Learning Guarantee of SDP Relaxation on Directed Stochastic Block Models

Name:	Yanlang Chen
Province:	Guangdong
Country:	China
Supervisor's Name:	Jianzhong Chen

Learning Guarantee of SDP Relaxation on Directed Stochastic Block Models

Yanlang Chen

October 25, 2023

Abstract

Community detection in networks has been a focal point in various scientific domains, but the study of directed networks remains relatively under-explored despite their prevalence and importance in capturing real-world systems. This work addresses this research gap by focusing on Directed Stochastic Block Models (DSBMs), a natural extension of traditional Stochastic Block Models (SBMs) to directed graphs. The inherent complexity of directionality in DSBMs makes them challenging to analyze, requiring new mathematical frameworks and computational approaches. We introduce an augmented matrix to encapsulate the directional relationships within these networks, providing a nuanced perspective for further analysis. In this work, we prove the information-theoretical threshold for exact recovery in the DSBMs and propose an SDP relaxation that can achieve this threshold, thereby contributing to the theoretical understanding of community detection in the realm of directed graphs.

Keywords: Directed Stochastic Block Models, Semi-Definite Programming(SDP)Relaxation, clustering, random graph, unsupervised learning, community detection, directed graphs.

1 Introduction

Community detection and clustering are pivotal challenges in an array of disciplines, ranging from machine learning and data science to the study of complex networks [5], [7]. One of the most striking features of any network is its unique structure, which becomes evident through the patterns of interaction among its vertices. For instance, certain subsets of vertices in a vast network are tightly interlinked, while their connections to vertices outside this cluster are notably sparse. While substantial research has focused on undirected networks—such as geographical maps, friendship circles, and familial connections—there is a compelling yet under-explored frontier in the realm of directed networks. Directed networks, evident in phenomena like social media interactions, web page hyperlinks, and aviation routes, more closely mimic the intricacies of real-world systems. Their inherent directionality not only makes them more relevant for practical applications but also significantly more challenging to analyze. This very novelty and complexity of directed networks serve as the driving force behind this thesis.

Stochastic Block Models (SBMs) [2] have traditionally been employed to examine random block structures, originally formulated to scrutinize social networks. This model serves as a powerful benchmark for assessing the performance of clustering algorithms. However, its main limitation lies in its oversimplification of real-world networks, particularly due to its strong homogeneity and lack of community structure. Moreover, the burgeoning research in this area has disproportionately focused on undirected SBMs [6] [4], thus leaving a crucial gap in our understanding of Directed Stochastic Block Models (DSBMs).

The challenge in DSBMs is not merely a replication of its undirected counterparts; it is profoundly exacerbated by the added complexity of directionality. In light of this, we leverage an augmented matrix to encapsulate these directional relationships, providing a nuanced mathematical framework to navigate this intricate landscape.

DSBMs also possess desirable consistency properties similar to undirected SBMs, but obtaining exact parameter estimates in both is generally an NP-hard problem. Inspired by semi-definite programming (SDP) relaxation techniques [3], our work aims to bypass this computational bottleneck. We offer a pioneering semi-definite relaxation approach to discern clustering thresholds in DSBMs, thereby overcoming the inherent NP-hardness.

In this work, we prove the information-theoretical threshold for exact recovery in the Directed Stochastic Block Models and propose an SDP relaxation that can achieve the threshold, which fills the gap of the theoretical understanding of community detection in the context of directed graphs.

1.1 Notations

Let $\boldsymbol{A} \in \mathbb{C}^{n\times m}$ be a complex matrix and denote its (i,j)-entry by A_{ij}. We denote its transpose and conjugate transpose as $\boldsymbol{A}^\top$ and $\boldsymbol{A}^H$ respectively. The ℓ_2-norm of a vector $\boldsymbol{v}$ is denoted by $\|\boldsymbol{v}\| = \sqrt{\boldsymbol{v}^H\boldsymbol{v}} = \sqrt{\sum_{j=1}^n |v_j|^2}$ and its ℓ_∞ norm is denoted by $\|\boldsymbol{v}\|_\infty = \max_{1\le k\le n}|v_k|$, where v_k is $\boldsymbol{v}$'s k-th entry. The inner product between two complex vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as $\langle \boldsymbol{u}, \boldsymbol{v}\rangle = \boldsymbol{u}^H\boldsymbol{v}$. For two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, we denote $\boldsymbol{u} \propto \boldsymbol{v}$ if they are parallel. We denote the operator 2-norm of $\boldsymbol{A}$ as $\|\boldsymbol{A}\|$ which is the largest singular value of $\boldsymbol{A}$. We denote the all-one vector in $\mathbb{R}^n$ as $\mathbf{1}_n$ and the all-one matrix in $\mathbb{R}^{n\times n}$ as $\boldsymbol{J}_n$.

For $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, if $\boldsymbol{A}$ is symmetric and all its eigenvalues are non-negative, we say $\boldsymbol{A}$ is positive semidefinite, denoted by $\boldsymbol{A} \succcurlyeq 0$.

2 Preliminaries

In this section, we will introduce the problem settings and the core definitions for the paper.

2.1 Directed Stochastic Block Models

The Directed Stochastic Block Models (or DSBM in short) is a generative model for modeling the community structures in directed networks, which is a benchmark for comparing different community detection methods. First, we define the DSBM as follows. Given an even integer $n \ge 2$, and $1 \ge p > q \ge 0$, we say that a directed random graph G is drawn from the Directed Stochastic Block Model with two communities (denoted as $\mathrm{DSBM}(n,p,q)$) with ground-truth $\boldsymbol{g}$, if G has n nodes, divided into two clusters of $n/2$ nodes each, and for each pair of vertices (i,j), (i,j) is an edge of G with probability p if i and j are in the same cluster and with probability q otherwise. The i-th entry of $\boldsymbol{g}$ is ±1 indicating the cluster to which the i-th node belongs. In particular, let $\boldsymbol{A}$ be the adjacency matrix of G. Each entry of $\boldsymbol{A}$ is given by

$$\mathbb{P}(\boldsymbol{A}_{ij} = 1) = \begin{cases} p & \text{if } i \text{ and } j \text{ are in the same community} \\ q & \text{otherwise} \end{cases} \qquad 1 \le i \le n, 1 \le j \le n.$$

Then the expected adjacency matrix $\boldsymbol{A}^* = \mathbb{E}\boldsymbol{A}$ is given by

$$\boldsymbol{A}^* = \mathbb{E}\boldsymbol{A} = \begin{bmatrix} p\boldsymbol{J}_{n/2\times n/2} & q\boldsymbol{J}_{n/2\times n/2} \\ q\boldsymbol{J}_{n/2\times n/2} & p\boldsymbol{J}_{n/2\times n/2} \end{bmatrix}.$$

This model can be seen as the concatenation of two directed Erdős-Rényi random graphs with parameter p (as two clusters) and the connection probability between these two graphs is q. To theoretically understand community detection in directed networks, we are interested in the information-theoretical threshold for exact recovery in DSBM, that is, we want to find a threshold as a function of (n,p,q), above which exactly recovering the membership of each node is possible with probability $1 - o(1)$, while impossible otherwise.

The definition of exact recovery is stated as follows. Let $\mathrm{Algo}(\cdot)$ be some community detection algorithm, and $\boldsymbol{A}$ be the adjacency matrix of $G \sim \mathrm{DSBM}(n,p,q)$ with ground-truth membership $\boldsymbol{g}$. Then we say $\mathrm{Algo}(\cdot)$ exactly recovers the membership if $\mathrm{Algo}(\boldsymbol{A}) = \boldsymbol{x} = \boldsymbol{g}$

where $\boldsymbol{x}$ is the membership estimated by Algo($\cdot$). Since it can be verified that connectedness is a sufficient condition for exact recovery in DSBM, we will choose the regime $p = a\log(n)/n$, $q = b\log(n)/n$ in the whole paper. In this work, we will propose the threshold of exact recovery for DSBM(n, a, b) and a clustering algorithm that can exactly recover the membership, and then prove the tightness of the threshold.

2.2 Co-clustering: community detection in directed networks

Co-clustering was a concept first proposed in 1972, where it clusters entries of a matrix $M \in \mathbb{R}^{n\times d}$. In the past, co-clustering has been applied to matrices where the rows and columns represent different meanings, and it clusters rows of matrix M into k_r communities, and columns into k_c communities. For example, in a matrix used for text processing, the rows represent documents, and the columns represent words. Therefore, each entry in (i, j) indicates how many time word j appears in document i.

However, in this thesis, we apply co-clustering to a matrix where the rows and columns index the same set of vertex. Specifically, the ith row of the matrix represents the connection of the ith vertex, where it shows the outgoing edges for vertex i. The ith column of the matrix represents the incoming edges of ith vertex. Therefore, each vertex i is included in two communities, one for the row and one for the column. It is worth mentioning that due to the directedness of the DSBM, the connectedness of the row community and the column community of ith vertex is not necessarily the same.

Specifically, we would like to find the Maximum Likelihood Estimation (MLE)to the communities in the DSBM. We assume u and v to be n by 1 matrix, where the first $\frac{n}{2}$ entries are 1, and others are -1. We have $\boldsymbol{A}$ as our adjacency matrix, which reflects the realistic connection behavior. Then we have the following equation:

$$\begin{aligned} \max \quad & \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v} \\ \text{s.t} \quad & u_i = \pm 1, 1 \le i \le n, \\ & v_i = \pm 1, 1 \le i \le n. \end{aligned} \tag{2.1}$$

in this multiplication, vertices within the same community would give a positive value, and we try to maximize the value for all vertices. However, this still remains a very challenging problem to tackle as the conditions remain discrete. Therefore, we need to use semi-definite programming (SDP) algorithm to loosen the conditions and find the solution under that condition, and lastly check whether the solution would work under the initial condition. The detailed process will be further explained in the following section.

2.3 SDP relaxation for co-clustering

The programming (2.1) is indeed finding the maximum likelihood estimation to the membership of the nodes, but it is challenging due to NP-hardness. We can simplify the algorithm (2.1) into:

$$\begin{aligned} \max \quad & \mathrm{Tr}\Big(\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v}\Big) \\ \text{s.t} \quad & u_i = \pm 1, 1 \le i \le n, \\ & v_i = \pm 1, 1 \le i \le n. \end{aligned} \tag{2.2}$$

However, finding the row membership vector $\boldsymbol{u}$ and column membership vector $\boldsymbol{v}$ from this programming is NP-hard due to the following reasons: (1) $\boldsymbol{A}$ is asymmetric, so there are limited linear algebra algorithms that can be used; (2) the problem is nonconvex; (3) there are limited prior knowledge about this model and there are no constraints in the model in (2.2). In order to tackle the third reason, we know that as defined, $\boldsymbol{u}$ and $\boldsymbol{v}$ are perpendicular to $\mathbf{1}$ matrix, so their dot product would equal 0. Therefore, we can penalize algorithm (2.2) with $u^T\mathbf{1}_n$ and

$v^T\mathbf{1}_n$ to the following function:

$$\begin{aligned} \max \quad & \mathrm{Tr}\Big(\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v}\Big) - \lambda(\langle \boldsymbol{u}, \mathbf{1}_n/\sqrt{n}\rangle + \langle \boldsymbol{v}, \mathbf{1}_n/\sqrt{n}\rangle) \\ \text{s.t} \quad & u_i = \pm 1, 1 \leq i \leq n, \\ & v_i = \pm 1, 1 \leq i \leq n. \end{aligned} \tag{2.3}$$

Another key characteristic of DSBM is its directedness, and in order to deal with the directness of G, we need to consider the symmetric augmented matrix defined below to represent the direction:

$$\widetilde{\boldsymbol{A}} = \begin{bmatrix} \mathbf{0}_{n\times n} & \boldsymbol{A}^\top \\ \boldsymbol{A} & \mathbf{0}_{n\times n} \end{bmatrix}.$$

Given the singular value decomposition (SVD) of $\boldsymbol{A}$ being

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top,$$

then the eigen-decomposition of $\widetilde{\boldsymbol{A}}$ can be formulated as

$$\widetilde{\boldsymbol{A}} = \frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{V} & \boldsymbol{V} \\ \boldsymbol{U} & -\boldsymbol{U} \end{bmatrix}\begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & -\boldsymbol{\Sigma} \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{V}^\top & \boldsymbol{U}^\top \\ \boldsymbol{V}^\top & -\boldsymbol{U}^\top \end{bmatrix}.$$

It is worth noting that conditions for algorithm (2.3) is discrete, where the algorithm is unsolvable in polynomial time. Hence, we need to loosen the constraints using SDP, converting them into semi-definite constraints that would be solvable in polynomial time. Using the well-known Goemans-Williams relaxation, we can formulate semi-definite programming to solve the NP-hardness:

$$\begin{aligned} \max \quad & \langle \widetilde{\boldsymbol{A}}, \boldsymbol{X}\rangle - \lambda\langle \boldsymbol{J}_{2n}, \boldsymbol{X}\rangle \\ \text{s.t.} \quad & X_{ii} = 1, 1 \leq i \leq 2n \\ & \boldsymbol{X} \succcurlyeq 0 \\ & \lambda > 0 \end{aligned} \tag{2.4}$$

(2.4) is a convex relaxation of (2.3). In order to recover the communities in the graph, we intend to maximize the difference between the in-community degree and the cross-community degree in rows and columns respectively. However, we don't want $\boldsymbol{u}$ and $\boldsymbol{v}$ to be too close to all-one vector or all-negative one vector. So we will take $\lambda = \frac{1}{2}$, and (2.4) becomes:

$$\begin{aligned} \max \quad & \mathrm{Tr}\Big((2\widetilde{\boldsymbol{A}} - \boldsymbol{J}_{2n})\boldsymbol{X}\Big) \\ \text{s.t.} \quad & X_{ii} = 1 \\ & \boldsymbol{X} \succcurlyeq 0 \end{aligned} \tag{2.5}$$

Note that

$$\begin{aligned} 2\widetilde{\boldsymbol{A}} - \boldsymbol{J}_{2n} &= \begin{bmatrix} -\boldsymbol{J}_n & \boldsymbol{B}^T \\ \boldsymbol{B} & -\boldsymbol{J}_n \end{bmatrix} \\ B_{ij} &= \begin{cases} 1 & \text{if } A_{ij} = 1 \\ -1 & \text{if } A_{ij} = 0 \end{cases}. \end{aligned} \tag{2.6}$$

3 Main Results

Given the the DSBM(n, p, q) defined in Section 2.1, in the regime $p = a\log(n)/n$, $q = b\log(n)/n$, we will present the main argument that $\sqrt{a} - \sqrt{b} = \sqrt{2}$ is the information-theoretical threshold for exact recovery in the DSBM. To be more specific, the argument will be presented from two perspectives, namely the impossibility part and the achievability part. In the impossibility part, we will show that when $\sqrt{a}-\sqrt{b} < \sqrt{2}$, even the MLE fails to recover the correct membership of each node. In the achievability part, we will show that when $\sqrt{a}-\sqrt{b} > \sqrt{2}$ the SDP relaxation can correctly recover the membership of each node with high probability. We only provide a proof sketch in this section and the detailed proofs are deferred to Section A and B.

3.1 Impossibility

Our goal in this section is to provide a proof sketch of the condition for which the MLE algorithm fails to recover the communities in DSBM, we first introduce the concept of bad vertices which is defined with respect to the connectedness of each node. Then using this concept of bad vertices, we will find under what condition this bad vertex definitely exists. Therefore, when this condition is met, the MLE will fail.

Theorem 3.1. *Let G be a graph drawn from DSBM(n, p, q), let $p = \frac{a\log(n)}{n}$ and $q = \frac{b\log(n)}{n}$, then exact recover is impossible if*

$$\sqrt{a} - \sqrt{b} < \sqrt{2}. \tag{3.1}$$

The following steps are the proof sketch to the above main theorem.

Definition 3.1. *We define the likelihood function of the DSBM as:*

$$L(x, y) = \prod_{i,j \in [n]^2} P_{i,j}^{A_{i,j}} (1 - P_{ij})^{1 - A_{i,j}} \tag{3.2}$$

where $\boldsymbol{A}$ is the adjacency matrix and

$$\boldsymbol{P} = \begin{bmatrix} p\boldsymbol{J}_{n/2} & q\boldsymbol{J}_{n/2} \\ q\boldsymbol{J}_{n/2} & p\boldsymbol{J}_{n/2} \end{bmatrix}.$$

Definition 3.2. *We define the degree matrices for DSBM as the following:*

$$\begin{aligned}
(\boldsymbol{D}_R^+)_{ii} &:= \begin{cases} \sum_{j=1}^{n/2} A_{ij} & i \in [1, \frac{n}{2}] \\ \sum_{j=n/2+1}^{n} A_{ij} & i \in [\frac{n}{2}+1, n] \end{cases} \\
(\boldsymbol{D}_R^-)_{ii} &:= \begin{cases} \sum_{j=n/2+1}^{n} A_{ij} & i \in [1, \frac{n}{2}] \\ \sum_{j=1}^{\frac{n}{2}} A_{ij} & i \in [\frac{n}{2}+1, n] \end{cases} \\
(\boldsymbol{D}_C^+)_{ii} &:= \begin{cases} \sum_{i=1}^{n/2} A_{ij} & i \in [1, \frac{n}{2}] \\ \sum_{i=\frac{n}{2}+1}^{n} A_{ij} & i \in [n/2+1, n] \end{cases} \\
(\boldsymbol{D}_C^-)_{ii} &:= \begin{cases} \sum_{i=\frac{n}{2}+1}^{n} A_{ij} & i \in [1, \frac{n}{2}] \\ \sum_{i=1}^{\frac{n}{2}} A_{ij} & i \in [\frac{n}{2}+1, n] \end{cases}
\end{aligned} \tag{3.3}$$

where each entry represents the number of connections that satisfies the condition described above. Therefore, for each vertex's connectedness, Then the in-degree matrix and the out-degree matrix for DSBM respectively as:

$$\boldsymbol{D}^+ := \begin{bmatrix} \boldsymbol{D}_C^+ & 0 \\ 0 & \boldsymbol{D}_R^+ \end{bmatrix} \quad \boldsymbol{D}^- := \begin{bmatrix} \boldsymbol{D}_C^- & 0 \\ 0 & \boldsymbol{D}_R^- \end{bmatrix}$$

For each vertex's connectedness, we can use $d(i)$ to represent the ith vertex's degree. Then we have $d_-(i)$ to represent the degree with cross-community vertex, $d_+(i)$ to represent the degree with the same community. $d^R(i)$ to represent the degree with the row, and $d^C(i)$ to represent the degree with the column.

The concept of bad vertex and bad edges is essential in the proof of the impossibility part. It can be verified that the presence of bad edges and bad vertices implies the impossibility of exact recovery and the condition $\sqrt{a} - \sqrt{b} < \sqrt{2}$ is the sufficient condition for the existence of bad vertices.

Definition 3.3. *Bad vertices is a type of vertices pair, where the two vertices' community in the pair are swapped, and the MLE of the swapped pair is larger than the MLE of the initial pair. Meaning that after the swap, vertices are connected better with their original community.*

Mathematically, we define a pair of bad vertices in rows, in columns, or in rows and columns respectively by:

$$\begin{aligned} B^R(G) &:= \big\{(u,v) : u \in C_1^R, v \in C_2^R, L(\widetilde{x},y) > L(x,y)\big\} \\ B^C(G) &:= \big\{(u,v) : u \in C_1^C, v \in C_2^C, L(x,\widetilde{y}) > L(x,y)\big\} \\ B^{R,C}(G) &:= \{(u,v) : u \in C_1, v \in C_2, L(\widetilde{x},\widetilde{y}) > L(x,y)\} \end{aligned} \tag{3.4}$$

Then we are trying to prove that for $\sqrt{a} - \sqrt{b} < \sqrt{2}$, there exists at least one bad vertex in rows or columns, which would result in $\max\{L(\widetilde{x},y), L(x,\widetilde{y}), L(\widetilde{x},\widetilde{y}) \geq L(x,y)\}$. We define a pair of bad vertex (u,v), the relationship between degrees can be inferred from the following relationship between MLE:

$$\begin{aligned} &L(\widetilde{x},y) > L(x,y) \to d_-^R(u) + d_-^R(v) > d_+^R(u) + d_+^R(v) \\ &L(x,\widetilde{y}) > L(x,y) \to d_-^C(u) + d_-^C(v) > d_+^C(u) + d_+^C(v) \\ &L(\widetilde{x},\widetilde{y}) > L(x,y) \to d_-^R(u) + d_-^R(v) + d_-^C(u) + d_-^C(v) > d_+^R(u) + d_+^R(v) + d_+^C(u) + d_+^C(v) \end{aligned} \tag{3.5}$$

Since if $L(\widetilde{x},y) > L(x,y)$, then $L(x,\widetilde{y}) > L(x,y) \text{or} L(\widetilde{x},\widetilde{y}) > L(x,y)$. Hence, it is enough to only study the bad vertices in rows or in columns.

Definition 3.4. *Using the concept of degree, we can define a set of bad vertices in rows:*

$$B_i^R(G) = \big\{\exists u : u \in C_i^R, d_+^R(u) \leq d_-^R(u) - 1\big\}, i = 1,2 \tag{3.6}$$

where i represents the community.

Lemma 3.2. *If $B_1^R(G)$ is non-empty and with high probability, then $B^R(G)$ is non-empty and with non-vanishing probability.*

3.2 Acheivability

Recall that our goal is to show that $\sqrt{a} - \sqrt{b} = \sqrt{2}$ is the tight threshold for exact recovery. After showing that when $\sqrt{a} - \sqrt{b} < \sqrt{2}$ MLE fails to recover the communities, we will show that the SDP relaxation (2.5) can recover the communities otherwise.

Theorem 3.3. *Let G be a graph drawn from DSBM(n,p,q), let $p = \frac{a\log(n)}{n}$ and $q = \frac{b\log(n)}{n}$,if*

$$\sqrt{a} - \sqrt{b} > \sqrt{2}, \tag{3.7}$$

then the SDP relaxation in (2.5) can recover the communities with high probability.

Without loss of generality, we suppose that the ground-truth community indicator, denoted by $\boldsymbol{g}$ is $(\mathbf{1}_{n/2}^\top, -\mathbf{1}_{n/2}^\top, \mathbf{1}_{n/2}^\top, -\mathbf{1}_{n/2}^\top)^\top$. Since we use the notion of co-clustering introduced Section 2.2, $\boldsymbol{g}$ is the stack of the column communities and the row communities. Then we claim that when $\sqrt{a} - \sqrt{b} > \sqrt{2}$, (2.5) has a unique solution equal to $\boldsymbol{g}\boldsymbol{g}^\top$. The following lemma uses a surrogate $\boldsymbol{\Lambda}$ to quantify the condition under which $\boldsymbol{g}\boldsymbol{g}^\top$ is the unique solution.

Definition 3.5. *Given a graph G drawn from DSBM with two clusters, we can have:*

$$\Gamma_{\text{DSBM}} = \boldsymbol{D}^+ - \boldsymbol{D}^- - \boldsymbol{A}^* \tag{3.8}$$

Lemma 3.4. *Let $\boldsymbol{\Lambda} = 2\Gamma_{\text{DSBM}} + \boldsymbol{I}_{2n} + \begin{bmatrix} 2\boldsymbol{J}_n & \boldsymbol{J}_n - \boldsymbol{I}_n \\ \boldsymbol{J}_n - \boldsymbol{I}_n & 2\boldsymbol{J}_n \end{bmatrix}$, if*

$$\boldsymbol{\Lambda} \succcurlyeq 0, \quad \lambda_2(\boldsymbol{\Lambda}) > 0.$$

then $\boldsymbol{g}\boldsymbol{g}^\top$ is the unique solution to the SDP relaxation (2.5).

Since $\boldsymbol{\Lambda}$ is random, the second smallest eigenvalue of it is hard to calculate. However, thanks to Weyl's inequality, we can estimate it using its population counterpart, i.e., the second smallest eigenvalue of $\mathbb{E}\,\boldsymbol{\Lambda}$. The following lemma specifies the distance between the two quantities.

Lemma 3.5. *Let $n > 4$ be even and*

$$asqwz\lambda_{\max}(-\Gamma_{SBM} + \mathbb{E}[\Gamma_{SBM}]) < n(p - q), \tag{3.9}$$

then the SDP relaxation for DSBM can achieve exact recovery, meaning that $gg^\top$ is the only solution.

The following theorem finalizes the proof in this part by connecting the degree and community detection in DSBM.

Theorem 3.6. *Let $n \geq 4$ be even and $\boldsymbol{G}$ be a graph of directed stochastic block model drawn from $\mathcal{G}(n, p, q)$, where $p > q$. Only when $\frac{\log(n)}{n} < q < p < \frac{1}{2}$, and for some constant $c > 1$, then $\boldsymbol{\Delta} > 0$ such that, with high probability, the following equation holds: If*

$$\min_{i \in [2n]} (\mathcal{D}^+_{ii} - \mathcal{D}^-_{ii}) \geq \frac{\boldsymbol{\Delta}}{\log(n)} \mathbb{E}\left[\deg^+_C(i) - \deg^-_C(i)\right] \tag{3.10}$$

then the semidefinite program achieve exact recovery. Now we have the equation represented by degree, which is easier and more straightforward to solve than the previous lemma.

Lemma 3.7. *Let G be a random graph with n node drawn accordingly to the directed stochastic block model on two communities with in-community edge probability p and cross-community edge probability q. Let $p = alog(n)/n$ and $q = blog(n)/n$, where $a > b$ are constant. Then for any constant $\boldsymbol{\Delta} > 0$:*
If

$$\sqrt{a} - \sqrt{b} > \sqrt{2} \tag{3.11}$$

then with high probability

$$\min_{i \in [2n]} (\mathcal{D}^+_{ii} - \mathcal{D}^-_{ii}) \geq \frac{\boldsymbol{\Delta}}{\log(n)} \mathbb{E}\left[\deg^+_C(i) - \deg^-_C(i)\right]. \tag{3.12}$$

4 Experiments

The goal of the numerical experiments is to confirm our theoretical results above. Let $n = 100$. For each (a, b) pair we generate the DSBM 25 times and apply the SDP relaxation to recover the community. Figure 1 visualizes the accuracy v.s. (a, b). The accuracy is calculated as follows

$$\frac{1}{2n} \sum_{i=1}^{n} \mathbf{1}_{\{g_i = x_i\}}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function and $\boldsymbol{g}$ is the ground-truth and $\boldsymbol{x}$ is the solution of SDP relaxation. As depicted in Figure 1, the empirical boundary between the success region and the failure region almost aligns with the curve $\sqrt{a} - \sqrt{b} = \sqrt{2}$, which suggests that our proved information-theoretical threshold is tight.

Acknowledgement

I would like to express my gratitude to my mentor, Mr. Chen, who serves as my mathematics instructor at Shenzhen College of International Education. He generously offered his guidance in helping me navigate the research process, fostering my curiosity to delve deeper into the realm of linear algebra, and ultimately steering me toward the captivating subject of clustering.

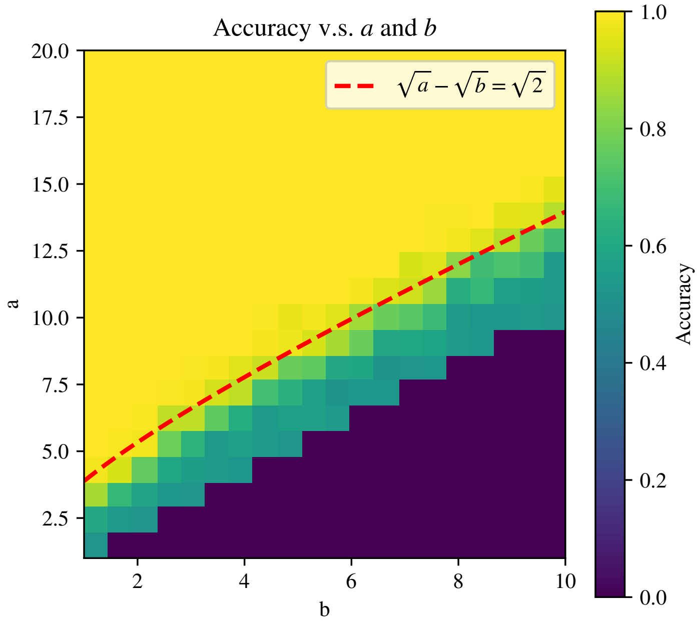

Figure 1: Accuracy of the SDP relaxation under different a and b's. Each (a, b) pair is subject to 25 experiments.

A Proof for Theorem 3.1

In this section, we are trying to discover the condition when MLE can not fully recover the communities in DSBM and the condition's proof. Let G be a graph drawn from $\mathcal{G}(n,p,q)$, let $p=\frac{a\log(n)}{n}$ and $q=\frac{b\log(n)}{n}$, $a>b$, if $\sqrt{a}-\sqrt{b}<\sqrt{2}$, then we need to prove that for this condition the exact recovery of DSBM is unsolvable, and therefore MLE fails. We define MLE for DSBM to be:

$$L(x,y)=\prod_{i,j}P_{i,j}^{A_{i,j}}(1-P_{ij})^{1-A_{i,j}} \tag{A.1}$$

Definition A.1. *We define a pair of bad vertices in rows, in columns, or in rows and columns respectively by:*

$$\begin{aligned} B^R(G) &:= \left\{(u,v): u\in C_1^R, v\in C_2^R, L(\widetilde{x},y)>L(x,y)\right\} \\ B^C(G) &:= \left\{(u,v): u\in C_1^C, v\in C_2^C, L(x,\widetilde{y})>L(x,y)\right\} \\ B^{R,C}(G) &:= \{(u,v): u\in C_1, v\in C_2, L(\widetilde{x},\widetilde{y})>L(x,y)\} \end{aligned} \tag{A.2}$$

Then we are trying to prove that for $\sqrt{a}-\sqrt{b}<\sqrt{2}$, there exists at least one bad vertex in rows or columns, which would result in $\max\left\{L(\widetilde{x},y),L(x,\widetilde{y}),L(\widetilde{x},\widetilde{y})\geq L(x,y)\right\}$. We define a pair of bad vertex (u,v), the relationship between degrees can be inferred from the following relationship between MLE:

$$\begin{aligned} &L(\widetilde{x},y)>L(x,y)\to d_-^R(u)+d_-^R(v)>d_+^R(u)+d_+^R(v) \\ &L(x,\widetilde{y})>L(x,y)\to d_-^C(u)+d_-^C(v)>d_+^C(u)+d_+^C(v) \\ &L(\widetilde{x},\widetilde{y})>L(x,y)\to d_-^R(u)+d_-^R(v)+d_-^C(u)+d_-^C(v)>d_+^R(u)+d_+^R(v)+d_+^C(u)+d_+^C(v) \end{aligned} \tag{A.3}$$

Since if $L(\widetilde{x},y)>L(x,y)$, then $L(x,\widetilde{y})>L(x,y)\text{or}L(\widetilde{x},\widetilde{y})>L(x,y)$. Hence, it is enough to only study the bad vertices in rows or in columns.

Definition A.2. *We define a set of bad vertices in rows:*

$$B_i^R(G)=\left\{\exists u: u\in C_i^R, d_+^R(u)\leq d_-^R(u)-1\right\}, i=1,2 \tag{A.4}$$

where i represents the community.

Lemma A.1. *If $B_1^R(G)$ is non-empty and with high probability, then $B^R(G)$ is non-empty and with non-vanishing probability.*

Proof. *If $u \in C_1^R$ and $v \in C_2^R$ such that $d_+^R(u) \leq d_-^R(u) - 1$ and $d_+^R(v) \leq d_-^R(v) - 1$, then combining these we get $d_-^R(u) + d_-^R(v) > d_+^R(u) + d_+^R(v)$. Then we have:*

$$\begin{aligned}
&\mathbb{P}(\exists u \in B^R or\ \exists v \in B^R) = \mathbb{P}(\exists u \in B_1^R(G)) + \mathbb{P}(\exists v \in B_1^R(G)) - \mathbb{P}(\exists (u,v) \in B^R(G)) \\
&\mathbb{P}(\exists (u,v) \in B^R(G)) = \mathbb{P}(\exists u \in B_1^R(G)) + \mathbb{P}(\exists v \in B_1^R(G)) - \mathbb{P}(\exists u \in B^R or\ \exists v \in B^R)
\end{aligned} \tag{A.5}$$

because the possibility of node u is a bad vertex is the same as node v is a bad vertex, so we can write:

$$\mathbb{P}(\exists (u,v) \in B^R(G)) \leq 2\mathbb{P}(\exists u \in B_1^R(G)) - 1 \tag{A.6}$$

Lemma A.2. *Let G be a graph drawn from $\mathcal{G}(n,p,q)$, let $p = \frac{a\log(n)}{n}$ and $q = \frac{b\log(n)}{n}$, $a > b$, if $\sqrt{a} - \sqrt{b} < \sqrt{2}$, then:*

$$\mathbb{P}(\exists u \in B_1^R(G)) = 1 - o(1) \tag{A.7}$$

Proof. *Note that $\mathbb{P}(\exists u \in B_1^R(G))$ can be written as:*

$$\mathbb{P}(\exists u \in B_1^R(G)) = n\mathbb{P}(d_-^R > d_+^R) = n\mathbb{P}(Bin(\frac{n}{2}, q) > Bin(\frac{n}{2}, p)) \tag{A.8}$$

Then, we introduce a new definition.

Definition A.3. *Let m be a natural number, $p, q \in [0,1]$, and $\delta \in \mathbb{R}$, we define*

$$T(m,p,q,\delta) = \mathbb{P}[\sum_{i=1}^{m}(Z_i - W_i \geq \delta)] \tag{A.9}$$

where $W_1, ..., W_m$ are i.i.d. Bernoulli(p) and $Z_1, ..., Z_m$ are i.i.d. Bernoulli(q), independent of $W_1, ..., W_m$.

Definition A.4. *We define:*

$$\begin{aligned}
&V(m,p,q,t,c) \\
&= \binom{m}{(t+c)\frac{m}{n}\log(n)}\binom{m}{t\frac{m}{n}\log(n)} p^{t\frac{m}{n}\log(n)} q^{(t+c)\frac{m}{n}\log(n)} (1-p)^{m-t\frac{m}{n}\log(n)} (1-q)^{(t+c)\frac{m}{n}\log(n)}
\end{aligned} \tag{A.10}$$

Where $c = O(1)$. We also define the function:

$$g(a,b,c) = (a+b) - c\log(b) - 2\sqrt{(\frac{c}{2})^2 + ab} + \frac{c}{2}\log\left(ab\frac{\sqrt{(\frac{c}{2})^2 + ab} + \frac{c}{2}}{\sqrt{(\frac{c}{2})^2 + ab} - \frac{c}{2}}\right) \tag{A.11}$$

Then we have the following results for $T^*(m,p,q,c) = max_{t>0}V(m,p,q,t,c)$: [1]
For $m \in \mathbb{N}$ and $\forall t > 0$:

$$-\log(T^*(m,p,q,c)) \geq \frac{m}{n}\log(n) * g(m,n,c) - o(\frac{m}{n}\log(n)) \forall m \in \mathbb{N} \tag{A.12}$$

In the following proof of this lemma, we will omit the ceiling symbol for clarity. In the case of the directed stochastic block model, recall definition A.3, we have:

$$T(m,p,q,0) = \mathbb{P}[Z - W \geq 0)] \tag{A.13}$$

where Z is a Binomial(m,q) and W is a Binomial (m,p), $p = \frac{a\log(n)}{n}, q = \frac{b\log(n)}{n}$. We can re-write (A.10) into:

$$T(m,p,q,0) = \sum_{k_1=0}^{m}(\sum_{k_2=k_1}^{m}\mathbb{P}(Z = k_2))\mathbb{P}(W = k_1) \tag{A.14}$$

Where each term in the double summation can be upper-bounded by $T^*(m,p,q,0)$. Using $c=0$, we have

$$\begin{aligned} T(m,p,q,0) &\leq m^2 T^*(m,p,q,0) \\ -\log(T(m,p,q,0)) &\geq -2\log(m) - \log(T^*(m,p,q,0)) \\ &\geq -2\log(m) + \frac{2m}{n}(\frac{a+b}{2} - \sqrt{ab})\log(n) \end{aligned} \tag{A.15}$$

As long as $\frac{m}{n} > \log\log(n)$ and $m \leq \frac{n^2}{4}$, then we have $\log(m) = o(\frac{m}{n}\log(n))$. Hence,

$$-\log(T(m,p,q,0)) \geq \frac{2m}{n}(\frac{a+b}{2} - \sqrt{ab})\log(n) - o(\frac{m}{n}\log(n)) \tag{A.16}$$

In this case, $T(m,p,q,0)$ is equivalent to $\mathbb{P}(Bin(\frac{n}{2},q) > Bin(\frac{n}{2},p))$, and continuing (A.8), as n approaches infinity, we get:

$$\mathbb{P}(\exists u \in B_1^R(G)) = n\mathbb{P}(d_-^R > d_+^R) = n\mathbb{P}(Bin(\frac{n}{2},q) > Bin(\frac{n}{2},p)) = n^{1-(\frac{\sqrt{a}-\sqrt{b}}{\sqrt{2}})^2 + o(1)} \tag{A.17}$$

Therefore, when $\sqrt{a} - \sqrt{b} < \sqrt{2}$, $\mathbb{P}(\exists u \in B_1^R(G)) = 1 - o(1)$, there exists a bad vertex and exact recovery is unachievable.

B Proof for Theorem 3.3

Proof for Lemma 3.4 Let $\boldsymbol{g} = (1, ..., 1, -1, ..., -1, 1, .., 1, -1, ..., -1)$. without loss of generality. By KKT condition, we obtain a sufficient condition for $\boldsymbol{g}\boldsymbol{g}^\top$ to be the solution of the SDP relaxation (2.5). Therefore, we have $\boldsymbol{\Lambda} \succcurlyeq 0$, and $\boldsymbol{g}\boldsymbol{g}^\top$ is guaranteed to be the optimal solution to SDP relaxation (2.5) if:

1. $\boldsymbol{g}\boldsymbol{g}^\top$ is a solution to the primal problem,
2. There exists a matrix $\boldsymbol{Y}$ feasible for the dual problem such that $Tr((2\boldsymbol{A}^* - \boldsymbol{J}_{2n})\boldsymbol{g}\boldsymbol{g}^\top) = Tr(\boldsymbol{Y})$.

The first condition is already satisfied by the given background, then we need to find a $\boldsymbol{Y}$ (also known as dual certificate) that would satisfy the second condition. We can use $\boldsymbol{C}$ to substitute $2\boldsymbol{A}^* - \boldsymbol{J}_{2n}$, then we have:

$$\begin{aligned} &(\boldsymbol{C}\boldsymbol{g}\boldsymbol{g}^\top)_{ii} = \text{correct edges + correct non-edges - incorrect edges - incorrect non-edges} \\ &= (\boldsymbol{D}_C^+)_{ii} + (\frac{n}{2} - (\boldsymbol{D}_C^-)_{ii}) - (\frac{n}{2} - 1 - (\boldsymbol{D}_C^+)_{ii}) - (\boldsymbol{D}_C^-)_{ii} + 1 \\ &= 2((\boldsymbol{D}_C^+)_{ii} - (\boldsymbol{D}_C^-)_{ii}) + 1 \end{aligned} \tag{B.1}$$

for $i \in [n+1, 2n]$, we let $j = i - n$, then we have:

$$(\boldsymbol{C}\boldsymbol{g}\boldsymbol{g}^\top)_{ii} = 2((\boldsymbol{D}_R^+)_{jj} - (\boldsymbol{D}_R^-)_{jj}) + 1 \tag{B.2}$$

Therefore, we get $Tr(\boldsymbol{C}\boldsymbol{g}\boldsymbol{g}^\top) = Tr(2(\boldsymbol{D}_C^+ - \boldsymbol{D}_C^-) + \boldsymbol{I}_n) + Tr(2(\boldsymbol{D}_R^+ - \boldsymbol{D}_R^-) + \boldsymbol{I}_n)$, and we are able to find a matrix $\boldsymbol{Y}$ that is feasible for the dual problem and satisfies the proposed condition:

$$\boldsymbol{Y} = \begin{bmatrix} 2(\boldsymbol{D}_C^+ - \boldsymbol{D}_C^-) + \boldsymbol{I}_n & 0 \\ 0 & 2(\boldsymbol{D}_R^+ - \boldsymbol{D}_R^-) + \boldsymbol{I}_n \end{bmatrix}.$$

As a result, if $\boldsymbol{\Lambda} \succcurlyeq 0$, then $\boldsymbol{g}\boldsymbol{g}^\top$ is the optimal solution to the SDP.
In addition, $\lambda_2(\boldsymbol{\Lambda}) > 0$ ensures that $\boldsymbol{g}\boldsymbol{g}^\top$ is the only solution to SDP. Imagine there is another optimal solution $\boldsymbol{X}^*$ to the SDP, then we get $\mathrm{Tr}\Big(\boldsymbol{X}'\boldsymbol{\Lambda}\Big) = 0$ by complementary slackness. By assumption, the second smallest eigenvalue of $\boldsymbol{\Lambda}$ is non-zero, together with the complementary slackness, the fact that $\boldsymbol{X}' \succcurlyeq 0$ and $\boldsymbol{\Lambda} \succcurlyeq 0$, we have $\boldsymbol{X}' = k\boldsymbol{g}\boldsymbol{g}^\top$. Since $\boldsymbol{X}'_{ii} = 1, \boldsymbol{X}' = \boldsymbol{g}\boldsymbol{g}^\top$ by contradiction.

Then we need to estimate $\mathbb{E}\,\boldsymbol{\Lambda}$, then we have the following:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\Lambda}] &= \mathbb{E}[2\Gamma_{SBM} + \boldsymbol{I}_{2n} + \begin{bmatrix} 0 & \boldsymbol{J}_n - \boldsymbol{I}_n \\ \boldsymbol{J}_n - \boldsymbol{I}_n & 0 \end{bmatrix} + 2\begin{bmatrix} \boldsymbol{J}_n & 0 \\ 0 & \boldsymbol{J}_n \end{bmatrix}] \\
&= 2(\frac{n}{2}(p-q)\boldsymbol{I}_{2n} - (\frac{p+q}{2}\begin{bmatrix} 0 & \boldsymbol{J}_n \\ \boldsymbol{J}_n & 0 \end{bmatrix} + \frac{p-q}{2}\boldsymbol{g}\boldsymbol{g}^\top)) \\
&+ \begin{bmatrix} 0 & \boldsymbol{J}_n \\ \boldsymbol{J}_n & 0 \end{bmatrix} + \boldsymbol{I}_{2n} - \begin{bmatrix} 0 & \boldsymbol{I}_n \\ \boldsymbol{I}_n & 0 \end{bmatrix} + (p-q)\begin{bmatrix} \boldsymbol{g}'\boldsymbol{g}'^\top & 0 \\ 0 & \boldsymbol{g}'\boldsymbol{g}'^\top \end{bmatrix} + 2\begin{bmatrix} \boldsymbol{J}_n & 0 \\ 0 & \boldsymbol{J}_n \end{bmatrix} \\
&= n(p-q)(\boldsymbol{I}_{2n} - \frac{\begin{bmatrix} 0 & \boldsymbol{g}'\boldsymbol{g}'^\top \\ \boldsymbol{g}'\boldsymbol{g}'^\top & 0 \end{bmatrix}}{n}) + (1-(p+q))\begin{bmatrix} 0 & \boldsymbol{J}_n \\ \boldsymbol{J}_n & 0 \end{bmatrix} + \boldsymbol{I}_{2n} - \begin{bmatrix} 0 & \boldsymbol{I}_n \\ \boldsymbol{I}_n & 0 \end{bmatrix} + 2\begin{bmatrix} \boldsymbol{J}_n & 0 \\ 0 & \boldsymbol{J}_n \end{bmatrix}
\end{aligned} \tag{B.3}$$

Suppose $p < \frac{1}{2}$ and $\lambda_2 = n(p-q)$, whose eigenvector is perpendicular to $(\boldsymbol{g}^{'}, \boldsymbol{g}^{'})^\top$and$(\mathbf{1}, \mathbf{1})^\top$ $\boldsymbol{\Delta}$ can be re-written as:

$$\boldsymbol{\Lambda} = 2\Gamma_{\text{DSBM}} + \boldsymbol{I}_{2n} + \begin{bmatrix} 0 & \boldsymbol{J}_n - \boldsymbol{I}_n \\ \boldsymbol{J}_n - \boldsymbol{I}_n & 0 \end{bmatrix} + 2\begin{bmatrix} \boldsymbol{J}_n & 0 \\ 0 & \boldsymbol{J}_n \end{bmatrix} \tag{B.4}$$

Using Weyl's inequalities, we get:

$$\begin{aligned}
\lambda_2 > \lambda_{max}(\mathbb{E}[\boldsymbol{\Lambda}] - \boldsymbol{\Lambda}) &= \|\mathbb{E}[\boldsymbol{\Lambda}] - \boldsymbol{\Lambda}\| \\
&\geq \sigma_2(\mathbb{E}[\boldsymbol{\Lambda}] - \sigma_2(\boldsymbol{\Lambda})) \\
&= \lambda_2(\mathbb{E}[\boldsymbol{\Lambda}] - \lambda_2(\boldsymbol{\Lambda})) \\
&\geq \lambda_2(\mathbb{E}[\boldsymbol{\Lambda}] - \lambda_2(\boldsymbol{\Lambda}))
\end{aligned} \tag{B.5}$$

This implies that $\boldsymbol{g}\boldsymbol{g}^\top$ is the unique solution to the semidefinite programming.

Proof for Theorem 3.6 . The method to approach the problem is by applying spectral approximation of random Laplacian matrix algorithm. However, Γ_{DSBM} is not a Laplacian matrix. Therefore, we will try to construct a Laplacian matrix $\Gamma^{'}_{\text{DSBM}}$ to help solve the problem. W.L.O.G. we let $\boldsymbol{g} = (\mathbf{1}_{n/2}, -\mathbf{1}_{n/2}, \mathbf{1}_{n/2}, -\mathbf{1}_{n/2})$, and we define:

$$\Gamma^{'}_{\text{DSBM}} = \text{diag}(\boldsymbol{g})\Gamma_{\text{DSBM}}\text{diag}(\boldsymbol{g}) \tag{B.6}$$

Both the eigenvalue and the diagonal elements of $\mathbb{E}[\Gamma^{'}_{\text{DSBM}}] - \Gamma^{'}_{\text{DSBM}}$ are the same as those of $\mathbb{E}\left[\Gamma_{\text{DSBM}}\right] - \Gamma_{\text{DSBM}}$. The off-diagonal entries of $\Gamma^{'}_{\text{DSBM}} = -\boldsymbol{A}_{ij}g_ig_i$. Then we apply spectral approximation of random Laplacian matrix algorithm, we let $\boldsymbol{L} = \mathbb{E}[\Gamma^{'}_{\text{DSBM}}] - \Gamma^{'}_{\text{DSBM}}$, where $\boldsymbol{L}$ has independent off-diagonal entries. Then we have:

$$\sum_{j\in[2n]/i} \mathbb{E}[\boldsymbol{L}_{ij}^2] = (\frac{n}{2}-1)p(1-p) + \frac{n}{2}q(1-q) \leq \frac{n}{2}*\frac{1}{4}(p+q) > \frac{n}{8}\frac{2\log n}{cn}(1-q)^2 = \frac{\log n}{4c}max_{i\neq j}||\boldsymbol{L}_{ij}||_\infty^2 \tag{B.7}$$

Where it exists a constant $\boldsymbol{\Delta}^{'}$ such that with high probability,

$$\lambda_{max}(\mathbb{E}[\Gamma^{'}_{\text{DSBM}}] - \Gamma^{'}_{DSBM}) \leq (1 + \frac{\boldsymbol{\Delta}^{'}}{\sqrt{\log n}})max_{i\in[2n]}[\mathbb{E}[(\Gamma^{'}_{\text{DSBM}})_{ii}] - (\Gamma^{'}_{\text{DSBM}})_{ii}] \tag{B.8}$$

and it is the same as the following:

$$\lambda_{max}(\mathbb{E}[\Gamma_{\text{DSBM}}] - \Gamma_{DSBM}) \leq (1 + \frac{\boldsymbol{\Delta}^{'}}{\sqrt{\log n}})max_{i\in[2n]}[\mathbb{E}[(\Gamma_{\text{DSBM}})_{ii}] - (\Gamma_{\text{DSBM}})_{ii}] \tag{B.9}$$

It is worth mentioning that

$$\begin{aligned}
min_{i\in[2n]}((\mathcal{D}^+)_{ii} - (\mathcal{D}^-)_{ii}) \geq (1 + \frac{\boldsymbol{\Delta}^{'}}{\sqrt{\log n}})max_{i\in[2n]}[\mathbb{E}[(\Gamma_{\text{DSBM}})_{ii}] - (\Gamma_{\text{DSBM}})_{ii}] \\
= (1 - \frac{\boldsymbol{\Delta}^{'}}{\sqrt{\log n}})(\frac{n}{2}(p-q) - p) \\
\geq max_{i\in[2n]}(\mathbb{E}[(\Gamma^{'}_{\text{DSBM}})_{ii}] - (\Gamma^{'}_{DSBM})_{ii}
\end{aligned} \tag{B.10}$$

Therefore we have:

$$\lambda_{max}(\mathbb{E}[\Gamma^{'}_{\mathrm{DSBM}}] - \Gamma^{'}_{DSBM}) \leq (1 + \frac{\mathbf{\Delta}^{'}}{\sqrt{\log n}})(1 - \frac{\mathbf{\Delta}^{'}}{\sqrt{\log n}})(\frac{n}{2}(p-q) - p) \tag{B.11}$$

For each $\mathbf{\Delta}^{'}$, there exists at least one $\mathbf{\Delta}^{'} > 0$ such that:

$$(1 - \frac{\mathbf{\Delta}^{'}}{\sqrt{\log n}})(1 + \frac{\mathbf{\Delta}^{'}}{\sqrt{\log n}}) < 1 \tag{B.12}$$

Hence

$$\lambda_{max}(\mathbb{E}[\Gamma^{'}_{\mathrm{DSBM}}] - \Gamma^{'}_{DSBM}) < \frac{n}{2}(p-q) \tag{B.13}$$

can guarantee the exact recovery of DSBM.

Proof for Lemma 3.7

Lemma B.1. *Recall definition A.3, let a, b and $\mathbf{\Delta}^{'}$ be constants. We have,*

$$T(\frac{n}{2}, \frac{a\log(n)}{n}, \frac{b\log(n)}{n}, -\mathbf{\Delta}^{'}\sqrt{\log(n)}) \leq \exp\left[-(\frac{a+b}{2} - \sqrt{ab} - \delta(n))\log(n)\right] \tag{B.14}$$

with $\lim_{n\to\infty}\delta(n)$

Proof of this lemma can be found in [2]. We are now ready to prove Lemma 3.7 Let $a > b$ be constants and satisfy $\sqrt{a} - \sqrt{b} > \sqrt{2}$. Given $\Delta > 0$, we want to prove that with high probability

$$\min_{i\in[2n]}(\mathcal{D}^{+}_{ii} - \mathcal{D}^{-}_{ii}) \geq \frac{\mathbf{\Delta}}{\log(n)}\mathbb{E}\left[\deg^{+}_{C}(i) - \deg^{-}_{C}(i)\right] = \frac{\mathbf{\Delta}}{\log(n)}\frac{n}{2}(p-q) \tag{B.15}$$

For fixed i throughout the proof. We can write

$$(D^{+})_{ii} - (D^{-})_{ii} = (\sum_{i=1}^{n/2-1} W_i) - (\sum_{i=1}^{n/2} Z_i) = \sum_{i=1}^{n/2-1}(W_i - Z_i) + Z_{n/2} \tag{B.16}$$

Hence, we substitute p and q with $\frac{a\log(n)}{n}$ and $\frac{b\log(n)}{n}$ respectively

$$\frac{\mathbf{\Delta}}{\sqrt{\log(n)}}(\frac{n}{2}(p-q)) = \mathbf{\Delta}\sqrt{\log(n)}(\frac{a-b}{2}) \tag{B.17}$$

We have the probability of $deg_{in}(i) - deg_{out}(i) < \frac{\mathbf{\Delta}}{\sqrt{\log(n)}}(n/2(p-q))$ is equal to

$$\begin{aligned} &\mathbb{P}(\sum_{i=1}^{n/2-1}(W_i - Z_i) + Z_{n/2} < \mathbf{\Delta}\sqrt{\log(n)}(\frac{a-b}{2})) \\ &= \mathbb{P}(\sum_{i=1}^{n/2-1}(Z_i - W_i) - Z_{n/2} > -\mathbf{\Delta}\sqrt{\log(n)}(\frac{a-b}{2})) \end{aligned} \tag{B.18}$$

which is upper bounded by,

$$\mathbb{P}[\sum_{i=1}^{n/2}(Z_i - Wi) > -\mathbf{\Delta}\sqrt{\log(n)}(\frac{a-b}{2})] \tag{B.19}$$

We let $\mathbf{\Delta}^{'} = \mathbf{\Delta}(\frac{a-b}{2}) + 1$, then using the previous definition, we can obtain the following inequalities:

$$\begin{aligned} &\mathbb{P}((\mathcal{D}^{+}_{ii} - \mathcal{D}^{-}_{ii}) < \frac{\mathbf{\Delta}}{\log(n)}\mathbb{E}\left[\deg^{+}_{C}(i) - \deg^{-}_{C}(i)\right]) \\ &\leq T(\frac{n}{2}, \frac{a\log(n)}{n}, \frac{b\log(n)}{n}, -\mathbf{\Delta}^{'}\sqrt{\log(n)}) \\ &\leq \exp\left[-(\frac{a+b}{2} - \sqrt{ab} - \delta(n))\log(n)\right] \end{aligned} \tag{B.20}$$

By using union bound, we can have:

$$\begin{aligned}&\mathbb{P}\left[\min_{i\in[2n]}(\boldsymbol{D}_{ii}^{+}-\boldsymbol{D}_{ii}^{-}<\frac{\boldsymbol{\Delta}}{\sqrt{\log(n)}}\frac{n}{2}(p-q))\right]\\&\leq\exp\left[-(\frac{a+b}{2}-\sqrt{ab}-1-\delta(n))\log(n)\right]\end{aligned}\tag{B.21}$$

from this, if we have $\frac{a+b}{2}-\sqrt{ab}>1$, which can be re-written into $\sqrt{a}-\sqrt{b}>\sqrt{2}$, then this means that the probability of $\mathbb{P}\left[\min_{i\in[2n]}(\boldsymbol{D}_{ii}^{+}-\boldsymbol{D}_{ii}^{-}<\frac{\boldsymbol{\Delta}}{\sqrt{\log(n)}}\frac{n}{2}(p-q))\right]$ is negative. Then when

$$\sqrt{a}-\sqrt{b}>\sqrt{2}\tag{B.22}$$

it holds true with high probability that

$$\min_{i\in[2n]}(\mathcal{D}_{ii}^{+}-\mathcal{D}_{ii}^{-})\geq\frac{\boldsymbol{\Delta}}{\log(n)}\mathbb{E}\left[\deg_{C}^{+}(i)-\deg_{C}^{-}(i)\right]\tag{B.23}$$

References

[1] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. *Exact Recovery in the Stochastic Block Model*. 2014. arXiv: 1405.3267 [cs.SI].

[2] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. “Exact Recovery in the Stochastic Block Model”. In: *CoRR* abs/1405.3267 (2014). arXiv: 1405.3267. URL: http://arxiv.org/abs/1405.3267.

[3] Afonso S. Bandeira. *Random Laplacian matrices and convex relaxations*. 2015. arXiv: 1504.03987 [math.PR].

[4] Santo Fortunato. “Community detection in graphs”. In: *Physics Reports* 486.3-5 (Feb. 2010), pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002. URL: https://doi.org/10.1016%2Fj.physrep.2009.11.002.

[5] M. Girvan and M. E. J. Newman. “Community structure in social and biological networks”. In: *Proceedings of the National Academy of Sciences* 99.12 (June 2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799. URL: https://doi.org/10.1073%2Fpnas.122653799.

[6] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. “Benchmark graphs for testing community detection algorithms”. In: *Physical Review E* 78.4 (Oct. 2008). DOI: 10.1103/physreve.78.046110. URL: https://doi.org/10.1103%2Fphysreve.78.046110.

[7] M. E. J. Newman. “The Structure and Function of Complex Networks”. In: *SIAM Review* 45.2 (Jan. 2003), pp. 167–256. DOI: 10.1137/s003614450342480. URL: https://doi.org/10.1137%2Fs003614450342480.