

# CHERNOFF INFORMATION IN COMMUNITY DETECTION

by  
Cong Mu

A dissertation submitted to The Johns Hopkins University in conformity  
with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland  
February, 2023

© 2023 Cong Mu  
All rights reserved

# Abstract

In network inference applications, it is desirable to detect community structure, i.e., cluster vertices into potential blocks. Beyond adjacency matrices, many real-world networks also involve vertex covariates that may carry information about underlying block structure. Since accurate inference on random networks depends on exploiting all available signal, we need scalable algorithms that can incorporate both network connectivity data and additional insight from vertex covariates. In addition, it can be prohibitively expensive to observe the entire graph in many real applications, especially for large graphs. Thus it becomes essential to identify vertices that have the most impact on block structure and only check whether there are edges between them given a limited budget.

To assess the effects of vertex covariates on block recovery, we consider two model-based spectral algorithms. The first algorithm uses only the adjacency matrix, and directly estimates the block assignments. The second algorithm incorporates both the adjacency matrix and the vertex covariates into the estimation of block assignments. We employ Chernoff information to analytically compare the algorithms' performance and derive the information-theoretic Chernoff ratio for certain models of interest. Analytic results and simulations suggest that the second algorithm is often preferred: one can better estimate the induced block assignments by first estimating the effect of vertex covariates. In addition, real data experiments also indicate that the second algorithm has the advantage of revealing underlying block structure while considering observed vertex heterogeneity in real applications.

Moreover, we propose a dynamic network sampling scheme to optimize block recovery for stochastic blockmodel in the case where it is prohibitively expensive to observe the entire graph. Theoretically, we provide justification of our proposed Chernoff-optimal dynamic sampling scheme via the Chernoff information. Practically, we evaluate the performance of our method on several real datasets from different domains. Both theoretically and practically results suggest that our method can identify vertices that have the most impact on block structure so that one can only check whether there are edges between them to save significant resources but still recover the block structure.

## Thesis Readers

Dr. Carey Priebe (Primary Advisor)

Professor

Department of Applied Mathematics and Statistics

Johns Hopkins University

Dr. Avanti Athreya

Associate Research Professor

Department of Applied Mathematics and Statistics

Johns Hopkins University

*Dedicated to my parents, Yongshi Mu and Hong Xu,  
for their love, trust and support.*

# Acknowledgements

There are so many people to thank. To begin with, I would like to thank my advisor, Professor Carey Priebe, for his unconditional support and valuable guidance during my journey at Hopkins. Carey introduces me to the world of graph inference and leads me to the famous quote by British statistician George E. P. Box –“All models are wrong, but some are useful”, which provides me with a unique perspective to revisit all the stuff that I have learned in statistics. It is always enjoyable to listen to his lecture, the topics may look similar, but each time I could learn something new. It is also a great experience to work with him, he could always find a creative way to formulate the problems, provide insightful suggestion, and make the work meaningful no matter what research topics are. I am really grateful to have Carey as my advisor.

I want to thank Professor Tamas Budavari and Doctor Anton Dahbura for the opportunity to work with them when I first come to Hopkins and their help throughout my stay here. Thanks to Professor Budavari, I have the chance to work on building deep learning models for real applications and develop my interest in computer vision. Thanks to Doctor Dahbura, I have the opportunity to apply mathematical methods and statistical models to solve real world problems.

I would like to thank Professor Avanti Athreya, Professor Angelo Mele, Professor Lingxin Hao, Professor Joshua Cape, Doctor Youngser Park for their help and all the insightful discussion we have over the years. I always feel lucky to have the chance to work in this interdisciplinary group, which gives me the opportunity to learn various concepts and methods from different areas, formulate and solve problems from different

perspectives. Thanks to Avanti, I get to know more about lots of graph models and learn how to express the idea for broad audience. Thanks to Angelo, I have the chance to work with this group at the very beginning and get to know more about network research. Thanks to Lingxin, I have the opportunity to apply our models on various real world datasets from different domains. Thanks to Joshua, I get to learn about Chernoff information and start my dissertation. Thanks to Youngser, I have the chance to learn and practice in scientific computing.

I also want to thank Xiaoyong Zhu, Perry Skountrianos, Mario Inchiosa, Lindsey Allen and all the team members for their help and support during my internship at Microsoft. I am very grateful to have such opportunity to spend my summers there. Thanks to these internships, I get to learn more about natural language processing, pre-trained language models and lots of useful tools to develop deep learning models for real applications.

I would like to thank my girlfriend Tianyao Qu and our beloved cat friend Jasper (Husky) Qu. I couldn't appreciate more for having you always on my side. It is you who make my life full of happiness and warmth. Tianyao and I met in China when we were undergraduates and then we went to America to pursue our graduate degrees. I really appreciate all your understanding over the years. Jasper joined us during the pandemic and made our life much more comfortable for that hard time. I couldn't thank more for your company and support.

Finally, I would like to thank all my committee members and department staffs for their valuable and insightful comments, advice and help over the years. I couldn't appreciate more.

# Contents

<b>Abstract</b> . . . . .	<b>ii</b>
<b>Dedication</b> . . . . .	<b>iv</b>
<b>Acknowledgements</b> . . . . .	<b>v</b>
<b>Contents</b> . . . . .	<b>vii</b>
<b>List of Tables</b> . . . . .	<b>x</b>
<b>List of Figures</b> . . . . .	<b>xi</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
Background and Motivation . . . . .	2
Significance and Contributions . . . . .	5
Overview . . . . .	6
<b>Chapter 2 Random Graph Models, Spectral Methods, Chernoff</b>	
<b>Analysis and Community Detection</b> . . . . .	<b>8</b>
Random Graph Models . . . . .	9
Spectral Methods . . . . .	15
Chernoff Analysis . . . . .	18
Community Detection . . . . .	22
<b>Chapter 3 Community Detection with Vertex Covariates</b> . . . . .	<b>25</b>

Model-based Spectral Inference . . . . .	26
Spectral Inference Performance . . . . .	29
Chernoff Ratio . . . . .	29
2-block Rank One Model with One Binary Covariate . . . . .	30
2-block Homogeneous Model with One Binary Covariate . . . . .	32
$K$ -block Homogeneous Model with One Binary Covariate . . . . .	34
Experiments . . . . .	36
2-block Rank One Model with One 5-categorical Covariate . . . . .	38
2-block Homogeneous Model with One 5-categorical Covariate . . . . .	41
Connectome Data . . . . .	44
Social Network Data . . . . .	46
Latent Position Geometry and Chernoff Ratio . . . . .	48
Proof of Proposition 1 . . . . .	49
Proof of Corollary 1 . . . . .	51
Proof of Theorem 2 . . . . .	53
<b>Chapter 4    Dynamic Network Sampling for Community Detection</b>	<b>62</b>
Dynamic Network Sampling Scheme . . . . .	63
Initial Sampling . . . . .	63
Dynamic Sampling . . . . .	66
Experiments . . . . .	74
Simulations . . . . .	74
Connectome Data . . . . .	78
Microsoft Bing Entity Data . . . . .	80
Social Network Data . . . . .	82
Chernoff Information and Chernoff Superiority . . . . .	84
Proof of Theorem 3 . . . . .	84
Proof of Corollary 4 . . . . .	87

<b>Chapter 5</b>	<b>Conclusions and Discussion</b>	<b>88</b>
<b>Appendix I</b>	<b>Additional Preliminaries in Random Graph Models</b>	<b>91</b>
	Degree Corrected Stochastic Blockmodel	91
	Mixed Membership Stochastic Blockmodel	92
<b>Appendix II</b>	<b>Additional Preliminaries in Spectral Methods</b>	<b>93</b>
	Adjacency Spectral Embedding	93
	Laplacian Spectral Embedding	94
	Spectral Clustering	95
<b>Bibliography</b>		<b>98</b>
<b>Vita</b>		<b>109</b>

# List of Tables

<b>Table 1-I</b>	List of abbreviations. . . . .	7
<b>Table 3-I</b>	Detailed simulation results associated with Fig. 3-4 right panel	37
<b>Table 3-II</b>	Detailed simulation results associated with Fig. 3-5 . . . . .	39
<b>Table 3-III</b>	Detailed simulation results associated with Fig. 3-6 . . . . .	42
<b>Table 3-IV</b>	Algorithms' performance on social network data in terms of ARI. Best results in bold. . . . .	47

# List of Figures

<b>Figure 3-1</b>	Chernoff ratio as in Eq. (3.2) for 2-block rank one model with one binary covariate as in Example 2. $p = 0.3, q \in (0.3, 0.7), \beta \in (0.1, 0.5), \boldsymbol{\pi} = (\frac{1}{2}, \frac{1}{2}), \boldsymbol{\pi}_Z = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ . . . . .	31
<b>Figure 3-2</b>	Chernoff ratio as in Eq. (3.2) for 2-block homogeneous model with one binary covariate as in Example 3. $b = 0.1, a \in (0.1, 0.5), \beta \in (0.1, 0.5), \boldsymbol{\pi} = (\frac{1}{2}, \frac{1}{2}), \boldsymbol{\pi}_Z = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ . . . . .	33
<b>Figure 3-3</b>	Chernoff ratio as in Eq. (3.2) for 4-block homogeneous model with one binary covariate as in Example 3 and Remark 6. $b = 0.1, a \in (0.1, 0.5), \beta \in (0.1, 0.5), \boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}), \boldsymbol{\pi}_Z = (\frac{1}{8}, \dots, \frac{1}{8})$ . . . . .	35
<b>Figure 3-4</b>	Correspondence between Chernoff analysis and simulations.	37
<b>Figure 3-5</b>	Simulations for 2-block rank one model with one 5-categorical covariate, balanced case. . . . .	40
<b>Figure 3-6</b>	Simulations for 2-block homogeneous model with one 5-categorical covariate, balanced case. . . . .	43
<b>Figure 3-7</b>	Algorithms' comparative performance on connectome data.	45
<b>Figure 4-1</b>	Chernoff information $\rho$ as in Eq. (2.29) corresponding to $\mathbf{B}$ as in Eq. (4.2) and $p\mathbf{B}$ for $p \in (0, 1)$ . . . . .	65

<b>Figure 4-2</b>	Chernoff information $\rho$ as in Eq. (2.29) corresponding to $\mathbf{B}$ as in Eq. (4.2), $\mathbf{B}_0$ as in Eq. (4.1), $\mathbf{B}_1$ as in Eq. (4.3), and $\tilde{\mathbf{B}}_1$ as in Eq. (4.5) with initial sampling parameter $p_0 = 0.01$ and dynamic network sampling parameter $p_1 \in (0, p_1^*)$ where $p_1^*$ is defined as in Assumption 2. . . . .	69
<b>Figure 4-3</b>	Chernoff information $\rho$ as in Eq. (2.29) corresponding to $\mathbf{B}$ as in Eq. (4.2), $\mathbf{B}_0$ as in Eq. (4.1), $\mathbf{B}_1$ as in Eq. (4.3), and $\tilde{\mathbf{B}}_1^*$ as in Eq. (4.9) with initial sampling parameter $p_0 = 0.01$ and dynamic network sampling parameter $p_1 \in [p_1^*, p_{11}^{\max}]$ where $p_1^*$ is defined as in Assumption 2 and $p_{11}^{\max}$ is defined as in Eq. (4.10). . . . .	71
<b>Figure 4-4</b>	Simulations for 4-block SBM parameterized by block connectivity probability matrix $\mathbf{B}$ as in Eq. (4.2) with initial sampling parameter $p_0 = 0.01$ and dynamic network sampling parameter $p_1 \in (0, p_{11}^{\max}]$ where $p_{11}^{\max}$ is defined as in Eq. (4.10). . . . .	75
<b>Figure 4-5</b>	Simulations for 4-block SBM parameterized by block connectivity probability matrix $\mathbf{B}$ as in Eq. (4.2) with initial sampling parameter $p_0 = 0.15$ and dynamic network sampling parameter $p_1 \in (0, 0.85)$ . . . . .	77
<b>Figure 4-6</b>	Algorithms' comparative performance on diffusion MRI connectome data via ARI with initial sampling parameter $p_0 = 0.25$ and dynamic network sampling parameter $p_1 = 0.25$ . . . . .	79
<b>Figure 4-7</b>	Algorithms' comparative performance on Microsoft Bing entity data via ARI with different initial sampling parameter $p_0$ and dynamic network sampling parameter $p_1$ . . . . .	81

<b>Figure 4-8</b>	Algorithms' comparative performance on social network data via ARI with different initial sampling parameter $p_0$ and dynamic network sampling parameter $p_1$ . . . . .	83
-------------------	---	----

# Chapter 1

## Introduction

One of the important tasks in network inference is to identify potential block structure from the observed network, i.e., community detection. In addition to the setting for traditional community detection where only information about the connection between vertices, usually represented in the form of adjacency matrices, is considered to detect potential communities, there are other extended problems that have drawn more and more attention recently. For example, if we have extra observed data, how should we combine them with the information contained in adjacency matrices to better identify the block structure? Moreover, even only with adjacency matrices, there are cases when it is impossible to observe the entire graph. That is, one may only have adjacency matrices with a small amount of edges, how should we utilize the limited resources to still recover the potential block structure without knowing all existing edges? This disseration aims to investigate these problems from a unique perspective with the notion of Chernoff information.

## Background and Motivation

On one hand, network data which encodes interactions or relationships between different entities, often involves more than mere links or connections across vertices. In fact, a network dataset may contain not only an adjacency matrix, which consolidates information about the connection between vertices in the network, but additional vertex covariates as well. For example, diffusion magnetic resonance imaging (MRI) connectome datasets [1] where vertices represent sub-regions of the brain defined via spatial proximity and edges represent tensor-based fiber streamlines connecting these sub-regions, can also have brain hemisphere and tissue labels for each vertex, which can be considered as vertex covariates for inference tasks. In addition, social network datasets [2–4], in which vertices can represent users or web pages and edges can represent followers or relationships, may come with ancillary demographic information for each vertex that can also be treated as vertex covariates for inference tasks.

Since accurate inference on random networks depends on exploiting all available signal, scalable algorithms that can incorporate both network connectivity data and any additional insight from vertex covariates are desirable. For instance, in the well-known  $K$ -block stochastic blockmodel (SBM) [5], network vertices belong to  $K$  distinct blocks, or communities, and the probabilities of connection across vertices depend on their block memberships. That is, if  $\tau_i$  denotes the block assignment associated with vertex  $i$ , the connection probability between vertex  $i$  and  $j$  is a function of  $\tau_i$  and  $\tau_j$ . Typically, a vertex’s block membership depends on inherent but unobserved (latent) vertex features. Therefore, a classic inference task is to estimate block memberships from a realization of the resulting network. If, however, we observe both adjacency matrices and vertex covariates, and if both can contain information about the latent communities or blocks, we need models and scalable algorithms that can effectively incorporate information from both the adjacency structure and covariate data and

account for their potentially disparate effects.

In fact, vertex covariates can affect the number of communities that are detected in a blockmodel: for example, a 2-block SBM might bifurcate further into a 4-block SBM because of the impact of a binary covariate (with each block splitting according to this binary covariate). Standard community detection methods may yield a 4-block assignment, but understanding the underlying 2-block SBM is very important in inference applications as we show. To get to an estimate of the underlying two-block assignment, we need to understand the role of the vertex covariates.

Moreover, a problem of interest in network hypothesis testing is to assess the influence of latent blocks on downstream or outcome variables, controlling for vertex covariate effects [6, 7]. For example, assume  $\mathbf{y}_i$  represents some outcome variable associated to vertex  $i$  in a  $K$ -block SBM (for instance, in a demographic dataset,  $\mathbf{y}_i$  might represent the educational attainment or earnings for individual  $i$ ). Suppose the distribution of this outcome variable depends on the vertex’s block assignment within the network, that is if  $\tau_i = k$ , then  $\mathbf{y}_i$  follows some distribution  $F_k$  that can depend on this block. We denote this scenario by writing  $\mathbf{y}_i | (\tau_i = k) \sim F_k$ . A natural question to ask is whether the distributions of the outcome variables are the same for different blocks, i.e., to test whether  $F_k = F$  for  $k \in \{1, \dots, K\}$ . To achieve this goal when we have information from both adjacency matrices and vertex covariates, it is crucial to estimate the underlying block structure  $\hat{\tau}$ —namely, to obtain an estimate of the block structure after accounting for, and effectively “netting out” the vertex covariate effect. Here we write “induced block assignment” to refer to the block assignment after accounting for the vertex covariates.

On the other hand, it can be prohibitively expensive to observe the entire graph in many cases, especially when the number of vertices is very large. For example, in a network where vertices represent landline phones and edges represent whether there is a call between two landline phones. Based on the size of the network, in terms

of the number of vertices, it can be extremely expensive to check whether there is a call for every landline phone pairs. Therefore, if one can utilize the information carried by a partially observed graph, that is only a small number of landline phone pairs are verified, to identify the landline phones that may play a more important role in formulating communities. Then given limited resources, one can choose to only check whether there are calls between those landline phone pairs to achieve the goal of detecting potential block structure. Thus it is important and useful to design certain procedures that can help identify these vertices when we only have a limited budget to observe a partial graph.

## Significance and Contributions

As we discuss before, accurate inference on graphs or networks depends on exploiting all available signal, we need scalable algorithms that can incorporate both network connectivity data and any additional insight from vertex covariates. In addition, given limited resources that one can only observe the partial graph with a small portion of edges, it also becomes essential to identify vertices that have the most impact on block structure so that one can only check whether there are edges between these vertices to save significant resources but still detect the potential block structure.

The innovation of this dissertation is the application of Chernoff information to investigate problems in the area of community detection. To our knowledge, this is the first time that it has been applied to community detection with vertex covariates and network sampling problems. Motivated by the Chernoff analysis, we focus on models and methods that can incorporate information from both the adjacency matrices and the vertex covariates into the estimation of block assignments, and design algorithms and techniques that can still recover the block structure when it is prohibitively expensive to observe the entire graph. We also provide the framework and justification for using Chernoff information in subsequent inference for graphs.

# Overview

The rest of this dissertation is organized as follows. Table 1-I includes a list of abbreviations used in this dissertation.

**Chapter 2** introduces some preliminaries including popular random graph models, commonly used spectral methods, the notion of Chernoff analysis and related works in the area of community detection.

**Chapter 3** focus on the problem of community detection with vertex covarites. Specifically, it introduces model-based spectral algorithms for clustering vertices in stochastic blockmodel graphs with vertex covariates; analytically compares the algorithms' performance via Chernoff information and derives the Chernoff ratio expression for certain models of interest; conducts simulations and real data experiments to compare the algorithms' performance.

**Chapter 4** focus on the problem of dynamic network sampling for community detection. In particular, it introduces the dynamic network sampling scheme to optimize block recovery for partially observed graphs; provides associated theoretical results via the notion of Chernoff analysis; conducts simulations and real data experiments to measure the algorithms' performance in terms of empirical block recovery results.

**Chapter 5** summarizes our findings, discusses the limitations and possible direction for future work.

**Appendix I** provides additional preliminaries in random graph models.

**Appendix II** provides additional preliminaries in spectral methods.

**Table 1-I.** List of abbreviations.

Abbreviation	Description
SBM	Stochastic Blockmodel
DCSBM	Degree Corrected Stochastic Blockmodel
MMSBM	Mixed Membership Stochastic Blockmodel
PCABM	Pairwise Covariates-Adjusted Stochastic Blockmodel
RDPG	Random Dot Product Graph
GRDPG	Generalized Random Dot Product Graph
ASE	Adjacency Spectral Embedding
LSE	Laplacian Spectral Embedding
SCA	Spectral Clustering with Adjacency
SCC	Spectral Clustering with Covariates
CASC	Covariate-Assisted Spectral Clustering
SCWA	Spectral Clustering With Adjustment
CDF	Cumulative Distribution Function
CLT	Central Limit Theorem
MLE	Maximum Likelihood Estimation
GMM	Gaussian Mixture Modeling
BIC	Bayesian Information Criterion
ARI	Adjusted Rand Index
i.i.d.	independent and identically distributed
stderr	standard error
MRI	Magnetic Resonance Imaging
NDMG	NeuroData’s Magnetic Resonance Imaging to Graphs

## Chapter 2

# Random Graph Models, Spectral Methods, Chernoff Analysis and Community Detection

In this chapter, we provide preliminaries including popular random graph models such as generalized random dot product graph and stochastic blockmodel, commonly used spectral methods like adjacency spectral embedding and Laplacian spectral embedding, the notion of Chernoff information and Chernoff analysis, and related works in community detection from the classical methods to the approaches for more complicated scenarios. All of these are necessary elements of applying Chernoff information for solving problems in the area of community detection.

# Random Graph Models

To ground our analysis and results, we start with a particular family of models known as latent position models [8, 9] for edge-independent random graphs. In these models, each network vertex  $i$  is associated with a latent position  $\mathbf{X}_i \in \mathcal{X}$  where  $\mathcal{X}$  is some latent space such as  $\mathbb{R}^d$ , and edges between vertices arise independently with probability  $P_{ij} = \kappa(\mathbf{X}_i, \mathbf{X}_j)$  for some kernel function  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ . This is an appealing model to consider not only because of its wide applicability—as the kernel can be any reasonable regular function—but because it is easily interpretable as well. For example, social network connections are often a function of individual participants’ (potentially unobserved) interests in a core set of topics or hobbies, and levels of interest can be easily encoded in a low-dimensional space. Moreover, the kernel and this lower-dimensional space can possess intuitive geometry, wherein collinearity or other “closeness” of latent positions increases the probability of a connection between the associated vertices.

The core model we focus on here, the generalized random dot product graph (GRDPG), has precisely such a property: the kernel function is taken to be the (indefinite) inner product. As the name suggests, this model generalizes the random dot product graph (RDPG) by relaxing the restriction that the kernel function be the inner product, and this relaxation permits SBM with dissassortative structure, and in fact subsumes all SBMs as special cases.

**Definition 1** (Random Dot Product Graph [10]). *Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix and  $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_n]^\top \in \mathbb{R}^{n \times d}$  where each  $\mathbf{X}_i \in \mathbb{R}^d$  denotes the latent position for vertex  $i$  satisfying  $\mathbf{X}_i^\top \mathbf{X}_j \in [0, 1]$  for all  $i, j \in \{1, \dots, n\}$ . Then we say  $(\mathbf{A}, \mathbf{X}) \sim \text{RDPG}(n)$  if for any  $i, j \in \{1, \dots, n\}$*

$$\begin{aligned} A_{ij} &\sim \text{Bernoulli}(P_{ij}), \\ P_{ij} &= \mathbf{X}_i^\top \mathbf{X}_j. \end{aligned} \tag{2.1}$$

**Remark 1.** The RDPG model has an inherent nonidentifiability. Let  $\mathbf{X}$  be a matrix of latent positions as in Definition 1 and  $\mathbf{W}$  be a unitary matrix, i.e.,  $\mathbf{W}\mathbf{W}^\top = \mathbf{I}$ . Consider  $\mathbf{Y} = \mathbf{X}\mathbf{W}$ , then we have

$$\mathbf{Y}\mathbf{Y}^\top = (\mathbf{X}\mathbf{W})(\mathbf{X}\mathbf{W})^\top = \mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X} = \mathbf{X}\mathbf{X}^\top. \quad (2.2)$$

Thus latent positions  $\mathbf{X}$  and  $\mathbf{Y}$  could give rise to the same distribution over graphs.

**Definition 2** (Generalized Random Dot Product Graph [11]). Let  $d = d_+ + d_-$  with  $d_+ \geq 1$  and  $d_- \geq 0$ . Let  $\mathbf{I}_{d_+d_-} = \text{diag}(1, \dots, 1, -1, \dots, -1)$ , i.e., a  $d \times d$  diagonal matrix with 1 in first  $d_+$  entries and  $-1$  in the next  $d_-$  entries. Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix and  $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_n]^\top \in \mathbb{R}^{n \times d}$  where each  $\mathbf{X}_i \in \mathbb{R}^d$  denotes the latent position for vertex  $i$  satisfying  $\mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{X}_j \in [0, 1]$  for all  $i, j \in \{1, \dots, n\}$ . Then we say  $(\mathbf{A}, \mathbf{X}) \sim \text{GRDPG}(n, d_+, d_-)$  if for any  $i, j \in \{1, \dots, n\}$

$$\begin{aligned} A_{ij} &\sim \text{Bernoulli}(P_{ij}), \\ P_{ij} &= \mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{X}_j. \end{aligned} \quad (2.3)$$

**Remark 2.** Similarly as RDPG, the GRDPG model has an inherent nonidentifiability. Let  $\mathbf{X}$  be a matrix of latent positions as in Definition 2 and  $\mathbf{Q} \in \mathbb{O}(d_+, d_-) = \{\mathbf{M} : \mathbf{M}\mathbf{I}_{d_+d_-}\mathbf{M}^\top = \mathbf{I}_{d_+d_-}\}$ . Consider  $\mathbf{Y} = \mathbf{X}\mathbf{Q}$ , then we have

$$\mathbf{Y}\mathbf{I}_{d_+d_-}\mathbf{Y}^\top = (\mathbf{X}\mathbf{Q})\mathbf{I}_{d_+d_-}(\mathbf{X}\mathbf{Q})^\top = \mathbf{X}\mathbf{Q}\mathbf{I}_{d_+d_-}\mathbf{Q}^\top\mathbf{X} = \mathbf{X}\mathbf{I}_{d_+d_-}\mathbf{X}^\top. \quad (2.4)$$

Thus latent positions  $\mathbf{X}$  and  $\mathbf{Y}$  could give rise to the same distribution over graphs.

**Remark 3.** Also note that in addition to RDPG and GRDPG, most of the latent position models would suffer from similar types of non-identifiability since edge probabilities may be invariant to various transformations [12].

As a special case of random graph models, SBMs are popular in the literature for community detection [5, 13, 14]. Degree corrected stochastic blockmodel (DCSBM) [14] relaxes the standard SBM by allowing for vertices within each block to have different

expected degrees. Mixed membership stochastic blockmodel (MMSBM) [15] extends the standard SBM by allowing for each vertex to in a mixture of different blocks. Here we focus on the standard SBM, additional details on DCSBM and MMSBM can be found in Appendix I.

**Definition 3** (*K*-block Stochastic Blockmodel Graph [5]). *The K-block stochastic blockmodel (SBM) graph is an independent-edge random graph with each vertex belonging to one of K blocks. It can be parameterized by a block connectivity probability matrix  $\mathbf{B} \in [0, 1]^{K \times K}$  and a nonnegative vector of block assignment probabilities  $\boldsymbol{\pi} \in [0, 1]^K$  summing to unity. Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix and  $\boldsymbol{\tau} \in \{1, \dots, K\}^n$  be a vector of block assignments with  $\tau_i = k$  if vertex  $i$  is in block  $k$  (occurring with probability  $\pi_k$ ). We say  $(\mathbf{A}, \boldsymbol{\tau}) \sim \text{SBM}(n, \mathbf{B}, \boldsymbol{\pi})$  if for any  $i, j \in \{1, \dots, n\}$*

$$A_{ij} \sim \text{Bernoulli}(P_{ij}), \quad (2.5)$$

$$P_{ij} = B_{\tau_i \tau_j}.$$

**Remark 4.** *The SBM is a special case of the GRDPG model. Let  $(\mathbf{A}, \boldsymbol{\tau}) \sim \text{SBM}(n, \mathbf{B}, \boldsymbol{\pi})$  as in Definition 3 where  $\mathbf{B} \in (0, 1)^{K \times K}$  with  $d_+$  strictly positive eigenvalues and  $d_-$  strictly negative eigenvalues. To represent this SBM in the GRDPG model, we can choose  $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_K \in \mathbb{R}^d$  where  $d = d_+ + d_-$  such that  $\boldsymbol{\nu}_k^\top \mathbf{I}_{d_+ d_-} \boldsymbol{\nu}_\ell = B_{k\ell}$  for all  $k, \ell \in \{1, \dots, K\}$ . For example, we can take  $\boldsymbol{\nu} = \mathbf{U}_B |\mathbf{S}_B|^{1/2}$  where  $\mathbf{B} = \mathbf{U}_B \mathbf{S}_B \mathbf{U}_B^\top$  is the spectral decomposition of  $\mathbf{B}$  after re-ordering. Then we have the latent position of vertex  $i$  as  $\mathbf{X}_i = \boldsymbol{\nu}_k$  if  $\tau_i = k$  for  $i \in \{1, \dots, n\}$ .*

**Example 1** (2-block Rank One Model). *As an illustration, consider the prototypical 2-block SBM with rank one block connectivity probability matrix  $\mathbf{B}$  where  $B_{11} = p^2, B_{22} = q^2, B_{12} = B_{21} = pq$  with  $0 < p < q < 1$ . Let  $X_i$  be the latent position of vertex  $i$  where  $X_i = \nu_1 = p$  if  $\tau_i = 1$  and  $X_i = \nu_2 = q$  if  $\tau_i = 2$ . Then we can represent this SBM in the GRDPG model with latent positions  $\boldsymbol{\nu} = \begin{bmatrix} p & q \end{bmatrix}^\top$  as*

$$\mathbf{B} = \boldsymbol{\nu} \boldsymbol{\nu}^\top = \begin{bmatrix} p^2 & pq \\ pq & q^2 \end{bmatrix}. \quad (2.6)$$

Since one of our goals is to examine the impact of covariates on community detection, we also consider the extension of the GRDPG model to permit vertex covariates as follows.

**Definition 4** (GRDPG with Vertex Covariates [16]). *Consider GRDPG as in Definition 2. Let  $\mathbf{Z}$  denote the observed vertex covariate and  $\beta$  denote the effect of the vertex covariate. Then we say  $(\mathbf{A}, \mathbf{X}, \mathbf{Z}, \beta) \sim \text{GRDPG-Cov}(n, d_+, d_-)$  if for any  $i, j \in \{1, \dots, n\}$*

$$\begin{aligned} A_{ij} &\sim \text{Bernoulli}(P_{ij}), \\ P_{ij} &= \mathbf{X}_i^\top \mathbf{I}_{d_+ d_-} \mathbf{X}_j + \beta \mathbf{1}\{Z_i = Z_j\}. \end{aligned} \tag{2.7}$$

**Remark 5.** *In the case of an SBM as in Definition 3, Eq. (2.7) becomes*

$$\begin{aligned} A_{ij} &\sim \text{Bernoulli}(P_{ij}), \\ P_{ij} &= B_{\tau_i \tau_j} + \beta \mathbf{1}\{Z_i = Z_j\}. \end{aligned} \tag{2.8}$$

**Example 2** (2-block Rank One Model with One Binary Covariate). *As an illustration, consider the rank one matrix  $\mathbf{B}$  in Eq. (2.6) and the SBM model in Remark 5. Let  $\mathbf{Z} \in \{1, 2\}^n$  denote the observed binary covariate. Assume  $0 < \beta < 1$  with  $p^2 + \beta, q^2 + \beta, pq + \beta \in [0, 1]$ . Then we have the block connectivity probability matrix with the vertex covariate effect as*

$$\mathbf{B}_{\mathbf{Z}} = \begin{bmatrix} p^2 + \beta & p^2 & pq + \beta & pq \\ p^2 & p^2 + \beta & pq & pq + \beta \\ pq + \beta & pq & q^2 + \beta & q^2 \\ pq & pq + \beta & q^2 & q^2 + \beta \end{bmatrix}. \tag{2.9}$$

*Specifically,*

- $B_{Z,11} = p^2 + \beta$  implies  $P_{ij} = p^2 + \beta$  if  $\tau_i = 1, \tau_j = 1, Z_i = 1, Z_j = 1$ .
- $B_{Z,22} = p^2 + \beta$  implies  $P_{ij} = p^2 + \beta$  if  $\tau_i = 1, \tau_j = 1, Z_i = 2, Z_j = 2$ .
- $B_{Z,33} = q^2 + \beta$  implies  $P_{ij} = q^2 + \beta$  if  $\tau_i = 2, \tau_j = 2, Z_i = 1, Z_j = 1$ .

- $B_{Z,44} = q^2 + \beta$  implies  $P_{ij} = q^2 + \beta$  if  $\tau_i = 2, \tau_j = 2, Z_i = 2, Z_j = 2$ .
- $B_{Z,12} = B_{Z,21} = p^2$  implies  $P_{ij} = p^2$  if  $\tau_i = 1, \tau_j = 1, Z_i = 1, Z_j = 2$  or  $\tau_i = 1, \tau_j = 1, Z_i = 2, Z_j = 1$ .
- $B_{Z,13} = B_{Z,31} = pq + \beta$  implies  $P_{ij} = pq + \beta$  if  $\tau_i = 1, \tau_j = 2, Z_i = 1, Z_j = 1$  or  $\tau_i = 2, \tau_j = 1, Z_i = 1, Z_j = 1$ .
- $B_{Z,14} = B_{Z,41} = pq$  implies  $P_{ij} = pq$  if  $\tau_i = 1, \tau_j = 2, Z_i = 1, Z_j = 2$  or  $\tau_i = 2, \tau_j = 1, Z_i = 2, Z_j = 1$ .
- $B_{Z,23} = B_{Z,32} = pq$  implies  $P_{ij} = pq$  if  $\tau_i = 1, \tau_j = 2, Z_i = 2, Z_j = 1$  or  $\tau_i = 2, \tau_j = 1, Z_i = 1, Z_j = 2$ .
- $B_{Z,24} = B_{Z,42} = pq + \beta$  implies  $P_{ij} = pq + \beta$  if  $\tau_i = 1, \tau_j = 2, Z_i = 2, Z_j = 2$  or  $\tau_i = 2, \tau_j = 1, Z_i = 2, Z_j = 2$ .
- $B_{Z,34} = B_{Z,43} = q^2$  implies  $P_{ij} = q^2$  if  $\tau_i = 2, \tau_j = 2, Z_i = 1, Z_j = 2$  or  $\tau_i = 2, \tau_j = 2, Z_i = 2, Z_j = 1$ .

**Example 3** (2-block Homogeneous Model with One Binary Covariate). *As a second illustration, consider the rank two matrix  $\mathbf{B}$  where  $B_{11} = B_{22} = a, B_{12} = B_{21} = b$  with  $0 < b < a < 1$ . Assume  $0 < \beta < 1$  with  $a + \beta, b + \beta \in [0, 1]$ . We then have the block connectivity probability matrix with the vertex covariate effect as*

$$\mathbf{B}_{\mathbf{Z}} = \begin{bmatrix} a + \beta & a & b + \beta & b \\ a & a + \beta & b & b + \beta \\ b + \beta & b & a + \beta & a \\ b & b + \beta & a & a + \beta \end{bmatrix}. \quad (2.10)$$

*Specifically,*

- $B_{Z,11} = a + \beta$  implies  $P_{ij} = a + \beta$  if  $\tau_i = 1, \tau_j = 1, Z_i = 1, Z_j = 1$ .

- $B_{Z,22} = a + \beta$  implies  $P_{ij} = a + \beta$  if  $\tau_i = 1, \tau_j = 1, Z_i = 2, Z_j = 2$ .
- $B_{Z,33} = a + \beta$  implies  $P_{ij} = a + \beta$  if  $\tau_i = 2, \tau_j = 2, Z_i = 1, Z_j = 1$ .
- $B_{Z,44} = a + \beta$  implies  $P_{ij} = a + \beta$  if  $\tau_i = 2, \tau_j = 2, Z_i = 2, Z_j = 2$ .
- $B_{Z,12} = B_{Z,21} = a$  implies  $P_{ij} = a$  if  $\tau_i = 1, \tau_j = 1, Z_i = 1, Z_j = 2$  or  $\tau_i = 1, \tau_j = 1, Z_i = 2, Z_j = 1$ .
- $B_{Z,13} = B_{Z,31} = b + \beta$  implies  $P_{ij} = b + \beta$  if  $\tau_i = 1, \tau_j = 2, Z_i = 1, Z_j = 1$  or  $\tau_i = 2, \tau_j = 1, Z_i = 1, Z_j = 1$ .
- $B_{Z,14} = B_{Z,41} = b$  implies  $P_{ij} = b$  if  $\tau_i = 1, \tau_j = 2, Z_i = 1, Z_j = 2$  or  $\tau_i = 2, \tau_j = 1, Z_i = 2, Z_j = 1$ .
- $B_{Z,23} = B_{Z,32} = b$  implies  $P_{ij} = b$  if  $\tau_i = 1, \tau_j = 2, Z_i = 2, Z_j = 1$  or  $\tau_i = 2, \tau_j = 1, Z_i = 1, Z_j = 2$ .
- $B_{Z,24} = B_{Z,42} = b + \beta$  implies  $P_{ij} = b + \beta$  if  $\tau_i = 1, \tau_j = 2, Z_i = 2, Z_j = 2$  or  $\tau_i = 2, \tau_j = 1, Z_i = 2, Z_j = 2$ .
- $B_{Z,34} = B_{Z,43} = a$  implies  $P_{ij} = a$  if  $\tau_i = 2, \tau_j = 2, Z_i = 1, Z_j = 2$  or  $\tau_i = 2, \tau_j = 2, Z_i = 2, Z_j = 1$ .

**Remark 6.** The SBMs parameterized by  $\mathbf{B}$  in Example 3 lead to the notion of the homogeneous model [13, 17]. For  $K$ -block homogeneous model, we have  $B_{k\ell} = a$  for  $k = \ell$  and  $B_{k\ell} = b$  for  $k \neq \ell$  where  $k, \ell \in \{1, \dots, K\}$ . For example, the block connectivity probability matrix  $\mathbf{B}$  of the 4-block homogeneous model is given by

$$\mathbf{B} = \begin{bmatrix} a & b & b & b \\ b & a & b & b \\ b & b & a & b \\ b & b & b & a \end{bmatrix}. \quad (2.11)$$

## Spectral Methods

In Examples 2 and 3, an induced 2-block SBM becomes a 4-block SBM via the effect of a binary vertex covariate. One of our goals is to cluster each vertex into one of the two induced blocks after accounting for the vertex covariate effect. To this end, we need to first recover the latent positions of the underlying GRDPG using spectral methods, and then estimate the block assignments with certain clustering technique.

Spectral methods [18] that promise applicability to large graphs have been widely used in random graph models for a variety of subsequent inference tasks such as community detection [19–22], vertex nomination [23], nonparametric hypothesis testing [24], and multiple graph inference [25]. Two particular spectral embedding methods, adjacency spectral embedding (ASE) and Laplacian spectral embedding (LSE), which are spectral decompositions of the graph adjacency and graph Laplacian matrices, respectively, are popular, since they provide consistent [11, 26] and asymptotically normal [11, 27, 28] estimates of underlying graph parameters.

**Definition 5** (Adjacency Spectral Embedding). *Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix with eigendecomposition  $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ . Given the embedding dimension  $d < n$ , the adjacency spectral embedding (ASE) of  $\mathbf{A}$  into  $\mathbb{R}^d$  is the  $n \times d$  matrix  $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}_d |\widehat{\mathbf{\Lambda}}_d|^{1/2}$  where  $\widehat{\mathbf{\Lambda}}_d$  is a diagonal matrix with the  $d$  largest eigenvalues in magnitudes and  $\widehat{\mathbf{U}}_d$  contains the associated eigenvectors. Here hat notation suggests these terms estimate the eigenvectors and eigenvalues of the matrix  $\mathbf{P}$  as in Eq. (2.3).*

**Definition 6** (Laplacian Spectral Embedding). *Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix and  $\mathcal{L}(\mathbf{A})$  be the normalized Laplacian of  $\mathbf{A}$  defined by  $\mathcal{L}(\mathbf{A}) = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$  where  $\mathbf{D}$  is a diagonal matrix with  $D_{ii} = \sum_{j \neq i} A_{ij}$ . Assume the eigendecomposition of  $\mathcal{L}(\mathbf{A})$  is given by  $\mathcal{L}(\mathbf{A}) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ . Given the embedding dimension  $d < n$ , the Laplacian spectral embedding (LSE) of  $\mathbf{A}$  into  $\mathbb{R}^d$  is the  $n \times d$  matrix  $\widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}_d |\widetilde{\mathbf{\Lambda}}_d|^{1/2}$  where  $\widetilde{\mathbf{\Lambda}}_d$  is a diagonal matrix with the  $d$  largest eigenvalues in magnitudes and  $\widetilde{\mathbf{U}}_d$*

contains the associated eigenvectors. Here tilde notation suggests these terms estimate the eigenvectors and eigenvalues of the matrix  $\mathcal{L}(\mathbf{P})$  where  $\mathbf{P}$  is defined as in Eq. (2.3).

**Remark 7.** There are different methods for choosing the embedding dimension [29, 30]; we adopt the well-established and computationally efficient profile likelihood method [31] to automatically identify an elbow in the scree plot to select embedding dimension  $\hat{d}$  in real applications.

As mentioned above, ASE and LSE yield consistent and asymptotically normal estimates of underlying graph parameters under different random graph models. Here we review the central limit theorem (CLT) of ASE under the SBM model, which are essential when we introduce the concept of Chernoff analysis later. Appendix II provides additional details on asymptotic results of ASE and LSE under the GRDPG model.

**Theorem 1** (CLT of ASE for SBM [11]). *Let  $(\mathbf{A}^{(n)}, \mathbf{X}^{(n)}) \sim \text{GRDPG}(n, d_+, d_-)$  be a sequence of adjacency matrices and associated latent positions of a  $d$ -dimensional GRDPG as in Definition 2 from a distribution  $F$  where  $F$  is a mixture of  $K$  point masses in  $\mathbb{R}^d$ , i.e.,*

$$F = \sum_{k=1}^K \pi_k \delta_{\nu_k} \quad \text{with} \quad \forall k, \pi_k > 0 \quad \text{and} \quad \sum_{k=1}^K \pi_k = 1, \quad (2.12)$$

where  $\delta_{\nu_k}$  is the Dirac delta measure at  $\nu_k$ . Let  $\Phi(\mathbf{z}, \Sigma)$  denote the cumulative distribution function (CDF) of a multivariate Gaussian distribution with mean  $\mathbf{0}$  and covariance matrix  $\Sigma$ , evaluated at  $\mathbf{z} \in \mathbb{R}^d$ . Let  $\widehat{\mathbf{X}}^{(n)}$  be the ASE of  $\mathbf{A}^{(n)}$  with  $\widehat{\mathbf{X}}_i^{(n)}$  as the  $i$ -th row (same for  $\mathbf{X}_i^{(n)}$ ). Then there exists a sequence of matrices  $\mathbf{M}_n \in \mathbb{R}^{d \times d}$  satisfying  $\mathbf{M}_n \mathbf{I}_{d_+ d_-} \mathbf{M}_n^\top = \mathbf{I}_{d_+ d_-}$  such that for all  $\mathbf{z} \in \mathbb{R}^d$  and fixed index  $i$ ,

$$\mathbb{P} \left[ \sqrt{n} \left( \mathbf{M}_n \widehat{\mathbf{X}}_i^{(n)} - \mathbf{X}_i^{(n)} \right) \leq \mathbf{z} \mid \mathbf{X}_i^{(n)} = \nu_k \right] \rightarrow \Phi(\mathbf{z}, \Sigma_k), \quad (2.13)$$

where for  $\nu \sim F$

$$\Sigma_k = \Sigma(\nu_k) = \mathbf{I}_{d_+ d_-} \Delta^{-1} \mathbb{E} \left[ \left( \nu_k^\top \mathbf{I}_{d_+ d_-} \nu \right) \left( 1 - \nu_k^\top \mathbf{I}_{d_+ d_-} \nu \right) \nu \nu^\top \right] \Delta^{-1} \mathbf{I}_{d_+ d_-}, \quad (2.14)$$

with

$$\Delta = \mathbb{E} [\boldsymbol{\nu} \boldsymbol{\nu}^\top]. \quad (2.15)$$

**Remark 8.** If the adjacency matrix  $\mathbf{A}$  is sampled from an SBM parameterized by the block connectivity probability matrix  $\mathbf{B}$  in Eq. (2.6) and block assignment probabilities  $\boldsymbol{\pi} = (\pi_1, \pi_2)$  with  $\pi_1 + \pi_2 = 1$ , then as a special case for Theorem 1 [12, 28], we have for each fixed index  $i$ ,

$$\begin{aligned} \sqrt{n} (\widehat{X}_i - p) &\xrightarrow{d} \mathcal{N}(0, \sigma_p^2) && \text{if } X_i = p, \\ \sqrt{n} (\widehat{X}_i - q) &\xrightarrow{d} \mathcal{N}(0, \sigma_q^2) && \text{if } X_i = q. \end{aligned} \quad (2.16)$$

where

$$\begin{aligned} \sigma_p^2 &= \frac{\pi_1 p^4 (1 - p^2) + \pi_2 p q^3 (1 - pq)}{[\pi_1 p^2 + \pi_2 q^2]^2}, \\ \sigma_q^2 &= \frac{\pi_1 p^3 q (1 - pq) + \pi_2 q^4 (1 - q^2)}{[\pi_1 p^2 + \pi_2 q^2]^2}. \end{aligned} \quad (2.17)$$

## Chernoff Analysis

To analytically measure the performance of algorithms for block recovery, we consider the notion of Chernoff information among other possible metrics. The concept of Chernoff information is first employed for SBMs [13, 28] and then extended to consider the underlying graph structure [17]. The advantages of Chernoff information are that it is independent of the clustering procedure, i.e., it can be derived no matter which clustering methods are used, and it is intrinsically related to the Bayes risk [12, 14, 28].

Formally following the analysis in [12, 28], let  $F_1$  and  $F_2$  be two absolutely continuous multivariate distributions on  $\mathbb{R}^d$  with density functions  $f_1$  and  $f_2$ , respectively. Suppose  $Y_1, \dots, Y_m$  are independent and identically distributed (i.i.d.) random variables with  $Y_i \sim F$  where  $F$  is either  $F_1$  or  $F_2$  for  $i \in \{1, \dots, m\}$ . We want to test the following hypothesis:

$$H_0 : F = F_1 \quad \text{v.s.} \quad H_A : F = F_2. \quad (2.18)$$

By Neyman-Pearson lemma [32], given the data  $y_1, \dots, y_m$  and a threshold  $\eta_m \in \mathbb{R}$ , the most powerful test at significance level  $\alpha_m = \alpha(\eta_m)$  is the likelihood ratio test that rejects  $H_0$  when

$$\sum_{i=1}^m \log f_1(y_i) - \sum_{i=1}^m \log f_2(y_i) \leq \eta_m. \quad (2.19)$$

That is given the constraint that the Type I error is at most  $\alpha_m$ , this likelihood ratio test minimizes the Type II error  $\beta_m$ . Suppose that  $H_0$  is true with prior probability  $\pi \in (0, 1)$ . For a given significance level  $\alpha_m^* \in (0, 1)$ , let  $\beta_m^* = \beta_m^*(\alpha_m^*)$  denote the Type II error associated with the likelihood ratio test defined as in Eq. (2.19) subject to the constraint that the Type I error is at most  $\alpha_m^*$ . Then the Bayes risk of testing the hypothesis as in Eq. (2.18) is given by

$$\inf_{\alpha_m^* \in (0, 1)} \pi \alpha_m^* + (1 - \pi) \beta_m^*. \quad (2.20)$$

It has been shown [33, 34] that this Bayes risk is intrinsically linked to the Chernoff information defined as follows.

**Definition 7** (Chernoff Information [33, 34]). *Let  $F_1$  and  $F_2$  be two continuous multivariate distributions on  $\mathbb{R}^d$  with density functions  $f_1$  and  $f_2$ . The Chernoff information is defined as*

$$\begin{aligned} C(F_1, F_2) &= -\log \left[ \inf_{t \in (0,1)} \int_{\mathbb{R}^d} f_1^t(\mathbf{x}) f_2^{1-t}(\mathbf{x}) d\mathbf{x} \right] \\ &= \sup_{t \in (0,1)} \left[ -\log \int_{\mathbb{R}^d} f_1^t(\mathbf{x}) f_2^{1-t}(\mathbf{x}) d\mathbf{x} \right]. \end{aligned} \quad (2.21)$$

**Remark 9.** *Consider the special case where we take  $F_1 = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$  and  $F_2 = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ ; then the corresponding Chernoff information is*

$$C(F_1, F_2) = \sup_{t \in (0,1)} \left[ \frac{1}{2} t(1-t) (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_t^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_t|}{|\boldsymbol{\Sigma}_1|^t |\boldsymbol{\Sigma}_2|^{1-t}} \right], \quad (2.22)$$

where  $\boldsymbol{\Sigma}_t = t\boldsymbol{\Sigma}_1 + (1-t)\boldsymbol{\Sigma}_2$ .

Then by Eq. (2.20) and Eq. (2.21), we have

$$\lim_{m \rightarrow \infty} \frac{1}{m} \inf_{\alpha_m^* \in (0,1)} \log [\pi \alpha_m^* + (1-\pi) \beta_m^*] = -C(F_1, F_2). \quad (2.23)$$

In other words, the Chernoff information between  $F_1$  and  $F_2$  is the exponential rate at which the Bayes risk as in Eq. (2.20) decreases as  $m \rightarrow \infty$ .

Similarly, we could extend the analysis to  $K$  hypotheses. Let  $F_1, \dots, F_K$  be absolutely continuous multivariate distributions on  $\mathbb{R}^d$  with density functions  $f_1, \dots, f_K$ , respectively. Suppose  $Y_1, \dots, Y_m$  are i.i.d. random variables with  $Y_i \sim F$  where  $F \in \{F_1, \dots, F_K\}$  for  $i \in \{1, \dots, m\}$ . We want to test the following hypothesis:

$$H_0 : F = F_1 \quad \dots \quad H_{K-1} : F = F_K. \quad (2.24)$$

Suppose that  $H_k$  is true with prior probability  $\pi_k \in (0, 1)$  for  $k \in \{0, \dots, K-1\}$ . Given a decision rule  $\delta$ , let  $\alpha_{\ell k}(\delta)$  denote the probability of accepting  $H_\ell$  when  $H_k$  is

true. Then we have [35]

$$\inf_{\delta} \lim_{m \rightarrow \infty} \frac{r(\delta)}{m} = \min_{k \neq \ell} C(F_k, F_{\ell}), \quad (2.25)$$

where

$$r(\delta) = \sum_k \pi_k \sum_{\ell \neq k} \alpha_{\ell k}(\delta). \quad (2.26)$$

That is, for any decision rule  $\delta$ , the risk  $r(\delta)$  decreases to 0 as  $m \rightarrow \infty$  at a rate no faster than  $\exp[-m \min_{k \neq \ell} C(F_k, F_{\ell})]$  where  $k, \ell \in \{0, \dots, K-1\}$ .

**Remark 10.** *With the similar notation in Definition 7, the Chernoff divergence defined by*

$$C_t(F_1, F_2) = -\log \left[ \int_{\mathbb{R}^d} f_1^t(\mathbf{x}) f_2^{1-t}(\mathbf{x}) d\mathbf{x} \right] \quad (2.27)$$

*belongs to the family of  $f$ -divergence as defined in [36, 37]. Note that  $C_t(F_1, F_2)$  is the Bhattacharyya distance [38, 39] between  $F_1$  and  $F_2$  when  $t = \frac{1}{2}$ . As any  $f$ -divergence such as the Kullback-Liebler divergence [40, 41] is invariant with respect to invertible transformations [42], they can also be considered as alternative metrics. But the Chernoff information is particularly appealing because of its explicit relationship with the Bayes risk as we discuss before.*

The comparison of block recovery via Chernoff information is based on the statistical information between the limiting distributions of the blocks and smaller statistical information implies less information to discriminate between different blocks of the SBM. With Theorem 1, we can describe the error rate for estimating the block assignments using ASE via the notion of Chernoff information as follows.

For a  $K$ -block SBM, let  $\mathbf{B} \in (0, 1)^{K \times K}$  be the block connectivity probability matrix and  $\boldsymbol{\pi} \in (0, 1)^K$  be the vector of block assignment probabilities. Given an  $n$  vertex instantiation of the SBM parameterized by  $\mathbf{B}$  and  $\boldsymbol{\pi}$ , for sufficiently large  $n$ , the large sample optimal error rate for estimating the block assignments using ASE can be

measured via Chernoff information as [12, 28]

$$\rho = \min_{k \neq \ell} \sup_{t \in (0,1)} \left[ \frac{1}{2} n t (1-t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) + \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_{k\ell}(t)|}{|\boldsymbol{\Sigma}_k|^t |\boldsymbol{\Sigma}_\ell|^{1-t}} \right], \quad (2.28)$$

where  $\boldsymbol{\Sigma}_{k\ell}(t) = t\boldsymbol{\Sigma}_k + (1-t)\boldsymbol{\Sigma}_\ell$ ,  $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}(\boldsymbol{\nu}_k)$  and  $\boldsymbol{\Sigma}_\ell = \boldsymbol{\Sigma}(\boldsymbol{\nu}_\ell)$  are defined as in Eq. (2.14). Also note that as  $n \rightarrow \infty$ , the logarithm term in Eq. (2.28) will be dominated by the other term. Then we have the approximate Chernoff information as

$$\rho \approx \min_{k \neq \ell} C_{k,\ell}(\mathbf{B}, \boldsymbol{\pi}), \quad (2.29)$$

where

$$C_{k,\ell}(\mathbf{B}, \boldsymbol{\pi}) = \sup_{t \in (0,1)} \left[ t(1-t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) \right]. \quad (2.30)$$

We also introduce the following two notions based on Chernoff information, which will be used when we introduce our dynamic network sampling scheme for community detection.

**Definition 8** (Chernoff-active Blocks). *For  $K$ -block SBM parametrized by the block connectivity probability matrix  $\mathbf{B} \in (0,1)^{K \times K}$  and the vector of block assignment probabilities  $\boldsymbol{\pi} \in (0,1)^K$ . The Chernoff-active blocks  $(k^*, \ell^*)$  are defined as*

$$(k^*, \ell^*) = \arg \min_{k \neq \ell} C_{k,\ell}(\mathbf{B}, \boldsymbol{\pi}), \quad (2.31)$$

where  $C_{k,\ell}(\mathbf{B}, \boldsymbol{\pi})$  is defined as in Eq. (2.29).

**Definition 9** (Chernoff Superiority). *For  $K$ -block SBMs, given two block connectivity probability matrices  $\mathbf{B}, \mathbf{B}' \in (0,1)^{K \times K}$  and a vector of block assignment probabilities  $\boldsymbol{\pi} \in (0,1)^K$ . Let  $\rho_B$  and  $\rho_{B'}$  denote the Chernoff information obtained as in Eq. (2.29) corresponding to  $\mathbf{B}$  and  $\mathbf{B}'$  respectively. We say that  $\mathbf{B}$  is Chernoff superior to  $\mathbf{B}'$ , denoted as  $\mathbf{B} \succ \mathbf{B}'$ , if  $\rho_B > \rho_{B'}$ .*

**Remark 11.** *If  $\mathbf{B}$  is Chernoff superior to  $\mathbf{B}'$ , then we can have a better block recovery from  $\mathbf{B}$  than  $\mathbf{B}'$ . In addition, Chernoff superiority is transitive, which is straightforward from the definition.*

# Community Detection

The goal of community detection in network inference is to cluster vertices into several potential blocks. Many classical community detection methods only consider the adjacency or Laplacian matrices for clustering vertices [43–52]. Following the categories summarized in [45],

- Consensus clustering [53–55] tries to incorporate the information of various outputs into a new partition. The goal is to search for a better consensus partition compared with the input partitions. As this is a difficult combinatorial optimization problem, an alternative greedy strategy is proposed to investigate the consensus matrix based on the co-occurrence of vertices in communities from the input partitions. In other words, this consensus matrix is treated as the input of any graph clustering methods and the output is used to generate a new consensus matrix. This procedure is repeated until the partitions no longer change.
- Traditional spectral methods [18, 44] detect communities by investigating the spectral properties of the the adjacency or Laplacian matrices. In other words, these methods first embed the adjacency or Laplacian matrices into some lower dimensional space via matrix decomposition. Then certain clustering technique such as  $k$ -means is applied to the lower dimensional embeddings to identify potential communities. Appendix II also provides additional details on classical spectral clustering, which is the building block of our proposed methods.
- Statistical inference based methods [9, 14, 56–63] aim to identify communities by fitting a generative network model from the data. For example, the commonly used SBM and its extensions such as DCSBM and MMSBM. The estimation of the parameters for these models mostly relies on the likelihood based methods or their variations, which can be time consuming for large networks.

- Optimization based methods [64–69] rely on finding the extremum of certain function that could indicate the clustering quality, for example, modularity. Most of these functions are proposed and chosen to estimate the quality of a partition as potential communities for the network. Based on the choice of the objective function, finding the extremum can be NP-hard. Therefore, various approaches are also proposed to find the approximations of the extremum for these functions.
- Dynamics based methods [70–81] focus on running dynamical processes on the network to find communities. For example, some techniques first applies random walk dynamics to estimate the similarity between vertex pairs and then detects potential communities by standard hierarchical or partitional clustering methods.

On one hand, these methods are typically not designed to distinguishing the impact of covariates from the mechanism of network generation itself—that is, delineating in the observed data what may be underlying, or fundamental, network effects from characteristics that are more properly functions of the covariates. By contrast, covariate-aware inference in SBMs often relies on either variational methods [82–84] or spectral approaches [16, 85, 86]. For example, [85] proposed covariate-assisted spectral clustering (CASC) where the covariates are first parameterized as in linear regression, i.e., categorical covariates are represented with dummy variables and continuous covariates can go through standardization, and then combined with the graph for subsequent spectral clustering. The pairwise covariates-adjusted stochastic blockmodel (PCABM), in which pairwise covariate information is incorporated with the classical SBM, was introduced in [86]. There, model parameters can be solved via maximum likelihood estimation (MLE) or spectral clustering with adjustment (SCWA).

On the other hand, none of the classical methods focus on the problem of clustering vertices for partially observed graphs. To address this issue, existing methods propose

different types of random and adaptive sampling strategies to minimize the information loss from the data reduction [87]. For example, [88] considered adaptive sampling strategies to design a jointly sampling and clustering algorithms to recover the hidden communities.

## Chapter 3

# Community Detection with Vertex Covariates

In this chapter, we focus on the problem of community detection with vertex covariates. We start with two model-based spectral algorithms for clustering vertices in stochastic blockmodel graphs with vertex covariates. The first algorithm uses only the adjacency matrix, and directly estimates the block assignments. The second algorithm incorporates both the adjacency matrix and the vertex covariates into the estimation of block assignments, and moreover quantifies the explicit impact of the vertex covariates on the resulting estimate of the block assignments. Theoretically, we employ Chernoff information to analytically compare the algorithms' performance and derive the information-theoretic Chernoff ratio for certain models of interest. Practically, we evaluate the performance, in terms of block recovery, of the algorithms on several real datasets from different domains. Both theoretically and practically results suggest that the second algorithm is often preferred: one can better estimate the induced block assignments by first estimating the effect of vertex covariates. In the meantime, it also has the advantage of revealing underlying block structure while considering observed vertex heterogeneity in real applications.

## Model-based Spectral Inference

We are interested in estimating the induced block assignments (clustering vertices) in a SBM with vertex covariates. To that end, we want to consider algorithms for estimating the vertex covariate effect  $\beta$  as in Definition 4 and Remark 5, which can be further used to estimate the induced block assignments. Our model-based spectral algorithms take observed adjacency matrices (and vertex covariates) as inputs and estimated block assignments for each vertex as outputs.

---

**Algorithm 1:** Estimation of induced block assignment using only the adjacency matrix [89]

---

**Input:** Adjacency matrix  $\mathbf{A} \in \{0, 1\}^{n \times n}$ .

**Output:** Induced block assignments  $\hat{\tau}$ .

- 1 Estimate latent positions under the effects of both observed covariates and unobserved heterogeneity of vertices as  $\hat{\mathbf{Y}} \in \mathbb{R}^{n \times \hat{d}}$  using ASE of  $\mathbf{A}$  where  $\hat{d}$  is chosen as in Remark 7.
- 2 Cluster  $\hat{\mathbf{Y}}$  using Gaussian mixture modeling (GMM) to estimate the block assignments under the effects of both observed covariates and unobserved heterogeneity of vertices as  $\hat{\xi} \in \{1, \dots, \hat{K}\}^n$  where  $\hat{K}$  is chosen via Bayesian Information Criterion (BIC).
- 3 Compute the estimated block connectivity probability matrix including the vertex covariate effect as

$$\hat{\mathbf{B}}_Z = \hat{\boldsymbol{\mu}} \mathbf{I}_{\hat{d} \times \hat{d}} \hat{\boldsymbol{\mu}}^\top \in [0, 1]^{\hat{K} \times \hat{K}},$$

where  $\hat{\boldsymbol{\mu}} \in \mathbb{R}^{\hat{K} \times \hat{d}}$  is the matrix of estimated means of all clusters.

- 4 Cluster the diagonal of  $\hat{\mathbf{B}}_Z$  using GMM to estimate the cluster assignments of the diagonal as  $\hat{\phi} \in \{1, \dots, \frac{\hat{K}}{2}\}^{\hat{K}}$ .
  - 5 Estimate the induced block assignments as  $\hat{\tau}$  by  $\hat{\tau}_k = c$  for  $k \in \{i \mid \hat{\xi}_i = t \text{ for } t \in \{j \mid \hat{\phi}_j = c\}\}$  and  $c = 1, \dots, \frac{\hat{K}}{2}$ .
- 

In Algorithm 1, the estimation of the induced block assignments, i.e.,  $\hat{\tau}$ , depends on the estimated block connectivity probability matrix  $\hat{\mathbf{B}}_Z$  (see Step 4 of Algorithm 1 for details). This suggests that we may not obtain an accurate estimate of the induced block assignments if the diagonal of  $\hat{\mathbf{B}}_Z$  does not contain enough information to distinguish the induced block structure. To address this uncertainty, we consider a modified algorithm that uses the information from vertex covariates to estimate

the induced block assignments along with vertex covariate effect  $\beta$  summarized as in Algorithm 2.

As an illustration of estimating  $\beta$  (Step 2 in Algorithm 2), consider the block connectivity probability matrix  $\mathbf{B}_Z$  as in Eq. (2.10). To get  $\beta$ , we can take the difference between two specific entries of  $\mathbf{B}_Z$ . For example,

$$\begin{aligned} B_{Z,11} - B_{Z,12} &= (a + \beta) - a = \beta, \\ B_{Z,13} - B_{Z,14} &= (b + \beta) - b = \beta. \end{aligned} \tag{3.1}$$

We can then obtain  $\hat{\beta}$  by subtracting two specific entries of  $\hat{\mathbf{B}}_Z$ . However, the ASE and GMM under GRDPG model can lead to the re-ordering of  $\hat{\mathbf{B}}_Z$ . Thus we need to identify pairs first so that we subtract the correct entries. Two alternative ways to achieve this are described in Step 2(a) and 2(b) of Algorithm 2.

In Step 2(a), we find pairs in  $\hat{\mathbf{B}}_Z$  by first assigning each block common covariates using the mode. However, it is possible that we can not find any pairs using this approach, especially in the unbalanced case where the size of each block is different and/or the distribution of the vertex covariate is different. For example, one block size is much larger than the others and/or vertex covariates are all the same within one block.

In Step 2(b), instead of first finding pairs using the mode, we only compute the probability that two entries of  $\hat{\mathbf{B}}_Z$  form a pair. This will make the estimation more robust to extreme cases or special structure by giving different weights to pairs [16].

---

**Algorithm 2:** Estimation of induced block assignment incorporating both the adjacency matrix and the vertex covariates [89]

---

**Input:** Adjacency matrix  $\mathbf{A} \in \{0, 1\}^{n \times n}$ ; observed vertex covariates  $\mathbf{Z} \in \{1, 2\}^n$ .

**Output:** Estimated vertex covariate effect  $\hat{\beta}$ ; induced block assignments  $\tilde{\tau}$ .

- 1 Steps 1 – 4 in Algorithm 1.
- 2 Estimate the vertex covariate effect as  $\hat{\beta}$  using one of the following procedures [16].
  - (a) Assign the block covariates as  $\mathbf{Z}_B \in \{-1, 1\}^{\hat{K}}$  for each block using the mode, i.e.,

$$Z_{B,k} = \begin{cases} -1 & \text{if } n_{-1,k} \geq n_{1,k} \\ 1 & \text{if } n_{-1,k} < n_{1,k} \end{cases} \quad \text{where} \quad n_{z,k} = \sum_{i: \hat{\xi}_i = k} \mathbf{1}\{Z_i = z\}.$$

Construct pair set

$S = \{(k\ell, k\ell'), k, \ell, \ell' \in \{1, \dots, \hat{K}\} \mid \hat{\phi}_\ell = \hat{\phi}_{\ell'}, Z_{B,k} = Z_{B,\ell}, Z_{B,k} \neq Z_{B,\ell'}\}$ . Estimate the vertex covariate effect as

$$\hat{\beta}_{\text{SA}} = \frac{1}{|S|} \sum_{(k\ell, k\ell') \in S} \hat{B}_{Z,k\ell} - \hat{B}_{Z,k\ell'}.$$

- (b) Compute the probability that two entries from  $\hat{\mathbf{B}}_Z$  form a pair as

$$p_{k\ell, k\ell'} = \frac{n_{-1,k}n_{-1,\ell}n_{1,\ell'} + n_{1,k}n_{1,\ell}n_{-1,\ell'}}{n_k n_\ell n_{\ell'}} \quad \text{where} \quad n_k = \sum_{i=1}^n \mathbf{1}\{\hat{\xi}_i = k\}.$$

Construct pair set  $W = \{(\ell, \ell'), \ell, \ell' \in \{1, \dots, \hat{K}\} \mid \hat{\phi}_\ell = \hat{\phi}_{\ell'}\}$ . Estimate the vertex covariate effect as

$$\hat{\beta}_{\text{WA}} = \frac{1}{\hat{K}|W|} \sum_{k=1}^{\hat{K}} \sum_{(\ell, \ell') \in W} p_{k\ell, k\ell'} (\hat{B}_{Z,k\ell} - \hat{B}_{Z,k\ell'}).$$

- 3 Account for the vertex covariate effect by

$$\tilde{A}_{ij} = A_{ij} - \hat{\beta} \mathbf{1}\{Z_i = Z_j\},$$

where  $\hat{\beta}$  is either  $\hat{\beta}_{\text{SA}}$  or  $\hat{\beta}_{\text{WA}}$ .

- 4 Estimate latent positions after accounting for the vertex covariate effect as  $\tilde{\mathbf{Y}} \in \mathbb{R}^{n \times \tilde{d}}$  using ASE of  $\tilde{\mathbf{A}}$  where  $\tilde{d}$  is chosen as in Remark 7.
  - 5 Cluster  $\tilde{\mathbf{Y}}$  using GMM to estimate the induced block assignments as  $\tilde{\tau} \in \{1, \dots, \frac{\hat{K}}{2}\}^n$ .
-

# Spectral Inference Performance

## Chernoff Ratio

To analytically investigate the proposed algorithms, we employ Chernoff information to compare the performance of Algorithms 1 and 2 for estimating the induced block assignments in SBMs with vertex covariates. There are other metrics for comparing spectral inference performance such as within-class covariance. The advantages of Chernoff information are that it is independent of the clustering procedure, i.e., it can be derived no matter which clustering methods are used, and it is intrinsically related to the Bayes risk [12, 14, 28] as we discuss in Chapter 2.

In short, there will be a quantity associated with each algorithm, say  $\rho_1^*$  and  $\rho_2^*$  are associated with the Algorithms 1 and 2 respectively. The comparison is based on the ratio  $\rho^* = \frac{\rho_1^*}{\rho_2^*}$ . If  $\rho^* > 1$ , then Algorithm 1 is preferred, otherwise Algorithm 2 is preferred.

Formally by Eq. (2.29) and Eq. (2.30), we have the Chernoff ratio as

$$\rho^* = \frac{\rho_1^*}{\rho_2^*} \rightarrow \frac{\min_{k \neq \ell} \sup_{t \in (0,1)} \left[ t(1-t)(\boldsymbol{\nu}_{1,k} - \boldsymbol{\nu}_{1,\ell})^\top \boldsymbol{\Sigma}_{1,k\ell}^{-1}(t)(\boldsymbol{\nu}_{1,k} - \boldsymbol{\nu}_{1,\ell}) \right]}{\min_{k \neq \ell} \sup_{t \in (0,1)} \left[ t(1-t)(\boldsymbol{\nu}_{2,k} - \boldsymbol{\nu}_{2,\ell})^\top \boldsymbol{\Sigma}_{2,k\ell}^{-1}(t)(\boldsymbol{\nu}_{2,k} - \boldsymbol{\nu}_{2,\ell}) \right]}. \quad (3.2)$$

Here  $\rho_1^*$  and  $\rho_2^*$  are associated with the Algorithms 1 and 2 respectively. If  $\rho^* > 1$ , then Algorithm 1 is preferred, otherwise Algorithm 2 is preferred.

## 2-block Rank One Model with One Binary Covariate

As an illustration of using Chernoff ratio in Eq. (3.2) to compare the performance of Algorithms 1 and 2 for estimating the induced block assignments, we consider the 2-block SBM with one binary covariate as in Example 2.

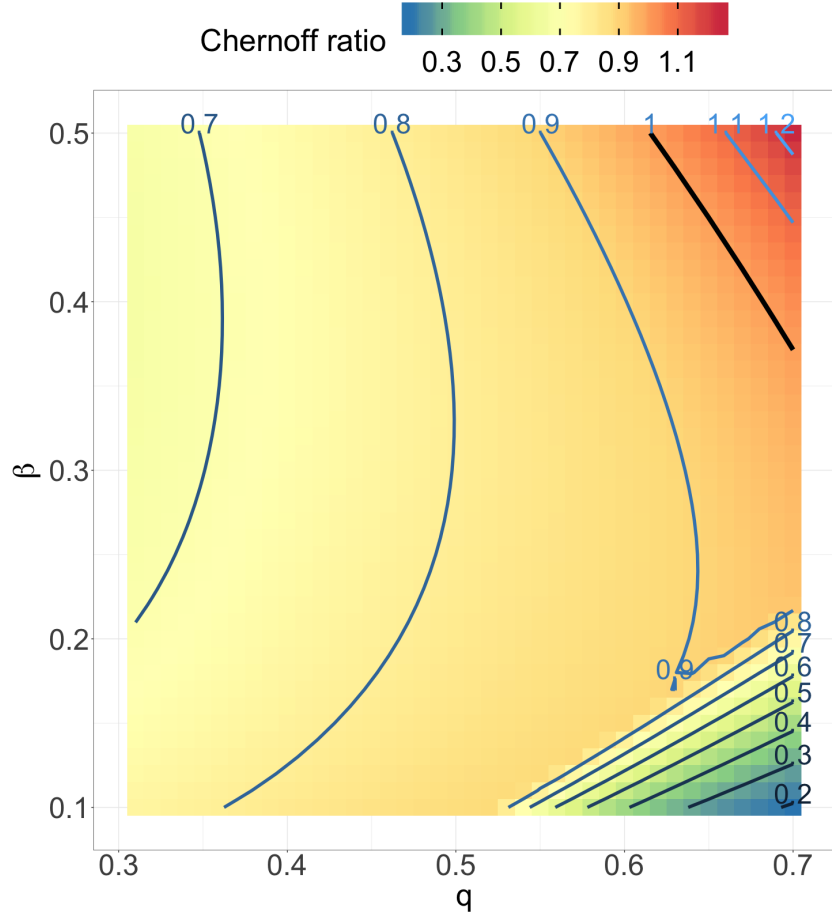
**Proposition 1.** *For 2-block rank one model with one binary covariate as in Example 2 with the assumption that  $n_i = n\pi_i$  and  $n_{Z,j} = n\pi_{Z,j}$  for  $i \in \{1, 2\}$  and  $j \in \{1, 2, 3, 4\}$  where  $\boldsymbol{\pi} = (\frac{1}{2}, \frac{1}{2})$  and  $\boldsymbol{\pi}_Z = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ , there is no tractable closed-form for Chernoff ratio as in Eq. (3.2) but numerical experiments can be used to obtain  $\rho_1^*$  and  $\rho_2^*$  can be derived analytically as*

$$\rho_2^* = \frac{(p - q)^2(p^2 + q^2)^2}{2 \left[ \sqrt{p^2\phi_p + q^2\phi_{pq}} + \sqrt{q^2\phi_q + p^2\phi_{pq}} \right]^2}, \quad (3.3)$$

where  $\sigma_p^2, \sigma_q^2$  are defined as in Eq. (2.17) and

$$\begin{aligned} \phi_p &= p^2(1 - p^2), \\ \phi_q &= q^2(1 - q^2), \\ \phi_{pq} &= pq(1 - pq). \end{aligned} \quad (3.4)$$

Technical details of Proposition 1 can be found at the end of this chapter. Figure 3-1 shows the Chernoff ratio as in Eq. (3.2) when we fix  $p = 0.3$  and take  $q \in (0.3, 0.7), \beta \in (0.1, 0.5)$  in the two-block rank one models with one binary covariate as in Example 2. Observe that  $\rho^* < 1$  for most of the region while  $\rho^* > 1$  only when  $q$  and  $\beta$  are relatively large. Recall that the performance of Algorithm 1 highly depends on the estimated block connectivity probability matrix  $\hat{\mathbf{B}}_Z$ . Large  $q$  and  $\beta$  lead to a relatively well-structured  $\hat{\mathbf{B}}_Z$  and thus Algorithm 1 can have better performance in this region.



**Figure 3-1.** Chernoff ratio as in Eq. (3.2) for 2-block rank one model with one binary covariate as in Example 2.  $p = 0.3, q \in (0.3, 0.7), \beta \in (0.1, 0.5), \pi = (\frac{1}{2}, \frac{1}{2}), \pi_Z = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .

## 2-block Homogeneous Model with One Binary Covariate

Now we consider the 2-block SBM with one binary covariate parameterized by the block connectivity probability matrix  $\mathbf{B}_Z$  as in Eq. (2.10).

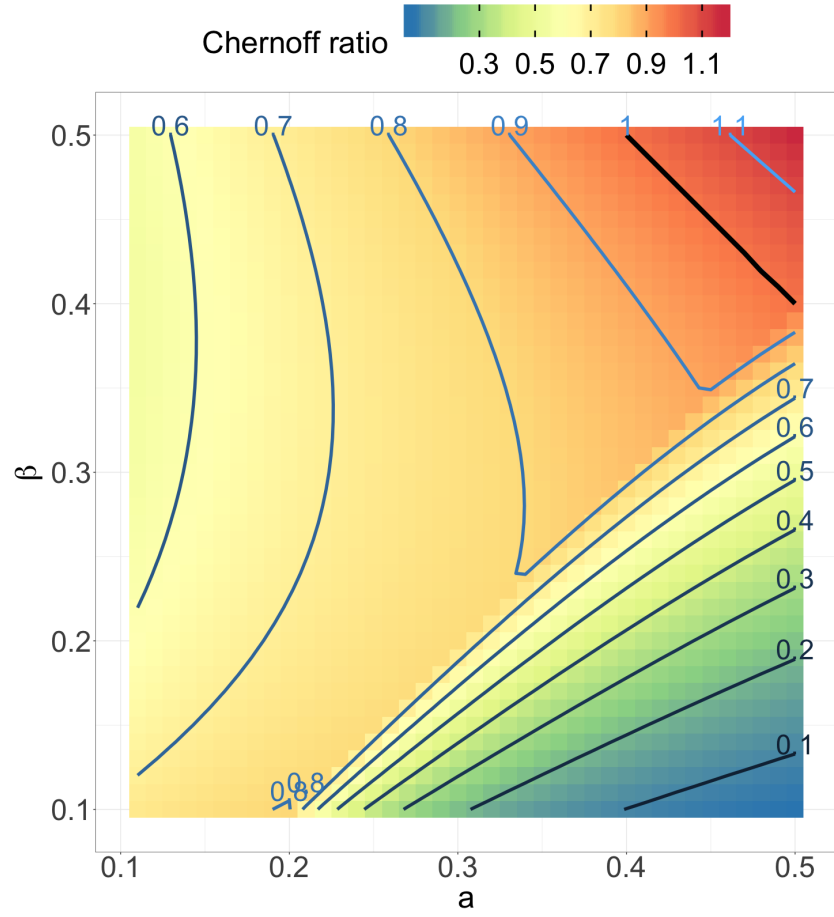
**Corollary 1.** *For 2-block homogeneous model with one binary covariate as in Example 3 with the assumption that  $n_i = n\pi_i$  and  $n_{Z,j} = n\pi_{Z,j}$  for  $i \in \{1, 2\}$  and  $j \in \{1, 2, 3, 4\}$  where  $\boldsymbol{\pi} = (\frac{1}{2}, \frac{1}{2})$  and  $\boldsymbol{\pi}_Z = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ . The Chernoff ratio as in Eq. (3.2) can be derived analytically as*

$$\rho^* = \frac{\rho_1^*}{\rho_2^*} \rightarrow \begin{cases} \frac{\beta^2(\phi_a + \phi_b)}{(a-b)^2(\phi_a + \phi_b + \phi_\beta)} & \text{if } \beta \leq a - b \\ \frac{\phi_a + \phi_b}{\phi_a + \phi_b + \phi_\beta} & \text{if } \beta > a - b \end{cases}, \quad (3.5)$$

where

$$\begin{aligned} \phi_a &= a(1 - a), \\ \phi_b &= b(1 - b), \\ \phi_\beta &= \beta(1 - a - b - \beta). \end{aligned} \quad (3.6)$$

Technical details of Corollary 1 can be found at the end of this chapter. Figure 3-2 shows Chernoff ratio as in Eq. (3.2) when we fix  $b = 0.1$  and take  $a \in (0.1, 0.5), \beta \in (0.1, 0.5)$  in the two-block homogeneous models with one binary covariate as in Example 3. Again observe that  $\rho^* < 1$  for most of the region while  $\rho^* > 1$  only when  $a$  and  $\beta$  are relatively large, which agrees with the general expression for Chernoff ratio as in Corollary 1. According to Eq. (3.5), we can have  $\rho^* > 1$  only when  $\phi_\beta < 0$  and this can happen only when  $a$  and  $\beta$  are relatively large. This implies that Algorithm 2 is often preferred for estimating the induced block assignments in terms of Chernoff information.



**Figure 3-2.** Chernoff ratio as in Eq. (3.2) for 2-block homogeneous model with one binary covariate as in Example 3.  $b = 0.1, a \in (0.1, 0.5), \beta \in (0.1, 0.5), \pi = (\frac{1}{2}, \frac{1}{2}), \pi_Z = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .

## **$K$ -block Homogeneous Model with One Binary Covariate**

We then extend the discussion from the two-block homogeneous model to the  $K$ -block homogeneous model with one binary covariate.

**Theorem 2.** *For the  $K$ -block homogeneous balanced model with one binary covariate as in Remark 6 with the assumption that that  $n_i = n\pi_i$  and  $n_{Z,j} = n\pi_{Z,j}$  for  $i \in \{1, \dots, K\}$  and  $j \in \{1, \dots, 2K\}$  where  $\boldsymbol{\pi} = (\frac{1}{K}, \dots, \frac{1}{K})$  and  $\boldsymbol{\pi}_Z = (\frac{1}{2K}, \dots, \frac{1}{2K})$ . The Chernoff ratio as in Eq. (3.2) can be derived analytically as*

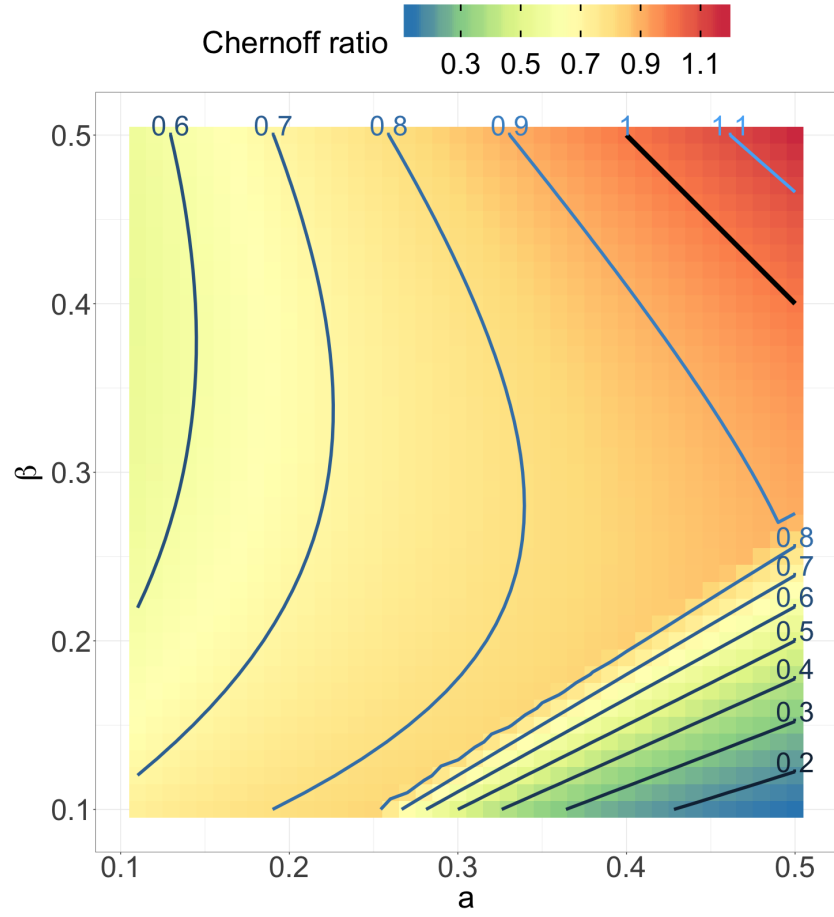
$$\rho^* = \frac{\rho_1^*}{\rho_2^*} \rightarrow \begin{cases} \frac{K^2\beta^2(\phi_a+\phi_b)}{2(a-b)^2D_4} & \text{if } \delta \leq 0 \\ \frac{\phi_a+\phi_b}{\phi_a+\phi_b+\phi_\beta} & \text{if } \delta > 0 \end{cases}, \quad (3.7)$$

where  $\phi_a, \phi_b, \phi_\beta$  are defined as in Eq. (3.6) and

$$\begin{aligned} D_3 &= K - 2a - 2(K-1)b - K\beta, \\ D_4 &= 2\phi_a + 2(K-1)\phi_b + \beta D_3, \\ \delta &= K^2\beta^2(\phi_a + \phi_b + \phi_\beta) - 2(a-b)^2D_4. \end{aligned} \quad (3.8)$$

**Remark 12.** *Theorem 2 generalizes Corollary 1 beyond  $K = 2$ .*

Technical details of Theorem 2 can be found at the end of this chapter. Figure 3-3 shows Chernoff ratio as in Eq. (3.2) when we fix  $b = 0.1$  and take  $a \in (0.1, 0.5), \beta \in (0.1, 0.5)$  in the 4-block homogeneous models with one binary covariate as in Example 3 and Remark 6. Note that  $\rho^* < 1$  for most of the region while  $\rho^* > 1$  only when  $a$  and  $\beta$  are relatively large. This implies again that Algorithm 2 is often preferred for estimating the induced block assignments in terms of Chernoff information.



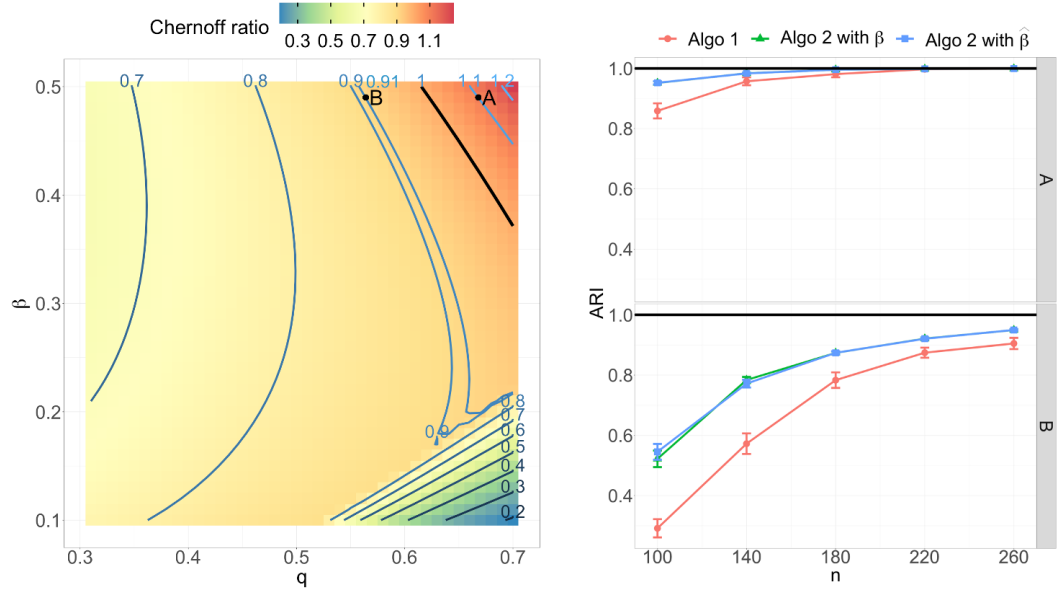
**Figure 3-3.** Chernoff ratio as in Eq. (3.2) for 4-block homogeneous model with one binary covariate as in Example 3 and Remark 6.  $b = 0.1, a \in (0.1, 0.5), \beta \in (0.1, 0.5), \boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}), \boldsymbol{\pi}_Z = (\frac{1}{8}, \dots, \frac{1}{8})$ .

## Experiments

In addition to comparing the two algorithms' performance analytically via the notion of Chernoff ratio, we also compare Algorithms 1 and 2 by empirical clustering results. Recall that the analytic comparison via Chernoff ratio is based on the limiting results of ASE for SBM when the number of vertices  $n \rightarrow \infty$ . The comparison via empirical clustering results can measure the performance of these two algorithms for finite  $n$ .

As an illustration of this correspondence, we start with the setting related to “A” ( $p = 0.3, q = 0.668, \beta = 0.49$  with  $\rho^* = 1.1 > 1$ ) and “B” ( $p = 0.3, q = 0.564, \beta = 0.49$  with  $\rho^* = 0.91 < 1$ ) in left panel of Figure 3-4 for 2-block rank one model with one binary covariate  $\mathbf{Z} \in \{1, 2\}^n$  as in Example 2. We consider the balanced case where  $n_1 = n_2 = \frac{n}{2}$  and  $n_{Z,1} = n_{Z,2} = n_{Z,3} = n_{Z,4} = \frac{n}{4}$ . For each  $n \in \{100, 140, 180, 220, 260\}$ , we simulate 100 adjacency matrices with  $\frac{n}{2}$  vertices in each block and generate binary covariate with  $\frac{n}{4}$  vertices having each value of  $\mathbf{Z}$  within each block. We then apply Algorithms 1 and 2 (with  $\beta$  and  $\hat{\beta}$  in Step 3 respectively) using embedding dimension  $\hat{d} = 3$  to estimate the induced block assignments where adjusted Rand index (ARI) [90] is used to measure the performance (ARI can take values from  $-1$  to  $1$  where larger value indicates a better alignment of the empirical clustering and the “truth”).

The upper right panel in Figure 3-4 shows that although  $\rho^* > 1$  and Algorithm 1 should be preferred in terms of Chernoff ratio, the ARI suggests that Algorithm 2 is preferred. While the Chernoff ratio is, in fact, a limit (computed as the sample size  $n$  increases to infinity), the region for which  $\rho^* > 1$  is so easy for clustering—e.g.,  $q - p$  is large for “A”—that both algorithms are essentially perfect even for small  $n$ . The lower right panel in Figure 3-4 shows that Algorithm 2 tends to have better performance than Algorithm 1, which agrees with the Chernoff ratio as in left figure where  $\rho^* < 1$  and Algorithm 2 is preferred.



**Figure 3-4.** Correspondence between Chernoff analysis and simulations.

Table 3-I summarizes the detailed simulation results associated with Fig. 3-4 right panel for correspondence between Chernoff analysis and simulations.

**Table 3-I.** Detailed simulation results associated with Fig. 3-4 right panel

$n$ <sup>1</sup>	$p$ <sup>2</sup>	$q$ <sup>2</sup>	$\beta$ <sup>3</sup>	$\hat{\beta}$ <sup>4</sup>	ARI (Algo 1) <sup>5</sup>	ARI (Algo 2 with $\beta$ ) <sup>5</sup>	ARI (Algo 2 with $\hat{\beta}$ ) <sup>5</sup>
100	0.3	0.668	0.49	0.489	0.858 ( $\pm 0.025$ )	0.951 ( $\pm 0.005$ )	<b>0.952 (<math>\pm 0.005</math>)</b>
140	0.3	0.668	0.49	0.487	0.957 ( $\pm 0.013$ )	<b>0.983 (<math>\pm 0.002</math>)</b>	<b>0.983 (<math>\pm 0.002</math>)</b>
180	0.3	0.668	0.49	0.488	0.980 ( $\pm 0.010$ )	<b>0.995 (<math>\pm 0.000</math>)</b>	<b>0.995 (<math>\pm 0.000</math>)</b>
220	0.3	0.668	0.49	0.489	0.997 ( $\pm 0.000$ )	<b>0.998 (<math>\pm 0.000</math>)</b>	<b>0.998 (<math>\pm 0.000</math>)</b>
260	0.3	0.668	0.49	0.489	<b>0.999 (<math>\pm 0.000</math>)</b>	<b>0.999 (<math>\pm 0.000</math>)</b>	<b>0.999 (<math>\pm 0.000</math>)</b>
100	0.3	0.564	0.49	0.478	0.291 ( $\pm 0.030$ )	0.522 ( $\pm 0.027$ )	<b>0.545 (<math>\pm 0.026</math>)</b>
140	0.3	0.564	0.49	0.485	0.572 ( $\pm 0.034$ )	<b>0.783 (<math>\pm 0.010</math>)</b>	0.771 ( $\pm 0.013$ )
180	0.3	0.564	0.49	0.486	0.783 ( $\pm 0.025$ )	0.873 ( $\pm 0.004$ )	<b>0.874 (<math>\pm 0.005</math>)</b>
220	0.3	0.564	0.49	0.490	0.874 ( $\pm 0.016$ )	<b>0.921 (<math>\pm 0.003</math>)</b>	0.920 ( $\pm 0.003$ )
260	0.3	0.564	0.49	0.489	0.905 ( $\pm 0.018$ )	<b>0.949 (<math>\pm 0.003</math>)</b>	<b>0.949 (<math>\pm 0.003</math>)</b>

<sup>1</sup> Number of vertices in the simulated adjacency matrices.

<sup>2</sup> Adjacency matrices are simulated from a two-block SBM where vertices within block 1 connect with probability  $p^2$ , vertices within block 2 connect with probability  $q^2$ , and vertices across two blocks connect with probability  $pq$ .

<sup>3</sup> Vertex covariate effect, see Definition 4 and Remark 5 for details.

<sup>4</sup> Estimated vertex covariate effect by Algorithm 2.

<sup>5</sup> Reported as  $\text{mean}(\pm \text{stderr})$  from 100 trials. Best results in bold.

## 2-block Rank One Model with One 5-categorical Covariate

To further investigate the flexibility of our models and algorithms, we also consider categorical vertex covariates in the simulations. We begin with the 2-block rank one model with one 5-categorical covariate  $\mathbf{Z} \in \{1, 2, 3, 4, 5\}^n$ , i.e., we have the block connectivity probability matrix  $\mathbf{B}_Z \in [0, 1]^{10 \times 10}$  with similar structure as in Eq. (2.9).

We first fix  $p = 0.3, \beta = 0.4$  and consider  $q \in \{0.35, 0.375, 0.4, 0.425, 0.45\}$ . For each  $q$ , we simulate 100 adjacency matrices with 1000 vertices in each block and generate 5-categorical covariate with 200 vertices having each value of  $\mathbf{Z}$  within each block. We then apply Algorithms 1 and 2 (with  $\beta$  and  $\hat{\beta}$  in Step 3 respectively) using embedding dimension  $\hat{d} = 6$  to estimate the induced block assignments. Figure 3-5a shows that both algorithms estimate more accurate induced block assignments as the latent positions of two induced block move away from each other, i.e., two induced blocks tend to be more separate, and Algorithm 2 can have better performance than Algorithm 1 in terms of empirical clustering results measured by ARI.

Next we fix  $p = 0.3, q = 0.375$  and consider  $\beta \in \{0.1, 0.15, 0.2, 0.25, 0.3\}$ . For each  $\beta$ , we simulate 100 adjacency matrices with 1000 vertices in each block and generate 5-categorical covariate with 200 vertices having each value of  $\mathbf{Z}$  within each block. We then apply both algorithms (with  $\beta$  and  $\hat{\beta}$  in Step 3 of Algorithm 2 respectively) using embedding dimension  $\hat{d} = 6$  to estimate the induced block assignments. Figure 3-5b shows Algorithm 1 can only estimate accurate induced block assignments when  $\beta$  is relatively small while Algorithm 2 can estimate accurate induced block assignments no matter  $\beta$  is small or large. Intuitively, as Algorithm 1 directly estimates the induced block assignments, when  $\beta$  is relatively large, i.e., vertex covariates can affect block structure significantly, it lacks the ability to distinguish this effect. However, Algorithm 2 can use additional information from vertex covariates to estimate  $\beta$ , taking this effect into consideration when estimating the induced block assignments.

Again, the overall performance of Algorithm 2 is better than that of Algorithm 1 in terms of empirical clustering results measured by ARI.

Table 3-II summarizes the detailed simulation results associated with Fig. 3-5 for 2-block rank one model with one 5-categorical covariate.

**Table 3-II.** Detailed simulation results associated with Fig. 3-5

$n$ <sup>1</sup>	$p$ <sup>2</sup>	$q$ <sup>2</sup>	$\beta$ <sup>3</sup>	$\hat{\beta}$ <sup>4</sup>	ARI (Algo 1) <sup>5</sup>	ARI (Algo 2 with $\beta$ ) <sup>5</sup>	ARI (Algo 2 with $\hat{\beta}$ ) <sup>5</sup>
2000	0.3	0.350	0.4	0.401	0.013 ( $\pm$ 0.000)	0.388 ( $\pm$ 0.012)	<b>0.395</b> ( $\pm$ <b>0.010</b> )
2000	0.3	0.375	0.4	0.400	0.283 ( $\pm$ 0.030)	<b>0.763</b> ( $\pm$ <b>0.002</b> )	<b>0.763</b> ( $\pm$ <b>0.002</b> )
2000	0.3	0.400	0.4	0.399	0.855 ( $\pm$ 0.014)	<b>0.931</b> ( $\pm$ <b>0.000</b> )	<b>0.931</b> ( $\pm$ <b>0.000</b> )
2000	0.3	0.425	0.4	0.399	0.967 ( $\pm$ 0.007)	<b>0.985</b> ( $\pm$ <b>0.000</b> )	<b>0.985</b> ( $\pm$ <b>0.000</b> )
2000	0.3	0.450	0.4	0.399	0.990 ( $\pm$ 0.005)	<b>0.998</b> ( $\pm$ <b>0.000</b> )	<b>0.998</b> ( $\pm$ <b>0.000</b> )
2000	0.3	0.375	0.10	0.097	0.774 ( $\pm$ 0.007)	<b>0.803</b> ( $\pm$ <b>0.002</b> )	0.802 ( $\pm$ 0.002)
2000	0.3	0.375	0.15	0.149	0.636 ( $\pm$ 0.026)	0.789 ( $\pm$ 0.001)	<b>0.790</b> ( $\pm$ <b>0.001</b> )
2000	0.3	0.375	0.20	0.199	0.452 ( $\pm$ 0.036)	0.779 ( $\pm$ 0.002)	<b>0.780</b> ( $\pm$ <b>0.002</b> )
2000	0.3	0.375	0.25	0.249	0.465 ( $\pm$ 0.034)	<b>0.770</b> ( $\pm$ <b>0.002</b> )	<b>0.770</b> ( $\pm$ <b>0.002</b> )
2000	0.3	0.375	0.30	0.300	0.340 ( $\pm$ 0.034)	0.766 ( $\pm$ 0.002)	<b>0.767</b> ( $\pm$ <b>0.001</b> )

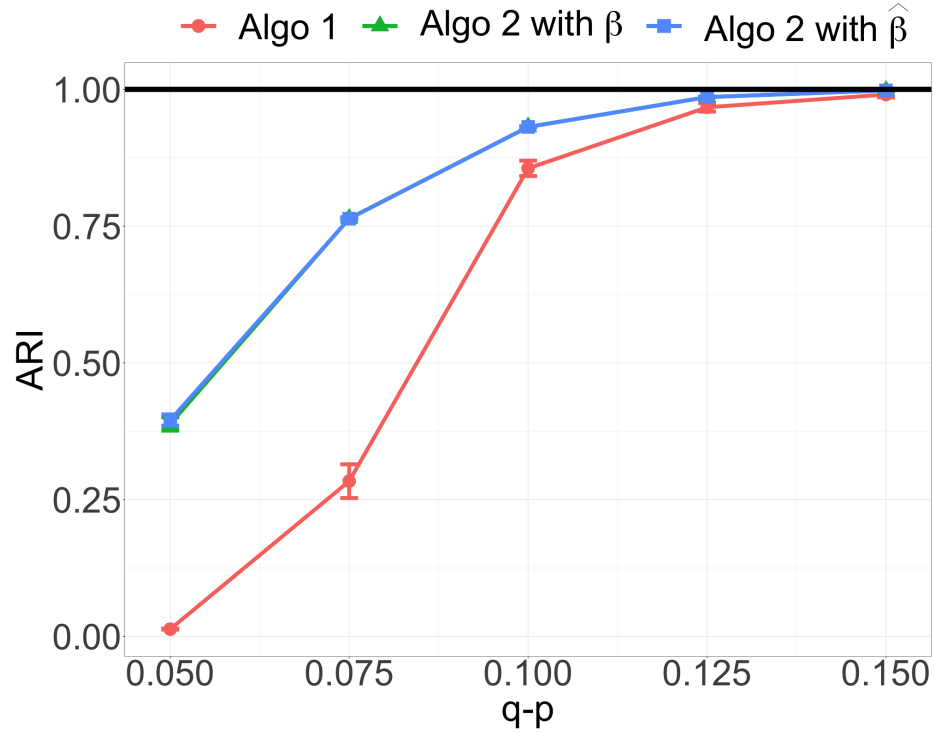
<sup>1</sup> Number of vertices in the simulated adjacency matrices.

<sup>2</sup> Adjacency matrices are simulated from a two-block SBM where vertices within block 1 connect with probability  $p^2$ , vertices within block 2 connect with probability  $q^2$ , and vertices across two blocks connect with probability  $pq$ .

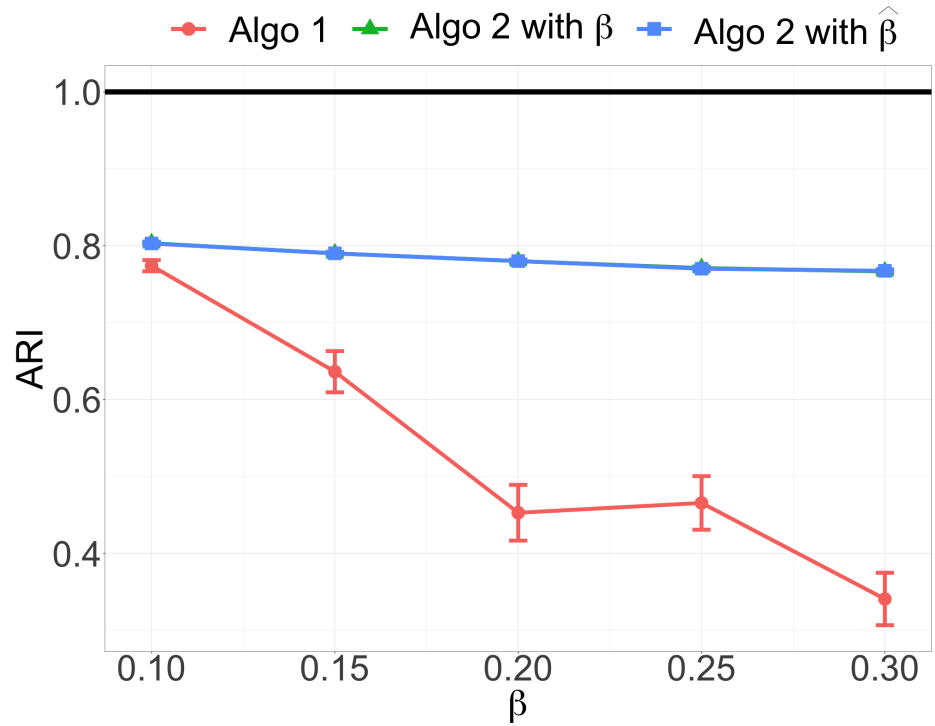
<sup>3</sup> Vertex covariate effect, see Definition 4 and Remark 5 for details.

<sup>4</sup> Estimated vertex covariate effect by Algorithm 2.

<sup>5</sup> Reported as **mean**( $\pm$ **stderr**) from 100 trials. Best results in bold.



(a) ARI as latent positions of two induced blocks move away from each other with  $\beta = 0.4$ .



(b) ARI as  $\beta$  increases with  $p = 0.3, q = 0.375$ .

**Figure 3-5.** Simulations for 2-block rank one model with one 5-categorical covariate, balanced case.

## 2-block Homogeneous Model with One 5-categorical Covariate

We now consider the 2-block homogeneous model with one 5-categorical covariate  $\mathbf{Z} \in \{1, 2, 3, 4, 5\}^n$ , i.e., we have the block connectivity probability matrix  $\mathbf{B}_Z \in [0, 1]^{10 \times 10}$  with the similar structure as in Eq. (2.10). Note that we can re-write  $\mathbf{B}$  like Eq. (2.6) as

$$\mathbf{B} = \boldsymbol{\nu} \boldsymbol{\nu}^\top = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \quad \text{with} \quad \boldsymbol{\nu} = \begin{bmatrix} \sqrt{a} & 0 \\ \frac{b}{\sqrt{a}} & \sqrt{\frac{(a-b)(a+b)}{a}} \end{bmatrix}. \quad (3.9)$$

With these canonical latent positions, the distance between two induced blocks can be measured by

$$\left( \sqrt{a} - \frac{b}{\sqrt{a}} \right)^2 + \left( 0 - \sqrt{\frac{(a-b)(a+b)}{a}} \right)^2 = 2(a-b). \quad (3.10)$$

We first fix  $b = 0.1, \beta = 0.2$  and consider  $a \in \{0.12, 0.125, 0.13, 0.135, 0.14\}$ . For each  $a$ , we simulate 100 adjacency matrices with 1000 vertices in each block and generate 5-categorical covariate with 200 vertices having each value of  $\mathbf{Z}$  within each block. We then apply both algorithms (with  $\beta$  and  $\hat{\beta}$  in Step 3 of Algorithm 2 respectively) using embedding dimension  $\hat{d} = 6$  to estimate the induced block assignments. Figure 3-6a shows that both algorithms estimate more accurate induced block assignments as the latent positions of two induced block move away from each other, i.e., two induced blocks tend to be more separate as measured by Eq. (3.10), and Algorithm 2 can have much better performance in terms of empirical clustering results measured by ARI. Recall that Algorithm 1 tries to estimate the induced block assignments by clustering the diagonal of  $\hat{\mathbf{B}}_Z$  and re-assigning the block assignments including the vertex covariate effect. For the homogeneous model, the diagonal of  $\mathbf{B}_Z$  are all the same, which can make it hard for Algorithm 1 to accurately estimate the induced block assignments. But Algorithm 2 is not affected by the homogeneous structure since it estimates the vertex covariate effect first and then estimates the induced block assignments by clustering the estimated latent positions like the canonical ones in Eq. (3.9).

Next we fix  $a = 0.135$ ,  $b = 0.1$  and consider  $\beta \in \{-0.09, -0.08, -0.07, -0.06, -0.05\}$ . For each  $\beta$ , we also simulate 100 adjacency matrices with 1000 vertices in each block and generate 5-categorical covariate with 200 vertices having each value of  $\mathbf{Z}$  within each block. We then apply both algorithms (with  $\beta$  and  $\hat{\beta}$  in Step 3 of Algorithm 2 respectively) using embedding dimension  $\hat{d} = 6$  to estimate the induced block assignments. Figure 3-6b shows that both algorithms are relative stable for this homogeneous model if we fix  $a$  and  $b$ , due to the special structure. Still, Algorithm 2 can have much better performance than Algorithm 1 in terms of empirical clustering results measured by ARI.

Table 3-III summarizes the detailed simulation results associated with Fig. 3-6 for 2-block homogeneous model with one 5-categorical covariate.

**Table 3-III.** Detailed simulation results associated with Fig. 3-6

$n$ <sup>1</sup>	$a$ <sup>2</sup>	$b$ <sup>2</sup>	$\beta$ <sup>3</sup>	$\hat{\beta}$ <sup>4</sup>	ARI (Algo 1) <sup>5</sup>	ARI (Algo 2 with $\beta$ ) <sup>5</sup>	ARI (Algo 2 with $\hat{\beta}$ ) <sup>5</sup>
2000	0.120	0.1	0.2	0.201	0.000 ( $\pm$ 0.000)	0.228 ( $\pm$ 0.014)	<b>0.230</b> ( $\pm$ <b>0.014</b> )
2000	0.125	0.1	0.2	0.200	0.001 ( $\pm$ 0.000)	<b>0.608</b> ( $\pm$ <b>0.006</b> )	0.583 ( $\pm$ 0.013)
2000	0.130	0.1	0.2	0.200	0.039 ( $\pm$ 0.007)	<b>0.790</b> ( $\pm$ <b>0.008</b> )	0.789 ( $\pm$ 0.008)
2000	0.135	0.1	0.2	0.199	0.058 ( $\pm$ 0.008)	<b>0.893</b> ( $\pm$ <b>0.009</b> )	<b>0.893</b> ( $\pm$ <b>0.009</b> )
2000	0.140	0.1	0.2	0.199	0.075 ( $\pm$ 0.010)	<b>0.954</b> ( $\pm$ <b>0.000</b> )	<b>0.954</b> ( $\pm$ <b>0.000</b> )
2000	0.135	0.1	-0.09	-0.088	0.098 ( $\pm$ 0.015)	<b>0.969</b> ( $\pm$ <b>0.000</b> )	<b>0.969</b> ( $\pm$ <b>0.000</b> )
2000	0.135	0.1	-0.08	-0.077	0.088 ( $\pm$ 0.012)	<b>0.967</b> ( $\pm$ <b>0.000</b> )	<b>0.967</b> ( $\pm$ <b>0.000</b> )
2000	0.135	0.1	-0.07	-0.065	0.073 ( $\pm$ 0.011)	<b>0.965</b> ( $\pm$ <b>0.000</b> )	<b>0.965</b> ( $\pm$ <b>0.000</b> )
2000	0.135	0.1	-0.06	-0.052	0.083 ( $\pm$ 0.013)	<b>0.963</b> ( $\pm$ <b>0.000</b> )	<b>0.963</b> ( $\pm$ <b>0.000</b> )
2000	0.135	0.1	-0.05	-0.037	0.069 ( $\pm$ 0.008)	<b>0.960</b> ( $\pm$ <b>0.000</b> )	<b>0.960</b> ( $\pm$ <b>0.000</b> )

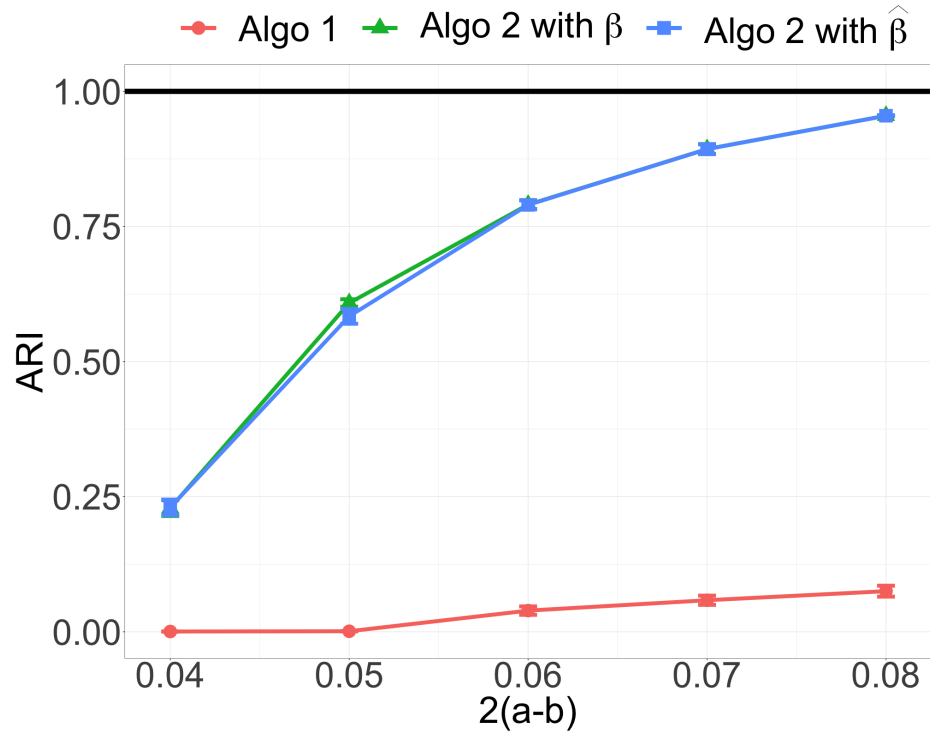
<sup>1</sup> Number of vertices in the simulated adjacency matrices.

<sup>2</sup> Adjacency matrices are simulated from a two-block SBM where vertices from the same block connect with probability  $a$ , and vertices across two blocks connect with probability  $b$ .

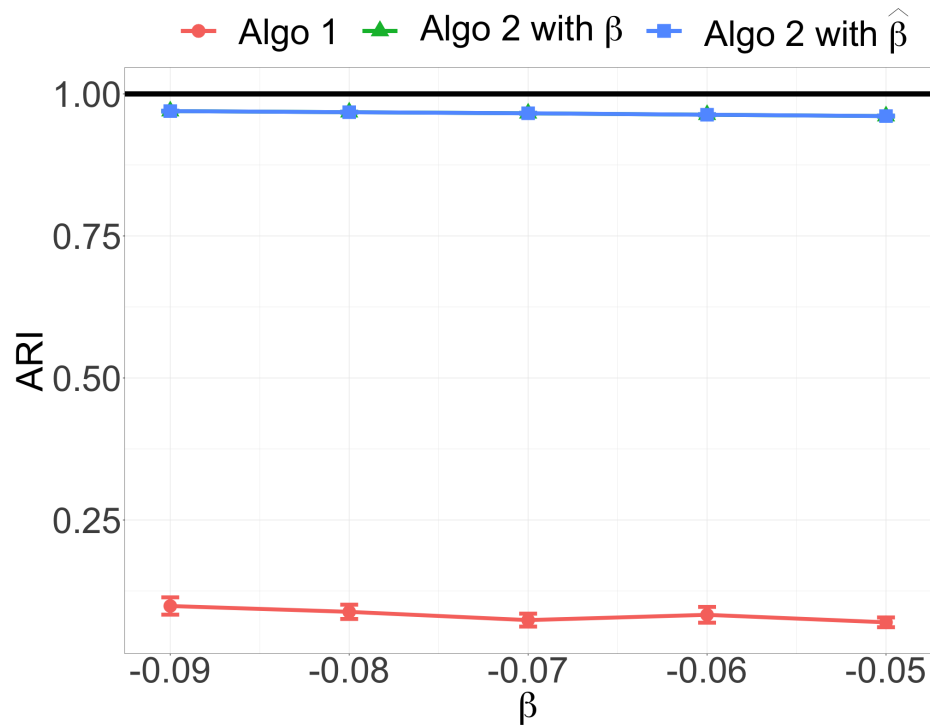
<sup>3</sup> Vertex covariate effect, see Definition 4 and Remark 5 for details.

<sup>4</sup> Estimated vertex covariate effect by Algorithm 2.

<sup>5</sup> Reported as `mean( $\pm$ stderr)` from 100 trials. Best results in bold.



(a) ARI as latent positions of two induced blocks move away from each other with  $\beta = 0.2$ .



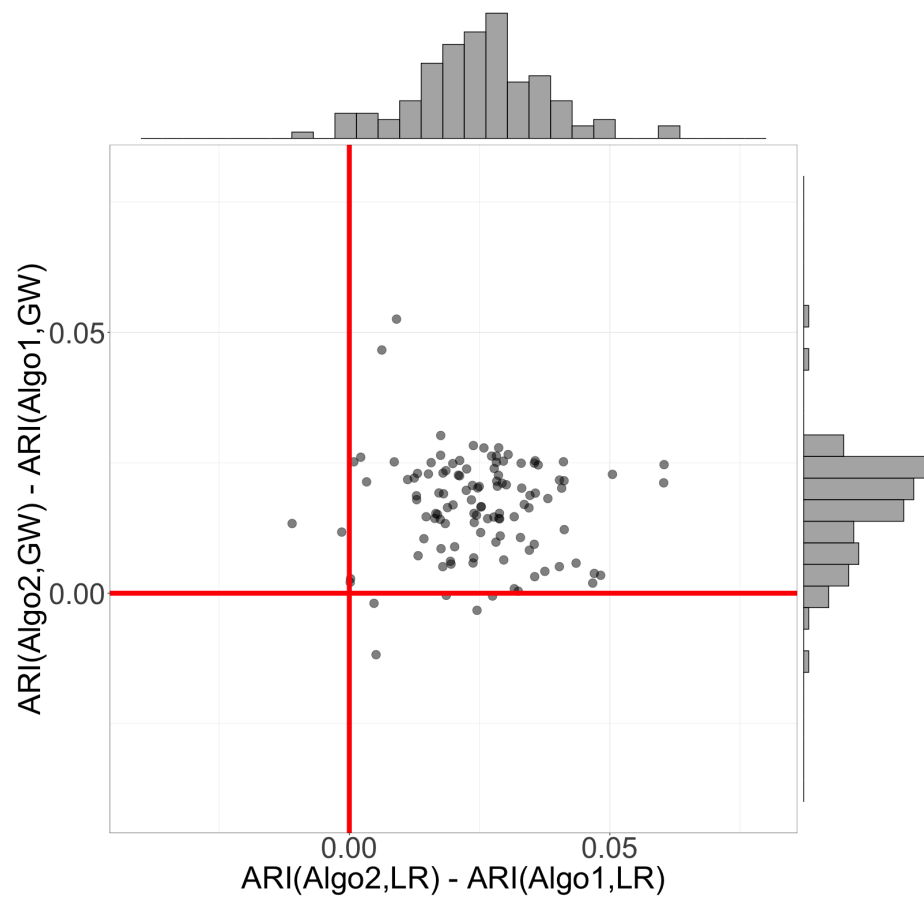
(b) ARI as  $\beta$  increases with  $a = 0.135, b = 0.1$ .

**Figure 3-6.** Simulations for 2-block homogeneous model with one 5-categorical covariate, balanced case.

## Connectome Data

Now we move from simulations to real data experiments. We start with the diffusion MRI connectome datasets [1]. There are 114 graphs (connectomes) estimated by the NDMG pipeline [91] in this data set where vertices represent brain sub-regions defined via spatial proximity and edges represent tensor-based fiber streamlines connecting these sub-regions. Each vertex in these graphs also has a  $\{\text{Left}, \text{Right}\}$  hemisphere label and a  $\{\text{Gray}, \text{White}\}$  tissue label. We treat one label as the induced block and the other one as the vertex covariate.

Each of the 114 connectomes (the number of vertices  $n$  varies from 23728 to 42022) is represented by a point in Figure 3-7 with  $x = \text{ARI}(\text{Algo2}, \text{LR}) - \text{ARI}(\text{Algo1}, \text{LR})$  and  $y = \text{ARI}(\text{Algo2}, \text{GW}) - \text{ARI}(\text{Algo1}, \text{GW})$  where  $\text{ARI}(\text{Algo1}, \text{LR})$  denotes the ARI when we apply Algorithm 1 and treat  $\{\text{Left}, \text{Right}\}$  as the induced block (with analogous notation for the rest). We see that most of the points lie in the  $(+, +)$  quadrant, indicating  $\text{ARI}(\text{Algo2}, \text{LR}) > \text{ARI}(\text{Algo1}, \text{LR})$  and  $\text{ARI}(\text{Algo2}, \text{GW}) > \text{ARI}(\text{Algo1}, \text{GW})$ . That is, Algorithm 2 is better at estimating the induced block assignments for this real application. Note that this claim holds no matter which label is treated as the induced block. This again emphasizes the importance of distinguishing different factors that can affect block structure in graphs. Algorithm 2 is able to identify particular block structure by using the observed vertex covariate information. That is, it is more likely to discover the  $\{\text{Left}, \text{Right}\}$  structure after accounting for the effect of  $\{\text{Gray}, \text{White}\}$  label and more likely to discover the  $\{\text{Gray}, \text{White}\}$  structure after accounting for the effect of  $\{\text{Left}, \text{Right}\}$  label.



**Figure 3-7.** Algorithms' comparative performance on connectome data.

## Social Network Data

We then utilize three social network datasets to compare our methods with several existing methods that also incorporate vertex covariates and can be scaled to deal with relatively large networks. Specifically, we compare with spectral clustering with adjacency matrix only (SCA) and spectral clustering with covariates only (SCC) [18], pairwise covariates-adjusted stochastic blockmodel via maximum likelihood estimation (PCABM.MLE) and spectral clustering with adjustment (PCABM.SCWA) [86], covariate-assisted spectral clustering (CASC) [85]. The description of these datasets are summarized as follows.

- LastFM asia social network dataset [2, 4]: there are 7624 vertices that represent LastFM users from asian countries and 27806 edges that represent mutual follower relationships. We treat the location of users, which are derived from the country field for each user, as the induced block. For the vertex covariate, we focus on the number of artists liked by users, which is discretized into four categories  $\{0-200, 200-400, 400-600, 600+\}$ .
- Facebook large page-page network dataset [2, 3]: there are 22470 vertices that represent official Facebook pages and 171002 edges that represent mutual likes. We treat four page types  $\{\text{Politician, Governmental Organization, Television Show, Company}\}$ , which are defined by Facebook, as the induced block. For the vertex covariate, we focus on the number of descriptions created by page owners to summarize the purpose of the site, which is discretized into two categories  $\{0-15, 15+\}$ .
- GitHub social network dataset [2, 3]: there are 37700 vertices that represent GitHub developers and 289003 edges that represent mutual follower relationships. We treat two developer types  $\{\text{Web, Machine Learning}\}$ , which are derived from the job title of each developer, as the induced block. For the vertex covariate, we

focus on the number of repositories starred by developers, which is discretized into two categories  $\{0-18, 18+\}$ .

Table 3-IV summarizes the algorithms’ comparative performances. Algorithm 2 is better at estimating the induced block assignments for all 3 datasets. This again suggests that we can better detect the block structure after accounting for the information contained in vertex covariates with our methods.

**Table 3-IV.** Algorithms’ performance on social network data in terms of ARI. Best results in bold.

	LastFM	Facebook	GitHub
SCA [18]	0.229	0.050	0.000
SCC [18]	0.012	0.038	0.001
PCABM.SCWA [86]	0.008	-0.002	0.000
PCABM.MLE [86]	0.000	0.004	-0.002
CASC [85]	0.020	0.053	-0.043
Algo 1 (ours)	0.090	0.036	0.001
Algo 2 (ours)	<b>0.297</b>	<b>0.076</b>	<b>0.013</b>

In real data, we may not have ground truth for the block structure. Our findings suggest that we are able to discover block structure by using observed vertex covariates, which can lead to meaningful insights in widely varying applications. That is, we can better reveal underlying block structure and thus better understand the data by accounting for the vertex covariate effect.

# Latent Position Geometry and Chernoff Ratio

We end this chapter with the investigation of the latent position geometry and the derivation of the Chernoff ratio for certain model of interest.

For convenience, we first introduce the following notations. Define for  $t \in (0, 1)$ ,

$$\begin{aligned} g(\boldsymbol{\nu}_k, \boldsymbol{\nu}) &= (\boldsymbol{\nu}_k^\top \boldsymbol{\nu}) (1 - \boldsymbol{\nu}_k^\top \boldsymbol{\nu}), \\ g_t(\boldsymbol{\nu}_k, \boldsymbol{\nu}_\ell, \boldsymbol{\nu}) &= t g(\boldsymbol{\nu}_k, \boldsymbol{\nu}) + (1 - t) g(\boldsymbol{\nu}_\ell, \boldsymbol{\nu}). \end{aligned} \quad (3.11)$$

Then we can re-write  $\boldsymbol{\Sigma}_k$  and  $\boldsymbol{\Sigma}_{k\ell}(t)$  as

$$\begin{aligned} \boldsymbol{\Sigma}_k &= \boldsymbol{\Delta}^{-1} \mathbb{E} [g(\boldsymbol{\nu}_k, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top] \boldsymbol{\Delta}^{-1}, \\ \boldsymbol{\Sigma}_{k\ell}(t) &= \boldsymbol{\Delta}^{-1} \mathbb{E} [g_t(\boldsymbol{\nu}_k, \boldsymbol{\nu}_\ell, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top] \boldsymbol{\Delta}^{-1}. \end{aligned} \quad (3.12)$$

We also define for  $1 \leq k < \ell \leq K$ ,

$$C_{k\ell} = \sup_{t \in (0,1)} t(1-t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell). \quad (3.13)$$

In addition, we review several useful results in linear algebra and matrix analysis that motivate our derivation of the Chernoff ratio for certain model of interest.

**Corollary 2** (Cholesky Decomposition [92]). *Let  $\mathbf{A}$  be a Hermitian matrix. Then  $\mathbf{A}$  is positive semi-definite (respectively, positive-definite) if and only if there exists a lower triangular matrix  $\mathbf{L}$  with nonnegative (respectively, positive) diagonal entries such that  $\mathbf{A} = \mathbf{L}\mathbf{L}^*$ . If  $\mathbf{A}$  is positive-definite, then  $\mathbf{L}$  is unique. If  $\mathbf{A}$  is real, then  $\mathbf{L}$  may be taken to be real.*

**Corollary 3** (Sherman–Morrison–Woodbury Formula [92]). *Let  $\mathbf{A}$  be a nonsingular matrix with a known inverse  $\mathbf{A}^{-1}$ . Let  $\mathbf{R}$  be a nonsingular matrix and consider  $\mathbf{B} = \mathbf{A} + \mathbf{X}\mathbf{R}\mathbf{Y}$ . If  $\mathbf{B}$  and  $\mathbf{R}^{-1} + \mathbf{Y}\mathbf{A}^{-1}\mathbf{X}$  are nonsingular, then*

$$\mathbf{B}^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{X} \left( \mathbf{R}^{-1} + \mathbf{Y}\mathbf{A}^{-1}\mathbf{X} \right)^{-1} \mathbf{Y}\mathbf{A}^{-1}. \quad (3.14)$$

## Proof of Proposition 1

*Proof.* Via the idea of Cholesky decomposition as in Corollary 2, we can re-write  $\mathbf{B}_Z$  as

$$\mathbf{B}_Z = \boldsymbol{\nu}_Z \boldsymbol{\nu}_Z^\top = \begin{bmatrix} \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_1 & \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_2 & \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_3 & \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_4 \\ \boldsymbol{\nu}_2^\top \boldsymbol{\nu}_1 & \boldsymbol{\nu}_2^\top \boldsymbol{\nu}_2 & \boldsymbol{\nu}_2^\top \boldsymbol{\nu}_3 & \boldsymbol{\nu}_2^\top \boldsymbol{\nu}_4 \\ \boldsymbol{\nu}_3^\top \boldsymbol{\nu}_1 & \boldsymbol{\nu}_3^\top \boldsymbol{\nu}_2 & \boldsymbol{\nu}_3^\top \boldsymbol{\nu}_3 & \boldsymbol{\nu}_3^\top \boldsymbol{\nu}_4 \\ \boldsymbol{\nu}_4^\top \boldsymbol{\nu}_1 & \boldsymbol{\nu}_4^\top \boldsymbol{\nu}_2 & \boldsymbol{\nu}_4^\top \boldsymbol{\nu}_3 & \boldsymbol{\nu}_4^\top \boldsymbol{\nu}_4 \end{bmatrix}, \quad (3.15)$$

where  $\boldsymbol{\nu}_Z = [\boldsymbol{\nu}_1 \ \boldsymbol{\nu}_2 \ \boldsymbol{\nu}_3 \ \boldsymbol{\nu}_4]^\top$ . Elementary calculations yield the canonical latent positions as

$$\boldsymbol{\nu}_Z = \begin{bmatrix} \sqrt{p^2 + \beta} & 0 & 0 \\ \frac{p^2}{\sqrt{p^2 + \beta}} & \sqrt{\frac{\beta(2p^2 + \beta)}{p^2 + \beta}} & 0 \\ \frac{pq + \beta}{\sqrt{p^2 + \beta}} & \sqrt{\frac{\beta p^2 (q - p)^2}{(p^2 + \beta)(2p^2 + \beta)}} & \sqrt{\frac{\beta(q - p)^2}{(2p^2 + \beta)}} \\ \frac{pq}{\sqrt{p^2 + \beta}} & \sqrt{\frac{\beta(p^2 + pq + \beta)^2}{(p^2 + \beta)(2p^2 + \beta)}} & \sqrt{\frac{\beta(q - p)^2}{(2p^2 + \beta)}} \end{bmatrix}. \quad (3.16)$$

For this model, the block connectivity probability matrix  $\mathbf{B}_Z$  is positive semi-definite with  $\text{rank}(\mathbf{B}_Z) = 3$ . Then we have  $\mathbf{I}_{d_+ d_-} = \mathbf{I}_3$  and we can omit it in our analytic derivations. With the canonical latent positions in Eq. (3.16), the only remaining term to derive for Chernoff ratio is  $\boldsymbol{\Sigma}_{kl}(t)$ .

By the symmetric structure of  $\mathbf{B}_Z$  and the balanced assumption, we observe that  $C_{13} = C_{24}, C_{14} = C_{23}$ . Thus we need only to evaluate  $C_{12}, C_{13}, C_{14}, C_{34}$ . Subsequent calculations and simplification yield

$$\begin{aligned} C_{12} &= \frac{\beta^2}{2[\phi_p + \phi_{pq} + \beta(1 - p^2 - pq - \beta)]}, \\ C_{34} &= \frac{\beta^2}{2[\phi_q + \phi_{pq} + \beta(1 - q^2 - pq - \beta)]}, \end{aligned} \quad (3.17)$$

where for  $0 < p < q < 1$

$$\begin{aligned}\phi_p &= p^2(1 - p^2), \\ \phi_q &= q^2(1 - q^2), \\ \phi_{pq} &= pq(1 - pq).\end{aligned}\tag{3.18}$$

Then we have the approximate Chernoff information for Algorithm 1 as

$$\rho_1^* \approx \min_{k \in \{1,3\}, k < \ell \leq 4} C_{k\ell},\tag{3.19}$$

where  $C_{k\ell}$  for  $k \in \{1,3\}, k < \ell \leq 4$  are defined as in Eq. (3.17). For this model, there is no tractable closed-form analytic expression for  $C_{13}$  and  $C_{14}$ , so we instead obtain values  $\rho_1^*$  by numerically solving the above optimization problem. By Remark 8 and similar calculations [12, 28], we have the approximate Chernoff information for Algorithm 2 as

$$\begin{aligned}\rho_2^* &\approx \sup_{t \in (0,1)} t(1-t)(p-q)^2 \left[ t\sigma_p^2 + (1-t)\sigma_q^2 \right]^{-1} \\ &= \frac{(p-q)^2(p^2+q^2)^2}{2 \left[ \sqrt{p^2\phi_p + q^2\phi_{pq}} + \sqrt{q^2\phi_q + p^2\phi_{pq}} \right]^2},\end{aligned}\tag{3.20}$$

where  $\phi_p, \phi_q, \phi_{pq}$  are defined as in Eq. (3.18) and

$$\begin{aligned}\sigma_p^2 &= \frac{\pi_1 p^4(1-p^2) + \pi_2 p q^3(1-pq)}{[\pi_1 p^2 + \pi_2 q^2]^2}, \\ \sigma_q^2 &= \frac{\pi_1 p^3 q(1-pq) + \pi_2 q^4(1-q^2)}{[\pi_1 p^2 + \pi_2 q^2]^2}.\end{aligned}\tag{3.21}$$

□

## Proof of Corollary 1

*Proof.* Similarly, the idea of Cholesky decomposition as in Corollary 2 and elementary calculations yield the canonical latent positions as

$$\boldsymbol{\nu}_Z = \begin{bmatrix} \sqrt{a+\beta} & 0 & 0 \\ \frac{a}{\sqrt{a+\beta}} & \sqrt{\frac{\beta(2a+\beta)}{a+\beta}} & 0 \\ \frac{b+\beta}{\sqrt{a+\beta}} & \sqrt{\frac{\beta(b-a)^2}{(a+\beta)(2a+\beta)}} & \sqrt{\frac{2(a-b)(a+b+\beta)}{(2a+\beta)}} \\ \frac{b}{\sqrt{a+\beta}} & \sqrt{\frac{\beta(a+b+\beta)^2}{(a+\beta)(2a+\beta)}} & \sqrt{\frac{2(a-b)(a+b+\beta)}{(2a+\beta)}} \end{bmatrix}. \quad (3.22)$$

Observe that for this model, the block connectivity probability matrix  $\mathbf{B}_Z$  is also positive semi-definite with  $\text{rank}(\mathbf{B}_Z) = 3$ . Then we have  $\mathbf{I}_{d_+d_-} = \mathbf{I}_3$  and we can omit it in the derivations as for two-block rank one model. To evaluate the Chernoff ratio, we also investigate the  $C_{k\ell}$  as defined in Eq. (3.13). Similar observations suggest that  $C_{12} = C_{34}, C_{13} = C_{24}, C_{14} = C_{23}$ . Thus we only need to evaluate  $C_{12}, C_{13}, C_{14}$ . Subsequent calculations and simplification yield

$$\begin{aligned} C_{12} &= \frac{\beta^2}{2(\phi_a + \phi_b + \phi_\beta)}, \\ C_{13} &= \frac{(a-b)^2}{2(\phi_a + \phi_b + \phi_\beta)}, \\ C_{14} &= \frac{\beta^2 N_1 + (a-b)N_2}{2[D_1 + (\phi_a + \phi_b)(\phi_a + \phi_b + 2\phi_\beta)]}, \end{aligned} \quad (3.23)$$

where for  $0 < b < a < 1$  and  $0 < \beta < 1$

$$\begin{aligned} \phi_a &= a(1-a), \\ \phi_b &= b(1-b), \\ \phi_\beta &= \beta(1-a-b-\beta), \\ N_1 &= a(1-b) + b(1-a) + \phi_\beta, \\ N_2 &= ab(a-b) + \phi_a(a+\beta) - \phi_b(b+\beta), \\ D_1 &= \beta^2(1-2a-\beta)(1-2b-\beta). \end{aligned} \quad (3.24)$$

Then we have the approximate Chernoff information for Algorithm 1 as given by

$$\rho_1^* \approx \min_{\ell \in \{2,3,4\}} C_{1\ell}, \quad (3.25)$$

where  $C_{1\ell}$  for  $\ell \in \{2, 3, 4\}$  are defined as in Eq. (3.23). Also observe that

$$\begin{aligned} C_{12} - C_{14} &= \frac{-(a-b)^2[\phi_a + \phi_b + \beta(1-a-b)]^2}{D_2}, \\ C_{13} - C_{14} &= \frac{-\beta^2 N_1^2}{D_2}, \end{aligned} \quad (3.26)$$

where

$$D_2 = 2(\phi_a + \phi_b + \phi_\beta)[D_1 + (\phi_a + \phi_b)(\phi_a + \phi_b + 2\phi_\beta)]. \quad (3.27)$$

Then we can further simplify  $\rho_1^*$  as

$$\rho_1^* \approx \begin{cases} \frac{\beta^2}{2(\phi_a + \phi_b + \phi_\beta)} & \text{if } \beta \leq a - b, \\ \frac{(a-b)^2}{2(\phi_a + \phi_b + \phi_\beta)} & \text{if } \beta > a - b. \end{cases} \quad (3.28)$$

By the same derivations [17], we have the approximate Chernoff information for Algorithm 2 as

$$\rho_2^* \approx \frac{(a-b)^2}{2[a(1-a) + b(1-b)]} = \frac{(a-b)^2}{2(\phi_a + \phi_b)}, \quad (3.29)$$

where  $\phi_a$  and  $\phi_b$  are defined as in Eq. (3.24). □

## Proof of Theorem 2

*Proof.* Observe that a  $K$ -block SBM can become a  $2K$ -block SBM when adding a binary covariate. To analytically derive the Chernoff ratio for the  $K$ -block homogeneous model with one binary covariate, we first investigate the canonical latent positions for this model via the idea of Cholesky decomposition as in Corollary 2.

Specifically, let  $\mathbf{B} \in [0, 1]^{K \times K}$  denote the block connectivity probability matrix after accounting for the vertex covariate effect and  $\mathbf{B}_Z \in [0, 1]^{2K \times 2K}$  denote the block connectivity probability matrix including the vertex covariate effect. Here we focus on canonical latent positions for  $\mathbf{B}_Z$ , details about the canonical latent positions for  $\mathbf{B}$  have been discussed [17]. Let  $\boldsymbol{\nu}_Z(K, 2K)$  denote the canonical latent position matrix, then we can re-write  $\mathbf{B}_Z$  as

$$\mathbf{B}_Z = \boldsymbol{\nu}_Z(K, 2K) \boldsymbol{\nu}_Z(K, 2K)^\top, \quad (3.30)$$

where  $\boldsymbol{\nu}_Z(K, 2K) = [\boldsymbol{\nu}_1 \ \cdots \ \boldsymbol{\nu}_{2K}]^\top$ . For  $K = 2$  we have via the idea of Cholesky decomposition as in Corollary 2

$$\boldsymbol{\nu}_Z(2, 4) = \begin{bmatrix} \sqrt{a+\beta} & 0 & 0 \\ \frac{a}{\sqrt{a+\beta}} & \sqrt{\frac{\beta(2a+\beta)}{a+\beta}} & 0 \\ \frac{b+\beta}{\sqrt{a+\beta}} & \sqrt{\frac{\beta(b-a)^2}{(a+\beta)(2a+\beta)}} & \sqrt{\frac{2(a-b)(a+b+\beta)}{(2a+\beta)}} \\ \frac{b}{\sqrt{a+\beta}} & \sqrt{\frac{\beta(a+b+\beta)^2}{(a+\beta)(2a+\beta)}} & \sqrt{\frac{2(a-b)(a+b+\beta)}{(2a+\beta)}} \end{bmatrix}. \quad (3.31)$$

And by induction, for  $K \geq 3$  we have

$$\begin{aligned}
\boldsymbol{\nu}_Z(K, 2K)_{\cdot,1} &= \begin{bmatrix} \boldsymbol{\nu}_Z(K-1, 2K-2)_{\cdot,1:(K-1)} \\ \boldsymbol{\nu}_Z(K-1, 2K-2)_{2K-3,1:(K-1)} \\ \boldsymbol{\nu}_Z(K-1, 2K-2)_{2K-2,1:(K-1)} \end{bmatrix}, \\
\boldsymbol{\nu}_Z(K, 2K)_{\cdot,2} &= \begin{bmatrix} \boldsymbol{\nu}_Z(K-1, 2K-2)_{\cdot,K} \\ \kappa \boldsymbol{\nu}_Z(K-1, 2K-2)_{2K-3,K} \\ \kappa \boldsymbol{\nu}_Z(K-1, 2K-2)_{2K-2,K} \end{bmatrix}, \\
\boldsymbol{\nu}_Z(K, 2K)_{\cdot,3} &= \begin{bmatrix} \mathbf{0} \\ \sqrt{\frac{(a-b)[2a+2(K-1)b+K\beta]}{2a+2(K-2)b+(K-1)\beta}} \\ \sqrt{\frac{(a-b)[2a+2(K-1)b+K\beta]}{2a+2(K-2)b+(K-1)\beta}} \end{bmatrix},
\end{aligned} \tag{3.32}$$

where

$$\kappa = \frac{2b + \beta}{2a + 2(K-2)b + (K-1)\beta}. \tag{3.33}$$

For this  $K$ -block homogeneous model with one binary covariate, the symmetric structure of  $\mathbf{B}_Z$  yields

$$\begin{aligned}
\boldsymbol{\nu}_1^\top \boldsymbol{\nu}_1 &= \boldsymbol{\nu}_2^\top \boldsymbol{\nu}_2 = \cdots = \boldsymbol{\nu}_{2K}^\top \boldsymbol{\nu}_{2K} = a + \beta, \\
\boldsymbol{\nu}_1^\top \boldsymbol{\nu}_2 &= \boldsymbol{\nu}_3^\top \boldsymbol{\nu}_4 = \cdots = \boldsymbol{\nu}_{2K-1}^\top \boldsymbol{\nu}_{2K} = a, \\
\boldsymbol{\nu}_1^\top \boldsymbol{\nu}_3 &= \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_5 = \cdots = \boldsymbol{\nu}_{2K-2}^\top \boldsymbol{\nu}_{2K} = b + \beta, \\
\boldsymbol{\nu}_1^\top \boldsymbol{\nu}_4 &= \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_6 = \cdots = \boldsymbol{\nu}_{2K-2}^\top \boldsymbol{\nu}_{2K-1} = b.
\end{aligned} \tag{3.34}$$

Along with the balanced assumption, i.e.,  $\boldsymbol{\pi}_Z = (\frac{1}{2K}, \dots, \frac{1}{2K})$ , the first four rows of  $\boldsymbol{\nu}_Z(K, 2K)$  are ideal for derivation as they have the fewest non-zero entries and can represent all the possible geometric structure. In other word, we can only evaluate  $C_{12}, C_{13}, C_{14}$  where  $C_{k\ell}$  is defined as in Eq. (3.13) to derive the Chernoff ratio.

For  $K$ -block homogeneous model with one binary covariate, we observe that  $\mathbf{B}_Z$  has eigenvalue 0 with algebraic multiplicity  $K-1$ , eigenvalue  $K\beta$  with algebraic

multiplicity 1, eigenvalue  $2(a - b)$  with algebraic multiplicity  $K - 1$  and eigenvalue  $2a + 2(K - 1)b + K\beta$  with algebraic multiplicity 1. Along with the assumption that  $0 < b < a < 1$  and  $0 < \beta < 1$ , we have among non-zero eigenvalues of  $\mathbf{B}_Z$

$$\begin{aligned}\lambda_{\max}(\mathbf{B}_Z) &= 2a + 2(K - 1)b + K\beta, \\ \lambda_{\min}(\mathbf{B}_Z) &= \begin{cases} K\beta & \text{if } \beta \leq \frac{2(a-b)}{K}, \\ 2(a - b) & \text{if } \beta > \frac{2(a-b)}{K}. \end{cases}\end{aligned}\quad (3.35)$$

Thus  $\mathbf{B}_Z$  is positive semi-definite with  $\text{rank}(\mathbf{B}_Z) = K + 1$ . Then we have  $\mathbf{I}_{d_+d_-} = \mathbf{I}_{K+1}$  which has no complicating effect on the subsequent derivations. As discussed in the previous section, we only consider the first four rows of the canonical latent position matrix  $\boldsymbol{\nu}_Z(K, 2K)$  and evaluate  $C_{12}, C_{13}, C_{14}$ . With the definition as in Eq. (3.11), we have

$$\begin{aligned}\mathbb{E} \left[ g_{\frac{1}{2}}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right] &= c_0 \boldsymbol{\Delta} + c_{12} \mathbf{N}_{12} \mathbf{N}_{12}^\top, \\ \mathbb{E} \left[ g_{\frac{1}{2}}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_3, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right] &= \boldsymbol{\Delta}_T + c_{13} \mathbf{N}_{13} \mathbf{N}_{13}^\top + c_{24} \mathbf{N}_{24} \mathbf{N}_{24}^\top, \\ \mathbb{E} \left[ g_{\frac{1}{2}}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_4, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right] &= c_0 \boldsymbol{\Delta} + c_{14} \mathbf{N}_{14} \mathbf{N}_{14}^\top + c_{23} \mathbf{N}_{23} \mathbf{N}_{23}^\top,\end{aligned}\quad (3.36)$$

where  $\boldsymbol{\Delta} \in \mathbb{R}^{(K+1) \times (K+1)}$  is defined as in Theorem 1,  $\boldsymbol{\nu}_Z \in \mathbb{R}^{2K \times (K+1)}$  is defined as in

Eq. (3.32) and

$$\begin{aligned}
\phi_a &= a(1 - a), \\
\phi_b &= b(1 - b), \\
\phi_{b\beta} &= (b + \beta)(1 - b - \beta), \\
c_0 &= \frac{\phi_b + \phi_{b\beta}}{2}, \\
c_{12} &= \frac{(a - b)(1 - a - b - \beta)}{2K}, \\
c_{13} = c_{14} &= \frac{(a - b)(1 - a - b - 2\beta)}{4K}, \\
c_{23} = c_{24} &= \frac{\phi_a - \phi_b}{4K}, \\
c_T &= \frac{\beta(1 - 2b - \beta)}{4K}, \\
\mathbf{N}_{k\ell} &= \begin{bmatrix} \boldsymbol{\nu}_k & \boldsymbol{\nu}_\ell \end{bmatrix} \in \mathbb{R}^{(K+1) \times 2}, \\
\mathbf{I}_T &= \text{diag}(1, -1, \dots, 1, -1) \in \mathbb{R}^{2K \times 2K}, \\
\boldsymbol{\Delta}_T &= \boldsymbol{\nu}_Z^\top \left( c_T \mathbf{I}_T + \frac{c_0}{2K} \mathbf{I}_{2K} \right) \boldsymbol{\nu}_Z \in \mathbb{R}^{(K+1) \times (K+1)}.
\end{aligned} \tag{3.37}$$

With the canonical latent position matrix  $\boldsymbol{\nu}_Z(K, 2K)$  as in Eq. (3.32), observe that

$$\begin{aligned}
\mathbf{N}_{12}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{12} &= \begin{bmatrix} K+1 & K-1 \\ K-1 & K+1 \end{bmatrix}, \\
\mathbf{N}_{13}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{13} &= \begin{bmatrix} K+1 & 1 \\ 1 & K+1 \end{bmatrix}, \\
\mathbf{N}_{14}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{14} &= \begin{bmatrix} K+1 & -1 \\ -1 & K+1 \end{bmatrix}, \\
\mathbf{N}_{13}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{24} &= \begin{bmatrix} K-1 & 1 \\ 1 & K-1 \end{bmatrix}, \\
\mathbf{N}_{13}^\top \boldsymbol{\Delta}_T^{-1} \mathbf{N}_{13} &= \frac{2}{n_{13}} \begin{bmatrix} \phi_b + K\phi_{b\beta} & \phi_b \\ \phi_b & \phi_b + K\phi_{b\beta} \end{bmatrix}, \\
\mathbf{N}_{24}^\top \boldsymbol{\Delta}_T^{-1} \mathbf{N}_{24} &= \frac{2}{n_{24}} \begin{bmatrix} K\phi_b + \phi_{b\beta} & \phi_{b\beta} \\ \phi_{b\beta} & K\phi_b + \phi_{b\beta} \end{bmatrix}, \\
\mathbf{N}_{13}^\top \boldsymbol{\Delta}_T^{-1} \mathbf{N}_{24} &= \frac{1}{c_0} \begin{bmatrix} K-1 & -1 \\ -1 & K-1 \end{bmatrix},
\end{aligned} \tag{3.38}$$

where  $c_0, \phi_b, \phi_{b\beta}$  are defined as in Eq. (3.37) and

$$\begin{aligned}
n_{13} &= 2\phi_b^2 + \beta^2(1-\beta)^2 + 3\beta\phi_b(1-2b-\beta) - 4b\beta^2(1-b-\beta), \\
n_{24} &= \phi_b(\phi_b + \phi_{b\beta}).
\end{aligned} \tag{3.39}$$

By the Sherman-Morrison-Woodbury formula as in Corollary 3, we have

$$\begin{aligned}
\mathbb{E} \left[ g_{\frac{1}{2}}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right]^{-1} &= \frac{1}{c_0} \boldsymbol{\Delta}^{-1} - \frac{1}{c_0^2} \boldsymbol{\Delta}^{-1} \mathbf{M}_{12} \boldsymbol{\Delta}^{-1}, \\
\mathbb{E} \left[ g_{\frac{1}{2}}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_3, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right]^{-1} &= \boldsymbol{\Delta}_T^{-1} - \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{13} \boldsymbol{\Delta}_T^{-1} - \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{24} \boldsymbol{\Delta}_T^{-1} \\
&\quad + \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{24} \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{13} \boldsymbol{\Delta}_T^{-1} + \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{13} \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{24} \boldsymbol{\Delta}_T^{-1} \\
&\quad - \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{13} \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{24} \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{13} \boldsymbol{\Delta}_T^{-1}, \\
\mathbb{E} \left[ g_{\frac{1}{2}}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_4, \boldsymbol{\nu}) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right]^{-1} &= \frac{1}{c_0} \boldsymbol{\Delta}^{-1} - \frac{1}{c_0^2} \boldsymbol{\Delta}^{-1} \mathbf{M}_{14} \boldsymbol{\Delta}^{-1} - \frac{1}{c_0^2} \boldsymbol{\Delta}^{-1} \mathbf{M}_{23} \boldsymbol{\Delta}^{-1} \\
&\quad + \frac{1}{c_0^3} \boldsymbol{\Delta}^{-1} \mathbf{M}_{23} \boldsymbol{\Delta}^{-1} \mathbf{M}_{14} \boldsymbol{\Delta}^{-1} + \frac{1}{c_0^3} \boldsymbol{\Delta}^{-1} \mathbf{M}_{14} \boldsymbol{\Delta}^{-1} \mathbf{M}_{23} \boldsymbol{\Delta}^{-1} \\
&\quad - \frac{1}{c_0^4} \boldsymbol{\Delta}^{-1} \mathbf{M}_{14} \boldsymbol{\Delta}^{-1} \mathbf{M}_{23} \boldsymbol{\Delta}^{-1} \mathbf{M}_{14} \boldsymbol{\Delta}^{-1},
\end{aligned} \tag{3.40}$$

where  $c_0, c_{12}, c_{13}, c_{14}, c_{23}, c_{24}$  are defined as in Eq. (3.37) and

$$\begin{aligned}
\mathbf{D}_{12} &= \frac{1}{c_{12}} \mathbf{I}_2 + \frac{1}{c_0} \mathbf{N}_{12}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{12}, \\
\mathbf{D}_{13} &= \frac{1}{c_{13}} \mathbf{I}_2 + \mathbf{N}_{13}^\top \boldsymbol{\Delta}_T^{-1} \mathbf{N}_{13}, \\
\mathbf{D}_{14} &= \frac{1}{c_{14}} \mathbf{I}_2 + \frac{1}{c_0} \mathbf{N}_{14}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{14}, \\
\mathbf{M}_{12} &= \mathbf{N}_{12} \mathbf{D}_{12}^{-1} \mathbf{N}_{12}^\top, \\
\mathbf{M}_{13} &= \mathbf{N}_{13} \mathbf{D}_{13}^{-1} \mathbf{N}_{13}^\top, \\
\mathbf{M}_{14} &= \mathbf{N}_{14} \mathbf{D}_{14}^{-1} \mathbf{N}_{14}^\top, \\
\mathbf{D}_{23} &= \frac{1}{c_{23}} \mathbf{I}_2 + \frac{1}{c_0} \mathbf{N}_{23}^\top \boldsymbol{\Delta}^{-1} \mathbf{N}_{23} - \frac{1}{c_0^2} \mathbf{N}_{23}^\top \boldsymbol{\Delta}^{-1} \mathbf{M}_{14} \boldsymbol{\Delta}^{-1} \mathbf{N}_{23}, \\
\mathbf{D}_{24} &= \frac{1}{c_{24}} \mathbf{I}_2 + \mathbf{N}_{24}^\top \boldsymbol{\Delta}_T^{-1} \mathbf{N}_{24} - \mathbf{N}_{24}^\top \boldsymbol{\Delta}_T^{-1} \mathbf{M}_{13} \boldsymbol{\Delta}_T^{-1} \mathbf{N}_{24}, \\
\mathbf{M}_{23} &= \mathbf{N}_{23} \mathbf{D}_{23}^{-1} \mathbf{N}_{23}^\top, \\
\mathbf{M}_{24} &= \mathbf{N}_{24} \mathbf{D}_{24}^{-1} \mathbf{N}_{24}^\top.
\end{aligned} \tag{3.41}$$

Again by canonical latent position matrix  $\boldsymbol{\nu}_Z(K, 2K)$  as in Eq. (3.32), we have

$$\begin{aligned}
(\boldsymbol{\nu}_1 - \boldsymbol{\nu}_2)^\top \boldsymbol{\Delta} (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_2) &= \beta^2, \\
(\boldsymbol{\nu}_1 - \boldsymbol{\nu}_3)^\top \boldsymbol{\Delta} (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_3) &= \frac{2}{K}(a - b)^2, \\
(\boldsymbol{\nu}_1 - \boldsymbol{\nu}_4)^\top \boldsymbol{\Delta} (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_4) &= \frac{2}{K}(a - b)^2 + \beta^2, \\
(\boldsymbol{\nu}_1 - \boldsymbol{\nu}_3)^\top \boldsymbol{\Delta} \boldsymbol{\Delta}_T^{-1} \boldsymbol{\Delta} (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_3) &= \frac{1}{c_0} \frac{2}{K}(a - b)^2.
\end{aligned} \tag{3.42}$$

Similarly, we have

$$\begin{aligned}
\mathbf{N}_{12}^\top (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_2) &= \beta \begin{bmatrix} 1 & -1 \end{bmatrix}^\top, \\
\mathbf{N}_{13}^\top (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_3) &= (a - b) \begin{bmatrix} 1 & -1 \end{bmatrix}^\top, \\
\mathbf{N}_{14}^\top (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_4) &= (a - b + \beta) \begin{bmatrix} 1 & -1 \end{bmatrix}^\top, \\
\mathbf{N}_{23}^\top (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_4) &= (a - b - \beta) \begin{bmatrix} 1 & -1 \end{bmatrix}^\top, \\
\mathbf{N}_{13}^\top \boldsymbol{\Delta}_T^{-1} \boldsymbol{\Delta} (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_3) &= \frac{(a - b)}{c_0} \begin{bmatrix} 1 & -1 \end{bmatrix}^\top.
\end{aligned} \tag{3.43}$$

Then with all the results above, we have

$$\begin{aligned}
C_{12} &= \frac{K\beta^2}{2D_4}, \\
C_{13} &= \frac{(a - b)^2}{K(\phi_a + \phi_b + \phi_\beta)}, \\
C_{14} &= \frac{K^2\beta^2(\phi_a + \phi_b + \phi_\beta) + 2KN_3 + 4N_4}{2K[2(\phi_a^2 - \phi_b^2) + D_5]},
\end{aligned} \tag{3.44}$$

where  $\phi_a, \phi_b$  are defined as in Eq. (3.37) and

$$\begin{aligned}
\phi_\beta &= \beta(1 - a - b - \beta), \\
D_3 &= K - 2a - 2(K - 1)b - K\beta, \\
D_4 &= 2\phi_a + 2(K - 1)\phi_b + \beta D_3, \\
N_3 &= (a - b)^2[2\phi_b + \beta(1 + \beta - 2b)], \\
N_4 &= (a - b)^3(1 - a - b - \beta), \\
D_5 &= 2\beta(a - b)[(1 - a - b - \beta) - 2(\phi_a + \phi_b) - \phi_\beta + 2b(a + \beta)] \\
&\quad + K\{2\phi_b(\phi_a + \phi_b) - 2b\beta(\phi_b + a - b^2) - 2ab\phi_\beta \\
&\quad + \beta(1 - \beta)[\phi_a + (3b + \beta)(1 - \beta) - a\beta - 5b^2]\}.
\end{aligned} \tag{3.45}$$

Then we have the approximate Chernoff information for Algorithm 1 as

$$\rho_1^* \approx \min_{\ell \in \{2, 3, 4\}} C_{1\ell}, \tag{3.46}$$

where  $C_{1\ell}$  for  $\ell \in \{2, 3, 4\}$  are defined as in Eq. (3.44). Also observe that

$$\begin{aligned}
C_{12} - C_{14} &= \frac{-(a - b)^2 N_6^2}{K D_4 [2(\phi_a^2 - \phi_b^2) + D_5]}, \\
C_{13} - C_{14} &= \frac{-\beta^2 [2(a - b)^2 + K(\phi_a + \phi_b + \phi_\beta)]^2}{2K(\phi_a + \phi_b + \phi_\beta) [2(\phi_a^2 - \phi_b^2) + D_5]},
\end{aligned} \tag{3.47}$$

where  $\phi_a, \phi_b$  are defined as in Eq. (3.37),  $\phi_\beta, D_4, D_5$  are defined as in Eq. (3.45) and

$$\begin{aligned}
N_5 &= \beta[K - 2a - 2(K - 1)b], \\
N_6 &= 2\phi_a + 2(K - 1)\phi_b + N_5.
\end{aligned} \tag{3.48}$$

Subsequent calculations and simplification yield  $\rho_1^*$  as

$$\rho_1^* \approx \begin{cases} \frac{K\beta^2}{2D_4} & \text{if } \delta \leq 0 \\ \frac{(a-b)^2}{K(\phi_a + \phi_b + \phi_\beta)} & \text{if } \delta > 0 \end{cases}, \tag{3.49}$$

where  $\phi_a, \phi_b, \phi_\beta$  are defined as in Eq. (3.24) and

$$\begin{aligned} D_3 &= K - 2a - 2(K - 1)b - K\beta, \\ D_4 &= 2\phi_a + 2(K - 1)\phi_b + \beta D_3, \\ \delta &= K^2\beta^2(\phi_a + \phi_b + \phi_\beta) - 2(a - b)^2 D_4. \end{aligned} \tag{3.50}$$

Again by the same derivations [17], we have the approximate Chernoff information for Algorithm 2 as

$$\rho_2^* \approx \frac{(a - b)^2}{K[a(1 - a) + b(1 - b)]} = \frac{(a - b)^2}{K(\phi_a + \phi_b)}, \tag{3.51}$$

where  $\phi_a$  and  $\phi_b$  are defined as in Eq. (3.24). □

## Chapter 4

# Dynamic Network Sampling for Community Detection

In this chapter, we focus on the problem of dynamic network sampling for community detection. Motivated by the Chernoff analysis, we propose a dynamic network sampling scheme to optimize block recovery for stochastic blockmodel in the case where it is prohibitively expensive to observe the entire graph. Theoretically, we provide justification of our proposed Chernoff-optimal dynamic sampling scheme via the notion of Chernoff information. Practically, we evaluate the performance, in terms of block recovery, of our method on several real datasets from different domains. Both theoretically and practically results suggest that our method can identify vertices that have the most impact on block structure so that one can only check whether there are edges between them to save significant resources but still recover the block structure.

# Dynamic Network Sampling Scheme

We start our analysis with the unobserved block connectivity probability matrix  $\mathbf{B}$  for SBM and then illustrate how to migrate the proposed methods for real applications when we have the observed adjacency matrix  $\mathbf{A}$ .

## Initial Sampling

Consider the  $K$ -block SBM parametrized by the block connectivity probability matrix  $\mathbf{B} \in (0, 1)^{K \times K}$  and the vector of block assignment probabilities  $\boldsymbol{\pi} \in (0, 1)^K$  with  $K > 2$ . Given initial sampling parameter  $p_0 \in (0, 1)$ , initial sampling is uniformly at random, i.e.,

$$\mathbf{B}_0 = p_0 \mathbf{B}. \quad (4.1)$$

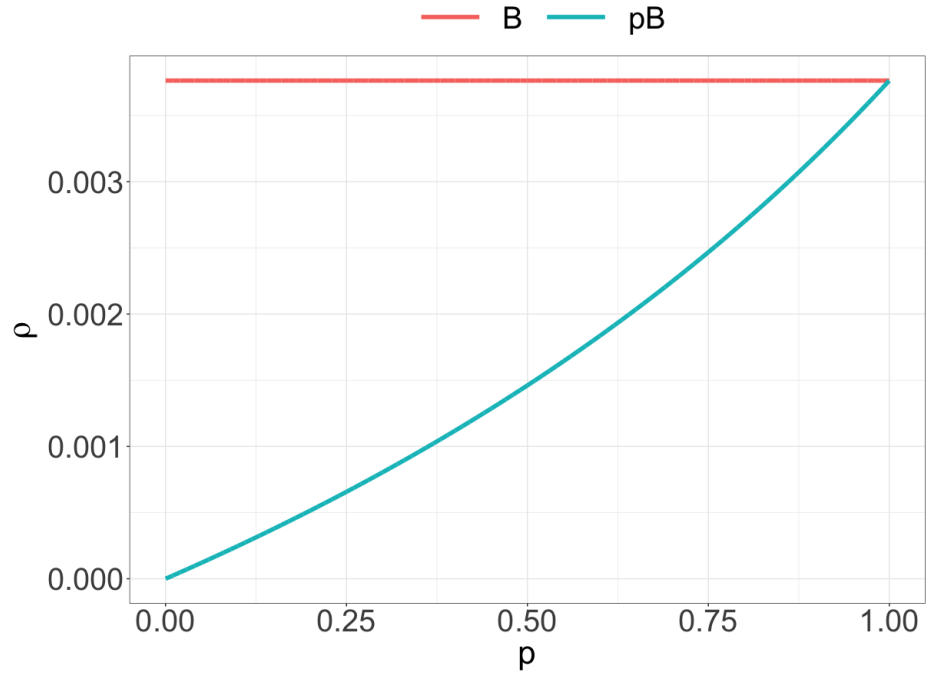
Note that this initial sampling simulates the case when one only observes a partial graph with a small portion of the edges instead of the entire graph with all existing edges.

**Theorem 3.** *For  $K$ -block SBMs, given two block connectivity probability matrices  $\mathbf{B}, p\mathbf{B} \in (0, 1)^{K \times K}$  with  $p \in (0, 1)$  and a vector of block assignment probabilities  $\boldsymbol{\pi} \in (0, 1)^K$ . We have  $\mathbf{B} \succ p\mathbf{B}$ .*

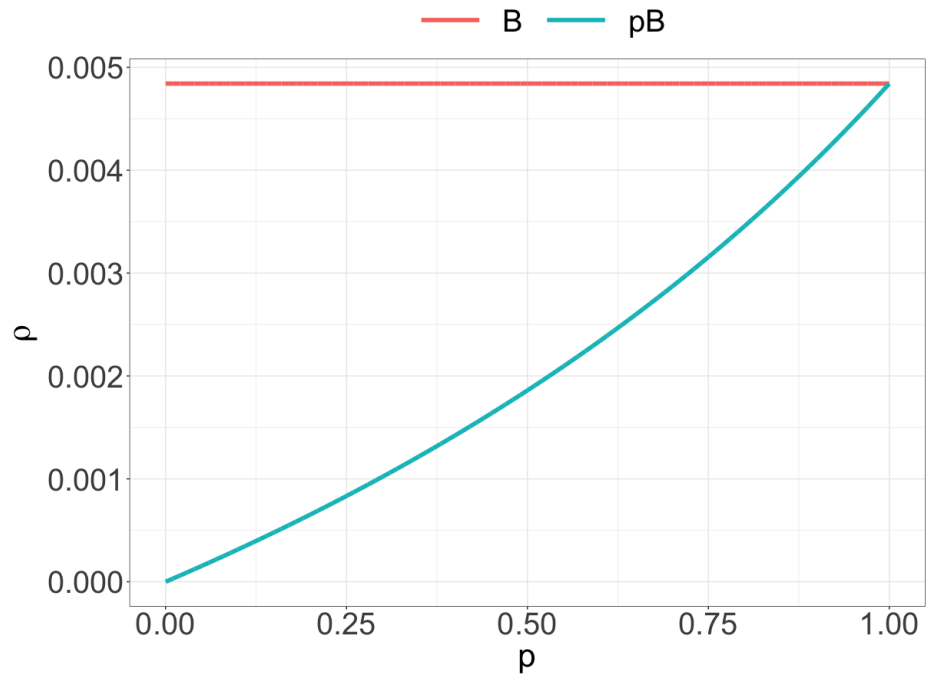
Technical details of Theorem 3 can be found at the end of this chapter. As an illustration, consider a 4-block SBM parametrized by block connectivity probability matrix  $\mathbf{B}$  as

$$\mathbf{B} = \begin{bmatrix} 0.04 & 0.08 & 0.10 & 0.18 \\ 0.08 & 0.16 & 0.20 & 0.36 \\ 0.10 & 0.20 & 0.25 & 0.45 \\ 0.18 & 0.36 & 0.45 & 0.81 \end{bmatrix}. \quad (4.2)$$

Figure 4-1 shows Chernoff information  $\rho$  as in Eq. (2.29) corresponding to  $\mathbf{B}$  as in Eq. (4.2) and  $p\mathbf{B}$  for  $p \in (0, 1)$ . In addition, Figure 4-1a assumes  $\boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$  and Figure 4-1b assumes  $\boldsymbol{\pi} = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ . As suggested by Theorem 3, for any  $p \in (0, 1)$  we have  $\rho_B > \rho_{pB}$  and thus  $\mathbf{B} \succ p\mathbf{B}$ . That is, we can have a better block recovery from  $\mathbf{B}$  than  $p\mathbf{B}$ . Intuitively, as we sample more edges, we will have more information that can be used to identify potential blocks and thus leads to the better performance.



(a) Balanced:  $\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .



(b) Unbalanced:  $\pi = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ .

**Figure 4-1.** Chernoff information  $\rho$  as in Eq. (2.29) corresponding to  $\mathbf{B}$  as in Eq. (4.2) and  $p\mathbf{B}$  for  $p \in (0, 1)$ .

## Dynamic Sampling

Now given dynamic network sampling parameter  $p_1 \in (0, 1 - p_0)$ , the baseline sampling scheme can proceed uniformly at random again, i.e.,

$$\mathbf{B}_1 = \mathbf{B}_0 + p_1 \mathbf{B} = (p_0 + p_1) \mathbf{B}. \quad (4.3)$$

Note that this dynamic network sampling simulates the situation when one is given limited resources to sample some extra edges after observing the partial graph with only a small portion of the edges. Since we only have limited budget to sample another small portion of edges, one would benefit from identifying vertex pairs that have much influence on the community structure. In other words, the baseline sampling scheme just randomly choosing vertex pairs without using the information from the initial observed graphs and our goal is to design an alternative scheme to optimize this dynamic network sampling procedure so that one could have a better block recovery even with limited resources to only observe a partial graph with a small portion of the edges.

**Corollary 4.** *For  $K$ -block SBMs, given block connectivity probability matrix  $\mathbf{B} \in (0, 1)^{K \times K}$  and a vector of block assignment probabilities  $\boldsymbol{\pi} \in (0, 1)^K$ . We have  $\mathbf{B} \succ \mathbf{B}_1 \succ \mathbf{B}_0$  where  $\mathbf{B}_0$  is defined as in Eq. (4.1) with  $p_0 \in (0, 1)$  and  $\mathbf{B}_1$  is defined as in Eq. (4.3) with  $p_1 \in (0, 1 - p_0)$ .*

Technical details of Corollary 4 can be found at the end of this chapter. This corollary implies that we can have a better block recovery from  $\mathbf{B}_1$  than  $\mathbf{B}_0$ . Similar intuition as before, as we sample more edges compared with the initial observed graphs, we could have more information to use for detecting block structure and thus leads to the better performance.

Motivated by the Chernoff analysis, we now describe the proposed dynamic network sampling scheme.

**Assumption 1.** *The Chernoff-active blocks after initial sampling is unique, i.e., there exists an unique pair  $(k_0^*, \ell_0^*) \in \{(k, \ell) \mid 1 \leq k < \ell \leq K\}$  such that*

$$(k_0^*, \ell_0^*) = \arg \min_{k \neq \ell} C_{k, \ell}(\mathbf{B}_0, \boldsymbol{\pi}), \quad (4.4)$$

where  $\mathbf{B}_0$  is defined as in Eq. (4.1) and  $\boldsymbol{\pi}$  is the vector of block assignment probabilities.

To improve this baseline sampling scheme, we concentrate on the Chernoff-active blocks  $(k_0^*, \ell_0^*)$  after initial sampling given Assumption 1 holds. Instead of sampling from the entire block connectivity probability matrix  $\mathbf{B}$  like the baseline sampling scheme as in Eq. (4.3), we only sample the entries associated with the Chernoff-active blocks. As a competitor to  $\mathbf{B}_1$ , our Chernoff-optimal dynamic network sampling scheme is then given by

$$\tilde{\mathbf{B}}_1 = \mathbf{B}_0 + \frac{p_1}{(\pi_{k_0^*} + \pi_{\ell_0^*})^2} \mathbf{B} \circ \mathbf{1}_{k_0^*, \ell_0^*}, \quad (4.5)$$

where  $\circ$  denotes Hadamard product,  $\pi_{k_0^*}$  and  $\pi_{\ell_0^*}$  denote the block assignment probabilities for block  $k_0^*$  and  $\ell_0^*$  respectively, and  $\mathbf{1}_*$  is the  $K \times K$  binary matrix with 0's everywhere except for 1's associated with the Chernoff-active blocks  $(k_0^*, \ell_0^*)$ , i.e., for any  $i, j \in \{1, \dots, K\}$

$$\mathbf{1}_{k_0^*, \ell_0^*}[i, j] = \begin{cases} 1 & \text{if } (i, j) \in \{(k_0^*, k_0^*), (k_0^*, \ell_0^*), (\ell_0^*, k_0^*), (\ell_0^*, \ell_0^*)\} \\ 0 & \text{otherwise} \end{cases}. \quad (4.6)$$

Note that the multiplier  $\frac{1}{(\pi_{k_0^*} + \pi_{\ell_0^*})^2}$  on  $p_1 \mathbf{B} \circ \mathbf{1}_*$  assures that we sample the same number of potential edges with  $\tilde{\mathbf{B}}_1$  as we do with  $\mathbf{B}_1$  in the baseline sampling scheme. In addition, to avoid over-sampling with respect to  $\mathbf{B}$ , i.e., to ensure  $\tilde{B}_1[i, j] \leq B[i, j]$  for any  $i, j \in \{1, \dots, K\}$ , we require

$$p_1 \leq p_1^{\max} = (1 - p_0) (\pi_{k_0^*} + \pi_{\ell_0^*})^2. \quad (4.7)$$

**Assumption 2.** *For  $K$ -block SBMs, given a block connectivity probability matrix  $\mathbf{B} \in (0, 1)^{K \times K}$  and a vector of block assignment probabilities  $\boldsymbol{\pi} \in (0, 1)^K$ . Let*

$p_1^* \in (0, p_1^{\max}]$  be the smallest positive  $p_1 \leq p_1^{\max}$  such that

$$\arg \min_{k \neq l} C_{k,\ell}(\tilde{\mathbf{B}}_1, \boldsymbol{\pi}) \quad (4.8)$$

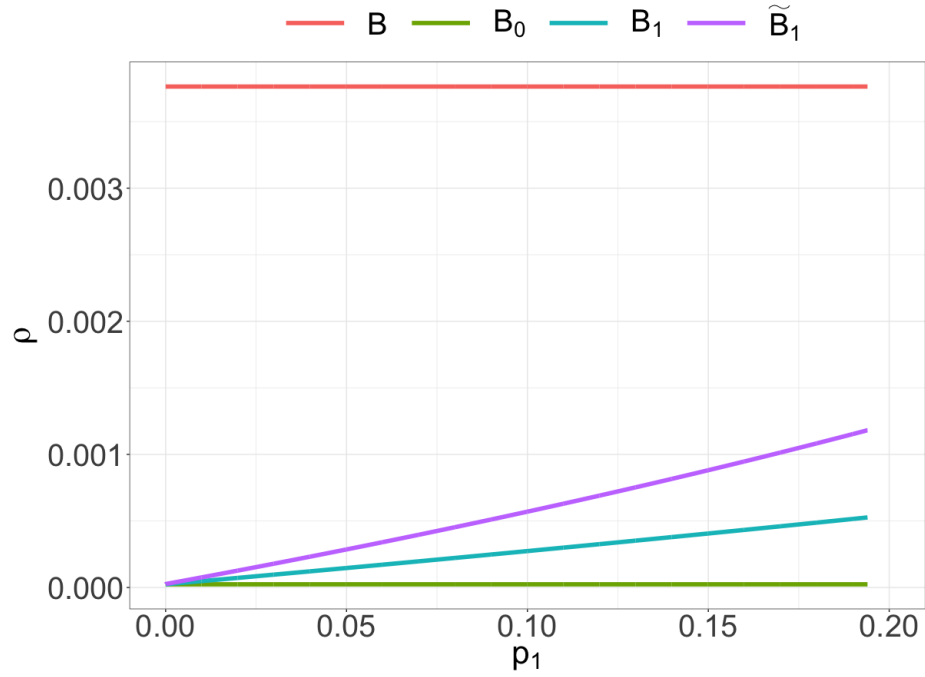
is not unique where  $p_1^{\max}$  is defined as in Eq. (4.7) and  $\tilde{\mathbf{B}}_1$  is defined as in Eq. (4.5). If the arg min is always unique, let  $p_1^* = p_1^{\max}$ .

For any  $p_1 \in (0, p_1^*)$ , we can have a better block recovery from  $\tilde{\mathbf{B}}_1$  than  $\mathbf{B}_1$ , i.e., our Chernoff-optimal dynamic network sampling scheme is better than the baseline sampling scheme in terms of block recovery.

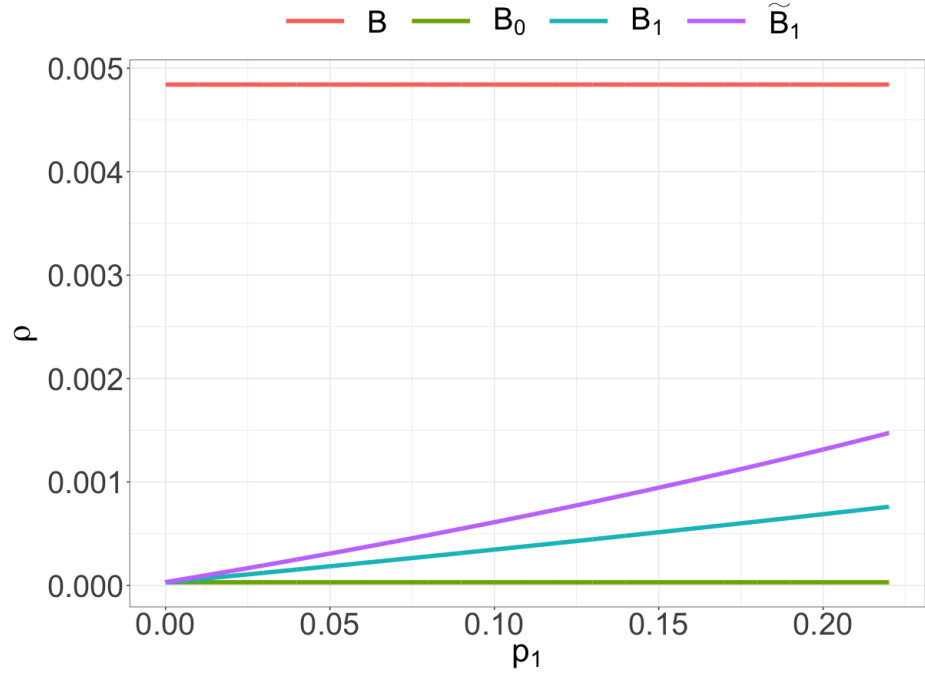
As an illustration, consider the 4-block SBM with initial sampling parameter  $p_0 = 0.01$  and block connectivity probability matrix  $\mathbf{B}$  as in Eq. (4.2). Figure 4-2 shows the Chernoff information  $\rho$  as in Eq. (2.29) corresponding to  $\mathbf{B}$  as in Eq. (4.2),  $\mathbf{B}_0$  as in Eq. (4.1),  $\mathbf{B}_1$  as in Eq. (4.3), and  $\tilde{\mathbf{B}}_1$  as in Eq. (4.5) with dynamic network sampling parameter  $p_1 \in (0, p_1^*)$  where  $p_1^*$  is defined as in Theorem 2. In addition, Figure 4-2a assumes  $\boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$  and Figure 4-2b assumes  $\boldsymbol{\pi} = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ . Note that for any  $p_1 \in (0, p_1^*)$  we have  $\rho_B > \rho_{\tilde{B}_1} > \rho_{B_1} > \rho_{B_0}$  and thus  $\mathbf{B} \succ \tilde{\mathbf{B}}_1 \succ \mathbf{B}_1 \succ \mathbf{B}_0$ . That is, in terms of Chernoff information, when given same amount of resources, the proposed Chernoff-optimal dynamic network sampling scheme can yield better block recovery results. In other words, to reach the same level of performance, in terms of Chernoff information, the proposed Chernoff-optimal dynamic network sampling scheme needs less resources.

Again at this stage, the comparison is based on theoretical analysis. Later we will show how the proposed method works for real applications.

As described earlier, it may be the case that  $p_1^* < p_1^{\max}$  at which point Chernoff-active blocks change to  $(k_1^*, \ell_1^*)$ . This potential non-uniqueness of the Chernoff argmin is a consequence of our dynamic network sampling scheme. In the case of  $p_1 > p_1^*$ , our



(a) Balanced:  $\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .



(b) Unbalanced:  $\pi = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ .

**Figure 4-2.** Chernoff information  $\rho$  as in Eq. (2.29) corresponding to  $B$  as in Eq. (4.2),  $B_0$  as in Eq. (4.1),  $B_1$  as in Eq. (4.3), and  $\tilde{B}_1$  as in Eq. (4.5) with initial sampling parameter  $p_0 = 0.01$  and dynamic network sampling parameter  $p_1 \in (0, p_1^*)$  where  $p_1^*$  is defined as in Assumption 2.

Chernoff-optimal dynamic network sampling scheme is adopted as

$$\tilde{\mathbf{B}}_1^* = \mathbf{B}_0 + (p_1 - p_1^*) \mathbf{B} + \frac{p_1^*}{(\pi_{k_0^*} + \pi_{\ell_0^*})^2} \mathbf{B} \circ \mathbf{1}_{k_0^*, \ell_0^*}, \quad (4.9)$$

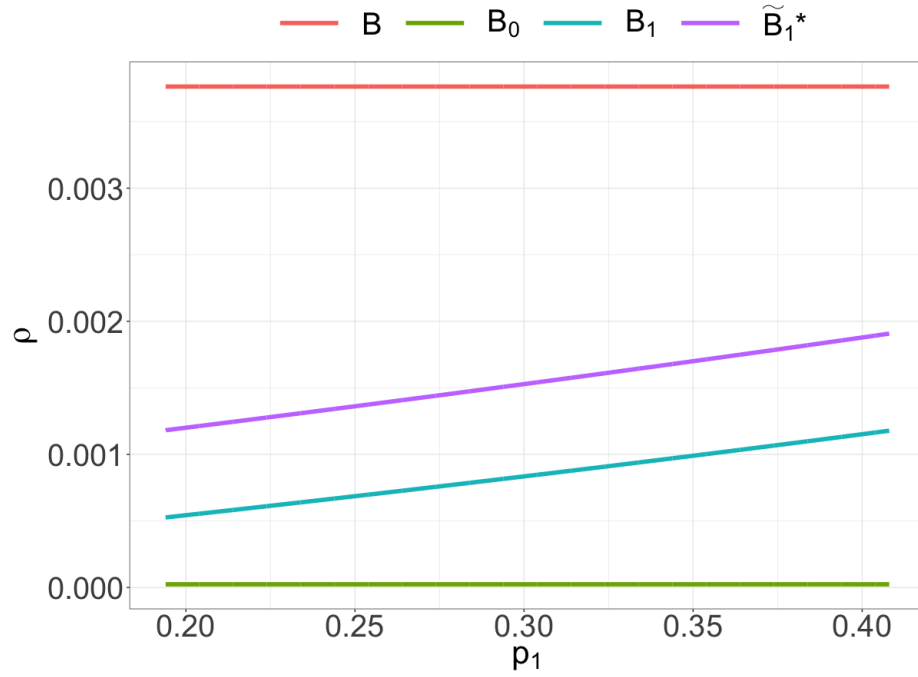
Similarly, the multiplier  $\frac{1}{(\pi_{k_0^*} + \pi_{\ell_0^*})^2}$  on  $p_1^* \mathbf{B} \circ \mathbf{1}_{k_0^*, \ell_0^*}$  assures that we sample the same number of potential edges with  $\tilde{\mathbf{B}}_1^*$  as we do with  $\mathbf{B}_1$  in the baseline sampling scheme. In addition, to avoid over-sampling with respect to  $\mathbf{B}$ , i.e.,  $\tilde{B}_1^*[i, j] \leq B[i, j]$  for any  $i, j \in \{1, \dots, K\}$ , we require

$$p_1 \leq p_{11}^{\max} = 1 - p_0 - \frac{p_1^*}{(\pi_{k_0^*} + \pi_{\ell_0^*})^2} + p_1^*. \quad (4.10)$$

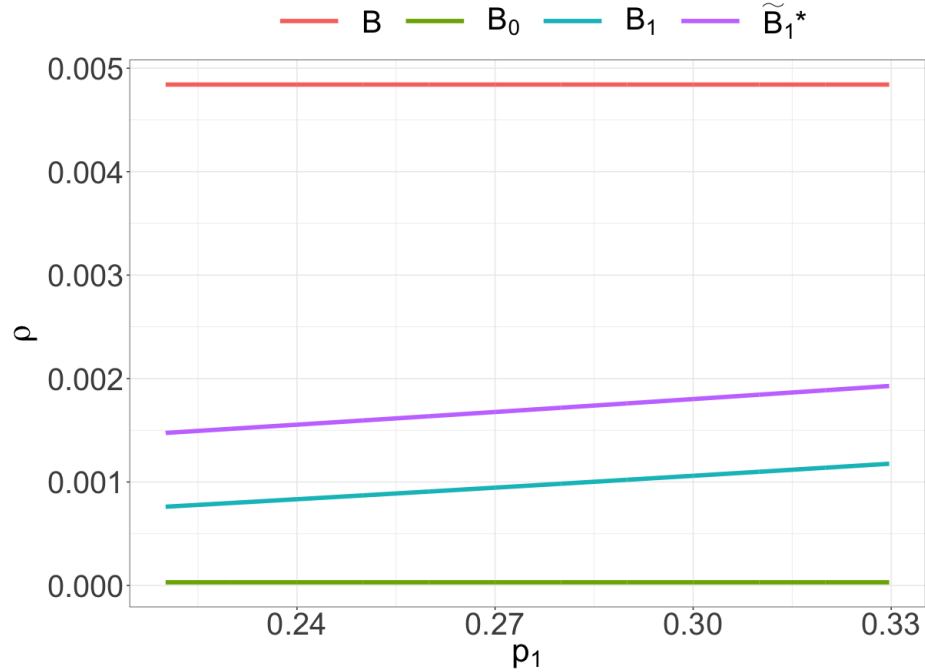
For any  $p_1 \in [p_1^*, p_{11}^{\max}]$ , we can have a better block recovery from  $\tilde{\mathbf{B}}_1^*$  than  $\mathbf{B}_1$ , i.e., our Chernoff-optimal dynamic network sampling scheme is again better than the baseline sampling scheme in terms of block recovery.

As an illustration, consider a 4-block SBM with initial sampling parameter  $p_0 = 0.01$  and block connectivity probability matrix  $\mathbf{B}$  as in Eq. (4.2). Figure 4-3 shows the Chernoff information  $\rho$  as in Eq. (2.29) corresponding to  $\mathbf{B}$  as in Eq. (4.2),  $\mathbf{B}_0$  as in Eq. (4.1),  $\mathbf{B}_1$  as in Eq. (4.3), and  $\tilde{\mathbf{B}}_1^*$  as in Eq. (4.9) with dynamic network sampling parameter  $p_1 \in [p_1^*, p_{11}^{\max}]$  where  $p_1^*$  is defined as in Assumption 2 and  $p_{11}^{\max}$  is defined as in Eq. (4.10). In addition, Figure 4-3a assumes  $\boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$  and Figure 4-3b assumes  $\boldsymbol{\pi} = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ . Note that for any  $p_1 \in [p_1^*, p_{11}^{\max}]$  we have  $\rho_B > \rho_{\tilde{B}_1^*} > \rho_{B_1} > \rho_{B_0}$  and thus  $\mathbf{B} \succ \tilde{\mathbf{B}}_1^* \succ \mathbf{B}_1 \succ \mathbf{B}_0$ . That is, the adopted Chernoff-optimal dynamic network sampling scheme can still yield better block recovery results, in terms of Chernoff information, given the same amount of resources.

Now we illustrate how the proposed Chernoff-optimal dynamic network sampling scheme can be migrated for real applications. We summarize the uniform dynamic sampling scheme (baseline) as Algorithm 3 and our Chernoff-optimal dynamic network sampling scheme as Algorithm 4. Recall given potential edge set  $E$  and initial sampling parameter  $p_0 \in (0, 1)$ , we have the initial edge set  $E_0 \subset E$  with  $|E_0| = p_0|E|$ . The



(a) Balanced:  $\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .



(b) Unbalanced:  $\pi = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ .

**Figure 4-3.** Chernoff information  $\rho$  as in Eq. (2.29) corresponding to  $\mathbf{B}$  as in Eq. (4.2),  $\mathbf{B}_0$  as in Eq. (4.1),  $\mathbf{B}_1$  as in Eq. (4.3), and  $\tilde{\mathbf{B}}_1^*$  as in Eq. (4.9) with initial sampling parameter  $p_0 = 0.01$  and dynamic network sampling parameter  $p_1 \in [p_1^*, p_{11}^{\max}]$  where  $p_1^*$  is defined as in Assumption 2 and  $p_{11}^{\max}$  is defined as in Eq. (4.10).

goal is to dynamically sample new edges from the potential edge set so that we can have a better block recovery.

In Algorithm 3, the dynamic sampling is done by randomly sampling edges from potential edge set without using any information from the initial observed edge sets.

In Algorithm 4, the Chernoff-optimal dynamic network sampling scheme first estimates the initial block assignments and block connectivity probability matrix from the initial observed edge sets. Then it identifies the Chernoff-active blocks and samples edges corresponding to these blocks from potential edge set.

---

**Algorithm 3:** Uniform dynamic network sampling scheme (baseline) [93]

---

**Input:** Number of vertices  $n$ ; potential edge set  $E = \{(i, j) \mid i, j \in \{1, \dots, n\}\}$ ; initial edge set  $E_0 \subset E$ ; dynamic network sampling parameter  $p_1 \in (0, 1 - \frac{|E_0|}{|E|})$ .

**Output:** Block assignments  $\hat{\tau}$ .

1 Construct dynamic edge set as

$$E_1 = \{(i, j) \mid (i, j) \in E \setminus E_0\} \quad \text{with} \quad |E_1| = p_1|E|.$$

2 Construct dynamic adjacency matrix as  $\mathbf{A} \in \{0, 1\}^{n \times n}$  where for any  $i, j \in \{1, \dots, n\}$

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E_0 \cup E_1 \text{ or } (j, i) \in E_0 \cup E_1 \\ 0 & \text{otherwise} \end{cases}.$$

3 Estimate dynamic latent positions as  $\hat{\mathbf{X}} \in \mathbb{R}^{n \times \hat{d}}$  using ASE of  $\mathbf{A}$  where  $\hat{d}$  is chosen as in Remark 7.

4 Cluster  $\hat{\mathbf{X}}$  using GMM to estimate the block assignments as  $\hat{\tau} \in \{1, \dots, \widehat{K}\}^n$  where  $\widehat{K}$  is chosen via BIC.

---

---

**Algorithm 4:** Chernoff-optimal dynamic network sampling scheme [93]

---

**Input:** Number of vertices  $n$ ; potential edge set  $E = \{(i, j) \mid i, j \in \{1, \dots, n\}\}$ ;  
initial edge set  $E_0 \subset E$ ; dynamic network sampling parameter  
 $p_1 \in (0, 1 - \frac{|E_0|}{|E|})$ .

**Output:** Block assignments  $\hat{\tau}$ .

- 1 Construct dynamic adjacency matrix as  $\mathbf{A} \in \{0, 1\}^{n \times n}$  where for any  $i, j \in \{1, \dots, n\}$

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E_0 \text{ or } (j, i) \in E_0 \\ 0 & \text{otherwise} \end{cases}.$$

- 2 Estimate dynamic latent positions as  $\hat{\mathbf{X}} \in \mathbb{R}^{n \times \hat{d}}$  using ASE of  $\mathbf{A}$  where  $\hat{d}$  is chosen as in Remark 7.
- 3 Cluster  $\hat{\mathbf{X}}$  using GMM to estimate the initial block assignments as  $\hat{\xi} \in \{1, \dots, \hat{K}\}^n$  where  $\hat{K}$  is chosen via BIC.
- 4 Estimate the dynamic block assignment probability vector as  $\hat{\pi} \in (0, 1)^K$  where for  $k \in \{1, \dots, K\}$

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{\hat{\xi}_i = k\}.$$

- 5 Estimate the dynamic block connectivity probability matrix as

$$\hat{\mathbf{B}} = \hat{\boldsymbol{\mu}} \mathbf{I}_{d_+ d_-} \hat{\boldsymbol{\mu}}^\top \in [0, 1]^{\hat{K} \times \hat{K}},$$

where  $\hat{\boldsymbol{\mu}} \in \mathbb{R}^{\hat{K} \times \hat{d}}$  is the estimated means of all clusters.

- 6 Find the Chernoff-active blocks as

$$(k^*, \ell^*) = \arg \min_{k \neq \ell} C_{k, \ell}(\hat{\mathbf{B}}, \hat{\pi}).$$

- 7 Construct dynamic edge set as

$$\begin{aligned} E_1 \subseteq E_* & \quad \text{with} \quad |E_1| = \min \left\{ p_1 |E| (\hat{\pi}_{k^*} + \hat{\pi}_{\ell^*})^2, |E_*| \right\}, \\ E_{11} \subset E \setminus (E_0 \cup E_1) & \quad \text{with} \quad |E_{11}| = p_1 |E| - |E_1|, \end{aligned}$$

where

$$E_* = \left\{ (i, j) \mid (i, j) \in E \setminus E_0 \text{ and } \hat{\xi}_i, \hat{\xi}_j \in \{k^*, \ell^*\} \right\}.$$

- 8 Update dynamic adjacency matrix as  $\mathbf{A} \in \{0, 1\}^{n \times n}$  where for any  $i, j \in \{1, \dots, n\}$

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E_0 \cup E_1 \cup E_{11} \text{ or } (j, i) \in E_0 \cup E_1 \cup E_{11} \\ 0 & \text{otherwise} \end{cases}.$$

- 9 Update dynamic latent positions as  $\hat{\mathbf{X}} \in \mathbb{R}^{n \times \hat{d}}$  using ASE of updated  $\mathbf{A}$  where  $\hat{d}$  is chosen as in Remark 7.
  - 10 Cluster  $\hat{\mathbf{X}}$  using GMM to estimate the block assignments as  $\hat{\tau} \in \{1, \dots, \hat{K}\}^n$  where  $\hat{K}$  is chosen via BIC.
-

# Experiments

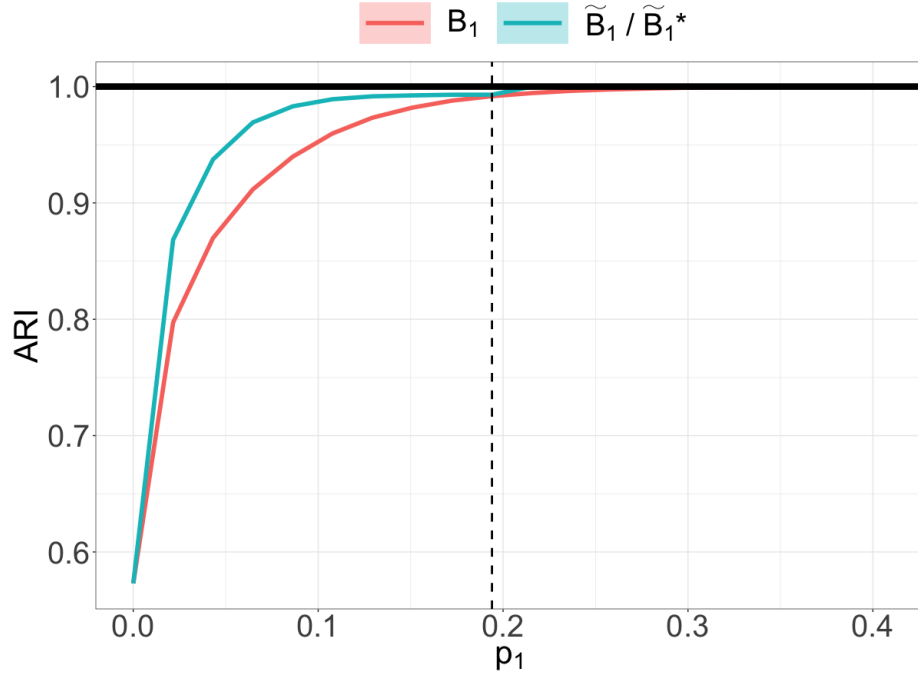
## Simulations

In addition to Chernoff analysis, we also evaluate our Chernoff-optimal dynamic network sampling scheme via simulations.

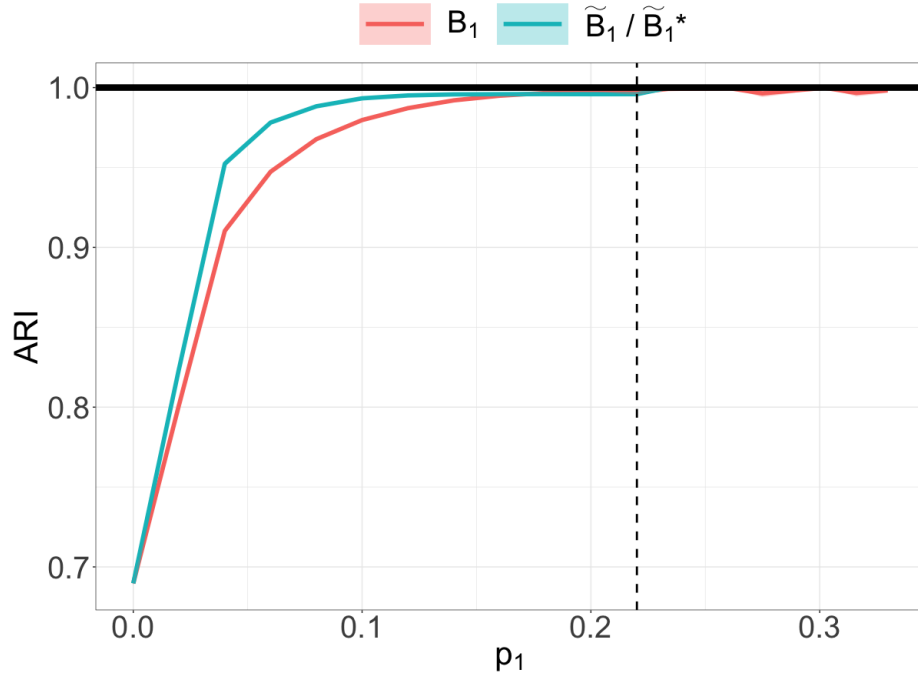
In particular, consider the 4-block SBM parameterized by block connectivity probability matrix  $\mathbf{B}$  as in Eq. (4.2) and dynamic network sampling parameter  $p_1 \in (0, p_{11}^{\max}]$  where  $p_{11}^{\max}$  is defined as in Eq. (4.10). We fix initial sampling parameter  $p_0 = 0.01$ . For each  $p_1 \in (0, p_1^*)$  where  $p_1^*$  is defined as in Assumption 2, we simulate 50 adjacency matrices with  $n = 12000$  vertices from  $\mathbf{B}_1$  as in Eq. (4.3) and  $\tilde{\mathbf{B}}_1$  as in Eq. (4.5) respectively. For each  $p_1 \in [p_1^*, p_{11}^{\max}]$ , we simulate 50 adjacency matrices with  $n = 12000$  vertices from  $\mathbf{B}_1$  as in Eq. (4.3) and  $\tilde{\mathbf{B}}_1^*$  as in Eq. (4.9) respectively. In addition, Figure 4-4a assumes  $\boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ , i.e., 3000 vertices in each block, and Figure 4-4b assumes  $\boldsymbol{\pi} = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ , i.e., 1500 vertices in two of the blocks and 4500 vertices in the other two blocks. We then apply ASE and GMM (Step 3 and 4 in Algorithm 3) to recover block assignments and adopt ARI to measure the performance.

Figure 4-4 shows ARI ( $\text{mean} \pm \text{stderr}$ ) associated with  $\mathbf{B}_1$  for  $p_1 \in (0, p_{11}^{\max}]$ ,  $\tilde{\mathbf{B}}_1$  for  $p_1 \in (0, p_1^*)$ , and  $\tilde{\mathbf{B}}_1^*$  for  $p_1 \in [p_1^*, p_{11}^{\max}]$  where the dashed lines denote  $p_1^*$ . Note that we can have a better block recovery from  $\tilde{\mathbf{B}}_1$  and  $\tilde{\mathbf{B}}_1^*$  than  $\mathbf{B}_1$ , which agree with our results from Chernoff analysis.

Now we compare the performance of Algorithms 3 and 4 by empirical block recovery results. In particular, we start with the 4-block SBM parameterized by block connectivity probability matrix  $\mathbf{B}$  as in Eq. (4.2). We consider dynamic network sampling parameter  $p_1 \in (0, 1 - p_0)$  where  $p_0$  is the initial sampling parameter. For each  $p_1$ , we simulate 50 adjacency matrices with  $n = 4000$  vertices and retrieve associated potential edge sets. We fix initial sampling parameter  $p_0 = 0.15$  and randomly sample initial edge sets. We then apply both algorithms to estimate the block assignments



(a) Balanced:  $\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .

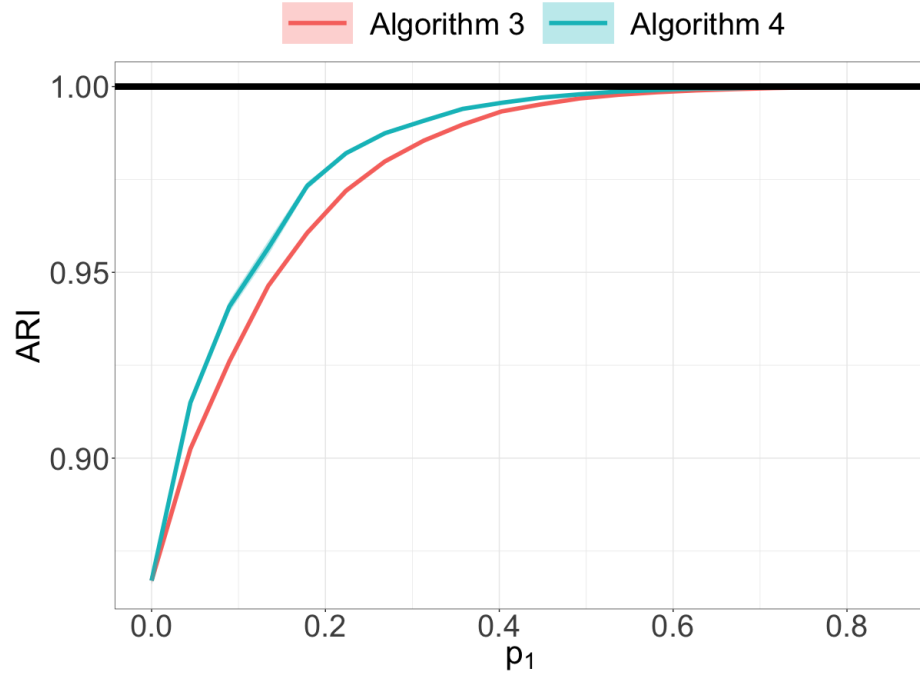


(b) Unbalanced:  $\pi = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ .

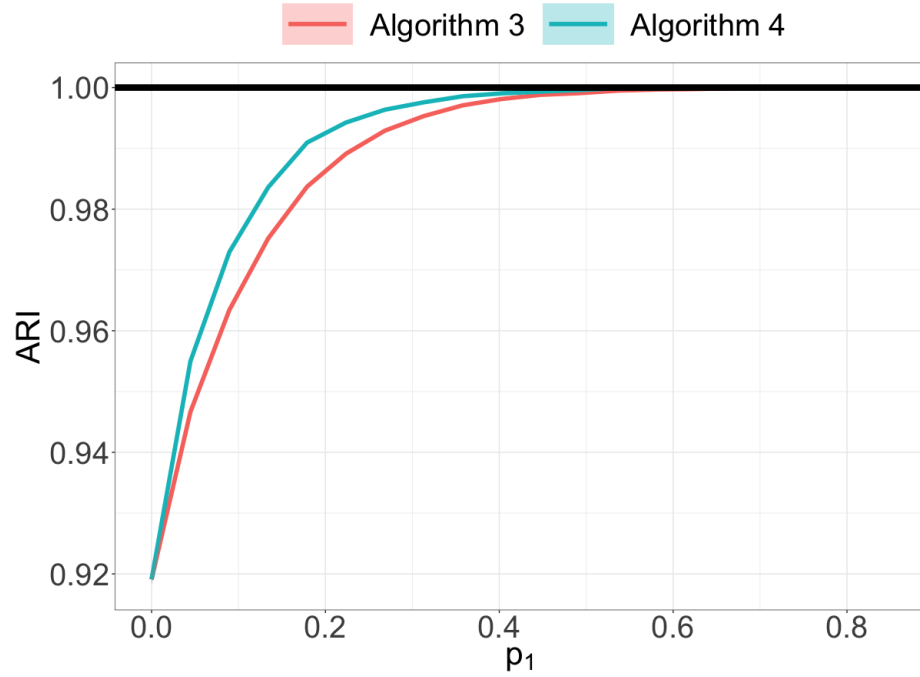
**Figure 4-4.** Simulations for 4-block SBM parameterized by block connectivity probability matrix  $\mathbf{B}$  as in Eq. (4.2) with initial sampling parameter  $p_0 = 0.01$  and dynamic network sampling parameter  $p_1 \in (0, p_{11}^{\max}]$  where  $p_{11}^{\max}$  is defined as in Eq. (4.10).

and adopt ARI to measure the performance.

Figure 4-5 shows ARI (`mean±stderr`) of two algorithms for  $p_1 \in (0, 0.85)$  where Figure 4-5a assumes  $\boldsymbol{\pi} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ , i.e., 1000 vertices in each block, and Figure 4-5b assumes  $\boldsymbol{\pi} = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ , i.e., 500 vertices in two of the blocks and 1500 vertices in the other two blocks. Note that both algorithms tend to have a better performance as  $p_1$  increases, i.e., as we sample more edges, and Algorithm 4 can always recover more accurate block structure than Algorithm 3. That is, given the same amount of resources, the proposed Chernoff-optimal dynamic network sampling scheme can yield better block recovery results. In other words, to reach the same level of performance, in terms of the empirical clustering results, the proposed Chernoff-optimal dynamic network sampling scheme needs less resources.



(a) Balanced:  $\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ .



(b) Unbalanced:  $\pi = (\frac{1}{8}, \frac{1}{8}, \frac{3}{8}, \frac{3}{8})$ .

**Figure 4-5.** Simulations for 4-block SBM parameterized by block connectivity probability matrix  $\mathbf{B}$  as in Eq. (4.2) with initial sampling parameter  $p_0 = 0.15$  and dynamic network sampling parameter  $p_1 \in (0, 0.85)$ .

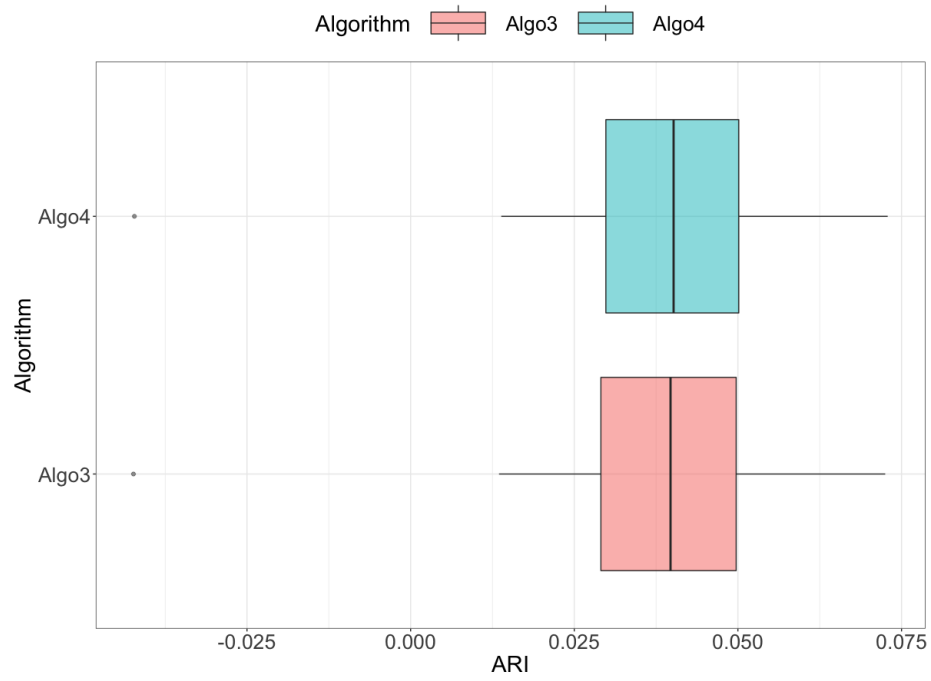
## Connectome Data

We also evaluate the performance of Algorithms 3 and 4 for real applications. We start the real data experiments with a diffusion MRI connectome dataset [1]. There are 114 graphs (connectomes) estimated by the NDMG pipeline [91] in this dataset. Each vertex in these graphs (the number of vertices  $n$  varies from 23728 to 42022) has a {Left, Right} hemisphere label and a {Gray, White} tissue label. We consider the potential 4 blocks as {LG, LW, RG, RW} where L and R denote the Left and Right hemisphere label, G and W denote the Gray and White tissue label.

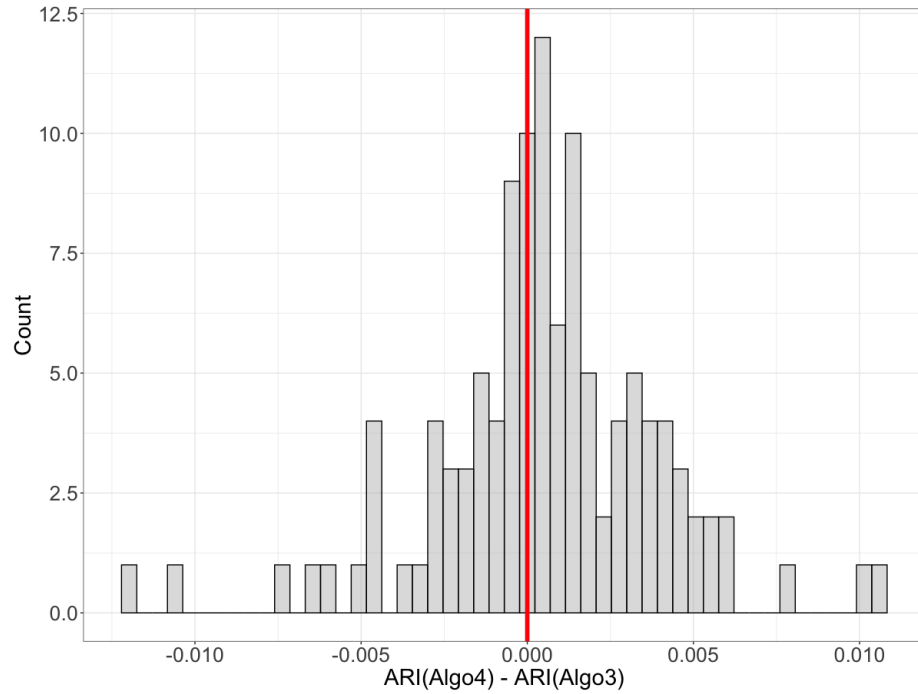
Here we consider initial sampling parameter  $p_0 = 0.25$  and dynamic network sampling parameter  $p_1 = 0.25$ . Let  $\Delta = \text{ARI}(\text{Algo4}) - \text{ARI}(\text{Algo3})$  where  $\text{ARI}(\text{Algo3})$  and  $\text{ARI}(\text{Algo4})$  denotes the ARI when we apply Algorithms 3 and 4 respectively. The following hypothesis testing yields **p-value=0.0184**.

$$H_0 : \text{median}(\Delta) \leq 0 \quad \text{v.s.} \quad H_A : \text{median}(\Delta) > 0. \quad (4.11)$$

It suggests that the proposed Chernoff-optimal dynamic network sampling scheme can yield better block recovery results given the same amount of resources.



(a) Boxplot of  $\text{ARI}(\text{Algo3})$  and  $\text{ARI}(\text{Algo4})$ .



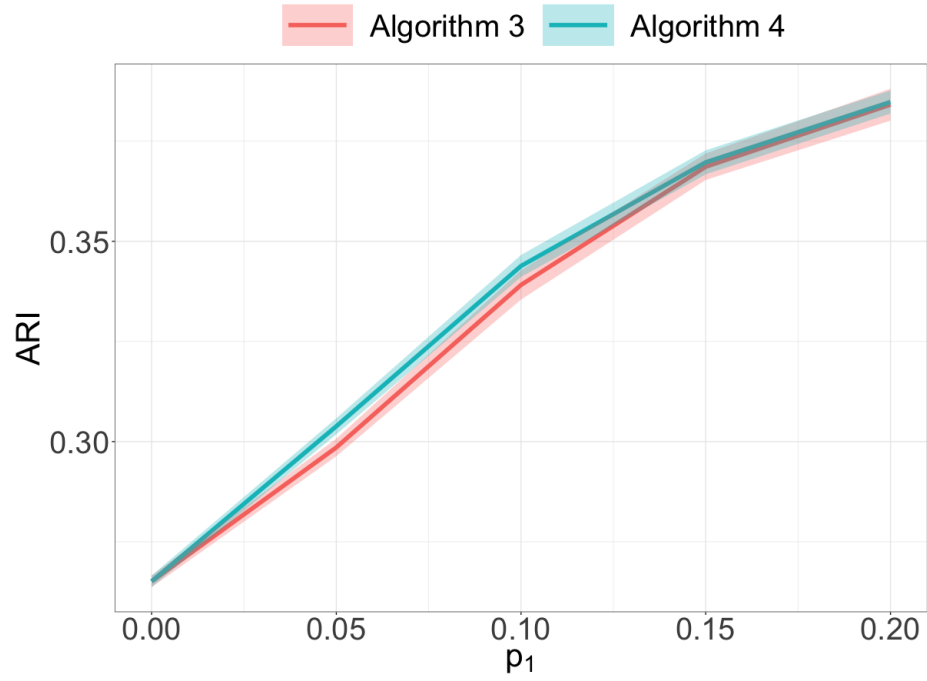
(b) Histogram of  $\Delta = \text{ARI}(\text{Algo4}) - \text{ARI}(\text{Algo3})$ .

**Figure 4-6.** Algorithms' comparative performance on diffusion MRI connectome data via ARI with initial sampling parameter  $p_0 = 0.25$  and dynamic network sampling parameter  $p_1 = 0.25$ .

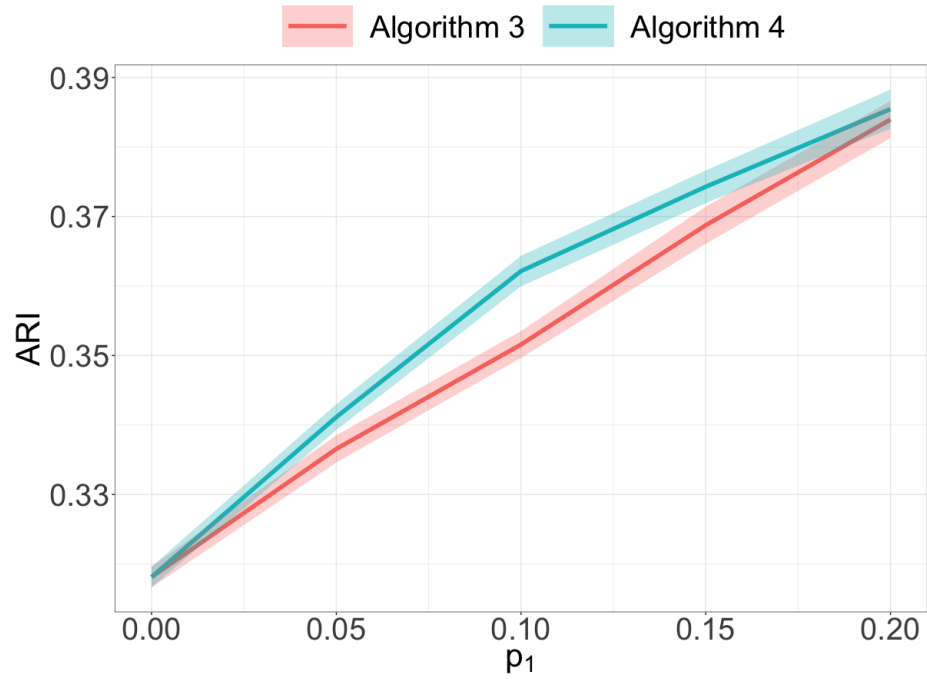
## Microsoft Bing Entity Data

Furthermore, we test our algorithms on a Microsoft bing entity dataset [94]. There are 2 graphs in this dataset where each has 13535 vertices. We treat block assignments estimated from the complete graph as ground truth. We consider initial sampling parameter  $p_0 \in \{0.2, 0.3\}$  and dynamic network sampling parameter  $p_1 \in \{0, 0.05, 0.1, 0.15, 0.2\}$ . For each  $p_1$ , we sample 100 times and compare the overall performance of Algorithm 3 and 4.

Figure 4-7 shows the results where ARI is reported as `mean( $\pm$ stderr)`. Still, it suggests that given the same amount of resources, the proposed Chernoff-optimal dynamic network sampling scheme can yield better block recovery results in general. In other words, to reach the same level of performance, in terms of the empirical clustering results, the proposed Chernoff-optimal dynamic network sampling scheme needs less resources in general.



(a)  $p_0 = 0.2$ ,  $p_1 \in \{0, 0.05, 0.1, 0.15, 0.2\}$ .



(b)  $p_0 = 0.3$ ,  $p_1 \in \{0, 0.05, 0.1, 0.15, 0.2\}$ .

**Figure 4-7.** Algorithms' comparative performance on Microsoft Bing entity data via ARI with different initial sampling parameter  $p_0$  and dynamic network sampling parameter  $p_1$ .

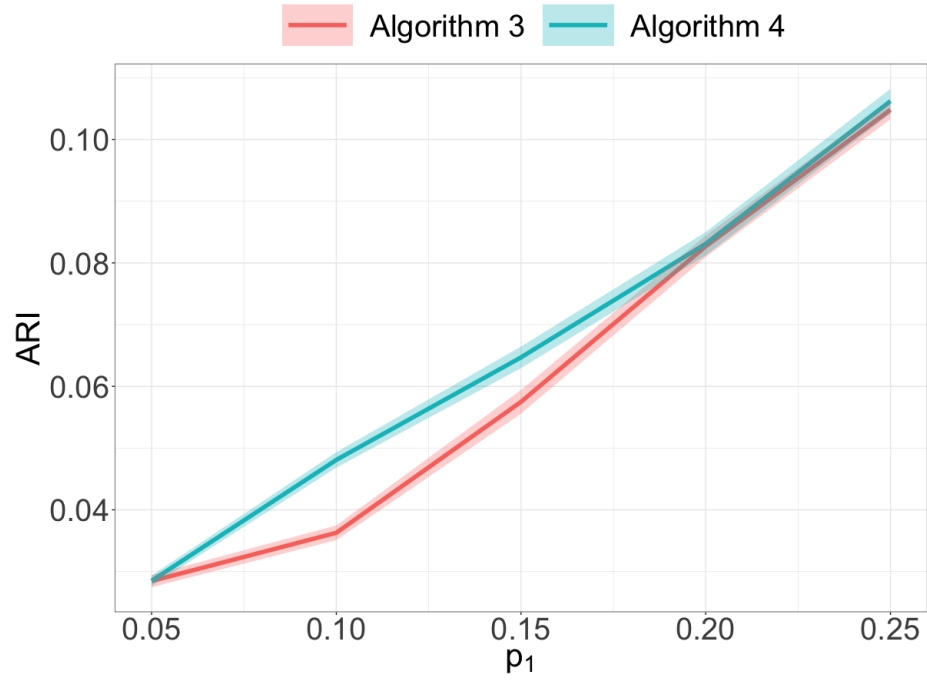
## Social Network Data

We also conduct real data experiments with 2 social network datasets. The description of these datasets are summarized as follows.

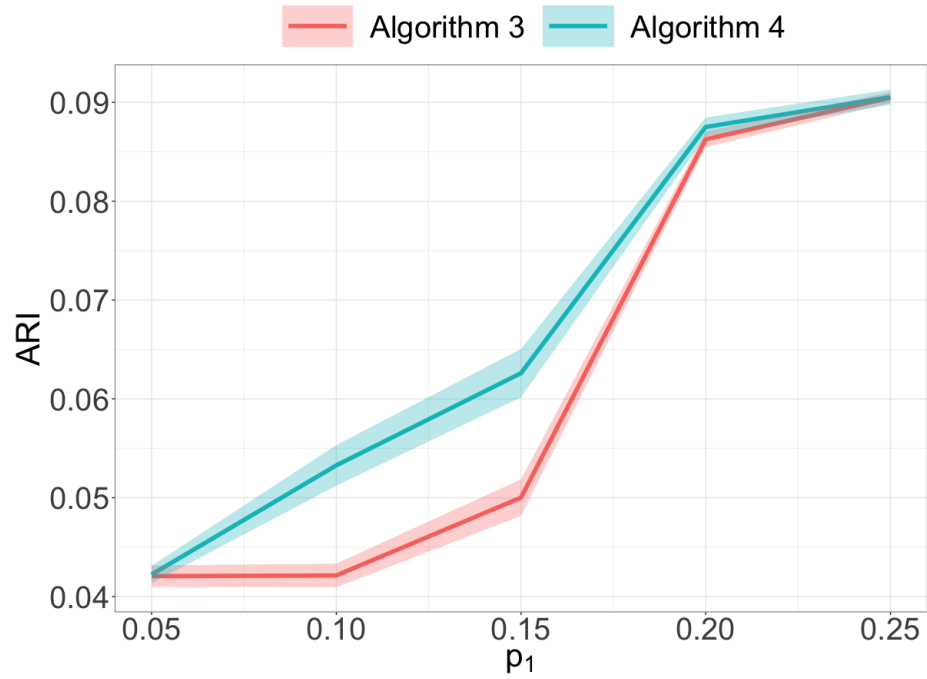
- LastFM asia social network data set [2, 4]: Vertices (the number of vertices  $n = 7624$ ) represent LastFM users from asian countries and edges (the number of edges  $e = 27806$ ) represent mutual follower relationships. We treat the location of users, which are derived from the country field for each user, as the potential block.
- Facebook large page-page network data set [2, 3]: Vertices (the number of vertices  $n = 22470$ ) represent official Facebook pages and edges (the number of edges  $e = 171002$ ) represent mutual likes. We treat 4 page types {Politician, Governmental Organization, Television Show, Company}, which are defined by Facebook, as the potential block.

We consider initial sampling parameter  $p_0 \in \{0.15, 0.35\}$  and dynamic network sampling parameter  $p_1 \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$ . For each  $p_1$ , we sample 100 times and compare the overall performance of Algorithm 3 and 4.

Figure 4-8 shows the results where ARI is reported as `mean( $\pm$ stderr)`. Again it suggests that given the same amount of resources, the proposed Chernoff-optimal dynamic network sampling scheme can yield better block recovery results. In other words, to reach the same level of performance, in terms of the empirical clustering results, the proposed Chernoff-optimal dynamic network sampling scheme needs less resources.



(a) LastFM:  $p_0 = 0.15$ ,  $p_1 \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$ .



(b) Facebook:  $p_0 = 0.35$ ,  $p_1 \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$ .

**Figure 4-8.** Algorithms' comparative performance on social network data via ARI with different initial sampling parameter  $p_0$  and dynamic network sampling parameter  $p_1$ .

# Chernoff Information and Chernoff Superiority

We end this chapter with investigation of Chernoff information and the analysis of Chernoff superiority for certain case of interest.

## Proof of Theorem 3

*Proof.* Let  $\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{U}^\top$  be the spectral decomposition of  $\mathbf{B}$  and  $\mathbf{B}' = p\mathbf{B}$  with  $p \in (0, 1)$ . Then we have

$$\mathbf{B}' = \mathbf{U}'\mathbf{S}(\mathbf{U}')^\top \quad \text{where} \quad \mathbf{U}' = \sqrt{p}\mathbf{U}. \quad (4.12)$$

By Remark 4, to represent these two SBMs parametrized by two block connectivity matrices  $\mathbf{B}$  and  $\mathbf{B}'$  respectively (with the same block assignment probability vector  $\boldsymbol{\pi}$ ) in the GRDPG models, we can take

$$\begin{aligned} \boldsymbol{\nu} &= [\boldsymbol{\nu}_1 \quad \cdots \quad \boldsymbol{\nu}_K]^\top = \mathbf{U}|\mathbf{S}|^{1/2} \in \mathbb{R}^{K \times d}, \\ \boldsymbol{\nu}' &= [\boldsymbol{\nu}'_1 \quad \cdots \quad \boldsymbol{\nu}'_K]^\top = \mathbf{U}'|\mathbf{S}|^{1/2} = \sqrt{p}\mathbf{U}|\mathbf{S}|^{1/2} = \sqrt{p}\boldsymbol{\nu} \in \mathbb{R}^{K \times d}. \end{aligned} \quad (4.13)$$

Then for any  $k \in \{1, \dots, K\}$ , we have  $\boldsymbol{\nu}'_k = \sqrt{p}\boldsymbol{\nu}_k \in \mathbb{R}^d$ . By Theorem 1, we have

$$\begin{aligned} \boldsymbol{\Delta} &= \sum_{k=1}^K \pi_k \boldsymbol{\nu}_k \boldsymbol{\nu}_k^\top \in \mathbb{R}^{d \times d}, \\ \boldsymbol{\Delta}' &= \sum_{k=1}^K \pi_k \boldsymbol{\nu}'_k (\boldsymbol{\nu}'_k)^\top = p \sum_{k=1}^K \pi_k \boldsymbol{\nu}_k \boldsymbol{\nu}_k^\top = p\boldsymbol{\Delta} \in \mathbb{R}^{d \times d}. \end{aligned} \quad (4.14)$$

Note that  $\mathbf{B}$  and  $\mathbf{B}'$  have the same eigenvalues, thus we have  $\mathbf{I}_{d_+d_-} = \mathbf{I}'_{d_+d_-}$ . See

also Lemma 2 of [95]. Then for  $k \in \{1, \dots, K\}$ , we have

$$\begin{aligned}
\Sigma_k &= \mathbf{I}_{d_+d_-} \Delta^{-1} \mathbb{E} \left[ \left( \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu} \right) \left( 1 - \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu} \right) \boldsymbol{\nu} \boldsymbol{\nu}^\top \right] \Delta^{-1} \mathbf{I}_{d_+d_-} \\
&= \mathbf{I}_{d_+d_-} \Delta^{-1} \left[ \sum_{\ell=1}^K \pi_\ell \left( \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \left( 1 - \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \boldsymbol{\nu}_\ell \boldsymbol{\nu}_\ell^\top \right] \Delta^{-1} \mathbf{I}_{d_+d_-} \in \mathbb{R}^{d \times d}, \\
\Sigma'_k &= \frac{1}{p^2} \mathbf{I}_{d_+d_-} \Delta^{-1} \left[ p^2 \sum_{\ell=1}^K \pi_\ell \left( \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \left( 1 - p \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \boldsymbol{\nu}_\ell \boldsymbol{\nu}_\ell^\top \right] \Delta^{-1} \mathbf{I}_{d_+d_-} \\
&= \mathbf{I}_{d_+d_-} \Delta^{-1} \left[ p \sum_{\ell=1}^K \pi_\ell \left( \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \left( 1 - \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \boldsymbol{\nu}_\ell \boldsymbol{\nu}_\ell^\top \right] \Delta^{-1} \mathbf{I}_{d_+d_-} \\
&\quad + \mathbf{I}_{d_+d_-} \Delta^{-1} \left[ (1-p) \sum_{\ell=1}^K \pi_\ell \left( \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell \right) \boldsymbol{\nu}_\ell \boldsymbol{\nu}_\ell^\top \right] \Delta^{-1} \mathbf{I}_{d_+d_-} \\
&= p \Sigma_k + \mathbf{V}^\top \mathbf{D}_k(p) \mathbf{V} \in \mathbb{R}^{d \times d},
\end{aligned} \tag{4.15}$$

where

$$\begin{aligned}
\mathbf{V} &= \boldsymbol{\nu} \Delta^{-1} \mathbf{I}_{d_+d_-} \in \mathbb{R}^{K \times d}, \\
\mathbf{D}_k(p) &= (1-p) \text{diag} \left( \pi_1 \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_1, \dots, \pi_K \boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_K \right) \in (0, 1)^{K \times K}.
\end{aligned} \tag{4.16}$$

Recall that by Remark 4, we have  $\boldsymbol{\nu}_k^\top \mathbf{I}_{d_+d_-} \boldsymbol{\nu}_\ell = \mathbf{B}_{k\ell} \in (0, 1)$  for all  $k, \ell \in \{1, \dots, K\}$ . Then we have  $\mathbf{D}_k(p)$  is positive-definite for any  $k \in \{1, \dots, K\}$  and  $p \in (0, 1)$ . For  $k, \ell \in \{1, \dots, K\}$  and  $t \in (0, 1)$ , let  $\Sigma_{k\ell}(t)$  and  $\Sigma'_{k\ell}(t)$  denote the matrices as in Eq. (2.29) corresponding to  $\mathbf{B}$  and  $\mathbf{B}'$  respectively, i.e.,

$$\begin{aligned}
\Sigma_{k\ell}(t) &= t \Sigma_k + (1-t) \Sigma_\ell \in \mathbb{R}^{d \times d}, \\
\Sigma'_{k\ell}(t) &= t \Sigma'_k + (1-t) \Sigma'_\ell \\
&= t \left[ p \Sigma_k + \mathbf{V}^\top \mathbf{D}_k(p) \mathbf{V} \right] + (1-t) \left[ p \Sigma_\ell + \mathbf{V}^\top \mathbf{D}_\ell(p) \mathbf{V} \right] \\
&= p \left[ t \Sigma_k + (1-t) \Sigma_\ell \right] + \mathbf{V}^\top \left[ t \mathbf{D}_k(p) + (1-t) \mathbf{D}_\ell(p) \right] \mathbf{V} \\
&= p \Sigma_{k\ell}(t) + \mathbf{V}^\top \mathbf{D}_{k\ell}(p, t) \mathbf{V} \in \mathbb{R}^{d \times d},
\end{aligned} \tag{4.17}$$

where

$$\mathbf{D}_{k\ell}(p, t) = t \mathbf{D}_k(p) + (1-t) \mathbf{D}_\ell(p) \in \mathbb{R}_+^{K \times K}. \tag{4.18}$$

Recall that  $\mathbf{D}_k(p)$  and  $\mathbf{D}_\ell(p)$  are both positive-definite for any  $k, \ell \in \{1, \dots, K\}$  and  $p \in (0, 1)$ , thus  $\mathbf{D}_{k\ell}(p, t)$  is also positive-definite for any  $k, \ell \in \{1, \dots, K\}$  and  $p, t \in (0, 1)$ . Now by the Sherman-Morrison-Woodbury formula as in Corollary 3, we have

$$\begin{aligned} [\boldsymbol{\Sigma}'_{k\ell}(t)]^{-1} &= \left[ p\boldsymbol{\Sigma}_{k\ell}(t) + \mathbf{V}^\top \mathbf{D}_{k\ell}(p, t) \mathbf{V} \right]^{-1} \\ &= \frac{1}{p} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) - \frac{1}{p^2} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \mathbf{V}^\top \left[ \mathbf{D}_{k\ell}^{-1}(p, t) + \frac{1}{p} \mathbf{V} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \mathbf{V}^\top \right]^{-1} \mathbf{V} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \quad (4.19) \\ &= \frac{1}{p} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) - \frac{1}{p^2} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \mathbf{V}^\top \mathbf{M}_{k\ell}^{-1}(p, t) \mathbf{V} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \in \mathbb{R}^{d \times d}, \end{aligned}$$

where

$$\mathbf{M}_{k\ell}(p, t) = \mathbf{D}_{k\ell}^{-1}(p, t) + \frac{1}{p} \mathbf{V} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \mathbf{V}^\top \in \mathbb{R}^{K \times K}. \quad (4.20)$$

Recall that for any  $k, \ell \in \{1, \dots, K\}$  and  $p, t \in (0, 1)$ ,  $\mathbf{D}_{k\ell}(p, t)$  and  $\boldsymbol{\Sigma}_{k\ell}(t)$  are both positive-definite, thus  $\mathbf{M}_{k\ell}(p, t)$  is also positive-definite. Then for any  $k, \ell \in \{1, \dots, K\}$  and  $p, t \in (0, 1)$ , we have

$$\begin{aligned} (\boldsymbol{\nu}'_k - \boldsymbol{\nu}'_\ell)^\top [\boldsymbol{\Sigma}'_{k\ell}(t)]^{-1} (\boldsymbol{\nu}'_k - \boldsymbol{\nu}'_\ell) &= p(\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \\ &\quad \left[ \frac{1}{p} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) - \frac{1}{p^2} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \mathbf{V}^\top \mathbf{M}_{k\ell}^{-1}(p, t) \mathbf{V} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) \right] \\ &\quad (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) \\ &= (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) - \frac{1}{p} \mathbf{x}^\top \mathbf{M}_{k\ell}^{-1}(p, t) \mathbf{x} \\ &= (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) - h_{k\ell}(p, t), \end{aligned} \quad (4.21)$$

where

$$\begin{aligned} \mathbf{x} &= \mathbf{V} \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) \in \mathbb{R}^K, \\ h_{k\ell}(p, t) &= \frac{1}{p} \mathbf{x}^\top \mathbf{M}_{k\ell}^{-1}(p, t) \mathbf{x}. \end{aligned} \quad (4.22)$$

Recall that for any  $k, \ell \in \{1, \dots, K\}$  and  $p, t \in (0, 1)$ ,  $\mathbf{M}_{k\ell}(p, t)$  is positive-definite, thus we have  $h_{k\ell}(p, t) > 0$ . Together with Eq. (4.21), we have

$$t(1-t)(\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t) (\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) > t(1-t)(\boldsymbol{\nu}'_k - \boldsymbol{\nu}'_\ell)^\top [\boldsymbol{\Sigma}'_{k\ell}(t)]^{-1} (\boldsymbol{\nu}'_k - \boldsymbol{\nu}'_\ell). \quad (4.23)$$

Thus for any  $k, \ell \in \{1, \dots, K\}$ , we have

$$\begin{aligned}
C_{k,\ell}(\mathbf{B}, \boldsymbol{\pi}) &= \sup_{t \in (0,1)} \left[ t(1-t)(\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell)^\top \boldsymbol{\Sigma}_{k\ell}^{-1}(t)(\boldsymbol{\nu}_k - \boldsymbol{\nu}_\ell) \right], \\
&> \sup_{t \in (0,1)} \left[ t(1-t)(\boldsymbol{\nu}'_k - \boldsymbol{\nu}'_\ell)^\top [\boldsymbol{\Sigma}'_{k\ell}(t)]^{-1} (\boldsymbol{\nu}'_k - \boldsymbol{\nu}'_\ell) \right] \\
&= C_{k,\ell}(\mathbf{B}', \boldsymbol{\pi}).
\end{aligned} \tag{4.24}$$

Let  $\rho_B$  and  $\rho_{B'}$  denote the Chernoff information obtained as in Eq. (2.29) corresponding to  $\mathbf{B}$  and  $\mathbf{B}'$  respectively (with the same block assignment probability vector  $\boldsymbol{\pi}$ ). Then we have

$$\rho_B \approx \min_{k \neq \ell} C_{k,\ell}(\mathbf{B}, \boldsymbol{\pi}) > \min_{k \neq \ell} C_{k,\ell}(\mathbf{B}', \boldsymbol{\pi}) \approx \rho_{B'}. \tag{4.25}$$

Thus we have  $\mathbf{B} \succ \mathbf{B}' = p\mathbf{B}$  for  $p \in (0, 1)$ . □

## Proof of Corollary 4

*Proof.* By Eq. (4.1) and Eq. (4.3), we have

$$\begin{aligned}
\mathbf{B}_0 &= \frac{p_0}{p_0 + p_1} \mathbf{B}_1, \\
\mathbf{B}_1 &= (p_0 + p_1) \mathbf{B}.
\end{aligned} \tag{4.26}$$

Recall that  $p_0 \in (0, 1)$  and  $p_1 \in (0, 1 - p_0)$ . Then by Theorem 3, we have  $\mathbf{B} \succ \mathbf{B}_1 \succ \mathbf{B}_0$ . □

# Chapter 5

## Conclusions and Discussion

This dissertation adopts the notion of Chernoff information to investigate two problems in the area of community detection. Motivated by the Chernoff analysis, we focus on models and algorithms that can incorporate information from both the adjacency matrices and the vertex covariates into the estimation of block structure, and design methods and procedures that can still identify the potential community structure when only a limited number of edges in graphs is observed.

On one hand, we study the problem of community detection for SBMs with vertex covariates. Specifically, we investigate two model-based spectral algorithms to assess the effect of observed and unobserved vertex heterogeneity on block structure in graphs. The major difference of these two algorithms in estimating the underlying block assignments is whether we estimate the vertex covariate effect using the information from the observed vertex covariates. To analyze the algorithms' performance, we employ the concept of Chernoff information and derive the Chernoff ratio expression for certain model of interest. We also simulate multiple adjacency matrices with varied type of covariates to compare the algorithms' performance via empirical clustering results measured by commonly used metrics such as ARI. In addition, we conduct real data experiments on diffusion MRI connectome datasets and social network datasets to evaluate the performance of these algorithms for real applications. Analytic results, simulations, and real data experiments suggest that the second algorithm is often

preferred: we can better estimate the induced block assignments and reveal underlying block structure by using additional information contained in vertex covariates. Our findings also emphasize the importance of distinguishing between observed and unobserved factors that can affect block structure in graphs.

At this stage, we focus on the model specified as in Definition 4 and Remark 5 where indicator function is used to measure the vertex covariate effect and identity function is used as the link between edge probabilities and latent positions. We also investigate the flexibility and generalizability of our approaches by considering categorical vertex covariates in the simulations. The extension from discrete vertex covariates to continuous vertex covariates is under investigation, for instance, via latent structure models [96]. The indicator function in our setting is used to measure the vertex covariate effect for binary and generally categorical vertex covariates under the intuition that vertices having the same covariates are more likely to form an edge between them. That being said, different functions can also be adopted for the continuous vertex covariates following the similar intuition. For example, similarity and distance functions can be chosen based on the nature of different vertex covariates to measure how they can influence graph structure in different ways. One another extension for this problem is to replace the identity link with, say, the logit link function. The idea of using Chernoff information to compare algorithms' performance can be adopted for all the above generalizations and numerical evaluations can be obtained in the absence of closed-form expressions, which in turn can reveal how graph structure will affect our algorithms and provide guidelines for real application.

On the other hand, we study the problem of dynamic network sampling for community detection. We propose a dynamic network sampling scheme to optimize block recovery for SBM when we don't have enough resources to observe the entire graphs. Theoretically, we provide justification of our proposed Chernoff-optimal dynamic sampling scheme via the notion of Chernoff information and Chernoff superiority.

Practically, we evaluate the performance, in terms of block recovery, of our method on several real datasets including diffusion MRI connectome dataset, Microsoft Bing entity graph transitions dataset and social network datasets. Both theoretically and practically results suggest that our proposed method can identify vertices that have the most impact on block structure and only check whether there are edges between them to save significant resources but still recover the block structure.

As discussed before, the Chernoff-optimal dynamic sampling scheme depends on the initial clustering results to identify Chernoff-active blocks and construct dynamic edge set. To that end, the performance could be impacted if the initial clustering is not very ideal. One of the future direction for this approach is to design certain strategy to reduce this dependency such that the proposed scheme is more robust.

# Appendix I

## Additional Preliminaries in Random Graph Models

### Degree Corrected Stochastic Blockmodel

**Definition 10** (Degree Corrected Stochastic Blockmodel [14]). *The  $K$ -block degree corrected stochastic blockmodel (DCSBM) generalizes the standard  $K$ -block SBM by allowing for vertices within each block to have different expected degrees. It can be parameterized by a block connectivity probability matrix  $\mathbf{B} \in [0, 1]^{K \times K}$ , a nonnegative vector of block assignment probabilities  $\boldsymbol{\pi} \in [0, 1]^K$  summing to unity, and a vector of weights  $\mathbf{w} \in [0, 1]^n$ . Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix and  $\boldsymbol{\tau} \in \{1, \dots, K\}^n$  be a vector of block assignments with  $\tau_i = k$  if vertex  $i$  is in block  $k$  (occurring with probability  $\pi_k$ ). We say  $(\mathbf{A}, \boldsymbol{\tau}) \sim \text{DCSBM}(n, \mathbf{B}, \boldsymbol{\pi}, \mathbf{w})$  if for any  $i, j \in \{1, \dots, n\}$*

$$\begin{aligned} A_{ij} &\sim \text{Bernoulli}(P_{ij}), \\ P_{ij} &= w_i w_j B_{\tau_i \tau_j}. \end{aligned} \tag{I.1}$$

**Remark 13.** *The DCSBM is a special case of the GRDPG model. Let  $(\mathbf{A}, \boldsymbol{\tau}) \sim \text{DCSBM}(n, \mathbf{B}, \boldsymbol{\pi}, \mathbf{w})$  as in Definition 10 where  $\mathbf{B} \in (0, 1)^{K \times K}$  with  $d_+$  strictly positive eigenvalues and  $d_-$  strictly negative eigenvalues. To represent this DCSBM in the GRDPG model, we can choose  $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_K \in \mathbb{R}^d$  where  $d = d_+ + d_-$  such that  $\boldsymbol{\nu}_k^\top \mathbf{I}_{d_+ d_-} \boldsymbol{\nu}_\ell = B_{k\ell}$  for all  $k, \ell \in \{1, \dots, K\}$ . For example, we can take  $\boldsymbol{\nu} = \mathbf{U}_B |\mathbf{S}_B|^{1/2}$  where  $\mathbf{B} = \mathbf{U}_B \mathbf{S}_B \mathbf{U}_B^\top$  is the spectral decomposition of  $\mathbf{B}$  after re-ordering. Then for any  $i, j \in \{1, \dots, n\}$ , given  $\tau_i = k$  and  $\tau_j = \ell$ , we have*

$$w_i w_j B_{\tau_i \tau_j} = w_i w_j B_{k\ell} = (w_i \boldsymbol{\nu}_k)^\top \mathbf{I}_{d_+ d_-} (w_j \boldsymbol{\nu}_\ell). \tag{I.2}$$

*Thus the latent position of vertex  $i$  for  $i \in \{1, \dots, n\}$  is given by  $\mathbf{X}_i = w_i \boldsymbol{\nu}_k$  if  $\tau_i = k$ .*

# Mixed Membership Stochastic Blockmodel

**Definition 11** (Mixed Membership Stochastic Blockmodel [15]). *The  $K$ -block mixed membership stochastic blockmodel (MMSBM) extends the standard  $K$ -block SBM by allowing for each vertex to be in a mixture of different blocks. It can be parameterized by a block connectivity probability matrix  $\mathbf{B} \in [0, 1]^{K \times K}$  and  $n$  nonnegative vectors of block assignment probabilities  $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n \in [0, 1]^K$  with each  $\boldsymbol{\pi}_i$  summing to unity for  $i \in \{1, \dots, n\}$ . Let  $\mathbf{A} \in \{0, 1\}^{n \times n}$  be an adjacency matrix and  $\boldsymbol{\pi} = [\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n]^\top \in [0, 1]^{n \times K}$  denote the matrix of block assignment probabilities. We say  $\mathbf{A} \sim \text{MMSBM}(n, \mathbf{B}, \boldsymbol{\pi})$  if for any  $i, j \in \{1, \dots, n\}$*

$$\begin{aligned} A_{ij} &\sim \text{Bernoulli}(P_{ij}), \\ P_{ij} &= \boldsymbol{\pi}_i^\top \mathbf{B} \boldsymbol{\pi}_j. \end{aligned} \tag{I.3}$$

**Remark 14.** *The MMSBM is a special case of the GRDPG model. Let  $\mathbf{A} \sim \text{MMSBM}(n, \mathbf{B}, \boldsymbol{\pi})$  as in Definition 11 where  $\mathbf{B} \in (0, 1)^{K \times K}$  with  $d_+$  strictly positive eigenvalues and  $d_-$  strictly negative eigenvalues. To represent this MMSBM in the GRDPG model, we can choose  $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_K \in \mathbb{R}^d$  where  $d = d_+ + d_-$  such that  $\boldsymbol{\nu}_k^\top \mathbf{I}_{d_+ d_-} \boldsymbol{\nu}_\ell = B_{k\ell}$  for all  $k, \ell \in \{1, \dots, K\}$ . For example, we can take  $\boldsymbol{\nu} = \mathbf{U}_B |\mathbf{S}_B|^{1/2}$  where  $\mathbf{B} = \mathbf{U}_B \mathbf{S}_B \mathbf{U}_B^\top$  is the spectral decomposition of  $\mathbf{B}$  after re-ordering. Then we have for any  $i, j \in \{1, \dots, n\}$*

$$\boldsymbol{\pi}_i^\top \mathbf{B} \boldsymbol{\pi}_j = \left( \sum_{k=1}^K \pi_{ik} \boldsymbol{\nu}_k \right)^\top \mathbf{I}_{d_+ d_-} \left( \sum_{k=1}^K \pi_{jk} \boldsymbol{\nu}_k \right). \tag{I.4}$$

*Thus the latent position of vertex  $i$  for  $i \in \{1, \dots, n\}$  is given by*

$$\mathbf{X}_i = \sum_{k=1}^K \pi_{ik} \boldsymbol{\nu}_k. \tag{I.5}$$

## Appendix II

# Additional Preliminaries in Spectral Methods

### Adjacency Spectral Embedding

**Theorem 4** (Consistency of ASE for GRDPG [11]). *Let  $(\mathbf{A}, \mathbf{X}) \sim \text{GRDPG}(n, d_+, d_-)$  be the adjacency matrix and associated latent positions of a  $d$ -dimensional GRDPG as in Definition 2. Let  $\widehat{\mathbf{X}}$  be the ASE of  $\mathbf{A}$  with  $\widehat{\mathbf{X}}_i$  as the  $i$ -th row (same for  $\mathbf{X}_i$ ). Under certain sparsity conditions [11], there exists a universal constant  $c > 1$  and a matrix  $\mathbf{M} \in \mathbb{R}^{d \times d}$  satisfying  $\mathbf{M}\mathbf{I}_{d_+d_-}\mathbf{M}^\top = \mathbf{I}_{d_+d_-}$  such that*

$$\max_{i \in \{1, \dots, n\}} \|\mathbf{M}\widehat{\mathbf{X}}_i - \mathbf{X}_i\| = O_P\left(\frac{\log^c n}{\sqrt{n}}\right). \quad (\text{II.1})$$

Here a random variable  $X$  is said to be  $O_P(f(n))$  if for any constant  $C > 0$  there exists an integer  $N_C$  and a constant  $M_C$  such that for all  $n \geq N_C$ ,  $|X| \leq M_C f(n)$  with probability at least  $1 - n^{-C}$ .

**Theorem 5** (CLT of ASE for GRDPG [11]). *Let  $(\mathbf{A}, \mathbf{X}) \sim \text{GRDPG}(n, d_+, d_-)$  be the adjacency matrix and associated latent positions of a  $d$ -dimensional GRDPG as in Definition 2 from a distribution  $F$ . Let  $\widehat{\mathbf{X}}$  be the ASE of  $\mathbf{A}$  with  $\widehat{\mathbf{X}}_i$  as the  $i$ -th row (same for  $\mathbf{X}_i$ ). Under certain sparsity conditions [11], there exists a matrix  $\mathbf{M} \in \mathbb{R}^{d \times d}$  satisfying  $\mathbf{M}\mathbf{I}_{d_+d_-}\mathbf{M}^\top = \mathbf{I}_{d_+d_-}$  such that for index  $i$ ,*

$$\sqrt{n}(\mathbf{M}\widehat{\mathbf{X}}_i - \mathbf{X}_i) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_i). \quad (\text{II.2})$$

where for  $\mathbf{Y} \sim F$

$$\Sigma_i = \Sigma(\mathbf{X}_i) = \mathbf{I}_{d_+d_-} \Delta^{-1} \mathbb{E} \left[ (\mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{Y}) (1 - \mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{Y}) \mathbf{Y} \mathbf{Y}^\top \right] \Delta^{-1} \mathbf{I}_{d_+d_-}, \quad (\text{II.3})$$

with

$$\Delta = \mathbb{E} [\mathbf{Y} \mathbf{Y}^\top]. \quad (\text{II.4})$$

## Laplacian Spectral Embedding

**Theorem 6** (Consistency of LSE for GRDPG [11]). *Let  $(\mathbf{A}, \mathbf{X}) \sim \text{GRDPG}(n, d_+, d_-)$  be the adjacency matrix and associated latent positions of a  $d$ -dimensional GRDPG as in Definition 2. Let  $\widetilde{\mathbf{X}}$  be the LSE of  $\mathbf{A}$  with  $\widetilde{\mathbf{X}}_i$  as the  $i$ -th row (same for  $\mathbf{X}_i$ ). Under certain sparsity conditions [11], there exists a universal constant  $c > 1$  and a matrix  $\mathbf{M} \in \mathbb{R}^{d \times d}$  satisfying  $\mathbf{M}\mathbf{I}_{d_+d_-}\mathbf{M}^\top = \mathbf{I}_{d_+d_-}$  such that*

$$\max_{i \in \{1, \dots, n\}} \left\| \mathbf{M}\widetilde{\mathbf{X}}_i - \frac{\mathbf{X}_i}{\sqrt{\sum_j \mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{X}_j}} \right\| = O_P \left( \frac{\log^c n}{n} \right). \quad (\text{II.5})$$

Here a random variable  $X$  is said to be  $O_P(f(n))$  if for any constant  $C > 0$  there exists an integer  $N_C$  and a constant  $M_C$  such that for all  $n \geq N_C$ ,  $|X| \leq M_C f(n)$  with probability at least  $1 - n^{-C}$ .

**Theorem 7** (CLT of LSE for GRDPG [11]). *Let  $(\mathbf{A}, \mathbf{X}) \sim \text{GRDPG}(n, d_+, d_-)$  be the adjacency matrix and associated latent positions of a  $d$ -dimensional GRDPG as in Definition 2 from a distribution  $F$ . Let  $\widetilde{\mathbf{X}}$  be the LSE of  $\mathbf{A}$  with  $\widetilde{\mathbf{X}}_i$  as the  $i$ -th row (same for  $\mathbf{X}_i$ ). Under certain sparsity conditions [11], there exists a matrix  $\mathbf{M} \in \mathbb{R}^{d \times d}$  satisfying  $\mathbf{M}\mathbf{I}_{d_+d_-}\mathbf{M}^\top = \mathbf{I}_{d_+d_-}$  such that for index  $i$ ,*

$$n \left( \mathbf{M}\widetilde{\mathbf{X}}_i - \frac{\mathbf{X}_i}{\sqrt{\sum_j \mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{X}_j}} \right) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \widetilde{\Sigma}_i). \quad (\text{II.6})$$

where for  $\mathbf{Y} \sim F$

$$\widetilde{\Sigma}_i = \widetilde{\Sigma}(\mathbf{X}_i) = \mathbf{I}_{d_+d_-} \widetilde{\Delta}^{-1} \widetilde{\Gamma}_i \widetilde{\Delta}^{-1} \mathbf{I}_{d_+d_-}, \quad (\text{II.7})$$

with

$$\begin{aligned} \boldsymbol{\mu} &= \mathbb{E}[\mathbf{Y}], \\ \widetilde{\Delta} &= \mathbb{E} \left[ \frac{\mathbf{Y}\mathbf{Y}^\top}{\boldsymbol{\mu}^\top \mathbf{I}_{d_+d_-} \mathbf{Y}} \right], \\ \mathbf{T}_1 &= \frac{\mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{Y} (1 - \mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \mathbf{Y})}{\mathbf{X}_i^\top \mathbf{I}_{d_+d_-} \boldsymbol{\mu}}, \\ \mathbf{T}_2 &= \frac{\mathbf{Y}}{\boldsymbol{\mu}^\top \mathbf{I}_{d_+d_-} \mathbf{Y}} - \frac{\widetilde{\Delta} \mathbf{I}_{d_+d_-} \mathbf{X}_i}{2 \boldsymbol{\mu}^\top \mathbf{I}_{d_+d_-} \mathbf{X}_i}, \\ \widetilde{\Gamma}_i &= \mathbb{E}[\mathbf{T}_1 \mathbf{T}_2 \mathbf{T}_2^\top]. \end{aligned} \quad (\text{II.8})$$

# Spectral Clustering

**Definition 12** (Graph Notation). Let  $G = (V, E)$  denote a graph with vertex set  $V$  and edge set  $E$ . Let  $\mathbf{W}$  denote the weighted adjacency matrix of the graph  $G$ , i.e.,  $W_{ij} > 0$  is the weight of the edge between vertex  $i$  and vertex  $j$  and  $W_{ij} = 0$  suggests that there is no edge between vertex  $i$  and vertex  $j$ . If the graph  $G$  is undirected, then  $\mathbf{W}$  is symmetric. Let  $\mathbf{D}$  denote the degree matrix of graph  $G$ , that is, a diagonal matrix with

$$D_{ii} = \sum_{j \neq i} W_{ij}. \quad (\text{II.9})$$

**Definition 13** (Similarity Matrix [18]). Given a data matrix  $\mathbf{A} \in \mathbb{R}^{n \times d}$  where  $n$  is the number of data points and  $d$  is the number of features for each data point. There are several commonly used approaches to construct a similarity matrix  $\mathbf{W}$  of the original data matrix  $\mathbf{A}$  where  $W_{ij}$  represents the similarities between data point  $i$  and data point  $j$ . For example,

- The  $\epsilon$ -neighborhood approach: first compute the pairwise distances between all data points and set  $W_{ij} = w_{ij}$  if the distance between data point  $i$  and data point  $j$  is smaller than  $\epsilon$  where  $w_{ij} = 1$  for unweighted case and some other positive value based on certain similarity measure (for example, distances) for weighted case,  $W_{ij} = 0$  otherwise.
- The  $k$ -nearest neighbor approach: first compute the pairwise distances between all data points and set  $W_{ij} = w_{ij}$  if data point  $j$  is among the  $k$ -nearest neighbors of data point  $i$  where  $w_{ij} = 1$  for unweighted case and some other positive value based on certain similarity measure (for example, distances) for weighted case,  $W_{ij} = 0$  otherwise. This strategy usually yields a non-symmetric similarity matrix as the neighborhood relationship may not be symmetric, i.e., data point  $j$  may be among the  $k$ -nearest neighbors of data point  $i$  but data point  $i$  may not be among the  $k$ -nearest neighbors of data point  $j$ . Alternatively, one can construct a symmetric similarity matrix by setting  $W_{ij} = w_{ij}$  if data point  $j$  is among the  $k$ -nearest neighbors of data point  $i$  or/and data point  $i$  is among the  $k$ -nearest neighbors of data point  $j$  where  $w_{ij} = 1$  for unweighted case and some other positive value based on certain similarity measure (for example, distances) for weighted case,  $W_{ij} = 0$  otherwise.
- The fully connected approach: construct the similarity matrix based on certain function such as Gaussian similarity function. That is,

$$W_{ij} = \exp \left( -\frac{\|\mathbf{A}_i - \mathbf{A}_j\|^2}{2\sigma^2} \right),$$

where  $\sigma$  is a hyperparameter to control the width of the neighborhoods, which plays a similar role as  $\epsilon$  in the  $\epsilon$ -neighborhood approach.

**Definition 14** (Graph Laplacian Matrix [18]). *Given a graph  $G$  and the associated weighted adjacency matrix  $\mathbf{W}$ . Let  $\mathbf{D}$  denote the degree matrix of graph  $G$  as in Definition 12. The unnormalized graph Laplacian matrix is given by*

$$\mathbf{L} = \mathbf{D} - \mathbf{W}. \quad (\text{II.10})$$

*There are several similar characterizations for the normalized graph Laplacian matrix, for example*

$$\begin{aligned} \mathbf{L}_{sym} &= \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}, \\ \mathbf{L}_{rw} &= \mathbf{D}^{-1} \mathbf{L} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{W}. \end{aligned} \quad (\text{II.11})$$

**Remark 15** (Properties of Laplacian Matrix [18]). *Let  $\mathbf{L}$  denote the unnormalized Laplacian matrix as in Eq. (II.10) and  $\mathbf{L}_{sym}$  denote the normalized Laplacian matrix as in Eq. (II.11).*

- *For any vector  $\mathbf{v} \in \mathbb{R}^n$ ,*

$$\begin{aligned} \mathbf{v}^\top \mathbf{L} \mathbf{v} &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} (v_i - v_j)^2, \\ \mathbf{v}^\top \mathbf{L}_{sym} \mathbf{v} &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} \left( \frac{v_i}{\sqrt{D_{ii}}} - \frac{v_j}{\sqrt{D_{jj}}} \right)^2. \end{aligned} \quad (\text{II.12})$$

- *$\mathbf{L}$  is symmetric and positive semi-definite.  $\mathbf{L}_{sym}$  is symmetric and positive semi-definite.*
- *$\mathbf{L}$  has  $n$  non-negative, real-valued eigenvalues. The smallest one is 0 and the associated eigenvector is the constant vector  $\mathbf{1}$ .  $\mathbf{L}_{sym}$  has  $n$  non-negative, real-valued eigenvalues. The smallest one is 0 and the associated eigenvector is the vector  $\mathbf{D}^{1/2} \mathbf{1}$ .*

*The spectral clustering algorithms are based on Laplacian matrices as they have these nice properties.*

---

**Algorithm 5:** Unnormalized Spectral Clustering [18]

---

- Input:** Data matrix  $\mathbf{A} \in \mathbb{R}^{n \times d}$ ; number of clusters  $K$ .  
**Output:** Cluster assignments  $\hat{\tau}$ .
- 1 Construct the similarity graph  $\mathbf{W}$  of  $\mathbf{A}$  as in Definition 13.
  - 2 Compute the unnormalized Laplacian matrix  $\mathbf{L}$  of  $\mathbf{W}$  as in Definition 14.
  - 3 Construct the matrix  $\mathbf{E}$  containing the first  $K$  eigenvectors of  $\mathbf{L}$ .
  - 4 Cluster  $\mathbf{E}$  using  $k$ -means to estimate the block assignments as  $\hat{\tau} \in \{1, \dots, K\}^n$ .
- 

---

**Algorithm 6:** Normalized Spectral Clustering [97]

---

- Input:** Data matrix  $\mathbf{A} \in \mathbb{R}^{n \times d}$ ; number of clusters  $K$ .  
**Output:** Cluster assignments  $\hat{\tau}$ .
- 1 Steps 1 – 2 in Algorithm 5.
  - 2 Compute the normalized Laplacian matrix  $\mathbf{L}_{\text{sym}}$  of  $\mathbf{W}$  as in Definition 14.
  - 3 Construct the matrix  $\mathbf{E}$  containing the first  $K$  eigenvectors of  $\mathbf{L}_{\text{sym}}$  and normalize the rows of  $\mathbf{E}$  to norm 1.
  - 4 Cluster  $\mathbf{E}$  using  $k$ -means to estimate the block assignments as  $\hat{\tau} \in \{1, \dots, K\}^n$ .
-

# Bibliography

- [1] C. E. Priebe, Y. Park, J. T. Vogelstein, J. M. Conroy, V. Lyzinski, M. Tang, A. Athreya, J. Cape, and E. Bridgeford, “On a two-truths phenomenon in spectral graph clustering,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 13, pp. 5995–6000, 2019.
- [2] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” <http://snap.stanford.edu/data>, Jun. 2014.
- [3] B. Rozemberczki, C. Allen, and R. Sarkar, “Multi-scale attributed node embedding,” 2019.
- [4] B. Rozemberczki and R. Sarkar, “Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models,” in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 2020, p. 1325–1334.
- [5] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.
- [6] L. Hao, A. Mele, J. Cape, A. Athreya, C. Mu, and C. E. Priebe, “Latent communities in employment relation and wage distribution: a network approach,” *submitted*, 2020.
- [7] C. R. Shalizi and E. McFowland III, “Estimating causal peer influence in homophilous social networks by inferring latent locations,” *arXiv preprint arXiv:1607.06565*, 2016.

- [8] P. D. Hoff, A. E. Raftery, and M. S. Handcock, “Latent space approaches to social network analysis,” *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1090–1098, 2002.
- [9] M. S. Handcock, A. E. Raftery, and J. M. Tantrum, “Model-based clustering for social networks,” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 170, no. 2, pp. 301–354, 2007.
- [10] S. J. Young and E. R. Scheinerman, “Random dot product graph models for social networks,” in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2007, pp. 138–149.
- [11] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe, “A statistical interpretation of spectral embedding: The generalised random dot product graph,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 84, no. 4, pp. 1446–1473, 2022. [Online]. Available: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12509>
- [12] A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, and Y. Qin, “Statistical inference on random dot product graphs: a survey,” *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 8393–8484, 2017.
- [13] E. Abbe, “Community detection and stochastic block models: recent developments,” *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6446–6531, 2017.
- [14] B. Karrer and M. E. Newman, “Stochastic blockmodels and community structure in networks,” *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.
- [15] E. M. Airoldi, D. Blei, S. Fienberg, and E. Xing, “Mixed membership stochastic blockmodels,” *Advances in neural information processing systems*, vol. 21, 2008.
- [16] A. Mele, L. Hao, J. Cape, and C. E. Priebe, “Spectral inference for large stochastic

- blockmodels with nodal covariates,” *arXiv preprint arXiv:1908.06438*, 2019.
- [17] J. Cape, M. Tang, and C. E. Priebe, “On spectral embedding performance and elucidating network structure in stochastic blockmodel graphs,” *Network Science*, vol. 7, no. 3, pp. 269–291, 2019.
  - [18] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
  - [19] V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe, “Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding,” *Electronic Journal of Statistics*, vol. 8, no. 2, pp. 2905–2922, 2014.
  - [20] V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe, “Community detection and classification in hierarchical stochastic blockmodels,” *IEEE Transactions on Network Science and Engineering*, vol. 4, no. 1, pp. 13–26, 2016.
  - [21] F. McSherry, “Spectral partitioning of random graphs,” in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 2001, pp. 529–537.
  - [22] K. Rohe, S. Chatterjee, and B. Yu, “Spectral clustering and the high-dimensional stochastic blockmodel,” *The Annals of Statistics*, vol. 39, no. 4, pp. 1878–1915, 2011.
  - [23] V. Lyzinski, K. Levin, and C. E. Priebe, “On consistent vertex nomination schemes,” *Journal of Machine Learning Research*, vol. 20, no. 69, pp. 1–39, 2019.
  - [24] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe, “A nonparametric two-sample hypothesis testing problem for random graphs,” *Bernoulli*, vol. 23, no. 3, pp. 1599–1630, 2017.
  - [25] S. Wang, J. Arroyo, J. T. Vogelstein, and C. E. Priebe, “Joint embedding of graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
  - [26] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe, “A consistent

- adjacency spectral embedding for stochastic blockmodel graphs,” *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1119–1128, 2012.
- [27] A. Athreya, C. E. Priebe, M. Tang, V. Lyzinski, D. J. Marchette, and D. L. Sussman, “A limit theorem for scaled eigenvectors of random dot product graphs,” *Sankhya A*, vol. 78, no. 1, pp. 1–18, 2016.
- [28] M. Tang and C. E. Priebe, “Limit theorems for eigenvectors of the normalized laplacian for random graphs,” *The Annals of Statistics*, vol. 46, no. 5, pp. 2360–2415, 2018.
- [29] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.
- [30] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.
- [31] M. Zhu and A. Ghodsi, “Automatic dimensionality selection from the scree plot via the use of profile likelihood,” *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 918–930, 2006.
- [32] J. Neyman and E. S. Pearson, “Ix. on the problem of the most efficient tests of statistical hypotheses,” *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, no. 694-706, pp. 289–337, 1933.
- [33] H. Chernoff, “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *The Annals of Mathematical Statistics*, vol. 23, no. 4, pp. 493–507, 1952.
- [34] —, “Large-sample theory: Parametric case,” *The Annals of Mathematical Statistics*, vol. 27, no. 1, pp. 1–22, 1956.

- [35] C. C. Leang and D. H. Johnson, “On the asymptotics of m-hypothesis bayesian detection,” *IEEE Transactions on Information Theory*, vol. 43, no. 1, pp. 280–282, 1997.
- [36] S. M. Ali and S. D. Silvey, “A general class of coefficients of divergence of one distribution from another,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.
- [37] I. Csiszár, “Information-type measures of difference of probability distributions and indirect observation,” *studia scientiarum Mathematicarum Hungarica*, vol. 2, pp. 229–318, 1967.
- [38] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.
- [39] —, “On a measure of divergence between two multinomial populations,” *Sankhyā: the indian journal of statistics*, pp. 401–406, 1946.
- [40] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [41] I. Csiszár, “I-divergence geometry of probability distributions and minimization problems,” *The annals of probability*, pp. 146–158, 1975.
- [42] F. Liese and I. Vajda, “On divergences and informations in statistics and information theory,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4394–4412, 2006.
- [43] M. E. Newman, “Modularity and community structure in networks,” *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [44] S. Fortunato, “Community detection in graphs,” *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

- [45] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” *Physics reports*, vol. 659, pp. 1–44, 2016.
- [46] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, “Metrics for community analysis: A survey,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, pp. 1–37, 2017.
- [47] M. Coscia, F. Giannotti, and D. Pedreschi, “A classification for community discovery methods in complex networks,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 4, no. 5, pp. 512–546, 2011.
- [48] F. D. Malliaros and M. Vazirgiannis, “Clustering and community detection in directed networks: A survey,” *Physics reports*, vol. 533, no. 4, pp. 95–142, 2013.
- [49] M. E. Newman, “Communities, modules and large-scale structure in networks,” *Nature physics*, vol. 8, no. 1, pp. 25–31, 2012.
- [50] S. Parthasarathy, Y. Ruan, and V. Satuluri, “Community discovery in social networks: Applications, methods and emerging trends,” *Social network data analytics*, pp. 79–113, 2011.
- [51] M. A. Porter, J.-P. Onnela, P. J. Mucha *et al.*, “Communities in networks,” *Notices of the AMS*, vol. 56, no. 9, pp. 1082–1097, 2009.
- [52] S. E. Schaeffer, “Graph clustering,” *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.
- [53] A. Goder and V. Filkov, “Consensus clustering algorithms: Comparison and refinement,” in *2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 2008, pp. 109–117.
- [54] A. Strehl and J. Ghosh, “Cluster ensembles—a knowledge reuse framework for combining multiple partitions,” *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.

- [55] A. Topchy, A. K. Jain, and W. Punch, “Clustering ensembles: Models of consensus and weak partitions,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.
- [56] R. Guimerà and M. Sales-Pardo, “Missing and spurious interactions and the reconstruction of complex networks,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22 073–22 078, 2009.
- [57] M. B. Hastings, “Community detection as an inference problem,” *Physical Review E*, vol. 74, no. 3, p. 035102, 2006.
- [58] M. E. Newman and E. A. Leicht, “Mixture models and exploratory analysis in networks,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 23, pp. 9564–9569, 2007.
- [59] T. P. Peixoto, “Hierarchical block structures and high-resolution model selection in large networks,” *Physical Review X*, vol. 4, no. 1, p. 011047, 2014.
- [60] E. Côme and P. Latouche, “Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood,” *Statistical Modelling*, vol. 15, no. 6, pp. 564–589, 2015.
- [61] J.-J. Daudin, F. Picard, and S. Robin, “A mixture model for random graphs,” *Statistics and computing*, vol. 18, no. 2, pp. 173–183, 2008.
- [62] P. Latouche, E. Birmele, and C. Ambroise, “Variational bayesian inference and complexity control for stochastic block models,” *Statistical Modelling*, vol. 12, no. 1, pp. 93–115, 2012.
- [63] M. E. Newman and G. Reinert, “Estimating the number of communities in a network,” *Physical review letters*, vol. 117, no. 7, p. 078301, 2016.
- [64] M. E. Newman, “Analysis of weighted networks,” *Physical review E*, vol. 70, no. 5, p. 056131, 2004.

- [65] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
- [66] M. J. Barber, “Modularity and community detection in bipartite networks,” *Physical Review E*, vol. 76, no. 6, p. 066102, 2007.
- [67] M. MacMahon and D. Garlaschelli, “Community detection for correlation matrices,” *arXiv preprint arXiv:1311.1924*, 2013.
- [68] V. A. Traag and J. Bruggeman, “Community detection in networks with positive and negative links,” *Physical Review E*, vol. 80, no. 3, p. 036115, 2009.
- [69] P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte, “Uncovering space-independent communities in spatial networks,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 19, pp. 7663–7668, 2011.
- [70] L. G. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney, “Think locally, act locally: Detection of small, medium-sized, and large communities in large networks,” *Physical Review E*, vol. 91, no. 1, p. 012821, 2015.
- [71] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” in *International symposium on computer and information sciences*. Springer, 2005, pp. 284–293.
- [72] M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [73] H. Zhou, “Distance, dissimilarity index, and network community structure,” *Physical review e*, vol. 67, no. 6, p. 061901, 2003.
- [74] —, “Network landscape from a brownian particle’s perspective,” *Physical Review E*, vol. 67, no. 4, p. 041908, 2003.
- [75] H. Zhou and R. Lipowsky, “Network brownian motion: A new method to measure

- vertex-vertex proximity and to identify communities and subcommunities,” in *International conference on computational science*. Springer, 2004, pp. 1062–1069.
- [76] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical review E*, vol. 76, no. 3, p. 036106, 2007.
- [77] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” *Physical review E*, vol. 74, no. 1, p. 016110, 2006.
- [78] P. Ronhovde and Z. Nussinov, “Local resolution-limit-free potts model for community detection,” *Physical Review E*, vol. 81, no. 4, p. 046114, 2010.
- [79] V. A. Traag, P. Van Dooren, and Y. Nesterov, “Narrow scope for resolution-limit-free community detection,” *Physical Review E*, vol. 84, no. 1, p. 016114, 2011.
- [80] A. Arenas, A. Diaz-Guilera, and C. J. Pérez-Vicente, “Synchronization reveals topological scales in complex networks,” *Physical review letters*, vol. 96, no. 11, p. 114102, 2006.
- [81] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, “Detecting complex network modularity by dynamical clustering,” *Physical Review E*, vol. 75, no. 4, p. 045102, 2007.
- [82] D. S. Choi, P. J. Wolfe, and E. M. Airoldi, “Stochastic blockmodels with a growing number of classes,” *Biometrika*, vol. 99, no. 2, pp. 273–284, 2012.
- [83] S. Roy, Y. Atchadé, and G. Michailidis, “Likelihood inference for large scale stochastic blockmodels with covariates based on a divide-and-conquer parallelizable algorithm with communication,” *Journal of Computational and Graphical Statistics*, vol. 28, no. 3, pp. 609–619, 2019.
- [84] T. M. Sweet, “Incorporating covariates into stochastic blockmodels,” *Journal of Educational and Behavioral Statistics*, vol. 40, no. 6, pp. 635–664, 2015.

- [85] N. Binkiewicz, J. T. Vogelstein, and K. Rohe, “Covariate-assisted spectral clustering,” *Biometrika*, vol. 104, no. 2, pp. 361–377, 2017.
- [86] S. Huang and Y. Feng, “Pairwise covariates-adjusted block model for community detection,” *arXiv preprint arXiv:1807.03469*, 2018.
- [87] S. Purohit, S. Choudhury, and L. B. Holder, “Application-specific graph sampling for frequent subgraph mining and community detection,” in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 1000–1005.
- [88] S.-Y. Yun and A. Proutiere, “Community detection via random and adaptive sampling,” in *Conference on learning theory*. PMLR, 2014, pp. 138–175.
- [89] C. Mu, A. Mele, L. Hao, J. Cape, A. Athreya, and C. E. Priebe, “On spectral algorithms for community detection in stochastic blockmodel graphs with vertex covariates,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3373–3384, 2022.
- [90] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.
- [91] G. Kiar, E. W. Bridgeford, W. R. Gray Roncal, V. Chandrashekhar, D. Mhembere, S. Ryman, X.-N. Zuo, D. S. Margulies, R. C. Craddock, C. E. Priebe, R. Jung, V. D. Calhoun, B. Caffo, R. Burns, M. P. Milham, and J. T. Vogelstein, “A high-throughput pipeline identifies robust connectomes but troublesome variability,” *bioRxiv*, p. 188706, 2018.
- [92] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge university press, 2012.
- [93] C. Mu, Y. Park, and C. E. Priebe, “Dynamic network sampling for community detection,” *Applied Network Science*, vol. 8, no. 5, 2023.
- [94] J. Agterberg, Y. Park, J. Larson, C. White, C. E. Priebe, and V. Lyzinski, “Vertex nomination, consistent estimation, and adversarial modification,” *Electronic*

*Journal of Statistics*, vol. 14, no. 2, pp. 3230–3267, 2020.

- [95] I. Gallagher, A. Bertiger, C. Priebe, and P. Rubin-Delanchy, “Spectral clustering in the weighted stochastic block model,” *arXiv preprint arXiv:1910.05534*, 2019.
- [96] A. Athreya, M. Tang, Y. Park, and C. E. Priebe, “On estimation and inference in latent structure random graphs,” *Statistical Science*, vol. accepted for publication, 2020.
- [97] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” *Advances in neural information processing systems*, vol. 14, 2001.

# Vita

Cong Mu received the B.S. degree in Statistics from Sun Yat-Sen University in 2017, the M.S.E. degree in Applied Mathematics and Statistics from Johns Hopkins University in 2019, the M.S.E. degree in Computer Science from Johns Hopkins University in 2022. He enrolled in the Applied Mathematics and Statistics Ph.D. program at Johns Hopkins University in 2019. His research interests include high-dimensional and graph inference, and deep learning applications.