^*/2 + \beta'\boldsymbol{\theta}^i - c_{\max}\mathbf{e_s}/2 + \Theta^T\mathbf{u}' \\
&= \mathbf{c} - \mathbf{r} + \beta'\boldsymbol{\theta}^i - c_{\max}\mathbf{e_s}/2 + \Theta^T\mathbf{u}' \\
&\qquad (\because (\beta^*, \mathbf{u}^*) \text{ is feasible for (Di)}) \\
&= \mathbf{c} - \mathbf{r} + I'^T\mathbf{z}^* - c_{\max}\mathbf{e_s}/2 \\
&\qquad (\text{using the definition of } (\beta', \mathbf{u}')) \\
&= \mathbf{c}. \\
&\qquad (\text{using (4.23)})
\end{aligned}$$

To argue about the nonnegativity of $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$, it suffices to argue that

1. $z_i^* + \beta^* \geq 0$
2. $\mathbf{z}^*([k] \setminus \{i\}) \geq \mathbf{0}$.

Note that our assumption on η implies $\eta < \dfrac{1}{2\sqrt{2}k}\dfrac{c_{\min}}{c_{\max}}$ and therefore Lemmas 4.5.2 and 4.5.3 apply.

We have

$$\begin{aligned}
z_i^* &= I'^{-T}(i,i)r_i + I'^{-T}(i,s)(r_s + c_{\max}/2) + \sum_{p\in[k]\setminus\{i,s\}} I'^{-T}(i,p)r_p \\
&\geq 0 + I'^{-T}(i,s)(r_s + c_{\max}/2) + \sum_{p\in[k]\setminus\{i,s\}} I'^{-T}(i,p)r_p \\
&\qquad (\because I'^{-T}(i,i) \geq 0, r_i \geq 0) \\
&\geq -2\sqrt{2}\eta k c_{\max}. \\
&\qquad (\text{using (4.24) and Lemma 4.5.3})
\end{aligned} \tag{4.26}$$

Combining the lower bound on z_i^* with the lower bound on β^* in Lemma 4.5.2 yields

$$\begin{aligned}
z_i^* + \beta^* &\geq c_i - 4\sqrt{2}\eta k c_{\max} \\
&\geq c_{\min} - 4\sqrt{2}\eta k c_{\max} \\
&> 0.
\end{aligned}$$

The last inequality above follows from our assumption on η. Indeed, we have

$$\begin{aligned}\eta &< \frac{1}{4k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right) \\ &< \frac{1}{4\sqrt{2}k}\frac{c_{\min}}{c_{\max}}. && \left(\because \frac{c_{\min}}{c_{\max}} \leq 1\right)\end{aligned}$$

Similarly, for any $t \in [k] \setminus \{i\}$ we have

$$\begin{aligned}z_t^* &\geq r_t + c_{\max} I(s,t)/2 - \|I'^{-T} - I\|_\infty \|\mathbf{r} + c_{\max}\mathbf{e_s}/2\|_\infty && \text{(using (4.23))} \\ &\geq r_t - \|I'^{-T} - I\|_\infty \|\mathbf{r} + c_{\max}\mathbf{e_s}/2\|_\infty \\ &\geq r_t - 2\sqrt{2}\eta k c_{\max} && \text{(using (4.24) and Lemma 4.5.4)} \\ &\geq c_{\min} - \frac{\eta}{1-\eta}c_{\max} - 2\sqrt{2}\eta k c_{\max}. && \text{(using Lemma 4.5.3)}\end{aligned} \tag{4.27}$$

Our assumption on η yields a positive lower bound on the above expression. Indeed, we have

$$\begin{aligned}c_{\min} &> \frac{c_{\max}}{2} + 4k\eta c_{\max} && \text{(by assumption on } \eta) \\ &\geq \frac{c_{\max}}{2} + (2+3k)\eta c_{\max} && (\because k \geq 2) \\ &= \frac{c_{\max}}{2} + 2\eta c_{\max} + 3\eta k c_{\max} \\ &\geq \frac{c_{\max}}{2} + \frac{\eta}{1-\eta}c_{\max} + 2\sqrt{2}\eta k c_{\max} && (\because \eta \leq 1/2)\end{aligned}$$

Using the above in (4.27), we get

$$z_t^* > c_{\max}/2. \tag{4.28}$$

Therefore $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$ is feasible for (Di-aux).

Now we argue that the objective value of (Di-aux) at $(\bar{\beta}, \bar{\gamma}, \bar{\mathbf{u}})$ is greater than β^*. Indeed note that

$$\begin{aligned}\beta' + (4\sqrt{2}\eta k)\frac{c_{\max}}{2} &= z_i^* + (4\sqrt{2}\eta k)\frac{c_{\max}}{2} \\ &> -2\sqrt{2}\eta k c_{\max} + (4\sqrt{2}\eta k)\frac{c_{\max}}{2} && \text{(using (4.26))} \\ &= 0.\end{aligned}$$

That is, $\beta' + (4\sqrt{2}\eta k)\frac{c_{\max}}{2} > 0$ or equivalently, $\bar{\beta} + (4\sqrt{2}\eta k)\bar{\gamma} > \beta^*$ thereby concluding the proof. □

Lemma 4.5.7. *Suppose* $\frac{c_{\min}}{c_{\max}} > \frac{1}{2}$ *and* $\eta < \frac{1}{4k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right)$. *Then*

$$\frac{1 - 4\sqrt{2}\eta^2 k}{\theta_{ii}} \leq y_i^* \leq \frac{1 + 4\sqrt{2}\eta^2 k}{\theta_{ii}}. \tag{4.29}$$

Proof. We note that the constraint $\mathbf{y}^T\boldsymbol{\theta}^i \geq 1$ in (Pi) is tight at optimality. Indeed otherwise one may scale the optimal solution so as to make that constraint tight and obtain a strictly smaller objective value, thereby contradicting optimality.

Then we have

$$\begin{aligned} 1 &= \mathbf{y}^{*T}\boldsymbol{\theta}^i \\ &= y_i^*\theta_{ii} + \sum_{s\in[k]\setminus\{i\}} y_s^*\theta_{is}. \end{aligned} \tag{4.30}$$

Moreover

$$\begin{aligned} \left|\sum_{s\in[k]\setminus\{i\}} y_s^*\theta_{is}\right| &\leq \|\mathbf{y}^*([k]\setminus\{i\})\|_\infty \|\boldsymbol{\theta}^i([k]\setminus\{i\})\|_1 && \text{(using Hölder's inequality)} \\ &\leq \eta\|\mathbf{y}^*([k]\setminus\{i\})\|_\infty && (\because \|\boldsymbol{\theta}^i([k]\setminus\{i\})\|_1 \leq \eta) \\ &\leq 4\sqrt{2}\eta^2 k. && \text{(using Lemmas 4.5.5 and 4.5.6)} \end{aligned} \tag{4.31}$$

Using (4.31) in (4.30) yields the desired result. □

Proof of Theorem 4.4.3. First note that (P) is both feasible and bounded below, which implies that it has an optimal solution. By assumption, there exists a $k \times k$ submatrix of Θ whose entrywise distance from I is at most η; this implies that the spectral norm distance of such a submatrix from I is at most ηk which is, by assumption, at most $(c_{\min}/c_{\max} - 1/2)/4$ which is at most one. This implies that the column rank of Θ is k. Therefore using the fact that B is full-rank, we conclude that the column range of Θ is equal to the range of P and consequently the rank of P is k. Therefore (P) may be rewritten as

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{y} \\ \text{s.t.} \quad & \Theta\mathbf{y} \geq \mathbf{0} \\ & \mathbf{y}^T\boldsymbol{\theta}^i \geq 1. \end{aligned} \tag{Py}$$

Note that (Py) is both feasible and bounded below, which implies that it has an optimal solution. Since $\mathbf{x}^*$ is an optimal solution to (P), there exists an optimal solution to (Py), called $\mathbf{y}^*$, satisfying $\Theta\mathbf{y}^* = \mathbf{x}^*$. Using Lemmas 4.5.5, 4.5.6, and 4.5.7, we conclude that

$$\left\|\mathbf{y}^* - \frac{\mathbf{e}_j}{\theta_{ij}}\right\|_\infty \leq \sqrt{2}\eta k \max\{2, 4, 4\eta/\theta_{ij}\} = 4\sqrt{2}\eta k. \tag{4.32}$$

The last equality above holds because $\theta_{ij} \geq 1 - \eta$ and $\eta < 1/2$. Then we have

$$\begin{aligned}
\left\|\mathbf{x}^* - \frac{\boldsymbol{\theta}_j}{\theta_{ij}}\right\|_\infty &= \left\|\Theta\mathbf{y}^* - \Theta\frac{\mathbf{e}_j}{\theta_{ij}}\right\|_\infty \\
&\leq \|\Theta\|_\infty \left\|\mathbf{y}^* - \frac{\mathbf{e}_j}{\theta_{ij}}\right\|_\infty \\
&\leq 4\sqrt{2}\eta k. \qquad (\|\Theta\|_\infty = 1 \text{ and using (4.32)})
\end{aligned} \tag{4.33}$$

Lastly, we have

$$\begin{aligned}
\left\|\frac{\mathbf{x}^*}{\|\mathbf{x}^*\|_\infty} - \boldsymbol{\theta}_j\right\|_\infty &\leq \left\|\frac{\mathbf{x}^*}{\|\mathbf{x}^*\|_\infty} - \mathbf{x}^*\right\|_\infty + \left\|\mathbf{x}^* - \frac{\boldsymbol{\theta}_j}{\theta_{ij}}\right\|_\infty + \left\|\frac{\boldsymbol{\theta}_j}{\theta_{ij}} - \boldsymbol{\theta}_j\right\|_\infty \\
&\qquad \text{(using triangle inequality)} \\
&= |1 - \|\mathbf{x}^*\|_\infty| + \left\|\mathbf{x}^* - \frac{\boldsymbol{\theta}_j}{\theta_{ij}}\right\|_\infty + \left\|\frac{\boldsymbol{\theta}_j}{\theta_{ij}} - \boldsymbol{\theta}_j\right\|_\infty \\
&\leq \left|1 - \frac{\|\boldsymbol{\theta}_j\|_\infty}{\theta_{ij}}\right| + \left|\|\mathbf{x}^*\|_\infty - \frac{\|\boldsymbol{\theta}_j\|_\infty}{\theta_{ij}}\right| + \left\|\mathbf{x}^* - \frac{\boldsymbol{\theta}_j}{\theta_{ij}}\right\|_\infty + \left\|\frac{\boldsymbol{\theta}_j}{\theta_{ij}} - \boldsymbol{\theta}_j\right\|_\infty \\
&\qquad \text{(using triangle inequality)} \\
&\leq \left|1 - \frac{\|\boldsymbol{\theta}_j\|_\infty}{\theta_{ij}}\right| + 2\left\|\mathbf{x}^* - \frac{\boldsymbol{\theta}_j}{\theta_{ij}}\right\|_\infty + \left\|\frac{\boldsymbol{\theta}_j}{\theta_{ij}} - \boldsymbol{\theta}_j\right\|_\infty \\
&\qquad \text{(using reverse triangle inequality)} \\
&\leq \left(\frac{\|\boldsymbol{\theta}_j\|_\infty}{\theta_{ij}} - 1\right) + 8\sqrt{2}\eta k + \left(\frac{1}{\theta_{ij}} - 1\right)\|\boldsymbol{\theta}_j\|_\infty \\
&\leq 8\sqrt{2}\eta k + 2\left(\frac{1}{\theta_{ij}} - 1\right) \\
&\leq 8\sqrt{2}\eta k + \frac{2\eta}{1-\eta} \\
&< 8\sqrt{2}\eta k + 4\eta \\
&= 4\eta(2\sqrt{2}k + 1)
\end{aligned}$$

where the inequality in the fifth line from bottom follows from using (4.33), the inequality in the fourth line from bottom follows because $\|\boldsymbol{\theta}_j\|_\infty \leq 1$, the inequality in the third line from bottom follows because $\theta_{ij} \geq 1-\eta$, and the inequality in the second line from bottom follows because $\eta < 1/2$.

Lastly we provide an argument for the time complexity claim. Since the rank of P is k, the column range of P is same as the column range of V where V is an $n \times k$ matrix whose columns contain the eigenvectors of P corresponding to its k nonzero eigenvalues. This implies that (P) is equivalent to $\{\min \mathbf{e}^T(V\mathbf{y})$ subject to $V\mathbf{y} \geq 0, (V\mathbf{y})_{\mathcal{J}(i)} \geq 1\}$ which contains $n+1$ constraints and k variables. Hence the result in [55] implies that (P) can be solved in $\mathcal{O}(n)$ time. Moreover, V can be obtained from P in $\mathcal{O}(n^2)$ time using, for instance, randomized SVD techniques [34]. □

Proof of Corollary 4.4.4. Define $P := \Theta B \Theta^T = \bar{\Theta}\bar{B}\bar{\Theta}^T$, and let $\mathbf{x}^*$ be an optimal solution to (P). From Theorem 4.4.3, we know that there exists a $j \in [k]$ such that the distance between vector $\mathbf{x}^*/\|\mathbf{x}^*\|_\infty$ and Θ_j is at most $4\eta(2\sqrt{2}k+1)$. Similarly, there also exists a $\bar{j} \in [k]$ such that the distance between vector $\mathbf{x}^*/\|\mathbf{x}^*\|_\infty$ and $\bar{\Theta}_{\bar{j}}$ is at most $4\bar{\eta}(2\sqrt{2}k+1)$.

Combining the above two observations with the triangle inequality yields the desired result. □

4.5.2 Some Concentration Properties in the MMSB

In this section, we show concentration properties of some key random variables associated with random matrices Θ and ΘB. We shall use these observations for our subsequent proofs, but they may also be of independent interest. Even though we work the equal parameter Dirichlet distribution, the proof techniques here easily extend to the case with different Dirichlet parameters.

Define $l := \sigma_{\min}(B)$ and $u := \sigma_{\max}(B)$. Suppose the k parameters of the Dirichlet distribution are all equal to α. We repeatedly use the facts that for any $i \in [n]$, $s \in [k]$,

$$\mathbb{E}[\theta_{is}] = \frac{1}{k} \tag{4.34}$$

and

$$\mathbb{E}[\theta_{is}^2] = \frac{\alpha+1}{k(\alpha k+1)}. \tag{4.35}$$

Moreover, if $s, t \in [k]$ such that $s \neq t$ then

$$\mathbb{E}[\theta_{is}\theta_{it}] = \frac{\alpha}{k(\alpha k+1)}. \tag{4.36}$$

Lemma 4.5.8. *For any $j \in [k]$, we have $\frac{9}{10}\frac{n}{k} \leq c_j \leq \frac{11}{10}\frac{n}{k}$ with probability at least $1 - 2\exp\left(\frac{-n}{50k^2}\right)$.*

Proof. For any $j \in [k]$, c_j is the sum of n independent bounded random variables $\{\theta_{ij}\}_{i=1}^{n}$. Indeed each row of Θ is sampled independently and each entry of Θ lies in $[0, 1]$. Moreover, using (4.34) we get that $\mathbb{E}[c_j] = n/k$. Thus, using Hoeffding's inequality, we have that for any $z > 0$

$$\Pr(|c_j - n/k| \geq z) \leq 2\exp\left(\frac{-2z^2}{n}\right). \tag{4.37}$$

Setting $z = n/10k$ in (4.37) yields the desired result. □

Corollary 4.5.9. *We have $c_{\min}/c_{\max} \geq 9/11$ with probability at least $1 - p_1$, where $p_1 := 2k\exp\left(\frac{-n}{50k^2}\right)$.*

Proof. Lemma 4.5.8 implies that with probability at least $1 - 2k\exp\left(\frac{-n}{50k^2}\right)$, both $c_{\min} \geq 9n/10k$ and $c_{\max} \leq 11n/10k$ hold. □

Lemma 4.5.10. *For any $\epsilon > 0$, $\|\Theta B\| \leq u\sqrt{\frac{2n}{k}} + \epsilon\|\Theta\|$ with probability at least $1 - \left(\frac{2u}{\epsilon} + 1\right)^k \exp\left(\frac{-2n}{k^2}\right)$.*

For proving Lemma 4.5.10, we first prove the following statements for set $\mathcal{C} := \{\mathbf{y} \in \mathbb{R}^k : \exists\, \mathbf{x} \in \mathbb{R}^k \text{ such that } B\mathbf{x} = \mathbf{y}, \|\mathbf{x}\| = 1\}$ defined as the image of the unit sphere under B.

Lemma 4.5.11. *If $\mathcal{E}$ is an ϵ-net of $\mathcal{C}$ of smallest possible cardinality, then*

$$|\mathcal{E}| \leq \left(\frac{2u}{\epsilon} + 1\right)^k.$$

Proof. Let $\mathcal{E}'$ be a maximal ϵ-separated subset of $\mathcal{C}$. Note that by definition of an ϵ-separated subset, for any distinct $\mathbf{x}, \mathbf{y} \in \mathcal{E}'$, we have $\|\mathbf{x}-\mathbf{y}\| > \epsilon$. Moreover, the maximality of $\mathcal{E}'$ implies that $\mathcal{E}'$ is also an ϵ-net of $\mathcal{C}$. Therefore

$$|\mathcal{E}| \leq |\mathcal{E}'|. \tag{4.38}$$

We also have that the union of $|\mathcal{E}'|$ disjoint balls $\bigcup_{\mathbf{x}\in\mathcal{E}'} \mathcal{B}(\mathbf{x}, \epsilon/2) \subseteq \mathcal{C} + \mathcal{B}(\mathbf{0}, \epsilon/2) \subseteq \mathcal{B}(\mathbf{0}, u + \epsilon/2)$. Therefore

$$\text{vol}\left(\bigcup_{\mathbf{x}\in\mathcal{E}'} \mathcal{B}(\mathbf{x}, \epsilon/2)\right) \leq \text{vol}(\mathcal{B}(\mathbf{0}, u + \epsilon/2)) \tag{4.39}$$

which implies that $|\mathcal{E}'|(\epsilon/2)^k \leq (u + \epsilon/2)^k$ which yields the desired result when combined with (4.38). □

Lemma 4.5.12. *Suppose* $\mathbf{y} \in \mathcal{C}$. *For any* $i \in [n]$*:*

1. $0 \leq \langle \boldsymbol{\theta}^i, \mathbf{y}\rangle^2 \leq u^2$
2. $\dfrac{l^2}{k(\alpha k + 1)} \leq \mathbb{E}[\langle \boldsymbol{\theta}^i, \mathbf{y}\rangle^2] \leq \dfrac{u^2}{k}$

Proof. Let $\mathbf{y} = B\mathbf{x}$ such that $\|\mathbf{x}\| = 1$. Then $l \leq \|\mathbf{y}\| \leq u$.

1. We have

$$\begin{aligned} \langle \boldsymbol{\theta}^i, \mathbf{y}\rangle^2 &\leq \|\boldsymbol{\theta}^i\|^2\|\mathbf{y}\|^2 && \text{(using Cauchy-Schwarz inequality)} \\ &\leq u^2 && (\|\boldsymbol{\theta}^i\| \leq 1). \end{aligned}$$

2. We have

$$\begin{aligned} \mathbb{E}[\langle \boldsymbol{\theta}^i, \mathbf{y}\rangle^2] &= \mathbb{E}[\theta_{i1}^2 y_1^2 + \cdots + \theta_{ik}^2 y_k^2] + \mathbb{E}\left[\sum_{\substack{s,t\in[k]:\\ s\neq t}} \theta_{is}\theta_{it} y_s y_t\right] \\ &= \frac{\alpha + 1}{k(\alpha k + 1)}\|\mathbf{y}\|^2 + \mathbb{E}\left[\sum_{\substack{s,t\in[k]:\\ s\neq t}} \theta_{is}\theta_{it} y_s y_t\right] && \text{(using (4.35))} \\ &= \frac{\alpha + 1}{k(\alpha k + 1)}\|\mathbf{y}\|^2 + \frac{\alpha}{k(\alpha k + 1)} \sum_{\substack{s,t\in[k]:\\ s\neq t}} y_s y_t && \text{(using (4.36))} \\ &= \frac{1}{k(\alpha k + 1)}\|\mathbf{y}\|^2 + \frac{\alpha}{k(\alpha k + 1)}(\mathbf{e}^T\mathbf{y})^2 && \text{(re-arranging terms).} \end{aligned}$$

Now noting the second term on the right hand side above is nonnegative yields the desired lower bound.

Similarly noting that $\mathbf{e}^T\mathbf{y} \leq u\sqrt{k}$ (using Cauchy-Schwarz inequality) yields the desired upper bound.

□

Proof of Lemma 4.5.10. We have

$$\|\Theta B\| = \sup_{\mathbf{x}\in S^{k-1}} \|\Theta B\mathbf{x}\| = \sup_{\mathbf{y}\in\mathcal{C}} \|\Theta\mathbf{y}\|. \tag{4.40}$$

Let $\mathcal{E}$ denote an ϵ-net of $\mathcal{C}$ of smallest possible cardinality. Then we have

$$\|\Theta B\| \leq \sup_{\mathbf{y}\in\mathcal{E}} \|\Theta\mathbf{y}\| + \epsilon\|\Theta\|. \tag{4.41}$$

Indeed if the supremum defining $\|\Theta B\|$ on the RHS in (4.40) is attained at $\mathbf{y}_s$, and if $\mathbf{y}_e$ is a point in $\mathcal{E}$ such that $\|\mathbf{y}_s - \mathbf{y}_e\| \leq \epsilon$, then

$$\begin{aligned}
\|\Theta B\| &= \|\Theta\mathbf{y}_s\| \\
&= \|\Theta\mathbf{y}_e + \Theta(\mathbf{y}_s - \mathbf{y}_e)\| \\
&\leq \|\Theta\mathbf{y}_e\| + \|\Theta(\mathbf{y}_s - \mathbf{y}_e)\| && \text{(using triangle inequality)} \\
&\leq \sup_{\mathbf{y}\in\mathcal{E}} \|\Theta\mathbf{y}\| + \epsilon\|\Theta\|.
\end{aligned}$$

For any $\mathbf{y} \in \mathcal{E}$, we have

$$\|\Theta\mathbf{y}\|^2 = \langle\boldsymbol{\theta}^1, \mathbf{y}\rangle^2 + \cdots + \langle\boldsymbol{\theta}^n, \mathbf{y}\rangle^2.$$

Now note that $\|\Theta\mathbf{y}\|^2$ is the sum of n independent random variables. Indeed using Lemma 4.5.12 we conclude that each of these random variables is bounded and that $\mathbb{E}[\|\Theta\mathbf{y}\|^2] \leq \frac{nu^2}{k}$. Thus, using Hoeffding's inequality, we have that for any $z > 0$,

$$\begin{aligned}
\Pr\left(\|\Theta\mathbf{y}\|^2 \geq \frac{nu^2}{k} + z\right) &\leq \Pr(\|\Theta\mathbf{y}\|^2 \geq \mathbb{E}[\|\Theta\mathbf{y}\|^2] + z) \\
&\leq \exp\left(\frac{-2z^2}{nu^4}\right).
\end{aligned}$$

Then using the union bound over the ϵ-net, we obtain that

$$\Pr\left(\sup_{\mathbf{y}\in\mathcal{E}}\|\Theta\mathbf{y}\| \geq \sqrt{\frac{nu^2}{k}+z}\right) \leq |\mathcal{E}|\exp\left(\frac{-2z^2}{nu^4}\right)$$
$$\leq \left(\frac{2u}{\epsilon}+1\right)^k \exp\left(\frac{-2z^2}{nu^4}\right) \qquad \text{(using Lemma 4.5.11)}$$

Setting $z = nu^2/k$ in the above, we note that $\sup_{\mathbf{y}\in\mathcal{E}}\|\Theta y\| \leq u\sqrt{\frac{2n}{k}}$ with probability at least $1-\left(\frac{2u}{\epsilon}+1\right)^k \exp\left(\frac{-2n}{k^2}\right)$, combining which with (4.41) yields the desired result. □

Corollary 4.5.13. *$\|\Theta\| \leq 2\sqrt{\frac{2n}{k}}$ with probability at least $1-p_2$, where*

$$p_2 := 5^k \exp\left(\frac{-2n}{k^2}\right).$$

Proof. Set $B = I$ and $\epsilon = 1/2$ in Lemma 4.5.10. □

Corollary 4.5.14. *$\|\Theta B\| \leq 2u\sqrt{\frac{2n}{k}}$ with probability at least $1-p_2$.*

Proof. This follows simply from using the inequality $\|\Theta B\| \leq \|\Theta\|\|B\|$ and the upper bound obtained in Corollary 4.5.13. □

Lemma 4.5.15. *$\sigma_k(\Theta B) \geq \frac{1}{4}\frac{l}{\sqrt{\alpha k+1}}\sqrt{\frac{2n}{k}}$ with probability at least $1-p_3$, where*

$$p_3 := p_2 + \left(\frac{16u\sqrt{\alpha k+1}}{l}+1\right)^k \exp\left(\frac{-nl^4}{2k^2u^4(\alpha k+1)^2}\right).$$

Proof. We have

$$\sigma_k(\Theta B) = \inf_{\mathbf{x}\in S^{k-1}}\|\Theta B\mathbf{x}\| = \inf_{\mathbf{y}\in\mathcal{C}}\|\Theta\mathbf{y}\|. \tag{4.42}$$

Let $\mathcal{E}$ denote an ϵ-net of $\mathcal{C}$ of smallest possible cardinality. Then we have

$$\sigma_k(\Theta B) \geq \inf_{\mathbf{y}\in\mathcal{E}}\|\Theta\mathbf{y}\| - \epsilon\|\Theta\|. \tag{4.43}$$

Indeed if the infimum defining $\sigma_k(\Theta B)$ on the RHS in (4.42) is attained at $\mathbf{y}_s$, and if $\mathbf{y}_e$ is a point in $\mathcal{E}$ such that $\|\mathbf{y}_s - \mathbf{y}_e\| \leq \epsilon$, then

$$\begin{aligned}
\sigma_k(\Theta B) &= \|\Theta \mathbf{y}_s\| \\
&= \|\Theta \mathbf{y}_e + \Theta(\mathbf{y}_s - \mathbf{y}_e)\| \\
&\geq |\|\Theta \mathbf{y}_e\| - \|\Theta(\mathbf{y}_s - \mathbf{y}_e)\|| \qquad \text{(using reverse triangle inequality)} \\
&\geq \|\Theta \mathbf{y}_e\| - \|\Theta(\mathbf{y}_s - \mathbf{y}_e)\| \\
&\geq \inf_{\mathbf{y} \in \mathcal{E}} \|\Theta \mathbf{y}\| - \epsilon \|\Theta\|.
\end{aligned}$$

For any $\mathbf{y} \in \mathcal{E}$, we have

$$\|\Theta \mathbf{y}\|^2 = \langle \boldsymbol{\theta}^1, \mathbf{y} \rangle^2 + \cdots + \langle \boldsymbol{\theta}^n, \mathbf{y} \rangle^2.$$

Now note that $\|\Theta \mathbf{y}\|^2$ is the sum of n independent bounded random variables. Indeed using Lemma 4.5.12 we conclude that each of these random variables is bounded and that $\mathbb{E}[\|\Theta y\|^2] \geq \dfrac{nl^2}{k(\alpha k + 1)}$. Thus, using Hoeffding's inequality, we have that for any $z > 0$,

$$\begin{aligned}
\Pr\left(\|\Theta y\|^2 \leq \frac{nl^2}{k(\alpha k + 1)} - z \right) &\leq \Pr(\|\Theta y\|^2 \leq \mathbb{E}[\|\Theta y\|^2] - z) \\
&\leq \exp\left(\frac{-2z^2}{nu^4} \right).
\end{aligned}$$

Then using the union bound over the ϵ-net, we obtain that

$$\begin{aligned}
\Pr\left(\inf_{y \in \mathcal{E}} \|\Theta y\| \leq \sqrt{\frac{nl^2}{k(\alpha k + 1)} - z} \right) &\leq |\mathcal{E}| \exp\left(\frac{-2z^2}{nu^4} \right) \\
&\leq \left(\frac{2u}{\epsilon} + 1 \right)^k \exp\left(\frac{-2z^2}{nu^4} \right) \quad \text{(using Lemma 4.5.11)}
\end{aligned}$$

Setting $z = \dfrac{1}{2} \dfrac{nl^2}{k(\alpha k + 1)}$, we note that $\inf\limits_{y \in \mathcal{E}} \|\Theta y\| \geq \sqrt{\dfrac{1}{2} \dfrac{nl^2}{k(\alpha k + 1)}}$ with probability at least $1 - \left(\dfrac{2u}{\epsilon} + 1 \right)^k \exp\left(\dfrac{-nl^4}{2k^2 u^4 (\alpha k + 1)^2} \right)$.

Using (4.43), we get that

$$\sigma_k(\Theta B) \geq \sqrt{\frac{1}{2}\frac{nl^2}{k(\alpha k+1)}} - \epsilon\|\Theta\| \tag{4.44}$$

with probability at least $1 - \left(\frac{2u}{\epsilon}+1\right)^k \exp\left(\frac{-nl^4}{2k^2u^4(\alpha k+1)^2}\right)$.

Lastly, using the upper bound on $\|\Theta\|$ derived in Corollary 4.5.13 in (4.44), we get that

$$\sigma_k(\Theta B) \geq \frac{1}{2}\frac{l}{\sqrt{\alpha k+1}}\sqrt{\frac{2n}{k}} - 2\epsilon\sqrt{\frac{2n}{k}}$$

with probability at least $1-p_2-\left(\frac{2u}{\epsilon}+1\right)^k \exp\left(\frac{-nl^4}{2k^2u^4(\alpha k+1)^2}\right)$. Setting $\epsilon = \frac{1}{8}\frac{l}{\sqrt{\alpha k+1}}$ yields the desired result. □

4.5.3 Proof of Main Theorem

In this section, we build the proof of Theorem 4.4.1.

Lemma 4.5.16. *Let $p, \gamma \in (0,1)$. If $n > \frac{\log(p/k)}{\log I_{1-\gamma}(\alpha,(k-1)\alpha)}$, then with probability at least $1-p$, for each $j \in [k]$, there exists a row vector $\mathbf{r}^T$ in Θ such that*

$$\|\mathbf{r} - \mathbf{e}_j\|_\infty < \gamma. \tag{4.45}$$

(Here $I_x(y,z)$ denotes the regularized incomplete beta function.)

Proof. For any $j \in [k]$, define E_j as the event that there exists a row $\mathbf{r}^T$ in Θ such that

$\|\mathbf{r} - \mathbf{e}_j\|_\infty < \gamma$. Then for any $j \in [k]$, we have

$$\begin{aligned}
\Pr(E_j^c) &= \prod_{i \in [n]} \Pr(\|\boldsymbol{\theta}^i - \mathbf{e}_j\|_\infty \geq \gamma) \\
&\qquad (\because \text{rows of } \Theta \text{ are independently sampled}) \\
&= \prod_{i \in [n]} \Pr(\theta_{ij} \leq 1 - \gamma) \\
&\qquad (\because \text{rows of } \Theta \text{ belong to unit simplex}) \\
&= [I_{1-\gamma}(\alpha, (k-1)\alpha)]^n \\
&\qquad (I_x(y,z) \text{ is the CDF of marginal of Dirichlet distribution}) \\
&< p/k. \\
&\qquad (\text{by assumption on } n)
\end{aligned} \tag{4.46}$$

Therefore

$$\begin{aligned}
\Pr(E_1 \cap \cdots \cap E_k) &= 1 - \Pr(E_1^c \cup \cdots \cup E_k^c) \\
&\geq 1 - \sum_{j \in [k]} \Pr(E_j^c) && \text{(using the union bound)} \\
&> 1 - p. && \text{(using (4.46))}
\end{aligned}$$

□

Proof of Theorem 4.4.1. Using the lower bound assumption on n and Lemma 4.5.16, we conclude that with probability at least $1 - p$, for each $j \in [k]$, there exists a row $\mathbf{r}^T$ in Θ such that

$$\|\mathbf{r} - \mathbf{e}_j\|_\infty < \epsilon. \tag{4.47}$$

Recalling the definition of Δ, we note that (4.47) is equivalent to

$$\|\Delta\|_{\max} < \epsilon. \tag{4.48}$$

Using Corollary 4.5.14 and Lemma 4.5.15, we conclude that

$$\kappa_0 \leq 8\kappa\sqrt{\alpha k + 1} \tag{4.49}$$

with probability at least $1 - p_2 - p_3$. Therefore (4.49) implies that

$$\begin{aligned}
\min\left(\frac{1}{\sqrt{k-1}}, \frac{1}{2}\right) \frac{1}{2\sqrt{2}\kappa_0(1 + 80\kappa_0^2)} &\geq \epsilon_1 && \text{(using the definition of } \epsilon_1) \\
&> \epsilon && \text{(using the assumption on } \epsilon) \\
&> \|\Delta\|_{\max} && \text{(using (4.48))}
\end{aligned} \tag{4.50}$$

with probability at least $1 - p - p_2 - p_3$.

Using (4.50), we note that the assumption of Theorem 4.4.2 is satisfied with probability at least $1 - p - p_2 - p_3$. Therefore the set $\mathcal{J}$ returned by Algorithm 2 satisfies

$$\begin{aligned} \|\Pi\Theta(\mathcal{J}, :) - I\|_{\max} &\leq 40\sqrt{2}\kappa_0^2 \|\Delta\|_{\max} \\ &< 40\sqrt{2}\kappa_0^2 \epsilon \end{aligned} \tag{4.51}$$

with probability at least $1 - p - p_2 - p_3$ for some $k \times k$ permutation matrix Π.

Now from Corollary 4.5.9, we know that

$$\frac{1}{4k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right) \geq \frac{7}{88k} \tag{4.52}$$

with probability at least $1 - p_1$.

Thus we have

$$\begin{aligned} 40\sqrt{2}\kappa_0^2\epsilon &< 40\sqrt{2}\kappa_0^2\epsilon_2 && \text{(using the assumption on } \epsilon) \\ &\leq 40\sqrt{2} \cdot 64\kappa^2(\alpha k + 1)\epsilon_2 && \text{(using (4.49))} \\ &= \frac{7}{88k} && \text{(using the definition of } \epsilon_2) \\ &\leq \frac{1}{4k}\left(\frac{c_{\min}}{c_{\max}} - \frac{1}{2}\right) && \text{(using (4.52))} \end{aligned} \tag{4.53}$$

with probability at least $1 - p - p_1 - p_2 - p_3$. Combining (4.51) and (4.53), we conclude that the assumption of Theorem 4.4.3 is satisfied with probability at least $1 - p - p_1 - p_2 - p_3$. Therefore for any $j \in [k]$, the vector $\hat{\boldsymbol{\theta}}_j$ returned by `SP+LP` satisfies

$$\begin{aligned} \|\hat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j\|_\infty &\leq 4 \cdot 40\sqrt{2}\kappa_0^2\epsilon \cdot (2\sqrt{2}k + 1) \\ &\leq 10240\sqrt{2}\kappa^2(\alpha k + 1)(2\sqrt{2}k + 1)\epsilon \quad \text{(using (4.49))} \\ &= \mathcal{O}(\alpha k^2 \kappa^2 \epsilon) \end{aligned}$$

with probability at least $1 - p - p_1 - p_2 - p_3$. Substituting the expressions for p_1, p_2 and p_3, the probability $1 - p - p_1 - p_2 - p_3$ can be expressed as $1 - p - c_1 e^{-c_2 n}$ such that c_1, c_2 are constants that depend on α, k, κ. □

4.6 Experiments

In this section, we compare the performance of SP+LP on both synthetic and real-world graphs with other popular algorithms. In practice, the user has access to the adjacency matrix, called A, of the observed weighted graph which is only an approximation of P. Matrix A may even be full-rank, and so for implementation we have to slightly modify the constraint $\mathbf{x} = P\mathbf{y}$ in the LP in SP+LP. (Indeed note that if A is full-rank, then the optimal solution to the LP is $\mathbf{e}_{\mathcal{J}(i)}$.) Specifically, we replace that constraint with $\mathbf{x} = V\mathbf{y}$ where V is an $n \times k$ matrix whose columns contain the eigenvectors of A corresponding to either its k largest eigenvalues or singular values. The intuition behind this is that we expect the range of V to approximate the k-dimensional subspace of $\mathbb{R}^n$ which is the range of P. For efficient computation of V, one may employ, for instance, the Lanczos method or randomized SVD [34].

4.6.1 Synthetic Graphs

We demonstrate the performance of SP+LP on artificial graphs generated according to the MMSB. In practice, the weighted adjacency matrix available is only approximately equal to P. Therefore for our experiments, we compute a weighted adjacency matrix by averaging s number of $0,1$-adjacency matrices, each of which is sampled according to P. That is, entry ij of a sampled adjacency matrix is a Bernoulli random variable with parameter P_{ij}. The diagonal entries in these adjacency matrices are all set to 1.

Evaluation Metrics: We evaluate SP+LP in terms of the entrywise error in the predicted columns of Θ and the wall-clock running time (Figure 4.1). The entrywise error is defined as $\min_{\Pi} \|\hat{\Theta} - \Theta\Pi\|_{\max}$ over all $k \times k$ permutation matrices Π, where $\hat{\Theta} := \begin{bmatrix}\hat{\boldsymbol{\theta}}_1 & \dots & \hat{\boldsymbol{\theta}}_k\end{bmatrix}$ contains as columns the predicted community characteristic vectors. For each plot, each point is determined by averaging the results over 10 samples and the error bars represent one standard deviation.

We compare our results with the GeoNMF algorithm which has been shown in [51] to computationally outperform popular methods such as Stochastic Variational Inference (SVI) by [33], a Bayesian variant of SNMF by [64], the OCCAM algorithm by [85], and the SAAC algorithm by [41]. We use the implementation of GeoNMF that is made available by the authors without any modification and also the provided default values for the tuning parameters.

Parameter Settings: Unless otherwise stated, the default parameter settings are $n = 5000, k = 3, \alpha = 0.5, s = \sqrt{n}$. Figures 4.1(d) and 4.1(f) show the performance of

the SP+LP for community interaction matrices B with higher off-diagonal elements. More specifically, for those plots, we set $B = (1-\delta)\cdot I + \delta \cdot \mathbf{e}\mathbf{e}^T$. For Figures 4.1(a), 4.1(b), 4.1(c), 4.1(e), we set $B = 0.5 \cdot I + 0.5 \cdot R$ where R is a $k \times k$ diagonal matrix whose each diagonal entry is generated from a uniform distribution over $[0, 1]$. One reason for choosing these parameter settings is to have a fair comparison. Indeed GeoNMF has already been shown to perform well over these parameter choices.

Figures 4.1(a), 4.1(b), 4.1(c), 4.1(d) demonstrate that SP+LP outperforms GeoNMF in terms of the entrywise error in the recovered MMSB communities with increasing n, k, α and δ. In particular, this implies that, compared to GeoNMF, SP+LP can handle larger graphs, more number of communities, more overlap among the communities, and a more general community interaction matrix B, while involving a lesser number of tuning parameters. However, Figure 4.1(e) shows that SP+LP is slower compared to GeoNMF and that opens up possibilities for future work to expedite SP+LP. On the other hand, Figure 4.1(f) shows that for a more general B, the time performances of GeoNMF and SP+LP are quite comparable.

4.6.2 Real-world Graphs

For practical application of SP+LP, we consider a well-studied problem in computational biology: that of clustering functionally similar proteins together based on protein-protein interaction (PPI) observations (see [58]). In the language of our problem setup, each node in the weighted graph represents a protein, and the weights represent the reliability with which any two proteins interact. The communities or clusters of similar proteins are called *protein complexes* in biology literature.

It is important to highlight that the PPI networks typically contain a large number of communities compared to the number of nodes and therefore our theory does not necessarily guarantee that SP+LP will succeed with high probability. Despite that, we observe that on some datasets, SP+LP matches or even outperforms commonly-used protein complex detection heuristics. Additionally, protein complex detection is a very well-studied problem in biology and there exist a vast number of heuristics which are tailored for this specific problem. For instance, recent works of [82], [83], and [84] incorporate existing ground truth knowledge of protein complexes in the algorithm to obtain a supervised-learning-based approach. Our goal in this paper is not to design a fine-tuned method specifically for protein complex detection. We are focused on studying the general purpose MMSB with minimal assumptions and demonstrating its applicability to a real-world problem of immense consequence. The connection of MMSB with protein complex detection was

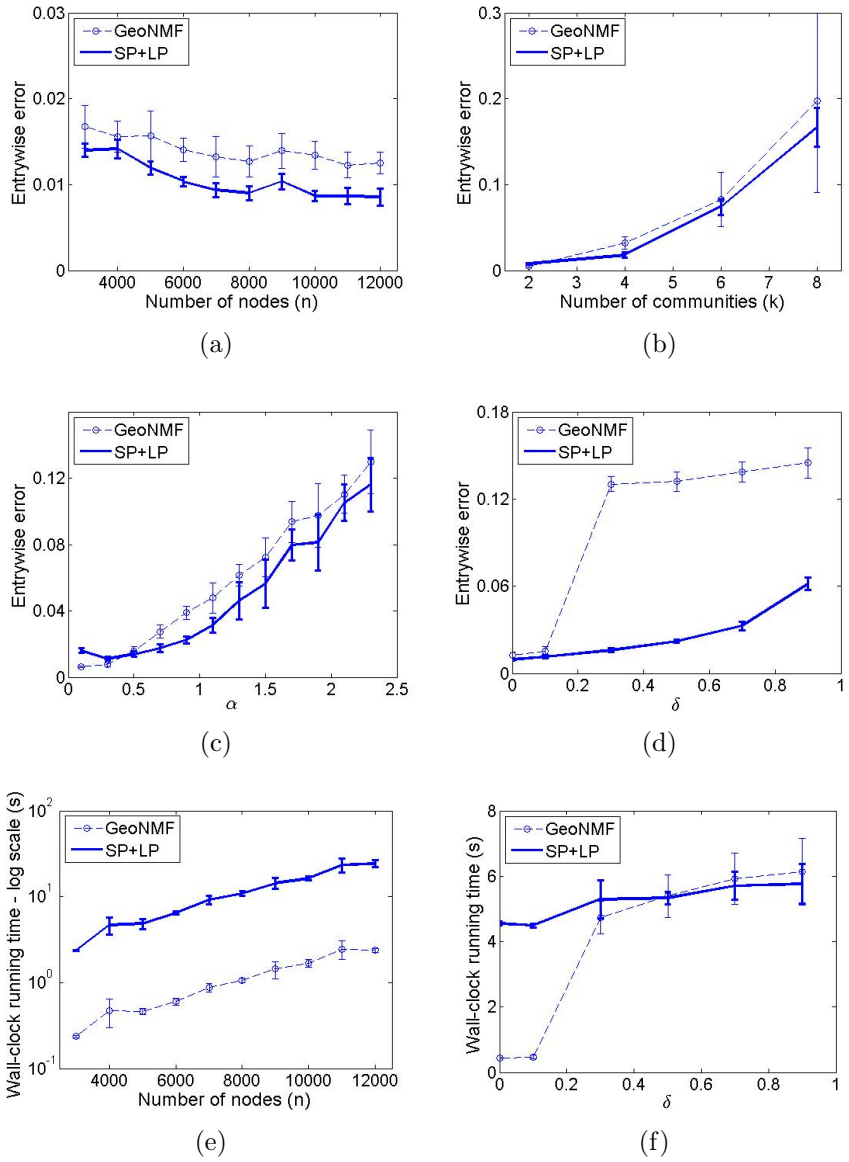

Figure 4.1: Performance of `SP+LP` on synthetic MMSB weighted graphs compared with GeoNMF.

also made in [3]; however, their theoretical and experimental results are quite preliminary compared to ours.

Datasets: We consider PPI datasets provided by [45] and [25], which are very popular among the biological community for the protein complex detection problem. The former contains two weighted graph datasets, which are referred to as Krogan core ($n = 2708$) and Krogan extended ($n = 3672$). The weighted graph dataset in the latter is referred to as Collins ($n = 1622$). The ground truth validation sets used are two standard repositories of protein complexes, which also appear to be the benchmarks in the biological community. These repositories are Munich Information Centre for Protein Sequence (MIPS) and *Saccharomyces* Genome Database (SGD). These repositories are manually curated and therefore are independent of the PPI datasets. We emphasize that protein complex detection is an ongoing research effort and that these repositories may not necessarily be considered complete as yet. This implies that `SP+LP` may find candidate complexes that are not known thus far but nonetheless do exist, thereby acting as a tool for biologists to make educated guesses.

Evaluation Metrics: The success of a protein complex detection algorithm is typically measured via a composite score which is the sum of three quantities: maximum matching ratio (MMR), fraction of detected complexes (frac), and geometric accuracy (GA). Intuitively, MMR captures how well the complexes in the validation set are predicted by computing a maximum matching in a bipartite graph in which the two vertex sets represent predicted and true complexes, and the weight of an edge denotes a similarity score between the predicted complex and the true complex on its endpoints, frac captures the fraction of true complexes for which a sufficiently good predicted complex exists, and GA is the geometric mean of clustering-wise sensitivity and positive predictive value. The reader may refer to [58] for an excellent in-depth discussion about these quantities. A higher score corresponds to better performance and the highest possible scores for MMR and frac are one each.

For the parameter k, we try different plausible values. The validation sets have binary memberships for the protein complexes, i.e. each protein is either present in a complex or it is not. The memberships determined via `SP+LP`, on the other hand, are fractional. However, the former can be easily binarized by rounding all entries that are at least 0.5 to 1 and rounding the remaining entries to 0. Additionally, we have performed another post-processing step after binarizing the result of `SP+LP` which appears quite commonly in the domain literature. Any pair of complexes that overlap significantly (as determined by a user-defined threshold) are merged. Tables 4.1 and 4.2 show the performance of `SP+LP` for protein complex detection, and we compare our results with one of the most popular problem-specific heuristics called ClusterONE. We highlight that, unlike MMR

Table 4.1: Comparision of SP+LP with ClusterONE on Krogan core, Krogan extended, and Gavin datasets using SGD repository as validation set.

Validation set	Metric	Krogan core		Krogan extended		Collins	
		SP+LP	ClusterONE	SP+LP	ClusterONE	SP+LP	ClusterONE
SGD	MMR	0.389	0.418	0.428	0.364	0.372	0.532
	frac	0.598	0.667	0.632	0.594	0.557	0.828
	GA	0.525	0.663	0.542	0.628	0.504	0.731
	Score	1.512	1.748	1.602	1.586	1.433	2.091

Table 4.2: Comparision of SP+LP with ClusterONE on Krogan core, Krogan extended, and Gavin datasets using MIPS repository as validation set.

Validation set	Metric	Krogan core		Krogan extended		Collins	
		SP+LP	ClusterONE	SP+LP	ClusterONE	SP+LP	ClusterONE
MIPS	MMR	0.285	0.317	0.319	0.282	0.275	0.418
	frac	0.537	0.669	0.576	0.573	0.547	0.782
	GA	0.331	0.438	0.336	0.422	0.397	0.555
	Score	1.153	1.424	1.231	1.277	1.219	1.755

and frac, GA penalizes predictions which contain complexes in addition to the true complexes. Therefore such a metric is not suitable for any algorithm which might predict new potentially valid complexes that do not yet exist in the validation sets.

4.7 Conclusions

In this chapter, we show how to detect potentially overlapping communities in a setup that is more plausible in real-world applications, i.e. in weighted graphs without assuming the presence of pure nodes. Our method uses linear programming, which is a relatively principled approach since the literature on the theory of convex optimization is quite rich. We show that our method performs excellently on synthetic datasets. Additionally, we also show that our method succeeds in solving an important problem in computational biology without any major domain-specific modifications to the algorithm.

Chapter 5

Robust Correlation Clustering with Asymmetric Noise

5.1 Problem Introduction

Suppose we have n objects and for any two objects i, j, a similarity score $p_{ij} \in [0, 1]$, and we wish to determine a clustering of the objects such that objects in the same cluster are similar and objects in different clusters are dissimilar. As discussed in Chapter 3, Correlation Clustering, first introduced by [12], formulates this problem as an optimization problem which does not require a priori knowledge about the number of clusters in the graph. The idea in Correlation Clustering is to first form a weighted graph on n nodes where the weight of edge ij is obtained from the similarity score using the transformation $\log(p_{ij}/(1-p_{ij}))$, and then to find a partition of the nodes which maximizes agreements, i.e. the sum of positive weights whose endpoints are put in the same cluster and the absolute values of negative weights whose endpoints are put in different clusters, (or, equivalently, minimizes disagreements, i.e. the sum of positive weights whose endpoints are put in different clusters and the absolute values of negative weights whose endpoints are put in the same cluster). In general, the aforementioned optimization problem is NP-hard. Interestingly, the objective functions for disagreement minimization and agreement maximization differ by a constant, and as a result, an approximation algorithm provides different approximation ratio guarantees for the two problems; however, for the work presented in this chapter, the two problems are equivalent. While there has been considerable interest in designing approximation algorithms, for example [12, 27, 50, 54, 75], there have been very few works focusing on average case analysis or recovery of a ground truth clustering, as discussed

in the following literature review. This style of analysis has recently gained popularity in tackling hard machine learning problems such as low-rank matrix completion [20, 69], dictionary learning [7, 74], and overlapping community detection [5, 49, 51], to name a few. In this chapter, we introduce a new graph generative model based on generating feature vectors/embeddings for the nodes in the graph, and propose a tuning-parameter-free semidefinite-programming (SDP)-based algorithm to recover nodes with sufficiently strong cluster membership.

Among existing provable methods, [40] propose a fully-random model which generates signed graphs, and show that the model ground truth clustering is close to the optimal solution of the combinatorial optimization problem of maximizing agreements. However, it is not shown how to provably recover the model ground truth efficiently.

In [54], the authors propose fully- and semi-random models, i.e. in which there is a probabilistic component and a deterministic adversarial component, which generate signed graphs. Their fully-random model can be interpreted as a special case of the planted partition model in which the noise probabilities $1-p$ and q are equal, and the lack of an edge is treated as an edge with weight -1. For graph instances generated by the fully-random model, they propose a recovery algorithm which uses a modification of the SDP formulation proposed in [75] followed by a novel randomized rounding procedure. However, their proposed SDP formulation suffers from the limitation that it has $\Theta(n^3)$ (where n is the number of nodes in the graph) constraints corresponding to triangle inequalities, making it almost unusable in practice for graphs with as low as 5000 nodes.

In [22], the authors consider the planted partition model with the added difficulty that some entries of the adjacency matrix are not known. The input graphs are signed as the lack of an edge is treated as an edge with weight -1, and their algorithm uses a matrix-splitting SDP, originally introduced to express a given matrix as the sum of a sparse and a low-rank matrix. They provide conditions under which the SDP solution is integral and therefore their algorithm requires no rounding. They also argue that the recovered model ground truth clustering coincides with the optimal solution of the combinatorial optimization problem of minimizing disagreements.

In [50], the authors propose a semi-random model which generates signed graphs. Their algorithm also uses the SDP proposed in [75] followed by a novel rounding procedure. Their recovery result states that if the input graph is generated from their semi-random model and additionally also satisfies some deterministic structural properties, then with constant probability at most a fraction of the nodes are mis-clustered.

5.2 Problem Formulation

5.2.1 Node Features Model (NFM)

We begin by defining the generative model, called the *Node Features Model (NFM)*, for which we formulate the Correlation Clustering recovery problem.

Definition 5.1 (Node Features Model (NFM))**.** *Let n and k be positive integers denoting the number of nodes and the number of clusters respectively. Let the nodes and the clusters be labelled using the sets $[n]$ and $[k]$ respectively. For each node $i \in [n]$, draw independently a feature vector $\boldsymbol{\theta}^i \in \mathbb{R}^k$ from a probability distribution on the unit simplex. Generate a weighted random graph G on the n nodes with weight matrix W defined as*

$$W_{ii'} = \begin{cases} \log\left(\dfrac{\boldsymbol{\theta}^{iT}\boldsymbol{\theta}^{i'}}{1-\boldsymbol{\theta}^{iT}\boldsymbol{\theta}^{i'}}\right) & \text{if } i \neq i' \\ 0 & \text{otherwise.} \end{cases}$$

For each $j \in [k]$, define cluster V_j as

$$V_j := \{i \in [n] : \theta^i_j > 0.5\}$$

and define the set of stray nodes V_{stray} as

$$V_{stray} := \{i \in [n] : \max(\boldsymbol{\theta}^i) \leq 0.5\}.$$

The intuition behind NFM is that first we generate a feature vector (or embedding) for each node in the graph, then for any pair $i, i' \in [n]$ of distinct nodes, we interpret $\boldsymbol{\theta}^{iT}\boldsymbol{\theta}^{i'}$ as a similarity score, i.e. the probability with which the two nodes belong to the same cluster, lastly we apply a logarithmic transformation on the similarity score which produces a positive weight if the score is greater than 0.5 and a negative weight if the score is less than 0.5. The transformation $h(x) = \log(x/(1-x))$ which maps the set $(0, 1)$ to arbitrary real values is called the *logit* or *log-odds function* in literature, and its inverse $h^{-1}(x) = 1/(1+e^{-x})$ is the so-called *logistic function*. These functions are commonly used in regression problems in which the output variable is interpreted as a probability and is therefore expected to belong to the set $(0, 1)$. For instance, a multivariate, vector-valued generalization of the logistic function, called the *softmax function*, is widely used in classification problems to transform arbitrary real-valued vectors into probabilities corresponding to class memberships.

We begin by asking the following question for the NFM described by (5.1):

Given W, how can we efficiently recover the sets $V_1, \ldots, V_k$ using no prior knowledge of k?

Using a combination of theoretical analyses and computational experiments, we make progress towards answering the question posed above by proposing two SDP-based recovery algorithms, called `1-diag` and ℓ_2`-norm-diag`. The first recovery algorithm, `1-diag`, is studied in Section 5.3 and is based on the SDP formulation of Swamy [75] whose variants have also been used in [50, 54]. Then we demonstrate a limitation of the aforementioned algorithm to handle certain noisy instances. Consequently, we propose and analyze the novel ℓ_2`-norm-diag` recovery algorithm in Section 5.4. Our theoretical analysis is not comprehensive and the deficiencies are taken care of using evidence from computational experiments. Before proceeding to the material on the two recovery algorithms, in the subsequent sections, we discuss structural properties of the NFM relevant to the recovery problem we are interested in solving.

5.2.2 Nature of Noise in the NFM

We discuss the nature of noise in our model. Define the *cluster set*

$$C_j := \{\mathbf{x} \in \Delta^{k-1} : x_j > 0.5\}$$

for each $j \in [k]$, and the *central set*

$$C := \{\mathbf{x} \in \Delta^{k-1} : \max(\mathbf{x}) \leq 0.5\}.$$

Figure 5.1 shows these sets for $k = 3$.

Note that in the light of the above definitions, we may equivalently redefine the sets V_j, for each $j \in [k]$, and V_{stray} in Definition 5.1 as

$$V_j := \{i \in [n] : \boldsymbol{\theta}^i \in C_j\}$$
$$V_{stray} := \{i \in [n] : \boldsymbol{\theta}^i \in C\}.$$

Observe that for any $\mathbf{x} \in C$ and $\mathbf{y} \in \Delta^{k-1}$, $\mathbf{x}^T\mathbf{y} \leq 0.5$. This suggests that in the weighted graphs generated by the NFM, the stray nodes form negative edges with all other nodes in the graph, hence justifying their name. Due to this property, such nodes are quite benign with regards to mathematical analysis as any reasonable clustering algorithm, including the ones proposed in this chapter, ought to be able to detect them exactly. For any

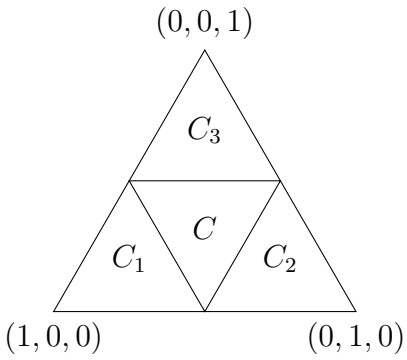

Figure 5.1: Central set C and cluster sets C_1, C_2, C_3 for the unit simplex in $\mathbb{R}^3$.

$\mathbf{x} \in C_j$, $\mathbf{y} \in C_{j'}$, for some distinct $j, j' \in [k]$, we have that $\mathbf{x}^T\mathbf{y} < 0.5$. This suggests that in the weighted graphs generated by the NFM, the clusters are *well-separated* in the sense that each pair of nodes lying in distinct clusters shares a negative weight edge. However, if both $\mathbf{x}, \mathbf{y} \in C_j$, for some $j \in [k]$, then $\mathbf{x}^T\mathbf{y}$ may or may not be larger than 0.5 and this is what introduces noise in our model. In other words, in the graphs generated by the NFM, it is possible for two nodes lying in the same cluster to share a negative weight edge. Therefore NFM models only one-sided noise. This behavior is well-motivated as real-world graphs do not always have a symmetric two-sided noise. For instance, consider a social network of researchers from the academic communities of mathematics, physics, history, and biology. Suppose the edge weights represent pair-wise similarities between any two researchers determined using the number of co-authored research articles. In this setting, we might have occasional collaborations amongst researchers of different communities; however, we almost certainly cannot expect all researchers in the same community to have collaborated with each other. In the language of weighted graphs, if the different academic communities represent the clusters in the graph, then we should expect a significantly high number of within-cluster negative edges compared to between-cluster positive edges. Due to such practical motivation, Correlation Clustering with asymmetric noise has also been studied in [38, 39].

5.2.3 Feature Space for a Cluster in the NFM

As briefly mentioned in Section 5.2.2, it is possible for two nodes belonging in the same cluster to share a negative edge. It is instructive to understand further the nature of such negative edges. For each $j \in [k]$, define a partition of the set C_j into *strong* and *fringe sets*

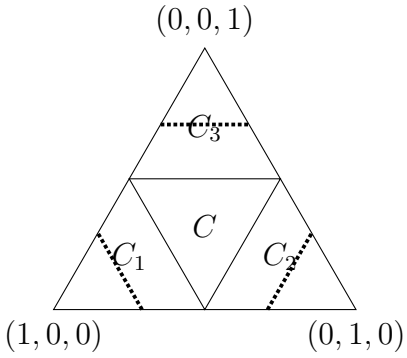

Figure 5.2: Central set C and the partition of corner sets C_1, C_2, C_3 into strong and fringe sets, shown using dotted lines, for the unit simplex in $\mathbb{R}^3$; for each corner set, the partition set containing a simplex vertex denotes the strong set.

as

$$C_j^{strong} := \{\mathbf{x} \in \Delta^{k-1} : x_j \geq 1/\sqrt{2}\}$$
$$C_j^{fringe} := \{\mathbf{x} \in \Delta^{k-1} : 0.5 \leq x_j < 1/\sqrt{2}\}.$$

Figure 5.2 shows these sets for $k = 3$.

Consequently, for each $j \in [k]$, we partition the cluster nodes V_j into *strong* and *fringe nodes* as

$$V_j^{strong} := \{i \in [n] : \boldsymbol{\theta}^i \in C_j^{strong}\}$$
$$V_j^{fringe} := \{i \in [n] : \boldsymbol{\theta}^i \in C_j^{fringe}\}.$$

These definitions are motivated by the intuition that the magnitude of the largest entry in the feature vector of a node quantifies the strength of cluster membership for that node. Moreover, the cut-off of $1/\sqrt{2}$ is chosen by noticing that any two points in the strong set of the same cluster have an inner product of at least 0.5. In other words, any two nodes which are strong for the same cluster share a non-negative edge. Therefore if the graph contains only strong nodes for each cluster, then it has no noise in the form of a negative within-cluster edge. Fringe nodes, however, may potentially share some negative edges among themselves and with other nodes in the same cluster because the memberships of such nodes in their respective clusters are not sufficiently strong. Therefore we may think that it is difficult to cluster all the fringe nodes correctly.

5.2.4 Relation to the MMSB

The problem setup developed using the NFM bears some resemblance with the weighted version of MMSB considered in Chapter 4. In particular, the graphs obtained by the NFM can be obtained by setting the community interaction matrix B to be the identity matrix in the weighted MMSB. In terms of recovery, because we are modeling Correlation Clustering using the NFM, our goal is to recover only the cluster labels without using an a priori estimate of the number of clusters k. The weighted MMSB models the overlapping community detection problem in which the goal was to recover the fractional memberships of each node in the different communities and we were allowed to use a parameter corresponding to k in the recovery algorithm.

5.3 `1-diag` Recovery Algorithm

We first present and analyze the `1-diag` recovery algorithm which uses the SDP relaxation (P-1D) first introduced in [75] to perform Correlation Clustering. For any node set V, we define the *cluster matrix* for some partition of V to be a $|V| \times |V|$, 0/1 matrix whose entry ii' is 1 if and only if nodes i and i' belong to the same partition set.

Algorithm 3 `1-diag`

Input: Graph $G = (V, W)$ generated according to NFM
Output: Symmetric matrix X^c of the same dimension as W whose each entry is in $\{0, 1\}$

1: $X^* = \arg\max \langle W, X \rangle$ s.t. $X \geq 0, X \succeq 0, X_{ii} = 1 \; \forall i \in [n]$
2: $X^c = \text{Round}(X^*, 0.5)$
3: **if** X^c is not the cluster matrix for some partition of V **then**
4: $\quad X^c = 0$
5: **end if**

Note that the output of `1-diag` can possibly be the zero matrix and therefore does not define a clustering for the input graph. However, the theory developed in Sections 5.3.1 and 5.3.2 provides conditions on the input graph sufficient for the output of `1-diag` to induce a clustering. We emphasize that since Correlation Clustering formulations do not require a prior estimate of the number of clusters in the graph, `1-diag` is designed to be free of a tuning parameter dependent on k.

Algorithm 4 Round

Input: Matrix X, scalar t

Output: 0/1 matrix X^r of the same dimension as X

1: **for** $i, j \in [n]$ **do**

2: $$X^r_{ij} = \begin{cases} 1 & \text{if } X_{ij} > t \\ 0 & \text{if } X_{ij} \leq t \end{cases}$$

3: **end for**

$$\begin{aligned} \max_X \quad & \langle W, X \rangle \\ \text{s.t.} \quad & X \geq 0 \\ & X \succeq 0 \\ & X_{ii} = 1 \ \forall i \in [n]. \end{aligned} \tag{P-1D}$$

5.3.1 Warmup

We begin by analyzing the scenarios in which the the SDP (P-1D) has a 0/1 solution. The following theorem provides a deterministic sufficient condition on the graph instances for (P-1D) to have a 0/1 solution. Subsequently, we discuss the deterministic sufficient condition in the context of the NFM.

Theorem 5.3.1. *Let $G = (V, W)$ be a graph generated using the NFM. Suppose that for each $j \in [k]$, $L(G[V_j]) \succeq 0$. Let X^* denote the cluster matrix corresponding to the partition $\{V_1, \ldots, V_k, \{v\}_{v \in V_{stray}}\}$. Then X^* is an optimal solution of (P-1D).*

Recall that $G[V_j]$ denotes the subgraph of G induced by the node set V_j and $L(G[V_j])$ denotes the Laplacian matrix of graph $G[V_j]$. The above theorem states that exact recovery of true clusters is achievable using (P-1D) provided each cluster Laplacian is positive semidefinite. To connect this result with the NFM, we may quantify the probability such that each cluster Laplacian, in a graph instance generated by the NFM, is positive semidefinite. Note that if all edges in a cluster are non-negative, then its Laplacian is necessarily positive semidefinite. However, since the NFM introduces noise in the form of negative within-cluster edges, the cluster Laplacians may not necessarily be positive semidefinite. Table 5.1 shows some computational experiments in this regard. Each row in the table corresponds to 10 cluster instances generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution. We fix $k = 3$ and the Dirichlet parameter

Table 5.1: Verification of positive semidefiniteness of cluster Laplacians.

Cluster size range	PSD success (/10)	Mean smallest Laplacian eigenvalue
$6-10$	4	-1.31
$11-15$	4	-1.31
$16-20$	0	-3.60
$21-25$	1	-2.82
$26-30$	1	-4.26
$31-35$	0	-5.96
$36-40$	0	-6.35

$\boldsymbol{\alpha} = 0.3\mathbf{e}$. The first column denotes the range in which the cluster size belongs, and the second column counts the *PSD success*, i.e. number of cluster instances, out of 10, which have a positive semidefinite Laplacian. Moreover, the third column contains the mean smallest eigenvalue of the Laplacian.

These computational results suggests a weakness of Theorem 5.3.1 in the sense that the determinstic condition required for exact recovery seems to hold with a probability converging to 0 for the NFM with the Dirichlet distribution as the size of input graph grows. Morever, the decreasing smallest eigenvalue of the cluster Laplacians may also be interpreted as an increasing amount of noise in the clusters which motivates the following conjecture.

Conjecture 5.3.2. *Let G be a graph generated according to NFM in which the simplex distribution is chosen to be the Dirichlet distribution with constant parameter. No algorithm can exactly recover the true clusters in G with probability not converging to 0 as $n \to \infty$.*

Conjecture 5.3.2 highlights information-theoretic limitations for exactly recovering the ground truth clusters, see [10, 11, 16, 18, 81] for instance, for results on information-theoretic limits for similar or related problems. The above observations also lead us to reformulate the central question posed in Section 5.2.1 as follows.

Given W, how can we efficiently recover exactly k disjoint node sets, such that each node set contains exactly one of $V_1^{strong}, \ldots, V_k^{strong}$, using no prior knowledge of k?

We may interpret this reformulation as: instead of attempting to exactly recover the true clusters, we focus on exactly recovering the strong nodes, possibly in the presence

of fringe nodes, which introduce noise in the form of negative within-cluster edges. The usage of the word "contains" in the above question indicates that recovery of any fringe node for a cluster is not necessarily intended but may happen. This perspective on robust Correlation Clustering, which involves clustering essentially only a subgraph of the input graph, is similar to that in [44], which provides an approximation algorithm for a generalized Correlation Clustering problem wherein the input graph is corrupted with a given number of noisy nodes which must be discarded before performing clustering.

To answer the reformulated question above, we adopt a two-step algorithm analysis approach described as follows. Let G be a graph generated by the NFM and let $\mathcal{A}$ be a cluster recovery algorithm of interest. For each $j \in [k]$, let V_j' be the union of strong nodes and possibly some fringe nodes for cluster j, such that we expect $\mathcal{A}$ to successfully recover node sets $V_1', \ldots, V_k'$ with a non-zero probability as the number of nodes $n \to \infty$. In other words, $\mathcal{A}$ is likely to fail on the sets $V_j \setminus V_j'$ for each $j \in [k]$. We may formalize the behavior of $\mathcal{A}$ using the following two steps.

1. For each $j \in [k]$, perturb the features of the node set $V_j \setminus V_j'$ to the central set to obtain the stray node set V_j^{stray}, and call the resulting graph G'. Prescribe deterministic conditions $\mathcal{C}'$ on node sets V_j', for each $j \in [k]$, which ensure their exact recoverability from G' by $\mathcal{A}$.

2. For each $j \in [k]$, re-perturb the features of the node set V_j^{stray} so as to obtain the node set $V_j \setminus V_j'$, which we may interpret as noisy nodes, i.e. we re-obtain graph G from G'. Prescribe deterministic conditions $\mathcal{C}$ under which $\mathcal{A}$ is robust to the presence of node sets $V_j \setminus V_j'$, for each $j \in [k]$. The desired robustness properties are established by applying perturbation arguments to the analysis of $\mathcal{A}$ on G' achieved in the previous step.

In terms of probability quantification, we must also argue that for a graph G generated by the NFM, the deterministic conditions required for provably robust recovery hold with probability not converging to 0 as $n \to \infty$.

5.3.2 Theoretical Guarantees

Using Theorem 5.3.1, we conclude that if each cluster Laplacian is positive semidefinite, then `1-diag` achieves exact recovery. Adopting the two-step approach outlined in the previous section, we are now interested in the following two questions:

- What is the probability that, for each cluster, the subgraph induced by the union of strong nodes and possibly some fringe nodes has a positive semidefinite Laplacian?
- Is the `1-diag` recovery algorithm robust to the presence of noisy nodes, i.e. fringe nodes that are close to being stray nodes?

In this section, we address the first question above, and in Section 5.3.4, we address the second question. Observe that if we restrict our attention to the cluster subgraph induced by merely the strong nodes, then with probability 1, the Laplacian is positive semidefinite because each edge has a non-negative weight. However, we are interested in extending this observation to a cluster subgraph induced by strong nodes and some fringe nodes which also possibly contains negative edges. (Based on the results in Table 5.1, we cannot expect to include all fringe nodes.) For the NFM, directly quantifying the probability of Laplacian positive semidefiniteness for a cluster subgraph comprised of strong nodes and some fringe nodes appears a difficult task. Therefore in the following, Theorems 5.3.3 and 5.3.4 provide combinatorial sufficient conditions for a graph Laplacian to be positive semidefinite.

Theorem 5.3.3. *Let $G = (V, W)$ be a signed graph. Suppose for each negative edge ii' where $i, i' \in [n]$, there exists a set of m disjoint two-edge ii'-paths $\{P_l^{ii'}\}_{l\in[m]}$ of positive weights such that*

$$-W_{ii'} \leq \sum_{l\in[m]} \frac{1}{2} \times \textit{harmonic mean of the two weights on } P_l^{ii'}$$

and the two-edge paths are disjoint across all negative edges, then $L(G) \succeq 0$.

The intuition behind the proof of Theorem 5.3.3 is to express the graph Laplacian as the sum of multiple graph Laplacians (corresponding to subgraphs of G), and then argue for the positive semidefiniteness of each summand Laplacian. Considering subgraphs in this way makes it easier to analyze negative edges; in particular a negative edge ii' is included in a subgraph which also contains an adequate number of positive ii'-paths so as to compensate the contribution of the edge ii' to the Laplacian. This idea is inspired by the support-graph technique used to design preconditioners for conjugate gradient [15].

Theorem 5.3.3 provides a sufficient condition to ensure Laplacian positive semidefiniteness, however, it is seemingly weak as described by the following example.

Example 5.1. *Generate a graph on n nodes using the NFM with the probability distribution over the unit simplex fixed as the Dirichlet distribution. Suppose cluster j of the graph contains f_j fringe nodes. Consider a case in which a constant fraction of all pairs*

of the f_j fringe nodes share a negative edge each. Then to use the sufficient condition in Theorem 5.3.3 to ensure positive semidefiniteness of the Laplacian of cluster j, we require $\Omega(f_j^2)$ strong nodes for that cluster. In other words, if the cluster contains n_j nodes, then Theorem 5.3.3 allows for only $\mathcal{O}(\sqrt{n_j})$ fringe nodes. However letting p be the probability of a feature vector lying in the fringe set for cluster j, we note that $\mathbb{E}[f_j] = np$. Moreover, using Hoeffding's inequality, we have that $f_j \in [np/2, 3np/2]$ with probability at least $1 - 2\exp(-np^2/2)$. That is, $f_j = \Theta(n)$, and consequently $f_j = \Omega(n_j)$, with probability converging to 1 as $n \to \infty$. This suggests a potential weakness of the sufficient condition presented in Theorem 5.3.3 for establishing positive semidefiniteness of cluster Laplacians.

The above shortcoming is addressed in the following theorem which provides a different combinatorial condition to ensure Laplacian positive semidefiniteness.

Theorem 5.3.4. *Let $G = (V, W)$ be a signed graph. Let $U \subseteq V$ contain all nodes of G adjacent to a negative edge. That is, $U := \{v \in V : W_{vw} < 0 \text{ for some } w \in V\}$. If there exists $S \subseteq V \setminus U$ such that for each $u \in U$ and $s \in S$, we have*

$$|S|W_{us} \geq -2\left(\sum_{\substack{u' \in U: \\ W_{uu'} < 0}} W_{uu'}\right) \tag{5.1}$$

then $L(G) \succeq 0$.

We revisit Example 5.1 in the light of Theorem 5.3.4. If we assume that all edges in cluster j other than the ones among the f_j fringe nodes have a non-negative weight, and that the positive and negative weight magnitudes are of the same order, then to ensure positive semidefiniteness of the Laplacian of cluster j using the sufficient condition obtained in Theorem 5.3.4, it suffices to have $f_j = \Theta(n_j)$. However, this example should not be interpreted to imply that Theorem 5.3.4 is a strengthening of Theorem 5.3.3. For example, if we have a cluster in which each node is adjacent to a negative edge, Theorem 5.3.3 may still be used to ensure positive semidefiniteness of the cluster Laplacian, but Theorem 5.3.4 does not apply due to the absence of a set S. But for the purpose of analyzing a generative model such as the NFM, Theorem 5.3.4 appears to be a better tool because of its tolerance to a number of fringe nodes that is linear in the size of the cluster, and because of the presence of strong nodes in the NFM. This is further corroborated by computational results shown in Table 5.2. Each row in the table corresponds to 10 cluster instances generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution. We fix $k = 3$ and the Dirichlet parameter $\boldsymbol{\alpha} = 0.3\mathbf{e}$. The first

Table 5.2: Verification of sufficient condition (5.1) for Laplacian positive semidefiniteness.

Cluster size range	Combinatorial condition success (/10)
$6-10$	9
$11-15$	9
$16-20$	7
$21-25$	9
$26-30$	9
$31-35$	8
$36-40$	9

column denotes the range corresponding to the size of the subgraph induced by strong nodes and fringe nodes whose feature vectors have largest entry at least 0.6; the cut-off of 0.6 is based on manual parameter search for the given setting of k and $\boldsymbol{\alpha}$. The second column counts the *combinatorial condition success*, i.e. number of instances, out of 10, for which the subgraph satisfies the combinatorial condition (5.1) in Theorem 5.3.4.

These computational results suggest that the probability with which the cluster subgraphs consisting of nodes whose feature vectors have largest entry at least 0.6 have a positive semidefinite Laplacian does not apparently converge to 0 as $n \to \infty$, and also motivate the following conjecture.

Conjecture 5.3.5. *Let $G = (V, W)$ be a graph generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution with constant parameter $\boldsymbol{\alpha}$. Then there exists a scalar $t(k, \boldsymbol{\alpha}) \in (0.5, 1/\sqrt{2})$ such that for each $j \in [k]$, with probability not converging to 0 as $n \to \infty$, $G[V_j']$ satisfies the hypothesis of Theorem 5.3.4 where*

$$V_j' := \{i \in [n] : \theta_j^i \geq t(k, \boldsymbol{\alpha})\}.$$

5.3.3 Proofs

In this section, we include proofs of Theorems 5.3.1, 5.3.3, and 5.3.4 stated in Section 5.3.2.

Proof of Theorem 5.3.1. Our analysis uses SDP duality and therefore note that the dual

of (P-1D) is

$$\begin{aligned} \min_{(Y,Z,\mathbf{y})} \quad & \mathbf{e}^T\mathbf{y} \\ \text{s.t.} \quad & Y \geq 0 \\ & Z \succeq 0 \\ & W + Y + Z = Diag(\mathbf{y}). \end{aligned} \tag{D-1D}$$

As mentioned in the theorem statement, X^* is the cluster matrix corresponding to the partition $\{V_1, \ldots, V_k, \{v\}_{v \in V_{stray}}\}$.

Both optimization problems (P-1D) and (D-1D) have strictly feasible solutions. For instance, $X' := 0.5I + 0.5E$ is a positive, positive definite matrix which is feasible for (P-1D). Similarly, $Y' := E$, $Z' := (\|W + E\| + \epsilon)I - (W + E)$ and $\mathbf{y}' := (\|W + E\| + \epsilon)\mathbf{e}$ gives a strictly feasible solution $(Y', Z', \mathbf{y}')$ for (D-1D) for any $\epsilon > 0$. Therefore using the Karush-Kuhn-Tucker (KKT) optimality conditions, we observe that $X^* \in \mathbb{S}^n$ is an optimal solution for (P-1D) if and only if X^* is feasible for (P-1D) and there exists a feasible solution of (D-1D), $(Y^*, Z^*, \mathbf{y}^*)$ such that:

- $X^*_{ij} Y^*_{ij} = 0, \forall i, j \in [n]$
- $\langle X^*, Z^* \rangle = 0.$

X^* has non-negative entries with each diagonal entry being equal to one. Additionally, up to a permutation of its rows and columns, it is a block diagonal matrix in which each non-zero diagonal block is the matrix of all ones. Therefore X^* is feasible for (P-1D), and in the rest of the proof, we explicitly construct $(Y^*, Z^*, \mathbf{y}^*)$.

For each $j \in [k]$, we set

$$\begin{aligned} Y^*(V_j, V_j) &= 0 \\ Z^*(V_j, V_j) &= L(G[V_j]) \\ \mathbf{y}^*(V_j) &= W(V_j, V_j)\mathbf{e}. \end{aligned}$$

For each distinct $j, j' \in [k]$, we set

$$\begin{aligned} Y^*(V_j, V_{j'}) &= -W(V_j, V_{j'}) \\ Z^*(V_j, V_{j'}) &= 0. \end{aligned}$$

For each stray node $v \in V_{stray}$, we set

$$\begin{aligned} Y^*(v,:) &= -W(v,:) && \text{(and } Y^*(:,v) = -W(:,v)\text{)} \\ Z^*(v,:) &= 0 && \text{(and } Z^*(:,v) = 0\text{)} \\ y^*(v) &= 0. \end{aligned}$$

Because each pair of nodes lying in distinct clusters shares a negative edge and because each stray node shares a negative edge with every other node in the graph, we have that $Y^* \geq 0$. Similarly, because $L(G[V_j]) \succeq 0$ for each $j \in [k]$, we have that $Z^* \succeq 0$.

Matrices X^* and Y^* have disjoint supports by construction, and therefore $X^*_{ij} Y^*_{ij} = 0$ for each $i, j \in [n]$. Moreover

$$\begin{aligned} \langle X^*, Z^* \rangle &= \sum_{j \in [k]} \langle X^*(V_j, V_j), Z^*(V_j, V_j) \rangle \\ &= \sum_{j \in [k]} \langle L(G[V_j]), E \rangle \\ &= 0 \end{aligned}$$

where the last line uses the fact that each row of a Laplacian matrix sums to zero.

Lastly, we show that the equation $W + Y^* + Z^* = Diag(\mathbf{y}^*)$ is satisfied. For each $j \in [k]$, we have

$$\begin{aligned} W(V_j, V_j) + Y^*(V_j, V_j) + Z^*(V_j, V_j) &= W(V_j, V_j) + L(G[V_j]) \\ &\qquad \text{(using the definitions of } Y^*, Z^*\text{)} \\ &= Diag(\mathbf{y}^*(V_j)). \\ &\qquad \text{(using the definition of } \mathbf{y}^*\text{)} \end{aligned}$$

For each distinct $j, j' \in [k]$, we have

$$W(V_j, V_{j'}) + Y^*(V_j, V_{j'}) + Z^*(V_j, V_{j'}) = 0$$

using the definitions of Y^*, Z^*. Similarly, for each stray node v, we have

$$\begin{aligned} W(v,:) + Y^*(v,:) + Z^*(v,:) &= 0 \\ W(:,v) + Y^*(:,v) + Z^*(:,v) &= 0 \end{aligned}$$

using the definitions of Y^*, Z^*. □

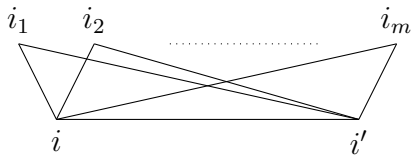

Figure 5.3: Subgraph of G containing negative edge ii' and m dijsoint two-edge ii'-paths of positive weights.

Now we provide proofs of Theorems 5.3.3 and 5.3.4 which provide combinatorial sufficient conditions for Laplacian positive semidefiniteness.

Proof of Theorem 5.3.3. Pick any negative edge ii' in G, and let $i - i_1 - i', \ldots, i - i_m - i'$ denote m disjoint two-edge ii'-paths of positive weights. Consider the subgraph of G containing edge ii' and these m disjoint paths as shown in Figure 5.3.

The contribution of this subgraph to the Laplacian of G is the matrix, padded appropriately with zeros,

$$\begin{bmatrix} W_{ii'} + \sum_{l\in[m]} W_{ii_l} & -W_{ii'} & -W_{ii_1} & -W_{ii_2} & \ldots & -W_{ii_m} \\ -W_{ii'} & W_{ii'} + \sum_{l\in[m]} W_{i'i_l} & -W_{i'i_1} & -W_{i'i_2} & \ldots & -W_{i'i_m} \\ -W_{ii_1} & -W_{i'i_1} & W_{ii_1} + W_{i'i_1} & 0 & \ldots & 0 \\ -W_{ii_2} & -W_{i'i_2} & 0 & W_{ii_2} + W_{i'i_2} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -W_{ii_m} & -W_{i'i_m} & 0 & 0 & \ldots & W_{ii_m} + W_{i'i_m} \end{bmatrix}.$$

Now since each of $W_{ii_1} + W_{i'i_1}, \ldots, W_{ii_m} + W_{i'i_m}$ is positive, using the Schur complement condition for positive semidefiniteness, the above matrix is positive semidefinite if and only if the 2×2 matrix

$$\begin{bmatrix} W_{ii'} + \sum_{l\in[m]} W_{ii_l} & -W_{ii'} \\ -W_{ii'} & W_{ii'} + \sum_{l\in[m]} W_{i'i_l} \end{bmatrix} - \sum_{l\in[m]} \left(\frac{\begin{bmatrix} W_{ii_l} \\ W_{i'i_l} \end{bmatrix} \begin{bmatrix} W_{ii_l} & W_{i'i_l} \end{bmatrix}}{W_{ii_l} + W_{i'i_l}} \right) \tag{5.2}$$

is positive semidefinite. However, the matrix in (5.2) can be rewritten as

$$\left(W_{ii'} + \sum_{l\in[m]} \frac{W_{ii_l} W_{i'i_l}}{W_{ii_l} + W_{i'i_l}} \right) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

which is positive semidefinite if and only if

$$-W_{ii'} \leq \sum_{l \in [m]} \frac{W_{ii_l} W_{i'i_l}}{W_{ii_l} + W_{i'i_l}}$$

which proves the desired statement. □

Proof of Theorem 5.3.4. For notational ease, define $L := L(G)$. Label the nodes of G using the set $[n]$ and assume, without loss of generality, that $U = [m]$ for some $m < n$, and $S = \{m+1, \ldots, m+|S|\}$. We define matrix $C \in \mathbb{S}^n$ as follows. For each $u, u' \in U$,

$$C_{uu'} := \begin{cases} L_{uu'} & \text{if } u \neq u' \\ \sum\limits_{l \in [m] \setminus \{u\}} |L_{ul}| & \text{if } u = u' \end{cases}$$

Moreover $C(U,S) := \dfrac{-C(U,U)\mathbf{e}\mathbf{e}^T}{|S|}$ and $C(S,U) := C(U,S)^T$. Lastly, we set $C(S,S) := \dfrac{\mathbf{e}^T C(U,U)\mathbf{e}}{|S|} I$, and we set all other entries of C to be zeros. That is,

$$C = \begin{bmatrix} C(U,U) & C(U,S) & 0 \\ C(S,U) & C(S,S) & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In the rest of the proof, we argue that each of C and $L - C$ is positive semidefinite, thereby proving the positive semidefiniteness of L.

To show the positive semidefiniteness of C, it suffices to show the positive semidefiniteness of $C(U \cup S, U \cup S)$. First note that using the diagonal dominance property in $C(U,U)$, we conclude that $C(U,U)$ is positive semidefinite. Morever, since each node in U is adjacent to at least one edge with a negative weight each entry of $C(U,U)\mathbf{e}$ is positive. This implies that $\mathbf{e}^T C(U,U)\mathbf{e}$ is positive which in turn implies that $C(S,S)$ is invertible. Using the Schur complement condition for positive semidefiniteness, C is positive semidefinite if and only if $C(U,U) - \dfrac{C(U,S)C(S,U)|S|}{\mathbf{e}^T C(U,U)\mathbf{e}}$ is positive semidefinite. Substituting for $C(U,S)$

and $C(S,U)$, we get

$$\begin{aligned} C(U,U) - \frac{C(U,S)C(S,U)|S|}{\mathbf{e}^T C(U,U)\mathbf{e}} &= C(U,U) - \frac{C(U,U)\mathbf{e}\mathbf{e}^T\mathbf{e}\mathbf{e}^T C(U,U)}{|S|\mathbf{e}^T C(U,U)\mathbf{e}} \\ &= C(U,U) - \frac{C(U,U)\mathbf{e}\mathbf{e}^T C(U,U)}{\mathbf{e}^T C(U,U)\mathbf{e}} \\ &= C(U,U)^{1/2}\left(I - \frac{C(U,U)^{1/2}\mathbf{e}\mathbf{e}^T C(U,U)^{1/2}}{\mathbf{e}^T C(U,U)\mathbf{e}}\right) C(U,U)^{1/2} \end{aligned}$$

where the last line from bottom uses $\mathbf{e}^T\mathbf{e} = |S|$ and the last line uses the positive semidefiniteness of $C(U,U)$. Therefore to argue for the positive semidefiniteness of the last term in the above chain, it suffices to show that $I - \dfrac{C(U,U)^{1/2}\mathbf{e}\mathbf{e}^T C(U,U)^{1/2}}{\mathbf{e}^T C(U,U)\mathbf{e}}$ is positive semidefinite. This follows from simply noticing that $\dfrac{C(U,U)^{1/2}\mathbf{e}\mathbf{e}^T C(U,U)^{1/2}}{\mathbf{e}^T C(U,U)\mathbf{e}}$ is a rank-one matrix with eigenvalue 1. Thus C is positive semidefinite.

To show that $L - C$ is also positive semidefinite, we show that it is the Laplacian of a graph with non-negative weights. First notice that $C\mathbf{e} = 0$. Indeed, we have

$$C\mathbf{e} = \begin{bmatrix} C(U,U)\mathbf{e} + C(U,S)\mathbf{e} \\ C(S,U)\mathbf{e} + C(S,S)\mathbf{e} \\ 0 \end{bmatrix}$$

where

$$C(U,U)\mathbf{e} + C(U,S)\mathbf{e} = C(U,U)\mathbf{e} - C(U,U)\mathbf{e} = 0$$

using the construction of $C(U,S)$, and

$$C(S,U)\mathbf{e} + C(S,S)\mathbf{e} = \frac{-\mathbf{e}\mathbf{e}^T C(U,U)\mathbf{e}}{|S|} + \frac{\mathbf{e}^T C(U,U)\mathbf{e}\mathbf{e}}{|S|} = 0$$

using the constructions of $C(S,U)$ and $C(S,S)$. Subsequently, using the fact that $L\mathbf{e} = 0$, we conclude that $(L - C)\mathbf{e} = 0$.

Defining set $R := V \setminus (U \cup S)$, we now show that each off-diagonal entry of

$$L - C = \begin{bmatrix} L(U,U) - C(U,U) & L(U,S) - C(U,S) & L(U,R) \\ L(S,U) - C(S,U) & L(S,S) - C(S,S) & L(S,R) \\ L(R,U) & L(R,S) & L(R,R) \end{bmatrix}$$

is non-positive. For each $u, u' \in U$, we have $L_{uu'} - C_{uu'} = 0$ by construction. For each $u \in U, s \in S$, we have

$$\begin{aligned}
L_{us} - C_{us} &= L_{us} + \frac{1}{|S|} \sum_{u' \in U} C_{uu'} && \text{(by construction of } C(U,S)) \\
&= L_{us} + \frac{C_{uu}}{|S|} + \frac{1}{|S|} \sum_{u' \in U \setminus \{u\}} C_{uu'} \\
&= L_{us} + \frac{1}{|S|} \left(\sum_{u' \in U \setminus \{u\}} |L_{uu'}| + L_{uu'} \right) && \text{(by construction of } C(U,U)) \\
&= L_{us} + \frac{1}{|S|} \sum_{\substack{u' \in U \setminus \{u\}: \\ L_{uu'} > 0}} 2L_{uu'} \\
&= -W_{us} - \frac{1}{|S|} \sum_{\substack{u' \in U \setminus \{u\}: \\ W_{uu'} < 0}} 2W_{uu'} && (\because L = L(G)) \\
&\leq 0. && \text{(using (5.1))}
\end{aligned}$$

Now it remains to consider the signs of the entries of $L(V,R)$ and the off-diagonal entries of $L(S,S) - C(S,S)$. Observe that each entry of $L(V,R)$ is non-negative since each entry of $W(V,R)$ is non-positive. Indeed any entry in $W(V,R)$ corresponds to an edge whose one endpoint lies in R; note that for a negative edge, both endpoints lie in U by definition. Lastly, every off-diagonal entry of $L(S,S) - C(S,S)$ is non-negative since such entries are 0 in $C(S,S)$, by construction, and non-negative in $L(S,S)$ since they correspond to edges whose both endpoints lie in S.

Therefore we have shown that $L - C$ is a Laplacian matrix for a graph with non-negative weights, and is consequently positive semidefinite. Since we have shown the positive semidefiniteness of both C and $L - C$, we conclude that L is positive semidefinite. □

5.3.4 Lack of Robustness

As discussed in Section 5.3.2, we are interested in understanding the robustness of `1-diag` recovery algorithm in the presence of noisy nodes, i.e. fringe nodes that are close to being stray nodes. However, through computational experiments, it is observed that `1-diag`

seems to have undesirable behavior in this setting. In particular, there exist pathological instances in which the output of `1-diag` contains groups of noisy nodes as spurious cluster. The following example further illustrates this phenomenon.

Example 5.2. *Consider a graph G on $n = 25$ nodes containing $k = 3$ clusters. Suppose G has a cluster j containing 6 nodes and the three-dimensional features of these nodes are as shown in the rows of the 6×3 matrix below.*

$$\begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.79 & 0.00 & 0.21 \\ 1.00 & 0.00 & 0.00 \\ 0.53 & 0.47 & 0.00 \\ 0.53 & 0.47 & 0.00 \\ 0.51 & 0.49 & 0.00 \end{bmatrix}$$

The submatrix of the output of `1-diag` corresponding to the nodes in cluster j is

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

The above matrix is not the matrix of all ones. In fact, it breaks down the true cluster into two clusters by creating one cluster each for strong and fringe nodes thereby creating a spurious cluster made up of the fringe nodes.

This apparent limitation of `1-diag` demotivates a theoretical analysis of cluster recovery using fractional optimal solutions of (P-1D).

5.4 ℓ_2-norm-diag Recovery Algorithm

We propose SDP formulation (P-ND) obtained by replacing the n diagonal constraints of (P-1D) with a single ℓ_2-norm constraint. Based on this formulation, we propose a novel recovery algorithm, called ℓ_2-`norm-diag`, which involves no tuning parameter.

Note that the output of ℓ_2-`norm-diag` can possibly be the zero matrix and therefore does not define a clustering for the input graph or any of its subgraphs. However, the

Algorithm 5 ℓ_2-norm-diag

Input: Graph $G = (V, W)$ generated according to NFM

Output: Symmetric matrix X^c of the same dimension as W whose each entry is in $\{0, 1\}$

1: $X^* = \arg\max \langle W, X \rangle$ s.t. $X \geq 0, X \succeq 0, \|diag(X)\| \leq 1$
2: **for** t in entries of X^* sorted in non-increasing order **do**
3: $X^c_{ij} = \text{Round}(X^*, t)$
4: **if** there exists a non-empty $V' \subseteq V$ such that $X^c(V', V')$ is the cluster matrix for some partition of V' **then**
5: **break**
6: **else**
7: $X^c = 0$
8: **end if**
9: **end for**

theory developed in Section 5.4.1 provides conditions on the input graph sufficient for the output of ℓ_2-norm-diag to induce a clustering for some subgraph of the input graph. We emphasize that since Correlation Clustering formulations do not require a prior estimate of the number of clusters in the graph, ℓ_2-norm-diag is designed to be free of a tuning parameter dependent on k.

$$\begin{aligned} \max_{X} \quad & \langle W, X \rangle \\ \text{s.t.} \quad & X \geq 0 \\ & X \succeq 0 \\ & \|diag(X)\| \leq 1. \end{aligned} \tag{P-ND}$$

SDPs (P-1D) and (P-ND) both have the non-negativity and positive semidefiniteness constraints on the variable matrix. The combination of these constraints, i.e. the set $\{X : X \geq 0, X \succeq 0\}$, forms the so-called *doubly non-negative (DNN) cone*. This cone has been studied in the context of SDP relaxations for other graph problems such as the minimum cut problem [48] and the quadratic assignment problem [35, 60].

The rounding procedure in ℓ_2-norm-diag is based on the observation from computational experiments that the entries of an optimal solution of (P-ND) corresponding to the recovered clusters are larger compared to, and therefore well-separated from, the rest of the entries. This implies the existence of a fixed threshold for rounding; however, computational experiments also suggest the dependence of this rounding threshold on problem parameters n, k and $\boldsymbol{\alpha}$. An algorithmic dependence on k and $\boldsymbol{\alpha}$ is undesirable, especially

in Correlation Clustering, since these parameters are latent and encode information about the number and size of clusters in the graph. It is not clear to us whether a fixed-threshold-based rounding procedure exists which does not require prior estimate of k and $\boldsymbol{\alpha}$, and this motivates the rounding procedure in ℓ_2-`norm-diag` which adapts to matrix being rounded.

5.4.1 Theoretical Guarantees

Similar to our approach for the `1-diag` recovery algorithm, we first prove exact cluster recovery under deterministic conditions on the input graph, followed by understanding the validity of these deterministic conditions for the NFM, and the robustness properties of (P-ND).

Assumption 5.1. *Suppose $G = (V, W)$ be a graph on n nodes where $W = \mathbf{q}\mathbf{q}^T - D + N$. Let $U \subseteq V$ contain all nodes of G adjacent to a negative edge. That is, $U := \{v \in V : W_{vw} < 0 \text{ for some } w \in V\}$. Suppose the following hold.*

1. *$\mathbf{q}$ is a positive n-dimensional vector satisfying*

$$\max(\mathbf{q}^{\circ 4/3}) \leq \frac{\|\mathbf{q}^{\circ 2/3}\|^2}{45}.$$

   Intuitively, this condition is likely to hold if the smallest entry in $\mathbf{q}$ is not too small.

2. *There exists $S \subseteq V \setminus U$ such that for each $u \in U$ and $s \in S$, we have*

$$|S| q_s^{1/3} W_{us} \geq -6 \left(\sum_{\substack{u' \in U: \\ W_{uu'} < 0}} q_{u'}^{1/3} W_{uu'} \right).$$

3. *N is a $n \times n$ symmetric matrix whose each diagonal entry is zero and, for each $i \in [n]$, satisfies*

$$\|\mathbf{n}^i\| \leq \frac{q_i \|\mathbf{q}^{\circ 2/3}\|^2}{45 \|\mathbf{q}^{\circ 1/3}\|}.$$

   (Recall the notation that for each $i \in [n]$, $\mathbf{n}^i$ denotes row i of matrix N.)

4. *$D = Diag(\mathbf{q} \circ \mathbf{q})$. This is a* diagonal correction matrix *chosen to ensure that the diagonal of W is indeed zero, as defined. Unlike (P-1D), the analysis of (P-ND) depends on the diagonal entries of W, and therefore it is reasonable to assume each of them to be 0.*

Theorem 5.4.1. *Let $G = (V, W)$ be a graph generated using the NFM, and suppose that for each $j \in [k]$, $W(V_j, V_j)$ satisfies Assumption 5.1. Then (P-ND) has an optimal solution X^* satisfying $X^*_{ii'} > 0$ if and only if nodes i and i' belong to the same cluster.*

In terms of proof techniques, unlike the SDP (P-1D), (P-ND) does not lend itself to an explicit construction of primal-dual optimal solutions and requires a more elaborate argument using Brouwer fixed-point theory.

Adopting the two-step approach outlined in Section 5.3.1, we are now interested in the following two questions:

- What is the probability that, for each cluster, the subgraph induced by the union of strong and some fringe nodes satisfies Assumption 5.1?
- Is the ℓ_2-`norm-diag` recovery algorithm robust to the presence of noisy nodes, i.e. fringe nodes that are close to being stray?

While we do not provide a precise answer to the first question above, we demonstrate, computationally, the connection between Assumption 5.1 and the NFM in which the distribution over the unit simplex is chosen to be the Dirichlet distribution. For a graph $G = (V, W)$ generated according to NFM, for each $j \in [k]$, define V'_j to be the union of strong nodes and some fringe nodes in cluster j such that the cut-off for selecting fringe nodes depends on problem parameters k and $\boldsymbol{\alpha}$. Let n_j be the cardinality of V'_j, and let Θ_j denote the $n_j \times k$ matrix whose rows contain the feature vectors corresponding to the nodes in V'_j. Observe that the univariate function $g : (0, 1) \to \mathbb{R}$ defined as $g(x) := \log(x/(1-x))$ can be approximated using a linear function $l : (0, 1) \to \mathbb{R}$ defined as $l(x) := c \cdot (2x - 1)$ for a suitably chosen positive constant c. Then for each $j \in [k]$, we have

$$W(V'_j, V'_j) \approx c \cdot (2\Theta_j \Theta_j^T - E). \tag{5.3}$$

Now through various computational experiments, we notice that the matrix $c \cdot (2\Theta_j \Theta_j^T - E)$ is almost a rank-one matrix such that eigenvector corresponding to the largest eigenvalue is a positive vector. The following example concretely illustrates these observations.

Example 5.3. *Consider a graph generated using the NFM with $n = 30$ and $k = 3$. The distribution over the unit simplex is chosen to be the Dirichlet distribution with parameter $\boldsymbol{\alpha} = 0.3\mathbf{e}$. For some cluster j, let V'_j be the union of strong nodes and fringe nodes whose feature vectors have largest entry at least 0.6. The three-dimensional features of the nodes*

in V_j' are shown in the rows of the matrix Θ_j below.

$$\Theta_j = \begin{bmatrix} 0.05 & 0.83 & 0.11 \\ 0.04 & 0.69 & 0.27 \\ 0.03 & 0.92 & 0.05 \\ 0.02 & 0.73 & 0.25 \\ 0.11 & 0.88 & 0.01 \\ 0.25 & 0.60 & 0.15 \\ 0.00 & 0.99 & 0.01 \\ 0.12 & 0.67 & 0.21 \\ 0.01 & 0.95 & 0.04 \end{bmatrix}$$

We notice that the matrix $c \cdot (2\Theta_j\Theta_j^T - E)$ with $c = 2.2$ has exactly three non-zero eigenvalues given by 8.57, 0.25 *and* -0.75. *Moreover the unit eigenvector, $\mathbf{v}_j$, corresponding to the eigenvalue* 8.57 *is*

$$\mathbf{v}_j = \begin{bmatrix} 0.33 \\ 0.17 \\ 0.43 \\ 0.22 \\ 0.38 \\ 0.06 \\ 0.51 \\ 0.15 \\ 0.45 \end{bmatrix}$$

which has all positive entries.

The observations made above regarding the spectral properties of the matrix $2.2 \cdot (2\Theta_j\Theta_j^T - E)$ are further shown to be consistent using the results in Table 5.3, 5.4, and 5.5. Each row in these tables corresponds to 10 cluster instances generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution. We fix $k = 3$ and the Dirichlet parameter $\boldsymbol{\alpha} = 0.3\mathbf{e}$. The first column denotes the range corresponding to the size of the subgraph induced by strong nodes and fringe nodes whose feature vectors have largest entry at least 0.6; the cut-off of 0.6 is based on manual parameter search for the given setting of k and $\boldsymbol{\alpha}$. The second column counts *eigenvector success*, i.e. the number of instances, out of 10, for which the eigenvector of $2.2 \cdot (2\Theta_j\Theta_j^T - E)$ corresponding to its largest eigenvalue is positive. For such instances, the third column contains the non-zero eigenvalues of $2.2 \cdot (2\Theta_j\Theta_j^T - E)$.

These computational results motivate the following conjecture.

Table 5.3: Structure of subgraph induced by strong nodes and some fringe nodes for each cluster (part 1/3).

Cluster size range	Eigenvector success (/10)	Non-zero eigenvalues
$6-10$	10	$7.8, 0.5, -1.1$
		$9.8, 0.4, -0.7$
		$9.2, 0.2, -0.4$
		$10.1, 0.4, -0.3$
		$10.1, 0.9, -0.6$
		$8.5, 0.1, -0.3$
		$12.7, 0.2, -0.6$
		$6.6, 0.5, -0.4$
		$9.8, 0.4, -0.5$
		$7.4, 0.3, -0.4$
$11-15$	10	$16.8, 0.2, -0.5$
		$14.2, 0.4, -0.4$
		$12.9, 0.6, -0.5$
		$16.7, 0.6, -0.7$
		$15.8, 0.3, -0.5$
		$15.9, 0.8, -0.8$
		$16.1, 0.2, -0.4$
		$14.0, 0.9, -0.8$
		$12.8, 0.5, -0.4$
		$14.0, 0.3, -0.5$
$16-20$	10	$16.3, 1.0, -1.3$
		$18.1, 0.7, -0.9$
		$17.0, 0.7, -0.7$
		$21.5, 1.2, -1.9$
		$14.5, 1.2, -1.3$
		$21.2, 0.6, -0.7$
		$19.4, 1.2, -1.0$
		$21.6, 0.3, -0.6$
		$23.5, 1.0, -1.0$
		$20.1, 0.6, -1.0$

Table 5.4: Structure of subgraph induced by strong nodes and some fringe nodes for each cluster (part 2/3).

Cluster size range	Eigenvector success (/10)	Non-zero eigenvalues
$21-25$	10	$26.2, 1.7, -1.8$
		$21.6, 1.5, -1.4$
		$23.9, 1.1, -1.2$
		$18.3, 1.6, -1.5$
		$26.5, 0.9, -1.2$
		$23.0, 2.5, -3.0$
		$23.8, 1.6, -1.7$
		$25.6, 1.0, -1.0$
		$20.0, 2.0, -2.2$
		$23.2, 1.4, -1.5$
$26-30$	10	$28.6, 1.2, -1.9$
		$34.0, 2.0, -2.0$
		$32.2, 1.3, -1.6$
		$29.0, 2.4, -2.2$
		$30.0, 1.6, -1.9$
		$30.6, 1.7, -1.8$
		$29.9, 1.4, -1.4$
		$35.9, 1.3, -1.3$
		$23.0, 1.7, -1.4$
		$29.1, 1.5, -1.5$
$31-35$	10	$38.8, 1.5, -1.5$
		$33.7, 1.8, -1.8$
		$38.4, 2.3, -2.2$
		$31.9, 1.9, -1.7$
		$38.3, 1.7, -1.9$
		$33.6, 1.9, -2.2$
		$36.3, 2.2, -1.9$
		$43.7, 1.1, -1.2$
		$29.0, 2.8, -3.2$
		$46.7, 0.9, -1.5$

Table 5.5: Structure of subgraph induced by strong nodes and some fringe nodes for each cluster (part 3/3).

Cluster size range	Eigenvector success (/10)	Non-zero eigenvalues
$36-40$	10	$35.6, 1.8, -2.3$
		$44.1, 2.0, -2.5$
		$40.2, 2.9, -2.8$
		$40.2, 2.5, -2.6$
		$44.5, 1.2, -2.0$
		$49.8, 1.6, -1.5$
		$33.5, 2.2, -2.6$
		$35.1, 2.1, -2.1$
		$46.3, 2.0, -2.1$
		$40.1, 1.8, -1.9$

Conjecture 5.4.2. *Let $G = (V, W)$ be a graph generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution with constant parameter $\boldsymbol{\alpha}$. Then there exists a scalar $t(k, \boldsymbol{\alpha}) \in (0.5, 1/\sqrt{2})$ such that for each $j \in [k]$, with probability not converging to 0 as $n \to \infty$, the largest eigenvalue of the matrix $4.4\Theta_j\Theta_j^T - 2.2E$ is well-separated from the remaining eigenvalues and the corresponding eigenvector is positive where*

$$V_j' := \{i \in [n] : \theta_j^i \geq t(k, \boldsymbol{\alpha})\}$$

and

$$\Theta_j := \Theta(V_j', :).$$

Now let $\mathbf{q}_j := \sqrt{\lambda_j}\mathbf{v}_j$ where λ_j and $\mathbf{v}_j$ denote the largest eigenvalue and the corresponding unit eigenvector respectively of $c \cdot (2\Theta_j\Theta_j^T - E)$. We rewrite (5.3) as

$$W(V_j', V_j') = \mathbf{q}_j\mathbf{q}_j^T - D_j + N_j \tag{5.4}$$

where D_j is the $n_j \times n_j$ diagonal matrix $Diag(\mathbf{q}_j\mathbf{q}_j^T)$ and N_j is the $n_j \times n_j$ symmetric matrix whose each diagonal entry is equal to 0 and the off-diagonal entries are chosen to make so as to make (5.3) hold. That is, matrix D_j applies diagonal correction to ensure $diag(W(V_j, V_j)) = \mathbf{0}$ and matrix N_j captures the error in approximating the logarithmic function g by the linear function l and the error in approximating $c \cdot (2\Theta_j\Theta_j^T - E)$ by $\mathbf{q}_j\mathbf{q}_j^T$. We show, using computational results, the validity of Assumption 5.1 for quantities $\mathbf{q}_j, N_j, W(V_j, V_j)$ described using (5.4) in Table 5.6. Each row in the table corresponds to

Table 5.6: Verification of Assumption 5.1

Cluster size range	C1, C2 success (/10)	Average C3 upper bound
$1201 - 1300$	4	13.3
$1301 - 1400$	4	14.1
$1401 - 1500$	3	13.2
$1501 - 1600$	4	13.3
$1601 - 1700$	5	13.4
$1701 - 1800$	8	13.5
$1801 - 1900$	3	13.6

10 cluster instances generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution. We fix $k = 3$ and the Dirichlet parameter $\boldsymbol{\alpha} = 0.3\mathbf{e}$. The first column denotes the range corresponding to the size of the subgraph induced by strong nodes and fringe nodes whose feature vectors have largest entry at least 0.6; the cut-off of 0.6 is based on manual parameter search for the given setting of k and $\boldsymbol{\alpha}$. The second column counts *C1, C2 success*, i.e. the number of instances, out of 10, for which vector $\mathbf{q}_j$ and matrix $W(V_j', V_j')$ as highlighted in (5.4) satisfy conditions 1 and 2 in Assumption 5.1. We notice that condition 3 is not satisfied for most instances; however, the violation is by a constant factor in the sense that the ratio

$$\frac{45\|\mathbf{n}^i\|\|\mathbf{q}^{\circ 1/3}\|}{q_i\|\mathbf{q}^{\circ 2/3}\|^2} \tag{5.5}$$

for each $i \in [n]$, is bounded above by a constant, albeit much larger than 1 as desired. Therefore in the fourth column of Table 5.6, we present *average C3 upper bound*, i.e. the quantity (5.5) first averaged over all nodes in the graph, then averaged over the instances out 10 runs in which both conditions 1 and 2 are satisfied.

These computational results partly justify Assumption 5.1. Now we turn to the robustness aspect, i.e. understanding the robustness of ℓ_2-`norm-diag` to the presence of noisy nodes. We begin by revisiting Example 5.2 mentioned in Section 5.3.4. In particular, the submatrix of the output of ℓ_2-`norm-diag` corresponding to the nodes in cluster j is

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This shows that ℓ_2-norm-diag is correctly able to cluster the strong nodes, for this example, despite the presence of fringe nodes without creating spurious clusters using the fringe nodes. This observation motivates the following results which show the robustness of the diagonal of an optimal solution of the SDP (P-ND) to perturbations of the weighted adjacency matrix W.

Theorem 5.4.3. *Let X^* be an optimal solution of the SDP (P-ND). If W contains at least one positive entry, then $diag(X^*)$ is uniquely determined by W.*

Theorem 5.4.4. *Let X^* and X' be optimal solutions to the SDP (P-ND) for weighted adjacency matrices W and $W + \Delta$ respectively. If each of W and $W + \Delta$ contains at least one positive entry, then*

$$\|diag(X^*) - diag(X')\| \leq \frac{2\ (2n)^{1/4}\|\Delta\|_F^{1/2}}{\langle W, X^*\rangle^{1/2}}.$$

While Theorem 5.4.4 shows the robustness of the diagonal of an optimal solution of (P-ND) to only perturbations of the weighted adjacency matrix, it is, in fact, observed using computational experiments that all entries of an optimal solution are robust to the presence of fringe nodes whose feature vectors have a relatively smaller largest entry, i.e. fringe nodes that are close to being stray nodes. In particular, the entries of an optimal solution corresponding to the cluster subgraphs comprised of strong nodes and fringe nodes close to the strong set are larger compared to, and therefore well-separated from, the rest of the entries. This is supported by computational results shown in Table 5.7 which show the performance of ℓ_2-norm-diag. Each row in the table corresponds to 10 graph instances generated using the NFM in which the simplex distribution is chosen to be the Dirichlet distribution. We fix $k = 3$ and the Dirichlet parameter $\boldsymbol{\alpha} = 0.3\mathbf{e}$. The first column denotes the size of graph, i.e. number of nodes n. The second column counts the ℓ_2-*norm-diag success*, i.e. the number of instances, out of 10, for which the number of recovered clusters is equal to the true number of clusters k such that the recovered clusters are disjoint and each recovered cluster contains all strong nodes (and possibly some fringe nodes) from exactly one ground-truth cluster.

The theoretical and computational results regarding the performance of ℓ_2-norm-diag presented in this section lead us to make the following conjecture.

Conjecture 5.4.5. *Let G be a graph generated according to NFM in which the simplex distribution is chosen to be the Dirichlet distribution with constant parameter. Then with probability not converging to 0 as $n \to \infty$, ℓ_2-norm-diag returns exactly k disjoint clusters $V_1', \ldots, V_k'$ such that, for each $j \in [k]$*

$$V_j^{strong} \subseteq V_j'.$$

Table 5.7: Performance of ℓ_2-norm-diag.

Graph size (number of nodes)	ℓ_2-norm-diag success (/10)
60	9
70	8
80	10
90	9
100	9
110	10
120	9
130	10
140	10

5.4.2 Proofs

In this section we build a proof of Theorem 5.4.1. Our strategy is to demonstrate the desired structure in each of the submatrices of an optimal solution corresponding to a cluster. One key ingredient for this approach is to determine a point $\mathbf{x}$ such that

$$(W\mathbf{x})^{\circ 1/3} = \mathbf{x}.$$

However, note that the exact solution to the system

$$(\mathbf{q}\mathbf{q}^T\mathbf{x})^{\circ 1/3} = \mathbf{q}$$

can be shown to be $\mathbf{x} = (\beta\mathbf{q})^{\circ 1/3}$ where $\beta = [\mathbf{q}^T(\mathbf{q}^{\circ 1/3})]^{3/2}$. Moreover, due to Assumption 5.1, W can be interpreted as a perturbation of the matrix $\mathbf{q}\mathbf{q}^T$ by matrices D and N thereby motivating the following lemma.

Lemma 5.4.6. *Let $G = (V, W)$ be a graph on n nodes satisfying conditions 1, 3, 4 in Assumption 5.4.1. Then the continuous function $f : \mathbb{R}^n \to \mathbb{R}^n$ defined as $f(\mathbf{x}) := (W\mathbf{x})^{\circ 1/3}$ maps the set*

$$S := \left[\frac{1}{2}(\beta\mathbf{q})^{\circ 1/3}, \frac{3}{2}(\beta\mathbf{q})^{\circ 1/3}\right], \text{ where } \beta = [\mathbf{q}^T(\mathbf{q}^{\circ 1/3})]^{3/2},$$

to itself.

Proof. We will show that for any $\mathbf{x} \in S$, each of $f(\mathbf{x}) \geq (\beta\mathbf{q})^{\circ 1/3}/2$ and $f(\mathbf{x}) \leq 3(\beta\mathbf{q})^{\circ 1/3}/2$

holds separately. Pick any $\mathbf{x} \in S$. We have

$$\begin{aligned}
(W\mathbf{x})^{\circ 1/3} &= (\mathbf{q}\mathbf{q}^T\mathbf{x} - D\mathbf{x} + N\mathbf{x})^{\circ 1/3} \\
&\geq \left[\frac{\beta^{1/3}\mathbf{q}\mathbf{q}^T(\mathbf{q}^{\circ 1/3})}{2} - D\mathbf{x} + N\mathbf{x}\right]^{\circ 1/3} && \text{(using the lower bound on } \mathbf{x}) \\
&= \left[\frac{\beta\mathbf{q}}{2} - D\mathbf{x} + N\mathbf{x}\right]^{\circ 1/3} && \text{(using the definition of } \beta) \\
&\geq \left[\frac{\beta\mathbf{q}}{2} - \frac{3\beta^{1/3}D(\mathbf{q}^{\circ 1/3})}{2} + N\mathbf{x}\right]^{\circ 1/3} && \text{(using the upper bound on } \mathbf{x}) \\
&\geq \left[\frac{\beta\mathbf{q}}{2} - \frac{3\beta^{1/3}\mathbf{q}^{\circ 7/3}}{2} + N\mathbf{x}\right]^{\circ 1/3}. && \text{(using the definition of } D)
\end{aligned} \tag{5.6}$$

Now we bound each of the second and third terms above separately. Condition 1 in Assumption 5.1 implies for each $i \in [n]$,

$$\begin{aligned}
q_i^{7/3} &\leq \frac{\|\mathbf{q}^{\circ 2/3}\|^2 q_i}{45} \\
&= \frac{\left(\sum_{i\in[n]} q_i^{4/3}\right) q_i}{45} \\
&= \frac{[\mathbf{q}^T(\mathbf{q}^{\circ 1/3})]q_i}{45} \\
&= \frac{\beta^{2/3}q_i}{45}. && \text{(using the definition of } \beta)
\end{aligned}$$

The above chain implies that

$$\begin{aligned}
&\mathbf{q}^{\circ 7/3} \leq \frac{\beta^{2/3}\mathbf{q}}{45} \\
\iff &\frac{3\beta^{1/3}\mathbf{q}^{\circ 7/3}}{2} \leq \frac{\beta\mathbf{q}}{30}. && \text{(multiplying both sides by } 3\beta^{1/3}/2)
\end{aligned} \tag{5.7}$$

Moreover, for each $i \in [n]$, we have

$$\begin{aligned}
|(N\mathbf{x})_i| &= |\mathbf{n}^i\mathbf{x}| \\
&\leq \|\mathbf{n}^i\|\|\mathbf{x}\| && \text{(using Cauchy-Schwarz inequality)} \\
&\leq \frac{q_i\|\mathbf{q}^{\circ 2/3}\|^2}{45\|\mathbf{q}^{\circ 1/3}\|}\|\mathbf{x}\| && \text{(using condition 3 in Assumption 5.1)} \\
&\leq \frac{q_i\|\mathbf{q}^{\circ 2/3}\|^2\beta^{1/3}}{30} && \text{(using the upper bound on } \mathbf{x}\text{)} \\
&= \frac{\beta q_i}{30}. && \text{(using the definition of } \beta\text{)}
\end{aligned} \tag{5.8}$$

Then using (5.7) and (5.8) in (5.6), we get

$$\begin{aligned}
(W\mathbf{x})^{\circ 1/3} &\geq \left(\frac{\beta\mathbf{q}}{2} - \frac{\beta\mathbf{q}}{15}\right)^{\circ 1/3} \\
&= \left(\frac{13}{30}\right)^{1/3}(\beta\mathbf{q})^{\circ 1/3} \\
&> \frac{1}{2}(\beta\mathbf{q})^{\circ 1/3}.
\end{aligned} \tag{5.9}$$

For any $\mathbf{x} \in S$, we also have

$$\begin{aligned}
(W\mathbf{x})^{\circ 1/3} &= (\mathbf{q}\mathbf{q}^T\mathbf{x} - D\mathbf{x} + N\mathbf{x})^{\circ 1/3} \\
&\leq \left[\frac{3\beta^{1/3}\mathbf{q}\mathbf{q}^T(\mathbf{q}^{\circ 1/3})}{2} - D\mathbf{x} + N\mathbf{x}\right]^{\circ 1/3} && \text{(using the upper bound on } \mathbf{x}\text{)} \\
&= \left[\frac{3\beta\mathbf{q}}{2} - D\mathbf{x} + N\mathbf{x}\right]^{\circ 1/3} && \text{(using the definition of } \beta\text{)} \\
&\leq \left[\frac{3\beta\mathbf{q}}{2} + N\mathbf{x}\right]^{\circ 1/3} && (\because D\mathbf{x} > 0) \\
&\leq \left(\frac{23}{15}\right)^{1/3}(\beta\mathbf{q})^{\circ 1/3} && \text{(using (5.8))} \\
&< \frac{3}{2}(\beta\mathbf{q})^{\circ 1/3}.
\end{aligned} \tag{5.10}$$

Combining (5.9) and (5.10), we conclude that the function f maps S to itself. □

To proceed with the proof of Theorem 5.4.1, in addition to Lemma 5.4.6, we also make use of the following Brouwer fixed-point theorem.

Theorem 5.4.7 (Brouwer Fixed-Point Theorem [19])**.** *Let $C \subseteq \mathbb{R}^n$ be a non-empty convex compact set and let $f : C \to C$ be a continuous function. Then there exists a point $\mathbf{x} \in C$ such that $f(\mathbf{x}) = \mathbf{x}$.*

Proof of Theorem 5.4.1. For each $j \in [k]$, let $\mathbf{q}_j, D_j, N_j, U_j$ and S_j denote the quantities mentioned in Assumption 5.1, and define n_j as the cardinality of V_j (i.e. the size of cluster j). Our analysis uses SDP duality and therefore note that the dual of (P-ND) is

$$\begin{aligned} \min_{(X,Y,Z,\lambda)} \quad & \lambda \|diag(X)\|^2 + \lambda \\ \text{s.t.} \quad & Y \geq 0 \\ & Z \succeq 0 \\ & \lambda \geq 0 \\ & W + Y + Z = \lambda \cdot Diag(X). \end{aligned} \tag{D-ND}$$

Both optimization problems (P-ND) and (D-ND) have strictly feasible solutions. For instance, $X' := (0.5I + 0.5E)/n$ is a positive, positive definite matrix which is feasible for (P-ND). Similarly, $X' := I, Y' := E$, $Z' := (\|W+E\|+\epsilon)I-(W+E)$ and $\lambda' := (\|W+E\|+\epsilon)$ gives a strictly feasible solution (X', Y', Z', λ') for (D-ND) for any $\epsilon > 0$. More specifically, we may express (P-ND) in the standard form as

$$\begin{aligned} \max_{M} \quad & \langle W^{SF}, M \rangle \\ \text{s.t.} \quad & M \in \mathbb{S}^{n^2+2n+1} \\ & M \succeq 0 \\ & M_{ab} + M_{ba} = 0 \ \forall a \in [n], b \in [n^2+2n+1] \setminus [n] \\ & M_{ab} + M_{ba} = 0 \ \forall a \in [n^2+n] \setminus [n], b \in \{i+1, \ldots, n^2+2n+1\} \\ & M_{ab} + M_{ba} = 2M_{ss} \ \forall a \in [n], b \in [n], s = (i-1)n + j + n \\ & M_{aa} = 1 \ \forall a \in [n^2+2n+1] \setminus [n^2+n] \\ & M_{ab} + M_{ba} = 0 \ \forall a \in [n^2+2n+1] \setminus [n^2+n], b \in \{i+1, \ldots, n^2+2n+1\} \\ & 2M_{aa} = M_{st} + M_{ts} \ \forall a \in [n], s = n^2+n+1, t = n^2+n+1+i \end{aligned} \tag{P-ND-SF}$$

where W^{SF} is the $(n^2+2n+1) \times (n^2+2n+1)$ symmetric matrix defined as

$$W^{SF}_{ij} := \begin{cases} W_{ab} & a \in [n], b \in [n] \\ 0 & \text{otherwise} \end{cases}$$

and construct strictly feasible solutions for (P-ND-SF) and its dual (which is not included here for brevity). Therefore using the Karush-Kuhn-Tucker (KKT) conditions for optimality, X^* is an optimal solution for (P-ND) if and only if X^* is feasible for (P-ND) and there exist a non-negative matrix $Y^* \in \mathbb{S}^n$, a positive semidefinite matrix Z^*, and a non-negative scalar λ^* such that:

- $X^*_{ij} Y^*_{ij} = 0, \forall i, j \in [n]$
- $\langle X^*, Z^* \rangle = 0$
- $\lambda^* \cdot (\|diag(X^*)\| - 1) = 0$
- $W + Y^* + Z^* = \lambda^* \cdot Diag(X^*)$

In the remainder of the proof, we will explicitly construct all the above mentioned quantities. Now for each $j \in [k]$, since $W(V_j, V_j)$ satisfies Assumption 5.1, using Lemma 5.4.6 and Theorem 5.4.7, we conclude that there exists an n_j-dimensional vector $\mathbf{r}_j \in \left[\frac{1}{2}(\beta_j \mathbf{q}_j)^{\circ 1/3}, \frac{3}{2}(\beta_j \mathbf{q}_j)^{\circ 1/3}\right]$, where $\beta_j = [\mathbf{q}_j^T (\mathbf{q}_j^{\circ 1/3})]^{3/2}$, such that

$$W(V_j, V_j) \cdot \mathbf{r}_j = \mathbf{r}_j^{\circ 3}. \tag{5.11}$$

We set

$$\lambda^* = \sqrt{\sum_{j \in [k]} \|\mathbf{r}_j \circ \mathbf{r}_j\|^2}.$$

For each $j \in [k]$, we set

$$\begin{aligned} X^*(V_j, V_j) &= \mathbf{r}_j \mathbf{r}_j^T / \lambda^* \\ Y^*(V_j, V_j) &= 0 \\ Z^*(V_j, V_j) &= Diag(\mathbf{r}_j \circ \mathbf{r}_j) - W(V_j, V_j). \end{aligned}$$

For each distinct $j, j' \in [k]$, we set

$$\begin{aligned} X^*(V_j, V_{j'}) &= 0 \\ Y^*(V_j, V_{j'}) &= -W(V_j, V_{j'}) \\ Z^*(V_j, V_{j'}) &= 0. \end{aligned}$$

For each stray node v, we set

$$\begin{aligned} &X^*(v,:) = 0 && (\text{and } X^*(:,v) = 0)\\ &Y^*(v,:) = -W(v,:) && (\text{and } Y^*(:,v) = -W(:,v))\\ &Z^*(v,:) = 0. && (\text{and } Z^*(:,v) = 0) \end{aligned}$$

First we show that the constructed X^* is feasible for (P-ND). Note that there exists a permutation of the rows (and columns) of X^* which yields a block diagonal matrix in which the non-zero blocks are given by the rank-one positive semidefinite matrices $\mathbf{r}_1\mathbf{r}_1^T/\lambda^*, \ldots, \mathbf{r}_k\mathbf{r}_k^T/\lambda^*$. This shows that X^* is positive semidefinite. Morever, since vectors $\mathbf{r}_1, \ldots, \mathbf{r}_k$ are positive, we conclude that X^* is non-negative. We also have

$$\begin{aligned} \|diag(X^*)\|^2 &= \frac{\sum_{j\in[k]} \|\mathbf{r}_j \circ \mathbf{r}_j\|^2}{{\lambda^*}^2}\\ &= 1. && (\text{using the definition of } \lambda^*) \end{aligned}$$

Therefore X^* is feasible for (P-ND). Note that this also implies that $\lambda^*(\|diag(X^*)\|-1) = 0$.

Now we show that the constructed Y^*, Z^*, λ^* satisfy the remaining desired properties. Because each pair of nodes lying in distinct clusters shares a negative edge and because each stray node shares a negative edge with every other node in the graph, we have that $Y^* \geq 0$. Also note that λ^* is a positive scalar by construction.

Matrices X^* and Y^* have disjoint supports by construction, and therefore $X^*_{ij}Y^*_{ij} = 0$ for each $i, j \in [n]$. Moreover, for each $j \in [k]$, using (5.11), we have

$$\begin{aligned} & W(V_j,V_j)\cdot\mathbf{r}_j = \mathbf{r}_j^{\circ 3}\\ \iff & W(V_j,V_j)\cdot\mathbf{r}_j = Diag(\mathbf{r}_j\circ\mathbf{r}_j)\cdot\mathbf{r}_j\\ \iff & Z^*(V_j,V_j)\cdot\mathbf{r}_j = 0 && (\text{using the definition of } Z^*)\\ \iff & Z^*(V_j,V_j)\cdot\mathbf{r}_j\mathbf{r}_j^T/\lambda^* = 0 && (\because \mathbf{r}_j > 0, \lambda^* > 0)\\ \iff & Z^*(V_j,V_j)\cdot X^*(V_j,V_j) = 0 && (\text{using the definition of } X^*)\\ \iff & \langle Z^*(V_j,V_j), X^*(V_j,V_j)\rangle = 0 && (\because X^*, Z^* \succeq 0) \end{aligned} \tag{5.12}$$

Therefore we have

$$\begin{aligned} \langle X^*, Z^*\rangle &= \sum_{j\in[k]} \langle X^*(V_j,V_j), Z^*(V_j,V_j)\rangle && (\text{using the definitions of } X^*, Z^*)\\ &= 0. && (\text{using (5.12)}) \end{aligned}$$

Note that there exists a permutation of the rows (and columns) of Z^* which yields a block diagonal matrix in which the non-zero blocks are given by the matrices $Diag(\mathbf{r}_1 \circ \mathbf{r}_1) - W(V_1, V_1), \ldots, Diag(\mathbf{r}_k \circ \mathbf{r}_k) - W(V_k, V_k)$. Therefore to show the positive semidefiniteness of Z^*, it suffices to show that for each $j \in [k]$, the matrix $Z^*(V_j, V_j) = Diag(\mathbf{r}_j \circ \mathbf{r}_j) - W(V_j, V_j)$ is positive semidefinite. From (5.12), we know that $Z^*(V_j, V_j) \cdot \mathbf{r}_j = 0$. This implies that $\mathbf{e}$ belongs to the null space of $Diag(\mathbf{r}_j) \cdot Z^*(V_j, V_j) \cdot Diag(\mathbf{r}_j)$. Consequently, we observe that

$$\bar{L}_j := Diag(\mathbf{r}_j) \cdot Z^*(V_j, V_j) \cdot Diag(\mathbf{r}_j)$$

is the Laplacian matrix of a graph, called $\bar{G}_j$, on n_j nodes whose weighted adjacency matrix is

$$\bar{W}_j := Diag(\mathbf{r}_j) \cdot W(V_j, V_j) \cdot Diag(\mathbf{r}_j).$$

Moreover, Z^* is positive semidefinite if and only if the Laplacian $\bar{L}_j$ is positive semidefinite since each entry of $\mathbf{r}_j$ is positive. Note that the sign of each edge in $\bar{G}_j$ is identical to that of the corresponding edge in $G[V_j]$ which implies that the set of all nodes in $\bar{G}_j$ adjacent to a negative edge is U_j. Now for any $u \in U_j$ and $s \in S_j$, we have

$$\begin{aligned}
|S_j| \bar{W}_j(u, s) &= |S_j| r_j(u) r_j(s) W(u, s) \\
&\geq |S_j| r_j(u) \frac{[\beta_j q_j(s)]^{1/3}}{2} W(u, s) \\
&\qquad \text{(using the lower bound on } \mathbf{r}_j\text{)} \\
&\geq -3\beta_j^{1/3} \left(\sum_{\substack{u' \in U_j: \\ W(u,u') < 0}} r_j(u) q_j(u')^{1/3} W(u, u') \right) \\
&\qquad \text{(using condition 2 in Assumption 5.1)} \\
&\geq 2 \left(\sum_{\substack{u' \in U_j: \\ W(u,u') < 0}} r_j(u) r_j(u') W(u, u') \right) \\
&\qquad \text{(using the upper bound on } \mathbf{r}_j\text{)} \\
&= 2 \left(\sum_{\substack{u' \in U_j: \\ \bar{W}_j(u,u') < 0}} \bar{W}_j(u, u') \right). \\
&\qquad \text{(using the definition of } \bar{W}\text{)}
\end{aligned}$$

Thus we have shown that graph $\bar{G}_j$ satisfies (5.1) stated in Theorem 5.3.4 using which we conclude that $\bar{L}_j$ is positive semidefinite.

Lastly, we show that the equation $W + Y^* + Z^* = \lambda^* \cdot Diag(X^*)$ is satisfied. For each $j \in [k]$, we have

$$\begin{aligned} W(V_j, V_j) + Y^*(V_j, V_j) + Z^*(V_j, V_j) &= Diag(\mathbf{r}_j \circ \mathbf{r}_j) \\ &\qquad \text{(using the definitions of } Y^*, Z^*) \\ &= \lambda^* \cdot Diag(X^*(V_j, V_j)). \\ &\qquad \text{(using the definition of } X^*) \end{aligned}$$

For each distinct $j, j' \in [k]$, we have

$$W(V_j, V_{j'}) + Y^*(V_j, V_{j'}) + Z^*(V_j, V_{j'}) = 0$$

using the definitions of Y^*, Z^*. Similarly, for each stray node v, we have

$$\begin{aligned} W(v, :) + Y^*(v, :) + Z^*(v, :) = 0 \\ W(:, v) + Y^*(:, v) + Z^*(:, v) = 0 \end{aligned}$$

using the definitions of Y^*, Z^*. □

Proof of Theorem 5.4.3. Observe that $X = 0$ is a feasible solution for (P-ND) which implies that the optimal value of (P-ND) is non-negative. This implies that for any optimal solution, without loss of generality, we may assume that the constraint $\|diag(X)\| \leq 1$ is tight. Since X^* is optimal for (P-ND), and since both (P-ND) and its dual have strictly feasible solutions, using the Karush-Kuhn-Tucker (KKT) conditions for optimality, there exist a non-negative matrix $Y^* \in \mathbb{S}^n$, a positive semidefinite matrix Z^*, and a non-negative scalar λ^* such that:

- $X^*_{ij} Y^*_{ij} = 0, \forall i, j \in [n]$
- $\langle X^*, Z^* \rangle = 0$
- $\lambda^* \cdot (\|diag(X^*)\| - 1) = 0$
- $W + Y^* + Z^* = \lambda^* \cdot Diag(X^*)$

We also note that λ^* is a positive scalar. Indeed if λ^* is zero, then the last condition above implies $diag(Z^*)$ is zero and consequently $Z^* = 0$ since Z^* is positive semidefinite. This implies that $Y^* = -W$ which contradicts the non-negativity of Y^* since W contains a positive entry.

Let X^{**} be another optimal solution of (P-ND). Then we have

$$\begin{aligned}
0 &= \langle W, X^* - X^{**} \rangle \\
&= \langle \lambda^* \cdot Diag(X^*) - Y^* - Z^*, X^* - X^{**} \rangle && \text{(substituting for } W) \\
&= \lambda^* - \lambda^* \cdot \langle Diag(X^*), X^{**} \rangle - \langle Y^* + Z^*, X^* - X^{**} \rangle && (\because \|diag(X^*)\| = 1) \\
&= \lambda^* - \lambda^* \cdot \langle Diag(X^*), X^{**} \rangle + \langle Y^* + Z^*, X^{**} \rangle && (\because \langle Y^*, X^* \rangle = \langle Z^*, X^* \rangle = 0) \\
&\geq \lambda^* - \lambda^* \cdot \langle Diag(X^*), X^{**} \rangle. && (\because \langle Y^*, X^{**} \rangle, \langle Z^*, X^{**} \rangle \geq 0)
\end{aligned}$$

Using the fact that λ^* is positive, the above implies that $\langle diag(X^*), diag(X^{**}) \rangle \geq 1$. However, since both $diag(X^*)$ and $diag(X^{**})$ lie on the unit sphere, we conclude that $diag(X^*) = diag(X^{**})$. □

Proof of Theorem 5.4.4. Note that since W has at least one positive entry, $\max(W_+)$ is a positive scalar. If $W_{ii'} > 0$ for some $i, i' \in [n]$, then $(\mathbf{e}_i \mathbf{e}_{i'}^T + \mathbf{e}_{i'} \mathbf{e}_i^T)/\sqrt{2}$ is feasible for (P-ND) and we have

$$\langle W, X^* \rangle \geq \sqrt{2} W_{ii'} > 0. \tag{5.13}$$

Using a similar argument, we also conclude that

$$\langle W + \Delta, X' \rangle > 0. \tag{5.14}$$

Moreover since the optimal values of the two programs are positive, we have that

$$\|diag(X^*)\| = \|diag(X')\| = 1.$$

Observe that

$$\begin{aligned}
|\langle \Delta, X^* \rangle| &\leq \|\Delta\|_F \|X^*\|_F && \text{(using Cauchy-Schwarz inequality)} \\
&\leq \sqrt{n} \|\Delta\|_F \|diag(X^*)\| && (\because \|X^*\|_F \leq \sqrt{n} \|diag(X^*)\|) \\
&= \sqrt{n} \|\Delta\|_F. && (\because \|diag(X^*)\| = 1)
\end{aligned} \tag{5.15}$$

Similarly, we also have that

$$|\langle \Delta, X' \rangle| \leq \sqrt{n} \|\Delta\|_F. \tag{5.16}$$

Now define

$$X'' := \frac{X^* + X'}{\|diag(X^*) + diag(X')\|}.$$

Noting that X'' is feasible for (P-ND), and therefore using the fact that $\langle W, X^* \rangle \geq \langle W, X'' \rangle$, we get

$$\begin{aligned}
\langle W, X^* \rangle \|diag(X^*) + diag(X')\| &\geq \langle W, X^* \rangle + \langle W, X' \rangle \\
&= \langle W, X^* \rangle + \langle W + \Delta, X' \rangle - \langle \Delta, X' \rangle \\
&\geq \langle W, X^* \rangle + \langle W + \Delta, X^* \rangle - \langle \Delta, X' \rangle \\
&\qquad \text{(using the optimality of } X') \\
&= 2\langle W, X^* \rangle + \langle \Delta, X^* \rangle - \langle \Delta, X' \rangle \\
&\geq 2\langle W, X^* \rangle - 2\sqrt{n}\|\Delta\|_F \\
&\qquad \text{(using (5.15) and (5.16))}
\end{aligned}$$

which is equivalent to

$$\|diag(X^*) + diag(X')\| \geq 2 - \frac{2\sqrt{n}\|\Delta\|_F}{\langle W, X^* \rangle} \tag{5.17}$$

since $\langle W, X^* \rangle$ is positive. Now we have

$$\begin{aligned}
\|diag(X^*) - diag(X')\| &= \sqrt{4 - \|diag(X^*) + diag(X')\|^2} \\
&\qquad (\because \|diag(X^*)\| = \|diag(X')\| = 1) \\
&\leq \frac{2\sqrt{2}n^{1/4}\|\Delta\|_F^{1/2}}{\langle W, X^* \rangle^{1/2}}. \\
&\qquad \text{(using (5.17))}
\end{aligned}$$

This concludes the proof. □

5.5 Conclusions

In this chapter, we propose a novel generative model, NFM, for graphs which, unlike the SBM, also generates feature vectors for each node in the graph. We analyze, theoretically and computationally, the performance of two different SDP formulations in recovering the true clusters in graph instances generated according to the NFM. In particular, we begin with an algorithm based on the SDP (P-1D), but then demonstrate its lack of robustness to certain noisy instances generated by the NFM. To overcome this shortcoming, we propose a new algorithm based on a different SDP (P-ND). We build theory towards showing that SDP (P-ND) can be used to provably recover, for each true cluster, nodes with sufficiently strong membership signal in their feature vectors, in the presence of noisy nodes, without involving any tuning parameters.

Chapter 6

Conclusions and Future Work Directions

In this thesis, we study two graph clustering problems, Overlapping Community Detection and Correlation Clustering, using the provable recovery framework. That is, for each problem, we consider a graph generative model, propose clustering algorithm(s), and develop theoretical guarantees regarding the performance of the proposed algorithm(s) in recovering the ground truth clustering posited by the considered generative model. The proposed algorithms rely on formulations and techniques from convex optimization.

For the Overlapping Community Detection problem, we consider the Mixed Membership Stochastic Blockmodel (MMSB), which is a generalization of the Stochastic Block Model (SBM) to allow overlapping communities. We propose a linear-programming-based algorithm which is relatively easy to implement, in part because it is almost tuning-parameter-free; indeed the algorithm requires only an a priori estimate of the number of communities in the input graph, which also appears as a parameter in other competing algorithms in the literature. We show theoretically that the proposed algorithm recovers an entrywise close approximation to the true mixed membership of each node. Our analysis does not explicitly require each community to have a node which belongs exclusively to that community. Indeed this assumption is often made in literature but is not realistic. We also show experimental performance of the proposed algorithm on synthetic and real-world datasets. This work leads to some interesting follow-up questions for future work. Firstly, it remains an open question to theoretically understand the robustness properties of the proposed algorithm. Indeed our analysis assumes access to the exact weighted adjacency matrix containing pairwise similarity scores generated according to the MMSB. However, in practice, the weighted adjacency matrix generated by the MMSB may be corrupted

with noise. We leave it as future work to extend the theoretical guarantees presented here to a setting in which the weighted adjacency matrix is, for instance, either uniformly corrupted with noise or available exactly but only partially, i.e. only some entries are available. Secondly, it is an interesting future work direction to provide a theoretical basis for the selection of estimated number of communities in the input graph, which appears as a parameter in the recovery algorithm.

For the Correlation Clustering problem, we introduce a novel graph generative model, Node Features Model (NFM), to generate signed random graphs in which the edge weights represent similarity and dissimilarity scores. The graph instances are obtained by generating random feature vectors for the nodes which can be interpreted as latent variables in the model. Moreover, the graph instances contain asymmetric noise in the sense that some pairs of nodes in the same cluster may potentially share a negative edge, but all pairs of nodes in different clusters share a negative edge. We first consider a semidefinite programming (SDP)-based algorithm which uses an SDP formulation that gives the best approximation ratio for the Correlation Clustering problem of maximizing agreements. We show the success of this algorithm in certain restrictive settings, but also demonstrate its potential lack of robustness to noisy instances generated by the NFM. Consequently, we propose a different SDP-based algorithm which appears to computationally address the robustness shortcoming and is tuning-parameter-free. We make progress towards showing that the proposed algorithm provably recovers at least the nodes whose feature vectors represent sufficiently strong cluster membership, in the presence of noisy nodes. In particular, we show exact recovery by the proposed algorithm if each cluster subgraph satisfies certain deterministic assumptions. We use computational experiments to show the validity of the aforementioned deterministic assumptions in the NFM. We also make progress towards theoretically explaining robustness of the proposed algorithm, as seen in computational experiments. We also show successful performance of the proposed algorithm on synthetic datasets. This work naturally poses interesting questions for future work. Firstly, providing a complete theoretical explanation for the robustness of the proposed algorithm is left as future work. Secondly, it is also an important open question to bridge the gap between the recovery guarantees and the NFM, i.e. to show that the deterministic conditions required for provable recovery are indeed satisfied by the NFM graph instances with probability not converging to zero asymptotically as the graph size grows. Lastly, it is a useful future exercise to understand, theoretically and computationally, the performance of the proposed algorithm using more flexible models than the NFM which are not restricted to asymmetric noise.

References

[1] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

[2] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.

[3] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, Eric P Xing, and Tommi Jaakkola. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the International Biometrics Society Annual Meeting*, volume 15, 2006.

[4] Brendan PW Ames and Stephen A Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical Programming*, 129(1):69–89, 2011.

[5] Animashree Anandkumar, Rong Ge, Daniel Hsu, and Sham Kakade. A tensor spectral approach to learning mixed membership community models. In *Conference on Learning Theory*, pages 867–881. PMLR, 2013.

[6] Animashree Anandkumar, Rong Ge, Daniel Hsu, and Sham M Kakade. A tensor approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15(1):2239–2312, 2014.

[7] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. More algorithms for provable dictionary learning. *arXiv preprint arXiv:1401.0579*, 2014.

[8] Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: Toward a rigorous approach. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 37–54, 2012.

[9] Maria-Florina Balcan, Christian Borgs, Mark Braverman, Jennifer Chayes, and Shang-Hua Teng. Finding endogenously formed communities. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 767–783. SIAM, 2013.

[10] Jess Banks, Cristopher Moore, Joe Neeman, and Praneeth Netrapalli. Information-theoretic thresholds for community detection in sparse networks. In *Conference on Learning Theory*, pages 383–416. PMLR, 2016.

[11] Jess Banks, Cristopher Moore, Roman Vershynin, Nicolas Verzelen, and Jiaming Xu. Information-theoretic bounds and phase transitions in clustering, sparse PCA, and submatrix localization. *IEEE Transactions on Information Theory*, 64(7):4872–4894, 2018.

[12] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[13] Punam Bedi and Chhavi Sharma. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135, 2016.

[14] Shai Ben-David, Ulrike Von Luxburg, and Dávid Pál. A sober look at clustering stability. In *International Conference on Computational Learning Theory*, pages 5–19. Springer, 2006.

[15] Marshall Bern, John R Gilbert, Bruce Hendrickson, Nhat Nguyen, and Sivan Toledo. Support-graph preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27(4):930–951, 2006.

[16] Quentin Berthet and Nicolai Baldin. Statistical and computational rates in graph logistic regression. In *International Conference on Artificial Intelligence and Statistics*, pages 2719–2730. PMLR, 2020.

[17] Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse PCA. *arXiv preprint arXiv:1304.0828*, 2013.

[18] Quentin Berthet, Philippe Rigollet, and Piyush Srivastava. Exact recovery in the Ising blockmodel. *The Annals of Statistics*, 47(4):1805–1834, 2019.

[19] L.E.J. Brouwer. Über abbildung von mannigfaltigkeiten. *Mathematische Annalen*, 71:97–115, 1912.

[20] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[21] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[22] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

[23] Yudong Chen, Xiaodong Li, and Jiaming Xu. Convexified modularity maximization for degree-corrected stochastic block models. *The Annals of Statistics*, 46(4):1573–1602, 2018.

[24] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[25] Sean R Collins, Kyle M Miller, Nancy L Maas, Assen Roguev, Jeffrey Fillingham, Clement S Chu, Maya Schuldiner, Marinella Gebbia, Judith Recht, Michael Shales, et al. Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, 446(7137):806–810, 2007.

[26] Vinh Loc Dao, Cécile Bothorel, and Philippe Lenca. Community structure: A comparative evaluation of community detection methods. *Network Science*, 8(1):1–41, 2020.

[27] Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006.

[28] Yon Dourisboure, Filippo Geraci, and Marco Pellegrini. Extraction and classification of dense implicit communities in the web graph. *ACM Transactions on the Web (TWEB)*, 3(2):1–36, 2009.

[29] Nan Du, Bin Wu, Xin Pei, Bai Wang, and Liutong Xu. Community detection in large-scale social networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 16–25, 2007.

[30] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[31] Anne-Claude Gavin, Patrick Aloy, Paola Grandi, Roland Krause, Markus Boesche, Martina Marzioch, Christina Rau, Lars Juhl Jensen, Sonja Bastuck, Birgit Dümpelfeld, et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631, 2006.

[32] Nicolas Gillis and Stephen A Vavasis. Fast and robust recursive algorithmsfor separable nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):698–714, 2013.

[33] Prem K Gopalan and David M Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

[34] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[35] Hao Hu, Renata Sotirov, and Henry Wolkowicz. Facial reduction for symmetry reduced semidefinite doubly nonnegative programs. *arXiv preprint arXiv:1912.10245*, 2019.

[36] Kejun Huang and Xiao Fu. Detecting overlapping and correlated communities without pure nodes: Identifiability and algorithm. In *International Conference on Machine Learning*, pages 2859–2868, 2019.

[37] Kejun Huang, Xiao Fu, and Nikolaos D Sidiropoulos. Anchor-free correlated topic modeling: Identifiability and algorithm. In *Advances in Neural Information Processing Systems*, pages 1786–1794, 2016.

[38] Jafar Jafarov, Sanchit Kalhan, Konstantin Makarychev, and Yury Makarychev. Correlation clustering with asymmetric classification errors. In *International Conference on Machine Learning*, pages 4641–4650. PMLR, 2020.

[39] Jafar Jafarov, Sanchit Kalhan, Konstantin Makarychev, and Yury Makarychev. Local correlation clustering with asymmetric classification errors. In *International Conference on Machine Learning*, pages 4677–4686. PMLR, 2021.

[40] Thorsten Joachims and John Hopcroft. Error bounds for correlation clustering. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 385–392, 2005.

[41] Emilie Kaufmann, Thomas Bonald, and Marc Lelarge. A spectral algorithm with additive clustering for the recovery of overlapping communities in networks. In *International Conference on Algorithmic Learning Theory*, pages 355–370. Springer, 2016.

[42] Ramya Korlakai Vinayak, Samet Oymak, and Babak Hassibi. Graph clustering with missing data: Convex algorithms and analysis. *Advances in Neural Information Processing Systems*, 27:2996–3004, 2014.

[43] Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L Johnson. *Continuous multivariate distributions, Volume 1: Models and applications*. John Wiley & Sons, 2004.

[44] Ravishankar Krishnaswamy, Nived Rajaraman, et al. Robust correlation clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[45] Nevan J Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

[46] Jing Lei, Alessandro Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

[47] Xiaodong Li, Yudong Chen, and Jiaming Xu. Convex relaxation methods for community detection. *Statistical Science*, 36(1):2–15, 2021.

[48] Xinxin Li, Ting Kei Pong, Hao Sun, and Henry Wolkowicz. A strictly contractive peaceman-rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem. *Computational Optimization and Applications*, 78(3):853–891, 2021.

[49] Jimit Majmudar and Stephen Vavasis. Provable overlapping community detection in weighted graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19028–19038. Curran Associates, Inc., 2020.

[50] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Correlation clustering with noisy partial information. In *Conference on Learning Theory*, pages 1321–1342, 2015.

[51] Xueyu Mao, Purnamrita Sarkar, and Deepayan Chakrabarti. On mixed memberships and symmetric nonnegative matrix factorizations. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2324–2333. JMLR. org, 2017.

[52] Xueyu Mao, Purnamrita Sarkar, and Deepayan Chakrabarti. Overlapping clustering models, and one (class) SVM to bind them all. In *Advances in Neural Information Processing Systems*, pages 2126–2136, 2018.

[53] Xueyu Mao, Purnamrita Sarkar, and Deepayan Chakrabarti. Estimating mixed memberships with sharp eigenvector deviations. *Journal of the American Statistical Association*, pages 1–13, 2020.

[54] Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 712–728. SIAM, 2010.

[55] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM (JACM)*, 31(1):114–127, 1984.

[56] Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E Tarjan. Clustering social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 56–67. Springer, 2007.

[57] James R Munkres. *Analysis on manifolds*. CRC Press, 2018.

[58] Tamás Nepusz, Haiyuan Yu, and Alberto Paccanaro. Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 9(5):471, 2012.

[59] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[60] Danilo Elias Oliveira, Henry Wolkowicz, and Yangyang Xu. ADMM for the SDP relaxation of the QAP. *Mathematical Programming Computation*, 10(4):631–658, 2018.

[61] Lorenzo Orecchia and Zeyuan Allen Zhu. Flow-based algorithms for local graph clustering. In *Proceedings of the Twenty-Fifth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1267–1286. SIAM, 2014.

[62] David Pollard. Strong consistency of k-means clustering. *The Annals of Statistics*, pages 135–140, 1981.

[63] Mason A Porter, Peter J Mucha, Mark EJ Newman, and Andrew J Friend. Community structure in the United States house of representatives. *Physica A: Statistical Mechanics and its Applications*, 386(1):414–438, 2007.

[64] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. Overlapping community detection using Bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114, 2011.

[65] Qing Qu, Ju Sun, and John Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems*, pages 3401–3409, 2014.

[66] Sara Rahiminejad, Mano R Maurya, and Shankar Subramaniam. Topological and functional comparison of community detection algorithms in biological networks. *BMC Bioinformatics*, 20(1):1–25, 2019.

[67] Ali Rahnavard, Suvo Chatterjee, Bahar Sayoldin, Keith A Crandall, Fasil Tekola-Ayele, and Himel Mallick. Omics community detection using multi-resolution clustering. *Bioinformatics*, 2021.

[68] Avik Ray, Javad Ghaderi, Sujay Sanghavi, and Sanjay Shakkottai. Overlap graph clustering via successive removal. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 278–285. IEEE, 2014.

[69] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12), 2011.

[70] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[71] Michael T Schaub, Jean-Charles Delvenne, Martin Rosvall, and Renaud Lambiotte. The many facets of community detection in complex networks. *Applied Network Science*, 2(1):4, 2017.

[72] Vatsal Sharan, Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Compressed factorization: Fast and accurate low-rank factorization of compressively-sensed data. In *International Conference on Machine Learning*, pages 5690–5700, 2019.

[73] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

[74] Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, 2012.

[75] Chaitanya Swamy. Correlation clustering: Maximizing agreements via semidefinite programming. In *SODA*, volume 4, pages 526–527. Citeseer, 2004.

[76] Levent Tunçel. *Polyhedral and semidefinite programming methods in combinatorial optimization*, volume 27. American Mathematical Soc., 2016.

[77] Stijn Van Dongen. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008.

[78] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[79] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[80] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[81] Tengyao Wang, Quentin Berthet, and Yaniv Plan. Average-case hardness of RIP certification. *Advances in Neural Information Processing Systems*, 29:3819–3827, 2016.

[82] Feng Ying Yu, Zhi Hao Yang, Xiao Hua Hu, Yuan Yuan Sun, Hong Fei Lin, and Jian Wang. Protein complex detection in PPI networks based on data integration and supervised learning method. *BMC Bioinformatics*, 16(12):S3, 2015.

[83] Feng Ying Yu, Zhi Hao Yang, Nan Tang, Hong Fei Lin, Jian Wang, and Zhi Wei Yang. Predicting protein complex in protein interaction network - a supervised learning based method. *BMC Systems Biology*, 8(S3):S4, 2014.

[84] Yang Yu, Xiaolong Wang, Lei Lin, Chengjie Sun, and Xuan Wang. A supervised approach to detect protein complex by combining biological and topological properties. *International Journal of Data Mining and Bioinformatics*, 8(1):105–121, 2013.

[85] Yuan Zhang, Elizaveta Levina, and Ji Zhu. Detecting overlapping communities in networks using spectral methods. *SIAM Journal on Mathematics of Data Science*, 2(2):265–283, 2020.