HYPOTHESIS TESTING AND COMMUNITY DETECTION ON NETWORKS WITH MISSINGNESS AND BLOCK STRUCTURE

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Guilherme Gomes

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

November 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jennifer Neville, Co-Chair

Departments of Computer Science and Statistics

Dr. Vinayak Rao, Co-Chair

Department of Statistics and Computer Science

Dr. Anindya Bhadra, Member

Department of Statistics

Dr. Bruno Ribeiro, Member

Departments of Computer Science

Approved by:

Dr. Hao Zhang

Department Head of Statistics

All knowledge is, in final analysis, history.
All sciences are, in the abstract, mathematics.
All judgements are, in their rationale, statistics.

Calyampudi Radhakrishna Rao

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

SYMBOLS

$\boldsymbol{M}_i$	i-th row of matrix $\boldsymbol{M}$
$\boldsymbol{M}[a, b]$	cell in the a-th row b-th column of matrix $\boldsymbol{M}$
$\mathbb{I}_{\text{condition}}$	Indicator function
$\underset{n\to\infty}{\overset{a.s.}{\to}}$	Converge almost surely
$\underset{n\to\infty}{\overset{P}{\to}}$	Convergence in probability
$\underset{n\to\infty}{\overset{D}{\to}}$	Convergence in distribution
V_n	Set of nodes in graph n
V	Overall set of nodes, usually $\bigcup_{n=1}^{N} V_n$
$\mathcal{G}$	Graph defined by the sets of nodes V and edges E
$\mathcal{G}_n$	Graph n
$\mathbf{A}$	$\lvert V\rvert \times \lvert V\rvert$ adjacency matrix
$\mathbf{A}_n$	$\lvert V_n\rvert \times \lvert V_n\rvert$ adjacency matrix of graph n
$\mathbf{X}_i$	latent (unobserved) features of node i
$\mathbf{Y}_i$	observed features of node i
H	number of cluster components
$\boldsymbol{\beta}$	$1 \times H$ vector of mixing probabilities
$\boldsymbol{\pi}_n$	user-specific mixing probability associate with user n (Chapter 2)
K	number of communities
$\boldsymbol{\Theta}$	$K \times K$ connectivity matrix (Chapter 3)
$\boldsymbol{\mu}$	$K \times K$ connectivity matrix (Chapter 4)
G_{nk}	Nodes in graph n and in community k
$G_{\cdot k}$	$\bigcup_{n=1}^{N} G_{nk}$
$\boldsymbol{\pi}_n$	distribution of nodes over cluster for graph n
$\mathbf{P}_n$	$\lvert V_n\rvert \times \lvert V_n\rvert$ edge probabilities matrix of graph n
Δ_n	$(\mathbf{X}_n'\mathbf{X}_n)^{1/2}$

$\mathbf{U}_n$	$\lvert V_n\rvert \times K$ matrix of eigenvectors of $\boldsymbol{P}_n$
$\lVert \cdot \rVert$	Euclidean (vector) and spectral (matrix) norm
$\lVert \boldsymbol{M} \rVert_F$	Frobenius norm of matrix $\boldsymbol{M}$
$\lVert \boldsymbol{M} \rVert_0$	The l_0-norm of matrix $\boldsymbol{M}$ (counts the nonzero elements)

ABSTRACT

Gomes, Guilherme PhD, Purdue University, November 2019. Hypothesis testing and community detection on networks with missingness and block structure. Major Professor: Jennifer Neville.

Statistical analysis of networks has grown rapidly over the last few years with increasing number of possible applications. Graph-valued data carries additional information of dependencies which opens the possibility of modeling highly complex types of problems in vast number of fields such as biology (e.g. brain networks , fungi networks, genes co-expression [1–3]), chemistry (e.g. molecules fingerprints [4, 5]), psychology (e.g. social networks [6]) and many others (e.g. citation networks, word co-occurrences, financial systems, anomaly detection [7–10]). While the inclusion of graph structure in the analysis can further help inference, simple statistical tasks in a network is very complex. For instance, the assumption of exchangeability of the nodes or the edges is quite strong, and it brings issues such as sparsity, size bias and poor characterization of the generative process of the data [11]. Solutions to these issues include constraint and assumptions to the data generation process [12, 13]. In this work, we approach this problem by assuming graphs are globally sparse, but locally dense which allows exchangeability assumption to hold in local regions of the graph. We consider problems in two types of locality structure: block structure also framed as multiple graphs (or population of networks) and unstructured which can be seen as missing data. For the former, we developed a powerful hypothesis testing framework for weighted aligned graphs; and spectral clustering method for community detection on population of non-aligned networks. For the latter, we derive an efficient spectral clustering approach to learn the parameters of the Zero inflated stochastic blockmodel. Overall, we found that incorporating multiple local dense structures leads to a more precise and powerful local and global inference. This result indicates that this general modeling scheme allows for exchangeability assumption on the edges to hold while generating more

realistic graphs. We give theoretical conditions for our proposed frameworks, and we evaluate them on synthetic and real-world datasets and show our models are able to outperform the baselines on each setting.

1. INTRODUCTION

Interest in the statistical analysis of network[1] data has grown rapidly over the last few years with increasing number of possible applications. Graph-valued data carries additional information of dependencies which opens the possibility of modeling highly complex types of problems in vast number of fields such as biology (e.g. brain networks , fungi networks, genes co-expression [1–3]), chemistry (e.g. molecules fingerprints [4, 5]), psychology (e.g. social networks [6]) and many others (e.g. citation networks, word co-occurrences, financial systems, anomaly detection [7–10]). While the inclusion of graph structure in the analysis can further help inference, simple statistical tasks in a network is very complex. For instance, the assumption of exchangeability of the nodes or the edges is quite strong, and it brings issues such as sparsity, size bias and poor characterization of the generative process of the data [11]. Solutions to these issues include constraint and assumptions to the data generation process [12, 13].

Prior to describe statistical models for networks, it is important to define statistical modeling in general, and then delineate how statistical models are applied to network data. Given a sample space ω, we observe the sequence $(X_1, X_2, ..., X_m) = \boldsymbol{X}(\omega) \in \mathbb{R}^n$. Then, we assume $\boldsymbol{X} \sim P_\theta$ and P_θ belongs to the family of distributions $\mathcal{P}$ where the family $\mathcal{P} = (P_\theta)_{\theta \in \Theta}$ is the statistical model. Formally, this model is defined as $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n), P_\theta)$, where $\mathcal{B}(\mathbb{R}^n)$ is the Borel σ-algebra which consists of all possible subsets of events in $\mathbb{R}^n$ [14]. The objective is to understand and predict new data values by retrieving θ that best fits the observed values. This process is called inference, i.e. use the observed data to infer from the population that this data points belongs to. There are two main perspectives to proceed with inference: classical (or frequentist) and Bayesian.

The classical approach is to choose a probability distribution $P_\theta \in \mathcal{P}$ and define a parametric space Θ where $P_\theta : \Theta \to \mathcal{P}$ and "search" the best estimator for $\theta \in \Theta$. The

[1] We use the terms *networks* and *graphs* interchangeably throughout this work.

most used and most efficient (lowest variance) is the maximum likelihood estimator (MLE). The Bayesian approach relies on *de Finetti*'s theorem which assumes the sequence of data points can be generated by first generating a random parameter value $\theta \sim \Theta$ and then sampling $\boldsymbol{X} \overset{iid}{\sim} P_\theta$. Using any MCMC scheme, one can sample from the posterior distribution $\theta|\boldsymbol{X} \sim \Theta$, and estimate θ by computing the sample mean. In order to have a reliable statistical inference, it is essential for both cases that 1) there is a reasonable enough assumption of an underlying generating process of the data and 2) the sequence $\boldsymbol{X}(\omega)$ is exchangeable. Exchangeability is defined as $(X_1, ..., X_m) \overset{D}{=} (X_{\pi(1)}, ..., X_{\pi(m)})$ where π is any permutation, i.e. the sequence has the same distribution regardless label ordering of each data point.

1.1 Statistical network analysis

In Networks, data points are not independent of each other and they have a structure to indicate their dependencies. Define a graph $\mathcal{G}$ by the pair $\mathcal{G} = (V, E)$ where V is a set of nodes (or vertices) and E a set of edges. Now, let $\theta \in \Theta$ be a parametrization family of data generating processes $\mathcal{M} \subseteq \mathcal{P}$ for a chosen unit $\mathcal{U}$ (vertex or edge), then the parametrized model is defined by the map $P_\theta : \Theta \to \mathcal{P}(\mathcal{G}_\mathcal{U})$ where $\mathcal{G}_\mathcal{U} \sim P_\theta$ is the set with all possible graphs using $\mathcal{U}$ as sample units. Also, we need to define a sampling mechanism $\Sigma_n : \mathcal{G}_\mathcal{U} \to \mathcal{G}_{[n]}$, i.e. the mechanism Σ_n sample items from $\mathcal{G}_\mathcal{U}$ where n is the target sample size of the unit $\mathcal{U}$. In summary, a network model is defined the same way as non structured data by the triplet $(\Theta, \mathcal{G}_\mathcal{U}, \Sigma_n)$, with an extra level of complexity given simple sampling mechanisms may lead to complete logical disconnection with the target generating process [13], i.e. $\mathcal{G}_{[n]} \notin \mathcal{M}$. In fact, [11] showed that any exchangeable edge generative model is fundamentally wrong since it leads to dense graphs as number of nodes increases, and real-world large graphs are known to be sparse.

In this work, we assume the sampling process Σ_n is also subject to a local mechanism $\mathcal{L}_l$ where $l \in E$. Precisely, consider a graph represented by its adjacency $\boldsymbol{A} \in \mathbb{R}^{|V|\times|V|}$

where V is the set of nodes. Write a_{ij} as the interaction between nodes $i \in V$ and $j \in V$. The generation process of a_{ij} is given by

$$a_{ij} = \mathcal{L}_{ij} \times \eta_{ij} \tag{1.1}$$

where the local term $\mathcal{L}_{ij} \in \{0, 1\}$ and the connectivity strength $\eta_{ij} \in \mathbb{R}$ are random variables defined as

$$\mathcal{L}_{ij} | \{\boldsymbol{W}_a\}_{a\in\{i,j\}}, \psi_1 \sim \text{Bern}(g(\{\boldsymbol{W}_a\}_{a\in\{i,j\}}, \psi_1)) \tag{1.2}$$

$$\eta_{ij} | \{\boldsymbol{W}_a\}_{a\in\{i,j\}}, \psi_2 \sim \text{Dist}(f(\{\boldsymbol{W}_a\}_{a\in\{i,j\}}, \psi_2)) \tag{1.3}$$

where $\{\boldsymbol{W}_a\}_{a\in\{i,j\}}$ represents node-level features (e.g. observed or latent), ψ_1 and ψ_2 represent graph-level parameters, Dist is some distribution (e.g. Bernoulli, Binomial, Poisson, Normal, Zipf), and $g(\cdot)$ and $f(\cdot)$ represents general functions. This general modeling scheme allows for exchangeability assumption on the edges to hold while generating more realistic graphs in terms of sparsity level. Our fundamental assumption is that graphs are locally dense, but globally sparse. For massive graphs, local parts of the network (which are allowed to be dense) are also very large which might indicates an escape of global sparsity assumption. Nevertheless, this is not an important concern in our applications since local regions are assumed to be small. At this point, the term *local* is deliberately ill-defined, and for each part of this work it will have a different characterization.

1.2 Contributions and contents

The process described in Eq. (1.1) can be seen as a missing data problem where the observed graph is subject to a general 'missingness' mechanism obfuscating potential interaction between nodes. In summary, the local term $\mathcal{L}_{ij}$ acts as a mask on the connectivity η_{ij} where nodes i and j are only able to connect when $\mathcal{L}_{ij} = 1$. More importantly, both $\mathcal{L}_{ij}$ and η_{ij} depends on node-level $\{\boldsymbol{W}_a\}_{a\in\{i,j\}}$ and global ψs parameters. In our applications, $\{\boldsymbol{W}_a\}_{a\in\{i,j\}}$ encompass observed, $\boldsymbol{Y}_a$, and latent, $\boldsymbol{X}_a$, features, ψ_1 represents global

parameters governing the missingness, $\boldsymbol{\phi}$, and ψ_2 the connectivity, $\boldsymbol{\Theta}$ and $\boldsymbol{\beta}$. Precisely, we have

$$\mathcal{L}_{ij}|\{\boldsymbol{Y}_a\}_{a\in\{i,j\}}, \{\boldsymbol{X}_a\}_{a\in\{i,j\}}, \boldsymbol{\phi} \sim \text{Bern}\left(g_{\boldsymbol{\phi}}\left(\{\boldsymbol{Y}_a\}_{a\in\{i,j\}}, \{\boldsymbol{X}_a\}_{a\in\{i,j\}}\right)\right) \tag{1.4}$$

$$\eta_{ij}|\{\boldsymbol{Y}_a\}_{a\in\{i,j\}}, \{\boldsymbol{X}_a\}_{a\in\{i,j\}}, \boldsymbol{\Theta}, \boldsymbol{\beta} \sim \sum_{h=1}^{H} \beta^{(h)} \text{Dist}\left(f\left(\boldsymbol{X}_{\boldsymbol{Y}_i}^{(h)} \boldsymbol{\Theta}^{(h)} \boldsymbol{X}_{\boldsymbol{Y}_j}^{T(h)}\right)\right). \tag{1.5}$$

Notice that, in our applications, the connectivity η_{ij} is distributed as mixture of H components of a given distribution (e.g. Bernoulli, Binomial, Poisson, Normal, Zipf) where $\boldsymbol{\beta} = [\beta^{(1)}, ..., \beta^{(H)}]$ is a vector of mixing probabilities, and $f(\cdot)$ is essentially a link function that maps the term $\boldsymbol{X}_{\boldsymbol{Y}_i} \boldsymbol{\Theta} \boldsymbol{X}_{\boldsymbol{Y}_j}^T$ to a suitable support.

We divide our applications based on whether the observed nodes features $\{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$ are independent of the local term $\mathcal{L}_{ij}$. More specifically, we consider two types of structures on our applications: 1) $\mathcal{L}$ depends on the observed $\boldsymbol{Y}$ that defines a block structure, and we focus on multiple (sub)graphs problems on Chapters 2 and 3 ; and 2) a unstructured setting where $\mathcal{L}$ is independent of node features which defines a zero inflated weights on Chapter 4. We can also break our work our work from the statistical task perspective which is based on whether the connectivity η_{ij} depends (or not) on observed nodes features $\{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$. For $\eta_{ij} \not\perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$, we work on hypothesis testing problems on Chapter 2, and for $\eta_{ij} \perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$ we worked on community detection on Chapters 3 and 4. Table 1.1 summarizes our applications by data structure and statistical tasks.

Table 1.1.: Types of applications considered divided by data structure and statistical tasks

	Block structure $\mathcal{L}_{ij} \not\perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$	**No structure** $\mathcal{L}_{ij} \perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$	**Statistical task**
$\eta_{ij} \not\perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$	Chapter 2	—	Hypothesis testing
$\eta_{ij} \perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$	Chapter 3	Chapter 4	Community detection

1.2.1 Block structure (multiple graphs)

In this case, each node belongs to one block, and each block is consider a smaller graph. Precisely, define the observed features as $\boldsymbol{Y}_a = [Z_a, l_a]$ where $Z_a = 1, ..., N$ indicates which graph (block) node a belongs, and $l_a \in L$ is the label of node a and L is the set of labels. This is a missing-not-at-random setting where $h(\cdot)$ is a function of the observed Z_a. Here, nodes i and j are *local* to each other $\iff$ i and j belong to the same graph, i.e.

$$\begin{aligned} \mathcal{L}_{ij} | \{\boldsymbol{Y}_a\}_{a \in \{i,j\}} &\sim \text{Bern}\left(g\left(Z_i, Z_j\right)\right) \\ &= \mathbb{I}_{Z_i = Z_j} \end{aligned} \tag{1.6}$$

In general, the work on network analysis is on a single graph, i.e. statistical inference is on node or edge level (e.g. link prediction [15, 16], collaborative filtering [17–19], community detection [20–22]). The current work in multiple graphs can be divided in two main areas: graphs evolving over time [23–26]; graphs as independent observations [2, 3, 8, 9]. Our work focus on the later type. Let graph n be represented by its adjacency matrix $\boldsymbol{A}_n \in \mathbb{R}^{|V_n| \times |V_n|}$where $n = 1, ..., N$. Also, write V_n as the set of nodes in graph n, and $V = \bigcup_{n=1}^{N} V_n$ where $V_n \bigcap V_{n'} = \emptyset$ for any $n \neq n'$. We further divide the multiple graph applications based on the structure of the labels $l_a \in L$.

Aligned graphs

We call *aligned graphs* settings when $f_{\boldsymbol{\Theta}}(\cdot) \not\perp\!\!\!\perp l_a \implies \eta_{ij} \not\perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a \in \{i,j\}}$. In this case, l_a defines an one-to-one mapping of labels across graphs and labels are *unique* within graphs. These can also be seen as a single graph with multiple types of edges (e.g., multi-view graphs [27–33]) or edges varying over time (e.g., dynamic networks [34–38]).

On Chapter 2, we examine inference issues on weighted aligned graphs, more specifically hypothesis testing on population of the networks. We propose a hypothesis testing framework for weighted graphs using a Bayesian approach. Here, $\boldsymbol{Y}_a = [Z_a, l_a, y_{Z_a}]$ where Z_a is the graph membership of node a, $l_a \in L$ is the label (e.g. a brain region in a brain

network, a word in a word co-occurrence graph), and y_{Z_a} indicates graph Z_a population membership. Also, η is distributed as a non binary random variables (e.g., Poisson, Binomial) associated with the link function $f(\cdot)$; the global connectivity Θ is the identity in this case; and the local mechanism $\mathcal{L}$ is defined in Eq. (1.6). For a population of N graphs where each graph has $|L|$ nodes, the data generating process is given by

$$\eta_{ij}|\{\boldsymbol{Y}_a\}_{a\in\{i,j\}}, \{\boldsymbol{X}_a\}_{a\in\{i,j\}}, \boldsymbol{\beta} \sim \sum_{h=1}^{H} \beta_{y_{Z_i}}^{(h)} \text{Poisson}\left(\exp\left(\boldsymbol{X}_{l_i}^{(h)} \boldsymbol{X}_{l_j}^{(h)T}\right)\right) \tag{1.7}$$

where $\boldsymbol{X}_a^{(h)}$ is a $1 \times R$ low dimension latent feature representation of node a in component h and $R \ll |L|$. The hypothesis testing method was build based on the distribution of the mixing probabilities across populations. Moreover, we propose an admixture model to deal with conditional hypothesis testing when a group information is also available, in other words we propose an hierarchical scheme to deal with conditional hypothesis testing. We show how this proposed methods performed compared to the alternatives in synthetic data and real world applications.

Non-aligned graphs

Here, $l_a = a$, therefore $f_{\boldsymbol{\Theta}}(\cdot) \perp\!\!\!\perp l_a \implies \eta_{ij} \perp\!\!\!\perp \{\boldsymbol{Y}_a\}_{a\in\{i,j\}}$. Unlike aligned graphs, *non-aligned* graphs have a fundamental issue – there is no known mapping of the nodes across the graphs. In general, the label has no specific meaning (e.g. ID of a person in a given village), and even if it does have a meaning there is no assumption of being unique within graphs (e.g. atom in a molecule). Typical examples include molecules, road networks, village networks.

On Chapter 3, we focus on the problem of community detection across a set of non-aligned graphs of varying size, i.e. $V_n \neq V_{n'}$ and $|V_n| \neq |V_{n'}|$. Here, $\boldsymbol{Y}_a = [Z_a, l_a]$ where Z_a is graph membership of node a, $l_a = a$; η is the Bernoulli distribution and $f(\cdot)$ is the identity; and we assume there is only one mixture component (i.e. $H = 1 \implies \boldsymbol{\beta} = 1$). In this case, the connectivity is governed by a well known community based model called

stochastic blockmodel [39, 40]. Precisely, $\boldsymbol{\Theta}$ is a $K \times K$ matrix of connectivities where K is the number of communities and the cell $\boldsymbol{\Theta}[k, l]$ represents the connectivity between communities k and l. The data generating process is given by

$$\eta_{ij} | \{\boldsymbol{X}_a\}_{a \in \{i,j\}}, \boldsymbol{\Theta} \sim \text{Bern}\left(\boldsymbol{X}_i \boldsymbol{\Theta} \boldsymbol{X}_j^T\right) \tag{1.8}$$

where $\boldsymbol{X}_a$ is a $1 \times K$ latent one hot vector where $\boldsymbol{X}_a[k] = 1$ if node a belongs to community k. We show that using pooled information across graphs and jointly estimating local and global structures is crucial to gain a precise understanding of heterogeneous networks. Our model outperforms current two-step approaches. Our algorithm is fast, efficient, robust, and has very few probabilistic assumptions. Next, we describe an unstructured $\mathcal{L}$ mechanism.

1.2.2 No structure (zero inflated)

In this case, the local mechanism is independent of any node feature, i.e. $\mathcal{L}_{ij} \perp\!\!\!\perp \left(\{\boldsymbol{Y}_a\}_{a \in \{i,j\}}, \{\boldsymbol{X}_a\}_{a \in \{i,j\}}\right)$. This can be seen as a missing-at-random problem where the observed graph is subject to a general 'missingness' mechanism which is independent of any node feature, i.e.

$$\mathcal{L}_{ij} | \boldsymbol{\phi} \sim \text{Bern}(\phi_{ij}) \tag{1.9}$$

where ϕ_{ij} is the probability that nodes i and j are local to each other. This is a sparsity perspective that has recent received attention specially in community detection tasks [41–43]. In these works, there is an additional structure assumption on ϕ_{ij} for all pair of nodes i and j (e.g. $\phi_{ij} = \phi$, $\phi_{ij} = \nu_i \nu_j$). On Chapter 4, we relax these assumptions by allowing $\phi_{ij} \neq \phi_{i'j'}$ for any $i, j, i', j' \in V$. Overall, we consider the following generative process

$$\eta_{ij} | \{\boldsymbol{X}_a\}_{a \in \{i,j\}}, \boldsymbol{\Theta} \sim \text{Dist}\left(\boldsymbol{X}_i \boldsymbol{\Theta} \boldsymbol{X}_j^T\right) \tag{1.10}$$

Notice that Eq. (1.10) is very similar to Eq. (1.8) where the main differences are: the local term $\mathcal{L}_{ij}$ (defined in Eq. (1.9)), and the distribution of the connectivity η_{ij} which, in this case, can be any distribution. We propose two inference schemes based on spectral

clustering: self-similar and ego-nets. The former focus on settings where $\mathcal{L}$ is observed and the later is used when there is no knowledge about $\mathcal{L}$.

1.2.3 Conclusion and future work

Finally on Chapter 5, we summarize and conclude our contribution, and discuss directions on future work.

2. ALIGNED GRAPHS

2.1 Introduction

Some important applications on multiple graph domain are only possible when graphs are aligned. Precisely, aligned graphs are defined as a set of graphs where there is a known one-to-one mapping of labels across graphs and labels are unique within graphs. We consider the problem of graph-based hypothesis testing, which tests whether two sets of graph-valued observation samples are drawn from the same distribution based on edge probabilities. This is a topic of growing interest [44–47]; however, there are only a few studies where the observations are *weighted graphs*. In this work, we address this gap, considering the problem of *hypothesis-testing for replicated weighted graph-valued data*. This is a challenging problem, since the average and atypical behavior of a sample of networks is difficult to characterize.

Figure 2.1 illustrates two example domains with populations of weighted graphs. The top row illustrates the word co-occurrence networks of two Twitter users in Brazil, one that is *pro-government* and one that is *anti-government*. Here the entity corresponds to a user on social media, the nodes in the graph are vocabulary words, and the edges weights reflect co-occurrences of words in the posts of the users. The bottom row illustrates brain connectivity networks of two individuals, one *female* and one *male*. Here the entity corresponds to an individual, the nodes in the graph are brain regions, and the edges weights represent functional connectivity strength as measured by functional magnetic resonance imaging (fMRI). In both cases, we would like to investigate how populations and entities differ. For example, whether brain activity differs with respect to sex or whether word usage differs with respect to political views.

There has been recent work on graph-based hypothesis testing (see [48] for a good survey). However, much of the work has focused on one-sample tests comparing a single

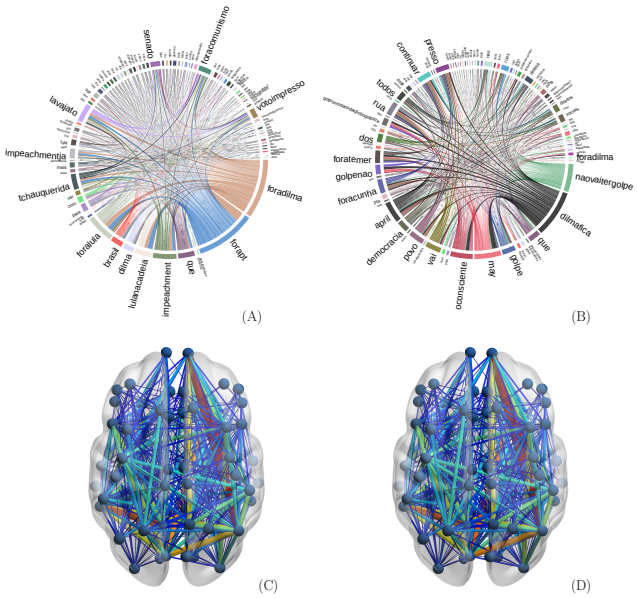

Figure 2.1.: Top row: word connectivity networks for two Brazilian Twitter users, (A) from the pro-government side and (B) from the anti-government side. Bottom row: brain connectivity network for two individuals, (C) female and (D) male.

graph to a null model (e.g., [49]). Work focusing on *populations* of graphs has received considerably less attention and falls into one of two categories: that of [3], which introduces a geometric characterization of the network using the so-called Fréchet mean, and that of [2], who proposed a Bayesian latent-variable model for unweighted graphs. We focus on the latter, which allows us to bring the powerful machinery of probabilistic hierarchical modeling to the table, allowing noisiness and missingness, and providing interpretable confidence scores. Unfortunately, existing work along this second direction is limited to modeling binary graphs, so that in practice, a *threshold* must be used to transform counts or continuous weights to 0/1 values. Such a thresholding operation discards valuable information about the strength of the edge-weights, and can also exhibit sensitivity towards the choice of threshold. Too small a threshold can result in a graph that is too dense, and too large, too sparse. Often, there does not even exist a single appropriate threshold across the entire graph.

In this chapter, we address the issues of previous work and develop a hypothesis testing framework that facilitates testing over graphs populations with edge-weights, which can follow any parametric distribution. Specifically, we propose a Bayesian hypothesis testing framework that uses a mixture of latent space models for weighted networks to test for population-differences. Our framework is capable of population-level, entity-specific as well as edge-specific hypothesis testing. We consider testing in two broad scenarios:

1. When all *observations* from the same population follow the same distribution, we can ask: Are the population distributions identical?

2. When all *entities* from the same population follow the same distribution, we can ask: Are the population distributions identical? Now, every entity has an associated set of graph-valued observations which are identically distributed, but are not exchangeable across entities.

Observe the first case is equivalent to the second when each entity has only one associated graph; however the latter allows heterogeneity among entities in the same population. For instance an entity in a population that is politically *conservative* might frame an issue they discuss from an economic perspective, while another entity in a population that is politically *liberal* might focus on the social aspects of the same issue. [23] proposed a strongly parametric time-varying framework to handle this important situation, our approach is significantly more flexible.

We apply our testing framework to problems from the types of domains summarized in Figure 2.1. First, we look at word co-occurrence network data from Twitter (on the political crisis in Brazil), as well as Instagram (on side effects of Adderall and Ritalin usage for Attention Deficit Hyperactivity Disorder [50]). In both datasets, we investigate how populations and entities differ based on the way they communicate—specially in the manner in which the usage of pairs of key words differs between groups. Standard methods such as unigram mixture models, latent Dirichlet allocation (LDA) [51] or N-gram language models [52], which are based just on word-frequency, do not capture the kind of contextual information we are interested in. While these methods can identify words that

'belong' to different groups, in our scenario there is a strong overlap in key words across groups, and such models will fail to differentiate between groups which share common themes and vocabularies. Second, we used functional magnetic resonance imaging (fMRI) data, to investigate how brain activity differs across populations like sex, age and personality traits like extroversion, conscientiousness and creativity. In both tasks, we show that graph-structure as well as graph weights are crucial to performance, and that we outperform baselines like latent Dirichlet allocation (LDA) [51], N-gram language models [52], as well as thresholding methods like [2].

Contributions: Our contribution is a multi-level statistical hypothesis testing framework for populations of weighted networks, concerning both the overall graph distributions, as well as two types of local hypotheses: entity-specific and edge-specific. The latter are important since a population might have networks, or a network edges, that are statistically different, and that might escape detection by a global test. Our hierarchical Bayesian mixed membership model allows statistical information to be shared across groups, increasing accuracy of hypothesis tests without loss of statistical power. This allows practitioners to evaluate anomalies in a principled manner, using statistical significance. Notably, our framework is more robust than previous methods developed for binary graphs, which require thresholding of weighted data before application.

2.2 The model

We are given a set $\mathbf{A}$ of undirected graphs, with graph A_{nt} belonging to entity n at index t (refered to as 'time'). Here $n \in [1, ..., N]$ and $t \in [1, ..., T_n]$, with $A_{nt}[i, j]$ giving the link strength between vertices i and j of entity n at time t ($i, j \in [1, ..., V]$). We also observe population information $y_n \in [1, ..., G]$ for each entity. For instance, each network might represent word co-occurrences in a user's social media messages over some time period, while the population might indicate whether the user's political leanings are 'Liberal' or 'Conservative'.

Underlying our testing framework is a probabilistic model which we now outline. We assume each observation A_{nt} comes from one of H clusters or mixture components, with cluster h having parameter $\boldsymbol{\theta}^{(h)}$. Each cluster has a distribution over graphs which we write as $F(\boldsymbol{\theta}^{(h)})$ (we specify $\boldsymbol{\theta}^{(h)}$ and F in the next paragraph). Clusters and cluster parameters are shared across populations, however each population y has its own Dirichlet-distributed probability over clusters, $\boldsymbol{\beta}_y$. At a high-level ours is a Bayesian hypothesis-testing approach which tests whether the $\boldsymbol{\beta}_y$'s are identical across populations. For the case of two populations, we place equal *a priori* probability on the null hypothesis $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$ and the alternative $H_1 : \boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$. Using the machinery of Bayesian inference, we evaluate the posterior probabilities of the two hypotheses given observations, and reject the null if its probability $P(H_0|-)$ is less than some specified threshold (e.g., 0.05 or 0.1).

We now describe the cluster-specific distribution over graphs, $F(\boldsymbol{\theta}^{(h)})$. For cluster h, $\boldsymbol{\theta}^{(h)}$ is a $V \times V$ matrix, whose (ij)th element parametrizes the probability of the weight on the edge between nodes i and j. In our applications, we looked at count-valued edges, and so assumed $F\left(\boldsymbol{\theta}^{(h)}\right)$ to be Binomial or Poisson distributed with parameter $\boldsymbol{\theta}^{(h)}[i,j]$ on edge (i,j). We define $\boldsymbol{\theta}^{(h)} = f\left(\boldsymbol{S}^{(h)}\right)$ where $f(\cdot)$ is some link function (e.g. the logistic or exponential function to ensure nonnegativity), and constrain $\boldsymbol{S}$ using a low-rank factorization scheme $\boldsymbol{S}^{(h)} = \mathbf{X}^{(h)}\mathbf{X}^{(h)T}$. Here $\mathbf{X}^{(h)} \in \mathcal{R}^{|V| \times R}$ and $R \ll |V|$, so that $\mathbf{X}_v^{(h)}$ gives the location of node v in some low-dimension space, and $\boldsymbol{S}^{(h)}$ is the proximity of all nodes. The number of parameters thus grows linearly, rather than quadratically with the number of vertices. In equations, we expand the upper plates in Figure 2.2(a) and (b), to get

$$\boldsymbol{\theta}^{(h)} = f(\boldsymbol{X}^{(h)}\boldsymbol{X}^{(h)T}), \quad \mathbf{X}_v^{(h)} \sim N_R\left(\mathbf{0}, \mathbb{I}\right), v = 1 \ldots, V \tag{2.1}$$

Each population y has a distribution over clusters $\boldsymbol{\beta}_y$. With prior probability half, the null hypothesis is true (we indicate this with the variable $\mathcal{T}$), in which case all populations have the same distribution $\boldsymbol{\beta}$. Otherwise, each population g has its own distribution, $\boldsymbol{\beta}_g$. Thus,

$$\begin{aligned} &\mathcal{T} \sim \text{Bern}(1/2) \\ &\text{If } \mathcal{T} = 0: \quad \boldsymbol{\beta}_1 = \ldots, = \boldsymbol{\beta}_G \sim \text{Dir}(\alpha, ..., \alpha) \\ &\qquad \text{Else:} \quad \boldsymbol{\beta}_g \overset{iid}{\sim} \text{Dir}(\alpha, ..., \alpha) \text{ for } g = 1 \ldots G. \end{aligned} \tag{2.2}$$

Now, consider the case where each entity has only a single associated graph. Then the nth entity (belonging to population Y_n) has a graph A_n distributed as

$$C_n | Y_n \sim \boldsymbol{\beta}_{Y_n} \quad \mathbf{A}_n | C_n \sim F\left(\boldsymbol{\theta}^{(C_n)}\right) \tag{2.3}$$

Here C_n refers to the latent variable that identifies the cluster membership of entity n, which depends on the population entity n is drawn from.

For the case where we have multiple network observations per entity, we add a layer to this hierarchical model. Now, each entity n has their own distribution over clusters $\boldsymbol{\pi}_n$ centered around the population distribution:

$$\boldsymbol{\pi}_n \sim Dir(\beta_{Y_n}). \tag{2.4}$$

The graph t of this entity is independently distributed as

$$C_{nt} | Y_n \sim \boldsymbol{\pi}_n, \quad \mathbf{A}_{nt} | C_{nt} \sim F\left(\boldsymbol{\theta}^{(C_{nt})}\right) \tag{2.5}$$

Figure (2.2) summarizes our generative process for both cases.

2.2.1 Model Inference

We are given a set of network observations $\boldsymbol{A}$, each written as A_{nt} where n indexes entities and t, 'time'. For each A_{nt}, we are also given a population assignment $Y_n \in \{1, 2\}$. Since we observe the population memberships $\boldsymbol{Y}$ and the networks $\boldsymbol{A}$, the inferential task is to learn C_{nt}, $\boldsymbol{\pi}_n$, $\boldsymbol{\beta}_y$ and $\boldsymbol{\theta}^{(h)}$. In the next section, we will use these variables as statistics in our hypothesis tests. For notational convenience, we will refer to a link between an arbitrary pair of nodes i and j with l, so that we can write $A_n[i, j]$ as $A_n[l]$. We also represent the weighted matrix with its vectorized lower triangular component $\mathcal{L}(\mathbf{A}_n) = (\mathcal{L}(A_{nt})_1, ..., \mathcal{L}(A_{nt})_{V(V-1)/2})$. For the general model specified above, we carry out posterior inference via a Gibbs sampler, whose individual updates we outline next:

1. *Sample the cluster indicator for each graph.*

   This comes from the multinomial:

$$P(C_{nt} = h|-) = \frac{\pi_{n_h}\beta_h \prod_l P\left(\mathcal{L}(A_{nt})_l = a_l|\theta_l^{(h)}\right)}{\sum_{m=1}^{H} \pi_{n_m}\beta_m \prod_l P\left(\mathcal{L}(A_{nt})_l = a_l|\theta_l^{(m)}\right)}$$

   where $l \in [1, ..., V(V-1)/2]$ and $a_l|\theta_l^{(h)} \sim \left[F(\boldsymbol{\theta}^{(h)})\right]_l$.

2. *Sample the mixing probabilities for each entity n.*

   With $\boldsymbol{m}_n$ a vector of cluster assignment counts of graphs of entity n

$$\boldsymbol{\pi}_n \sim \text{Dir}(\boldsymbol{\beta}_{Y_n} + \boldsymbol{m}_n)$$

3. *Sample the locations of the nodes for each cluster.*

   With a Gaussian prior over the locations $\boldsymbol{X}_n$, and the weight-distribution parametrized after transforming through a link function f, this is a straightforward exercise in sampling from the posterior of a Gaussian with a nonlinear link function. Standard techniques exist to do this [53, 54], though we followed a recent idea involving the Polya-Gamma data-augmentation scheme [55].

4. *Sample the testing indicator* $\mathcal{T} \sim$ *Bern*$(P(\mathcal{H}_1)|-)$. Since $\mathcal{T}$ is the test-statistic central to our methodology, we discuss this in a bit more detail in the next section. The update rule is given by Equation 2.7.

5. *Sample the mixing probabilities for each population* y.
   If $\mathcal{T} = 0$ then for all y, $\boldsymbol{\beta}_y = \boldsymbol{\beta} \sim \text{Dir}(\alpha + n_1, ..., \alpha + n_H)$ where n_h is the number of graphs in cluster h. If $\mathcal{T} = 1$ then $\boldsymbol{\beta}_y \sim \text{Dir}(\alpha + n_{1y}, ..., \alpha + n_{Hy})$ for each y, where n_{iy} counts the number of graphs from population y in cluster i.

2.3 Weighted-network comparison tests

For simplicity, we focus on the case where we have only two populations ($G = 2$). Under our formulation, the problem of hypothesis testing amounts to testing whether the population-level cluster assignment probabilities $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are significantly different under the posterior. We elaborate on this below.

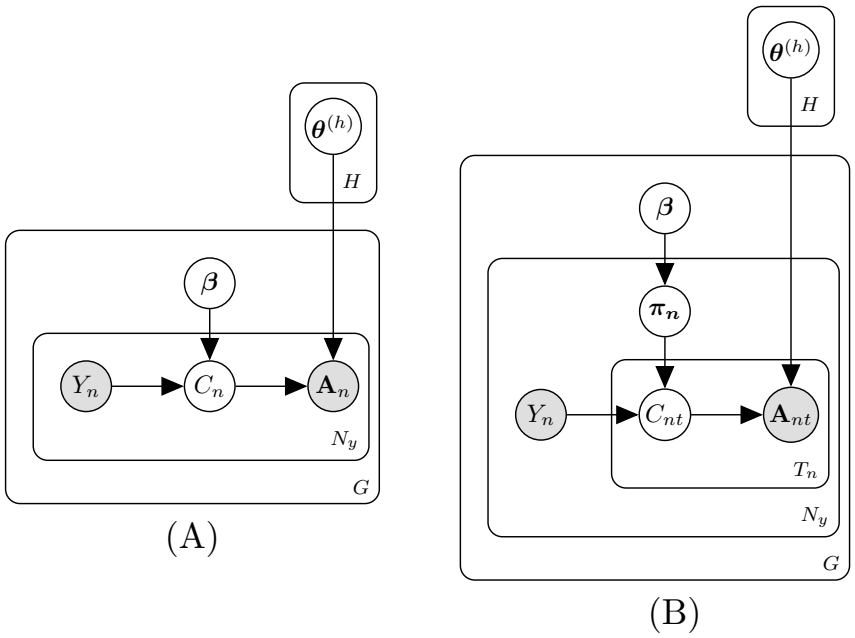

Figure 2.2.: The graphical models are given by (A) fixed and (B) with time-varying structures.

2.3.1 Population-level network comparison test

This task involves comparing the posterior probabilities of the two hypotheses, $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$ vs $H_1 : \boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$. Since H_0 being true amounts to $\mathcal{T} = 1$, our MCMC estimate of the probability equals the fraction of MCMC iterations where $\mathcal{T} = 1$. We first describe how our Gibbs sampler updates this variable (step 4 of our Gibbs sampler). At any MCMC iteration, let $\boldsymbol{m}_y$ be the vector of cluster assignment counts for population y, with component c giving the number of observations from population y assigned to cluster c: $\boldsymbol{m}_y = \left(\sum_{n;y_n=y}\sum_t \mathbb{I}_{C_{nt}=1}, ..., \sum_{n;y_n=y}\sum_t \mathbb{I}_{C_{nt}=H}\right)$. We write $\boldsymbol{m} = \boldsymbol{m}_1 + \boldsymbol{m}_2$ (for the two populations in G, i.e., 1 and 2). Then, under the two hypotheses, these counts are distributed as

$$\begin{aligned}
&\mathcal{H}_0 : \boldsymbol{m}_1, \boldsymbol{m}_2 \overset{iid}{\sim} \text{Mult}(\boldsymbol{\beta}), \\
&\qquad \boldsymbol{\beta} \sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
&\mathcal{H}_1 : \boldsymbol{m}_1 \sim \text{Mult}(\boldsymbol{\beta}_1) \;\; and \;\; \boldsymbol{m}_2 \sim \text{Mult}(\boldsymbol{\beta}_2) \\
&\qquad \boldsymbol{\beta_1}, \boldsymbol{\beta_2} \sim \text{Dirichlet}(\boldsymbol{\alpha})
\end{aligned} \tag{2.6}$$

Marginalizing out the $\boldsymbol{\beta}'s$, and recalling that both hypotheses have the same prior probability, we can specify the posterior

$$P(\mathcal{H}_1|-) = \frac{P(\boldsymbol{m}_1|\boldsymbol{\alpha})P(\boldsymbol{m}_2|\boldsymbol{\alpha})}{P(\boldsymbol{m}|\boldsymbol{\alpha}) + P(\boldsymbol{m}_1|\boldsymbol{\alpha})P(\boldsymbol{m}_2|\boldsymbol{\alpha})}$$

From the Dirichlet-multinomial conjugacy, we can write down the marginal probabilities of the observations, giving

$$P(\mathcal{H}_1|-) = \frac{\prod_{y=1}^{2} \frac{B(\boldsymbol{\alpha}+\boldsymbol{m}_y)}{B(\boldsymbol{\alpha})}}{\frac{B(\boldsymbol{\alpha}+\boldsymbol{m})}{B(\boldsymbol{\alpha})} + \prod_{y=1}^{2} \frac{B(\boldsymbol{\alpha}+\boldsymbol{m}_y)}{B(\boldsymbol{\alpha})}} \tag{2.7}$$

where $B(\cdot)$ is the multivariate beta function $B(x) = \prod_{i=1}^{q} \frac{\Gamma(x_i)}{\Gamma(\sum_{i=1}^{q} x_i)}$. Every Gibbs iteration samples $\mathcal{T}$ from this, with the posterior probability of the alternative hypothesis, $P(\mathcal{H}_1|-)$, being the fraction of MCMC samples where $\mathcal{T}$ equals 1. If the estimate from Equation (2.7) is larger than a specified threshold (e.g., 0.95), we reject the null hypothesis and conclude

that the populations are significantly different. We can use this *network comparison* (NC) test for both models in Figure 2.2. When we use the (fixed) model in 2.2a, we will refer to it as NC-F and when we use the (mixed-membership) model in 2.2b, we will refer to it as NC-M.

2.3.2 Entity-specific comparison test

This task refers to do following hypothesis test: $H_0^{n_1 n_2} : \boldsymbol{\pi}_{n_1} = \boldsymbol{\pi}_{n_2}$ Vs $H_1^{n_1 n_2} : \boldsymbol{\pi}_{n_1} \neq \boldsymbol{\pi}_{n_2}$ for any two users n_1 and n_2. Estimating this from our posterior samples is straightforward. Assuming multiple networks per entity, let $\tilde{\boldsymbol{m}}_n$ be a vector of counts for entity n, giving the number of observations assigned to each cluster. As mentioned earlier, the entity-specific distribution over clusters follows the distribution $\boldsymbol{\pi}_n \sim \text{Dir}(\boldsymbol{\beta})$. Following the earlier logic, Equation 2.8 gives the posterior probability two given entities have different cluster assignment probabilities:

$$P(\mathcal{H}_1^{n_1 n_2}|-) = \frac{\prod_{i=1}^{2} \frac{B(\boldsymbol{\beta}+\tilde{\boldsymbol{m}}_{n_i})}{B(\boldsymbol{\beta})}}{\frac{B(\boldsymbol{\beta}+\tilde{\boldsymbol{m}})}{B(\boldsymbol{\beta})} + \prod_{i=1}^{2} \frac{B(\boldsymbol{\beta}+\tilde{\boldsymbol{m}}_{n_i})}{B(\boldsymbol{\beta})}} \tag{2.8}$$

It is important to note that Equation 2.8 allows pairwise comparisons across populations, and therefore it is possible to have significantly similar entities from different populations and significantly different entities in the same population.

2.3.3 Edge-specific comparison test

This task refers to the following hypothesis test for an edge $l = (i, j)$, $H_0^l : \boldsymbol{\theta}_1[l] = \boldsymbol{\theta}_2[l]$ vs $H_1^l : \boldsymbol{\theta}_1[l] \neq \boldsymbol{\theta}_2[l]$. For the edge application we use an adjusted version of Cramer's V-statistic proposed by [10] given by Equation 2.9:

$$p_l^2 = \sum_{y=1}^{2} \mathbf{p}_{\mathcal{Y}} \sum_{a_l} \frac{P(\mathcal{L}(A)_l = a_l|\bar{\theta}_{y_l}) - P(\mathcal{L}(A)_l = a_l|\bar{\theta}_l)}{P(\mathcal{L}(A)_l = a_l|\bar{\theta}_l)} \tag{2.9}$$

where $\mathbf{p}_{\mathcal{Y}}$ is the sample size proportion of each population, $\bar{\boldsymbol{\theta}}_y = \sum_{n=1}^{N} \sum_{h=1}^{H} \beta_{yh} \boldsymbol{\theta}^{(h)} \mathbb{I}_{y_n=y}$, and $\bar{\boldsymbol{\theta}} = \sum_{y=1}^{2} \frac{\bar{\boldsymbol{\theta}}_y}{2}$. If $p_l \approx 1$ then there is evidence that edge weights differ across the populations.

2.4 Related work

Graph-based hypothesis testing and anomaly detection are topics of growing interest with diverse applications (see e.g., [48]). Many applications of hypothesis testing in network analysis focus on subgraphs *within* a larger graph (e.g., [56]), or one-sample tests comparing a single graph to a null model (e.g., [49]).

Work on populations of graphs can be divided on two areas: dynamic networks, in which one graph is replicated over time [24], [25] and [26]; and exchangeable graph modeling in which each graph is considered to be one observation for a single entity (see [2, 3, 9, 45] and [4]).

This chapter generalizes work from the latter category by allowing within-population heterogeneity, with each entity having multiple graphs with similar statistical properties. Both [3] and [45] deal with geometric characterizations of networks, and while their approaches are mathematically elegant, they are substantially less flexible than our work. [4] take a convolution neural network approach for non-aligned graphs, where there is no known mapping between nodes in each graph. This, coupled with the fact that their method requires the presense of node features, makes it unsuitable for our applications.

Most closely related to ours is the method presented in [2]. This method, which we will refer to DD, is a special case of our framework, where there is no within population variation, and where network edges are binary. For count or continuous-valued data, one might consider thresholding the edge weights of each entity and then applying the DD method. However this discards valuable information about the strength of the edge-weights, introduces sensitivity to threshold-level, and can reduce statistical power. Our method offers the ability to flexibly model such edge-weight information without any significant additional computational complexity. In particular, the computational time complexity of

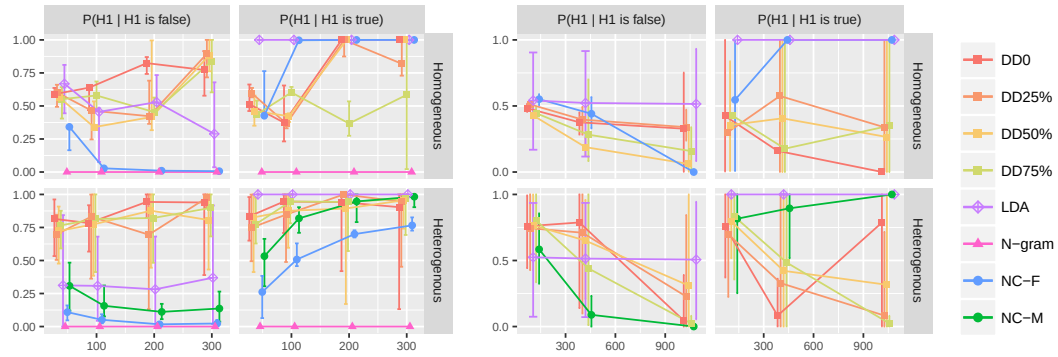

Figure 2.3.: Type I error and statistical power curves for the synthetic (left) and twitter (right) data for increasing sample sizes

both our method and DD is $\mathcal{O}(NHR|V|^2)$ per iteration. Here N is the number of networks, H is the number of clusters, R is the dimensionality of the low rank approximation, and $|V|$ is the number of nodes in each graph. In practice, H and R are small constants that do not grow with the data. In our experiments we compare with DD for different thresholds.

2.5 Experiments

In order to assess the efficacy of our method, we divided our analysis into four parts: statistical power and type-I error analyses, population-level hypothesis tests, edge-specific hypothesis tests, and additional exploratory analysis.

We start with statistical power and type-I error analyses, the most important measures of assessing hypothesis tests. We investigate the efficacy of our (and competing) methods for varying sample sizes when the ground truth in known. We show that when the data are generated from a known two-population setup, our hypothesis testing framework produces significantly more accurate results and has lower variance, with respect to type-I error and statistical power, compared to a number of other baselines. We show that for time-varying data, the mixed membership extension of our model is essential for reliable inference. We also study the sensitivity of the method of [2] (which requires unweighted graphs) to threshold settings, for population-level hypothesis tests. We show that for het-

erogeneous data, the hypothesis-test decisions are highly sensitive to threshold choice. We study the edge-specific hypothesis tests qualitatively, by visualizing the estimated model structure for each approach. We end by describing some additional insights that that our method gleans from the data. We start by describing the datasets.

2.5.1 Datasets

We generated synthetic weighted network data for two settings, the entity-homogeneous version from Figure 2.2a (NC-F, where each entity is represented by one graph) and the entity-heterogeneous version from Figure 2.2b (NC-M, where each entity is represented by multiple graphs). We also applied our framework to three real-world applications: a Twitter dataset from the political crisis in Brazil, two datasets about drugs usage on Instagram, and fMRI recordings of brains of human subjects.

Synthetic data (Homogeneous): We generate synthetic data from two populations whose underlying weight probability matrices $\boldsymbol{\theta}$ overlap around the middle set of nodes, but where population 1 has an elevated pattern of weight values for the first set of nodes, and population 2 in the final set. Figure A.1 in Appendix A.1 shows this structure. We simulated 200 entities per population, with 100 nodes for each network. Given the structure, the weights of the edges were distributed according to a multivariate Zipf distribution [57]. See Appendix section A.1 for more details.

Synthetic data (Heteromogeneous): Using the same population structure as above, we also construct a time-varying dataset where each individual has four time points, resulting in four different graphs per entity. In this case, we have 50 entities and 100 nodes for each network. Given the dependency structure, the weights for each entity at each time point were distributed according to a multivariate Zipf distribution. Figure A.2 in Appendix A.1 shows these structures. We use this dataset to compare the behavior of the NC-F and NC-M models under different scenarios.

Real data:

Twitter: Brazil has recently faced the worst economic/political crisis of its republic years. People were largely split into two sides: one who argued for the impeachment of the now former president, Dilma Rousseff, and the opposition who claimed that the process was a government coup. We crawled public Twitter posts from April 6th to May 31st 2016, using hashtags from both sides to collect tweets. The resulting dataset consists of $7,447$ users (entities), $4,233$ for the proposition and $3,214$ for opposition. In order to have appropriate data for the heterogeneous setting, we also split the dataset into time intervals, with each user having a network for every two weeks of tweets. We call this dataset "Twitter time". In this dataset, consisting of users with at least 15 days of tweets, we have a total of $2,098$ users, $1,255$ for proposition and 843 for opposition. Figure 2.1 shows sample co-occurrence networks from the two sides: Proposition and Opposition. Each edge-weight indicates the number of tweets of a user n containing two words (nodes) in a time interval t.

Instagram: We collected public Instagram comments with hashtags referring to the two most common drugs to treat ADHD (Adderall and Ritalin) and Depression (Prozac and Zoloft). These medications all have additional uses (and consequently symptoms), for instance, Adderall is known for loss of appetite, and as an aid for academic performance. Our dataset consists of 65 users with $44,408$ posts for #adderall, 21 with $17,466$ for #ritalin, 111 with $129,405$ for #prozac, and 35 with $29,357$ for #zoloft.

fMRI brain images: Functional magnetic resonance imaging (fMRI) captures activity in the brain by measuring blood flow from one region of the brain to another. We used the MRN-111 dataset[1] which consists of functional magnetic resonance images (fMRI) for 114 subjects (entities). As in [58] we used a total of 68 brain regions, 34 from the left hemisphere and 34 from the right. Nodes represent brain regions, and weights, white matter density across nodes. We compare brain activity across characteristics like Sex ($Male$ vs $Female$), and personality traits like creative level (≤ 90 vs ≥ 111), extroversion (≤ 30 vs ≥ 35). Values for creative level (CCI) and extroversion are given from a psychometric scale determined by the corresponding scientific literature, those thresholds were chosen

[1] `http://openconnecto.me/data/public/MR/`

to illustrate a clear *LowVsHigh* setting. Figure 2.1 shows sample brain networks of the MRN111 dataset for female and male individuals. We observe significant variability in these weights, suggesting that thresholding can lead to loss of information.

2.5.2 Baselines

We compared our NC-F and NC-M methods with the following baselines:

1. LDA (topic modeling) [51]: This treats each entity as a document made up of 'topics' (each corresponding to a distribution over word-count patterns).

2. N-gram language model [52]: We use observed bigrams frequencies to estimate co-occurrence probabilities.

3. DD network model [2]: As stated previously, DD forms a special instance of our more general framework for unweighted networks. In order to apply DD, we need first threshold the weighted network observations. We do so using the following criterion [59]:

$$p_{ij} = \frac{\text{co-occurances between words } i \text{ and } j}{\text{min (counts of words } i \text{ and } j)}$$

Then $A_n[i,j] = 1$ if $p_{ij} > threshold$ for a chosen threshold level.

Since N-gram and LDA do not directly allow us to estimate $P(H_1)$, we use a Kolmogorov-Smirnov test on the words distribution to perform an overall hypothesis test between populations for the N-gram model, and a chi-square test for topic assignments across populations for LDA.

2.5.3 Hyperparameters tunning

DD and our method require setting the number of clusters H, the dimensionality of the low-rank factorization R, the Dirichlet concentration parameters α for $\boldsymbol{\beta}$, and the prior probability of $P(H_1)$. For our experiments, we fixed $H = 15$, $R = 10$, $\alpha = 1/H$ and $P(H_1) = 0.5$. We found that $H = 15$ was more than enough clusters for all instances,

larger numbers resulting in empty clusters. Most of our experiments focus on settings with count-valued weights, and in the case of word co-occurrences, the weight between words i and j is bounded by the smaller of the number of occurrences of the two words i and j. For this setting, we therefore used the logit link and the binomial likelihood. On the other hand, the Brain dataset has count-valued weights with unbounded support, and we used Exponential link and the Poisson likelihood. Our results were fairly robust to hyperparameters settings. For our MCMC algorithm, we observed good mixing properties, and used 1300 Gibbs samples with an extra 200 burn-in samples.

For LDA and N-Gram, we used settings following implementations from [60] and [61], respectively.

2.5.4 Results

Type-I error and statistical power: Type-I errors or false positives arise when a model incorrectly marks two populations as different when actually the null hypothesis is true, i.e., $P(H_1|H_1\text{is false})$. Ideally, type-I error rates should be 0.05 or less. Statistical power shows if the models can correctly determine when the populations are different, and $P(H_1|H_1 \text{ is true})$ should be close to 1. Measuring these quantities requires access to ground truth. For the Brazil dataset, the disparity of political tendencies between opposition and proposition is clear enough that we treat it as ground truth (For the other real datasets, we do not have ground truth available).

We consider four sample sizes for the synthetic data: 50, 100, 200 and 300. For the twitter data, we consider three sample sizes: 105 ($\sim 5\%$), 420 ($\sim 20\%$) and 1049 ($\sim 50\%$). For DD which requires thresholding, we used four different threshold-levels, 0, 25%, 50% and 75%. For all methods, we compute $P(H_1|-)$ under different settings. In order to estimate variance $P(H_1|-)$, we generated 20 datasets for each sample size.

Figure 2.3 presents the Type-I error (i.e., $P(H_1|H_1\text{is false})$) and power curves (i.e., $P(H_1|H_1\text{is true})$) for increasing sample sizes for each method for synthetic data (left) and real world data (right). Each data point shown is the mean of the 20 trials for the respective

sample size, we also present the range of 5% to 95% percentiles. We see that NC-F has the best overall performance both when H_1 is true and for H_1 false in the homogeneous scenario. N-gram has the worst performance overall so we do not consider this baseline in the real data. LDA have a good overall power, however it performs poorly as far as Type I error goes, with the largest variance. Further, LDA is not able to capture probabilistic structure underlying the data (we discuss this later). DD's performance is not so good for Type I error in the synthetic data, but it is good overall for power. This is not the case for the real data though, where power is poor and varies significantly with threshold. This sensitivity to threshold-level confirms the original motivation for this work. NC-M outperformed all other methods for the heterogeneous data, and NC-F was the second best even though it does not account for heterogeneity. This is due to the fact that our method accounts for the weight distributions, and thus can handle overdispersed counts relatively well.

We also investigate how the number of active users in the dataset affects the overall power performance. First, we define "activeness" as a function of number of days a user tweeted. We created datasets restricting to users with at least 2, 5 and 10 days of tweets. Thus, we assume that in the dataset with users having at least 10 days of tweets, there are a significantly larger number of active users than in dataset with a minimum of 2 days of tweets. Note that we are assessing only statistical power here, therefore we are only looking to the case that H_1 is true, i.e., "PropositionVsOpposition". Figure 2.4 shows the power curve for each these datasets. As expected, our method needs fewer observations to find statistical difference between populations with more active users in the dataset, in other words NC-F has the ability to differentiate between populations easier as the proportion of active users increases. DD does not improve with larger sample size which suggests DD loses the ability to determine whether populations differ in a higher heterogeneous setting.

Population-level hypothesis test: In the previous results, we had a glimpse of the sensibility of threshold choice in terms of decision making on the hypothesis testing procedure. Here, we aim to analyze this further. We estimate the posterior probability of H_1 for *all observations* of all the datasets. We compare our results with DD for 10 different thresh-

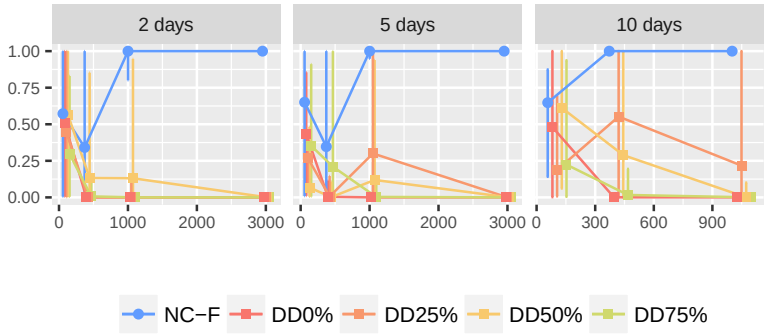

Figure 2.4.: Power curves of NC-F and DD (multiple threshold levels) for increasing proportion of active users

old levels $(0, 10\%, 20\%, ..., 90\%)$. Note that NC-F and NC-M do not vary with threshold level. Here, in addition to the Twitter data, we include results testing the fMRI dataset—comparing populations based on the creative index (CCI). [2] found a significant difference in Brain connectivity between non creative individuals ($CCI \leq 90$) vs creative subjects ($CCI \geq 111$). In their tests, the graphs were thresholded at 0.

Figure 2.5(left) shows DD represented as red solid squares, NC-F and NC-M as blue and green lines, respectively. Again, DD exhibits sensitivity to the threshold choice making inferences unreliable. For instance, if we consider testing whether populations Proposition and Opposition are significantly different and use a 60% threshold for Twitter time, we would reject the null, since the posterior of $P(H_1) \approx 1$. However, if we slightly change the threshold level to 50%, $P(H_1) \approx 0.5$ meaning that there is not enough evidence to support that they are statistically different and we accept the null. The same behavior can be seen on the fMRI dataset where the threshold at 0 is statistical significant for *LowVsHigh*, however it is not for any other threshold. Overall, we found that our methods NC-M and NC-F are more reliable, since they avoid the need for practitioners to make sensitive preprocessing choices.

Edge-specific level hypothesis test: Another important task is that of retrieving the structure of the co-occurrences probabilities. For better visualization, we generated a version of the synthetic homogeneous with 20 nodes, and we look at differences between true and

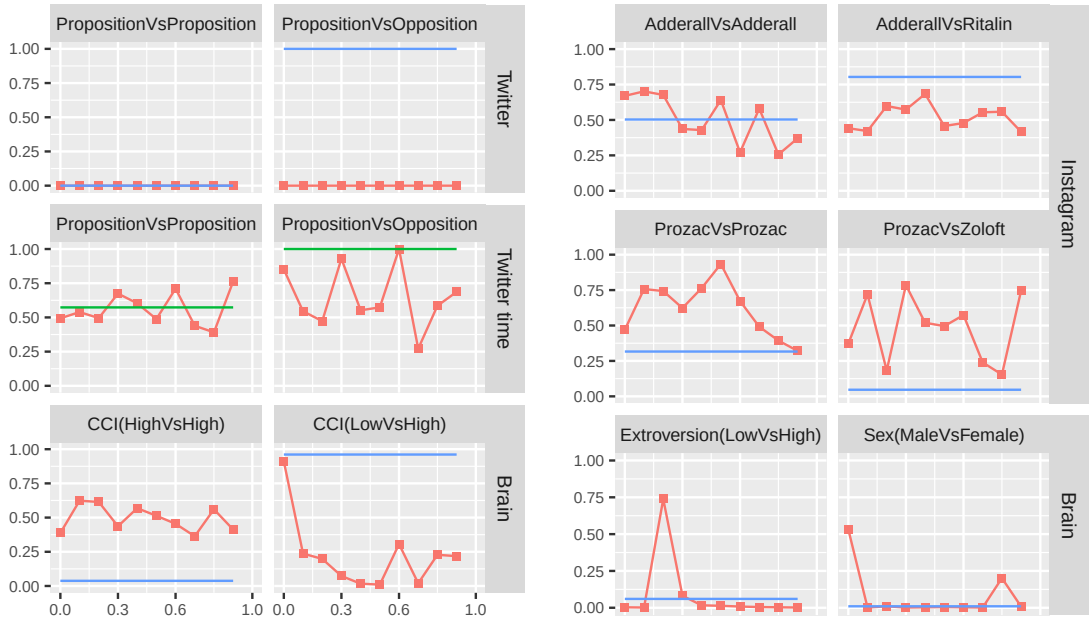

Figure 2.5.: **Left:** $P(H_1|-)$ across threshold levels for three datasets for when H_1 is false (left column) and H_1 is true (right column). NC-F and NC-M are represented as blue and green lines, respectively. **Right:** $P(H_1|-)$ across threshold levels for Instagram and fMRI datasets.

predicted edge probability matrices for both populations, i.e. we compute the estimated difference $\bar{\boldsymbol{\theta}}_1 - \bar{\boldsymbol{\theta}}_2$ for each model and compared with the ground truth. In Figure 2.6, we see that our proposed framework accurately recovers the structure of the ground truth. The DD0 also retrieves the structure of population 1, however it performs poorly for population 2. This is related to the sensitivity of results to the threshold-level, suggesting this needs to be chosen carefully across different scenarios. Our models NC-F and NC-M both do not require such hand-tuning, and further exploit values of the pre-thresholded counts for more accurate inference. Unsurprisingly, all the other models fail to correctly learn the structure used to simulate the data.

For the Twitter and Instagram datasets, we looked at each edge, and identified those that are different using a 0.1 significance level. Figure 2.7 shows that the NC-F model was able to capture a clear pattern of significantly different use of words among populations for the Brazil dataset, as opposed to the other modeling schemes which look almost random.

For the Instagram drug data, the edge-specific hypothesis testing matrix structure is much more significant compared with the Twitter case. One reason for this is that there is a group of Ritalin users that are German, and their words differ from others.

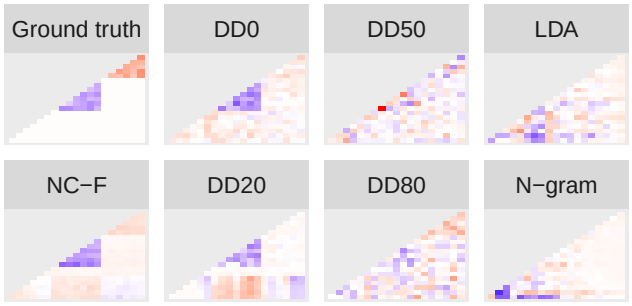

Figure 2.6.: Edge-specific probabilities difference

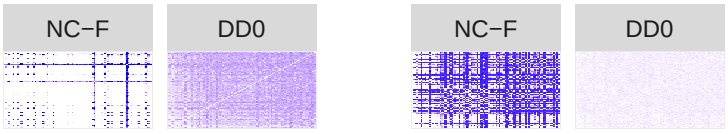

Figure 2.7.: Edge-specific tests for Twitter (left) and Instagram (right).

Exploratory analysis: Here, we show some additional insights that our methods are capable of capturing. One interesting fact of the Brazil political scenario is that many high frequency words were extensively used across both populations, examples being “impeachmentja” (impeachment now), “lavajato” (carwash), “golpenao” (no coup), “direitos” (rights). However, using the probability structure $\bar{\theta}$ estimated from our framework, we can make some interesting insights about how the two sides frame the issues differently. Figure 2.8 plots the difference of the link probability for each high frequency word used in conjunction (co-occurrence) with all other words, across the two sides—if the value is larger than zero then it is a ‘proposition expression’, otherwise it is an ‘opposition expression’. For instance, “lavajato” is the name of the investigation and if it is used with “motivo” (motive) is a proposition statement where if it is used with “luta” (fight) then it is a clear opposition

one. From Figure 2.8 it is clear that the two sides use sets of words (e.g., phrases) quite differently.

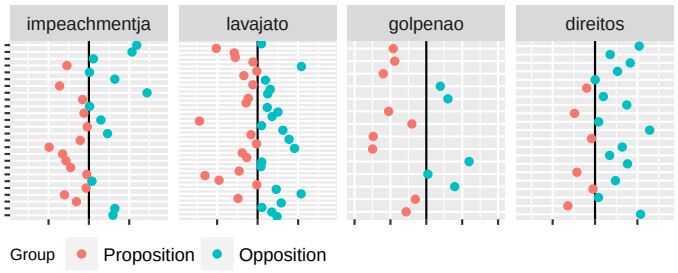

Figure 2.8.: Swing words: differences in edge probabilities for example high frequency terms.

Figure 2.5(right) presents additional results for the Instagram and fMRI datasets. In this case, we look again to the behavior of $P(H_1|-)$ across threshold levels. It is important to highlight that we do not have a ground truth information to compare our findings with, however it is an additional set of results to explore the assessment of significant difference between two populations, and to note the lack of robustness of DD wrt threshold choice.

2.6 Conclusion

This chapter presents the first steps towards routine and systematic hypothesis testing on populations of weighted networks. Our statistical framework applies both to settings where entities from each population have single graphs associated with them, as well as settings where each entity has associated a set of graphs (we call these without and with within-population heterogeneity). Through a flexible and general clustering mechanism for replicated weighted networks, our framework offers a powerful and accurate hypothesis testing at three levels: population-level, entity-specific and edge-specific. We applied our model to study communication behavior on real social media data (Instagram and Twitter), as well as for brain connectivity data. We saw that by not relying on a a user-specified threshold, our proposed method offers robustness over the methodology of [2], besides outperforming other baselines like LDA, N-gram language models.

3. NON-ALIGNED GRAPHS

3.1 Introduction

As discussed before, much of the work on networks has focused on the analysis of a single large graph and in the area of community detection and clustering is not different. Some efforts have moved beyond this to consider multiple graphs scenarios with include multi-view clustering [27], multi-layer community detection [28–33], and temporal clustering [34]. These assume the graphs are *aligned*, with a known mapping between nodes in each graph. The multiple graphs provide different information about the same set of nodes. Applications are time-evolving [23–26] as well as independent graph observations [2,3,8,9]. Such methods are not applicable to non-aligned graphs, from fields like biology (e.g., fungi networks, protein networks) and social media (e.g., social networks, word co-occurrences). Here, graph instances are often drawn from the same population, with limited or no correspondence between the nodes across graphs (the nodes have similar behavior but represent different entities). While one could cluster graphs separately, pooling information across graphs can improve estimation, particularly for sparse or imbalanced graphs.

In this chapter, we focus on the problem of community detection across a set of non-aligned graphs of varying size. We are given a set of N graphs, $\mathbf{\Omega} = \{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N\}$, where the nth graph is represented by its adjacency matrix $\boldsymbol{A}_n \in \{0,1\}^{|V_n| \times |V_n|}$ where V_n is the set of nodes. We do not require V_n to be shared across different graphs, however we assume they belong to a common set of K communities. Our goal is to identify these K communities. For instance, consider villages represented by social graphs where nodes represent individuals in a village and edges represent relationships between them. Although the people are different in each village and the sizes of villages vary, personal characteristics may impact the propensity of having some type of relationship, e.g. a younger individual is more willing to connect with other younger individuals, or an influential person such as a

priest might have more ties. Indeed, there are a vast number of factors, both observed and latent, shared among people across villages, that might influence relationship and community formation.

Our approach builds on the popular stochastic block model (SBM). SBMs have been studied extensively for single graph domains [62–64], and have been highly effective on real world problems [65, 66]. A simple extension to our setting separately estimates the SBM parameters per graph, and then attempt to find a correspondence between the estimated probability structures to determine a single *global* community structure. We call this *Isolated SBM* and show it is only accurate when each community is well represented in each graph. Moreover, the process of aligning the estimated probabilities can have $\mathcal{O}(NK!)$ complexity, which is only feasible when K is very small.

We propose a joint SBM model to estimate the joint connectivity structure among the communities in each graph, while allowing the number and sizes of clusters to vary across graphs. To estimate the parameters of the model, we relate the individual graph adjacency matrix eigendecompositions to the decomposition of the whole dataset $\mathbf{\Omega}$, and then derive an efficient joint spectral clustering approach. Our learning algorithm has complexity $\mathcal{O}(|V|K)$ per iteration where $|V|$ is total number of nodes across graphs, and does not need the $\mathcal{O}(NK!)$ alignment step of *Isolated SBM*. We evaluate our *Joint SBM* on synthetic and real data, and show it is able to more accurately recover the community structure than a number of baselines, particularly when graphs are highly heterogeneous. We defer all proofs to the appendix B.

3.2 Joint SBM for multiple graphs, and spectral clustering

The stochastic blockmodel (SBM) [39, 40] for a *single graph* $\boldsymbol{A}$ with K communities, is defined by a $(|V_n| \times K)$ membership matrix $\boldsymbol{X}$, and a connectivity matrix $\mathbf{\Theta} \in [0,1]^{K \times K}$. Here $\boldsymbol{X}[i,k] = 1$ if node i belongs to community k, and is 0 otherwise. $\mathbf{\Theta}[k,l] = \theta_{kl}$ is the edge-probability between nodes from communities k and l. Then, a graph represented by adjacency matrix $\boldsymbol{A}$ is generated as

$$a_{ij} \sim \text{Bern}(\boldsymbol{X}_i \boldsymbol{\Theta} \boldsymbol{X}_j^T) \quad \text{if } i < j. \tag{3.1}$$

Note that $a_{ij} = a_{ji}\ \forall i, j$, and since we do not consider self-edges, $a_{ii} = 0\ \forall i$. In settings with a single graph, theoretical properties like consistency and goodness-of-fit are well understood, and efficient polynomial-time algorithms with theoretical guarantees have been proposed for learning and inference. However, there is little work for the case of multiple graphs.

3.2.1 Multi-graph joint SBM

To address this, we consider an extension of the SBM in Eq.(3.1). Our model does not require vertices to be aligned across graphs, nor does it require different graphs to have the same number of vertices. Vertices from all graphs belong to one of shared set of groups, with membership of graph n represented by a membership matrix $\boldsymbol{X}_n$. Edge-probabilities between nodes are determined by a *global* connectivity matrix $\boldsymbol{\Theta}$ shared by all graphs. For notational convenience, we will refer to the set of stacked $\boldsymbol{X}_n$ matrices as the full membership matrix $\boldsymbol{X}$, with $\boldsymbol{X}_{ni}$ one-hot membership vector of the i-th node of graph n. The overall generative process assuming K blocks/communities is

$$a_{nij} \sim \text{Bern}(\boldsymbol{X}_{ni} \boldsymbol{\Theta} \boldsymbol{X}_{nj}^T) \quad \text{if } i < j \tag{3.2}$$

where again, a_{nij} is ij-th cell of the adjacency matrix $\mathbf{A}_n$. Note that $\boldsymbol{X}_n$ is a $(|V_n| \times K)$ binary matrix, and $\boldsymbol{\Theta}$ is a $K \times K$ matrix of probabilities. This model can easily be extended to edges with weights (replacing the Bernoulli distribution with some other distribution) or to include covariates (e.g. through another layer of coefficients relating covariates with membership or edge probabilities).

3.2.2 Spectral clustering for a single graph

First, we recall the spectral clustering method for SBMs for a single graph $\mathbf{A}_n$ [63, 67]. Let $\mathbf{P}_n$ refer to the edge probability matrix of graph n under an SBM, where $\mathbf{P}_n =$

$\boldsymbol{X}_n\boldsymbol{\Theta}\boldsymbol{X}_n^T$ (Eq.(3.1)). Since we do not consider self-loops, $\mathbb{E}[\mathbf{A}_n] = \mathbf{P}_n - \text{diag}(\mathbf{P}_n)$. Write the eigendecomposition of $\mathbf{P}_n$ as $\mathbf{P}_n = \mathbf{U}_n\mathbf{D}_n\mathbf{U}_n^T$. Here, $\mathbf{U}_n$ is a $(|V_n| \times K)$ matrix of eigenvectors related to the K largest absolute eigenvalues and $\mathbf{D}_n$ is a $K \times K$ diagonal matrix with the K non-zero eigenvalues of $\boldsymbol{P}_n$. Let $|G_{nk}|$ be the number of nodes that belong to cluster k, and $\Delta_n = \left(\boldsymbol{X}_n^T\boldsymbol{X}_n\right)^{1/2}$ define a $K \times K$ diagonal matrix with entries $\sqrt{|G_{nk}|}$. Letting $\mathbf{Z}_n\tilde{\mathbf{D}}_n\mathbf{Z}_n^T$ be the eigendecomposition of $\Delta_n\boldsymbol{\Theta}\Delta_n$,

$$\begin{aligned}\mathbf{U}_n\mathbf{D}_n\mathbf{U}_n^T = \mathbf{P}_n = \boldsymbol{X}_n\boldsymbol{\Theta}\boldsymbol{X}_n^T &= \boldsymbol{X}_n\Delta_n^{-1}\Delta_n\boldsymbol{\Theta}\Delta_n\Delta_n^{-1}\boldsymbol{X}_n^T \\ &= \boldsymbol{X}_n\Delta_n^{-1}\mathbf{Z}_n\tilde{\mathbf{D}}_n\mathbf{Z}_n^T\Delta_n^{-1}\boldsymbol{X}_n^T. \end{aligned} \tag{3.3}$$

Since $\mathbf{D}_n$ and $\tilde{\mathbf{D}}_n$ are diagonal, and $\boldsymbol{X}_n\Delta_n^{-1}\mathbf{Z}_n$ is orthonormal, $\mathbf{D}_n = \tilde{\mathbf{D}}_n$, $\mathbf{U}_n = \boldsymbol{X}_n\Delta_n^{-1}\mathbf{Z}_n$. In practice, we use the observed adjacency matrix $\mathbf{A}_n$ as a proxy for $\mathbf{P}_n$, and replace $\mathbf{U}_n$ with $\widehat{\mathbf{U}}_n$ calculated from the eigendecomposition $\mathbf{A}_n = \widehat{\mathbf{U}}_n\widehat{\mathbf{D}}_n\widehat{\mathbf{U}}_n^T$. Finally we note that each row of $\boldsymbol{X}_n$ has only one non-zero element, indicating which group that node belongs to. Thus, as in [63], we can use k-means clustering to recover $\boldsymbol{X}_n$ and $\mathbf{W}_n = \Delta_n^{-1}\mathbf{Z}_n$ from $\widehat{\mathbf{U}_n}$:

$$\left(\widehat{\boldsymbol{X}_n}, \widehat{\boldsymbol{W}_n}\right) = \underset{\boldsymbol{X}_n \in \mathcal{M}_{|V_n|,K}, \boldsymbol{W}_n \in \mathbb{R}^{K\times K}}{\arg\min} \|\boldsymbol{X}_n\boldsymbol{W}_n - \widehat{\boldsymbol{U}}_n\|_F^2 \tag{3.4}$$

Given $\left(\widehat{\boldsymbol{X}_n}, \widehat{\boldsymbol{W}_n}\right)$, we can estimate $\boldsymbol{\Theta}$ as:

$$\widehat{\boldsymbol{\Theta}}_n = \widehat{\boldsymbol{S}}_n + \widehat{\Delta}_n^{-2}\left[\mathbb{I}_k - \widehat{\Delta}_n^{-2}\right]^{-1}\text{diag}\left(\widehat{\boldsymbol{S}}_n\right) \tag{3.5}$$

where $\widehat{\boldsymbol{S}}_n = \widehat{\Delta}_n^{-2}\widehat{\boldsymbol{X}}_n^T\boldsymbol{A}_n\widehat{\boldsymbol{X}}_n\widehat{\Delta}_n^{-2}$ and $\widehat{\Delta}_n^2 = \widehat{\boldsymbol{X}}_n^T\widehat{\boldsymbol{X}}_n$. See Appendix B.3 for details of derivation.

3.2.3 Naive spectral clustering for multiple graphs

Given multiple unaligned graphs, the procedure above can be applied to each graph, returning a set of $\boldsymbol{X}_n$'s and $\boldsymbol{\Theta}_n$'s, one for each graph. The complexity of this is $\mathcal{O}(N\phi +$

$|V|K)$, where ϕ is the cost of eigendecomposition on a single graph (typically $\mathcal{O}(|E|)$ for sparse graphs [68,69]). This does not recognize that $\mathbf{\Theta}$ is shared across all graphs, and estimating a global $\mathbf{\Theta}$ from the individual $\mathbf{\Theta}_n$s requires an *alignment* step. Searching over all permutations for the best alignment has cost $\mathcal{O}(NK!)$ and is a two-stage procedure that results in loss of statistical efficiency, especially in settings with heterogenerous, imbalanced graphs. We refer to this approach as *Isolated SBM*. We propose a novel algorithm to get around these issues by understanding how each graph relates to the global structure.

3.2.4 Joint spectral clustering for multiple graphs

Let $|V| = \sum_n |V_n|$, where $|V_n|$ is the number of nodes in A_n. Consider the $|V| \times |V|$ block diagonal matrix $\mathbf{A}$ representing all adjacency matrices, and define an associated probability matrix $\mathbf{P}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & \mathbf{A}_n & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \mathbf{A}_N \end{bmatrix}, \mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & \dots & \mathbf{P}_{1n} & \dots & \mathbf{P}_{1N} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{P}_{n1} & \dots & \mathbf{P}_n & \dots & \mathbf{P}_{nN} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{N1} & \dots & \mathbf{P}_{Nn} & \dots & \mathbf{P}_N \end{bmatrix}. \tag{3.6}$$

Write the membership-probability decomposition of $\mathbf{P}$ as $\mathbf{P} = \boldsymbol{X}\mathbf{\Theta}\boldsymbol{X}^T$, here, $\boldsymbol{X}$ is the stacked $|V| \times K$ matrix of the $\boldsymbol{X}_n$'s for all graphs, and $\mathbf{\Theta}$ a $K \times K$ matrix of edge-probabilities among groups. Note that $i \neq j$, $\mathbf{P}_{ij}$ includes edge-probabilities between nodes *in different graphs*, something we cannot observe. As before, define $\Delta = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{1/2}$, and the eigendecomposition of $\mathbf{P}$ gives

$$\begin{aligned} \mathbf{U}\mathbf{D}\mathbf{U}^T = \mathbf{P} &= \boldsymbol{X}\mathbf{\Theta}\boldsymbol{X}^T = \boldsymbol{X}\Delta^{-1}\Delta\mathbf{\Theta}\Delta\Delta^{-1}\boldsymbol{X}^T \\ &= \boldsymbol{X}\Delta^{-1}\mathbf{Z}\tilde{\mathbf{D}}\mathbf{Z}^T\Delta^{-1}\boldsymbol{X}^T, \end{aligned} \tag{3.7}$$

with $\mathbf{Z}\tilde{\mathbf{D}}\mathbf{Z}^T$ corresponding to the eigendecomposition of $\Delta\boldsymbol{\Theta}\Delta$. Similar to the single graph case

$$\mathbf{D} = \tilde{\mathbf{D}}, \quad \mathbf{U} = \boldsymbol{X}\Delta^{-1}\mathbf{Z}, \tag{3.8}$$

Note that $\mathbf{D}$ is still a $K \times K$ diagonal matrix. Let $\boldsymbol{U}_{n*} = \boldsymbol{X}_{n*}\Delta^{-1}\mathbf{Z}$ refer to the subset of $\boldsymbol{U}$ corresponding to the nodes in graph n, and define $\boldsymbol{X}_{n*}$ similarly. Note that $\boldsymbol{X}_{n*} = \boldsymbol{X}_n$, though $\boldsymbol{U}_{n*}$ differs from $\boldsymbol{U}_n$ of Eq. (3.3). By selecting the decomposition related to graph n, we have,

$$\mathbf{P}_n = (\mathbf{P})_{n,n} = \left(\mathbf{U}\mathbf{D}\mathbf{U}^T\right)_{n,n} = \mathbf{U}_{n*}\mathbf{D}\mathbf{U}_{n*}^T. \tag{3.9}$$

From Eq.(3.3), $\mathbf{U}_{n*}\mathbf{D}\mathbf{U}_{n*}^T = \mathbf{U}_n\mathbf{D}_n\mathbf{U}_n^T$. Let $\mathbf{Q}_n := \mathbf{U}_n\mathbf{D}_n$

$$\begin{aligned}
\mathbf{U}_{n*}\mathbf{D}\mathbf{U}_{n*}^T &= \mathbf{Q}_n\mathbf{U}_n^T \\
\mathbf{U}_{n*}\mathbf{D}\mathbf{U}_{n*}^T\mathbf{U}_n &= \mathbf{Q}_n\mathbf{U}_n^T\mathbf{U}_n \\
\mathbf{U}_{n*}\mathbf{D}(\boldsymbol{X}_{n*}\Delta^{-1}\mathbf{Z})^T\boldsymbol{X}_n\Delta_n^{-1}\mathbf{Z}_n &= \mathbf{Q}_n \\
\mathbf{U}_{n*}\mathbf{D}\mathbf{Z}^T\Delta^{-1}\boldsymbol{X}_{n*}^T\boldsymbol{X}_n\Delta_n^{-1}\mathbf{Z}_n &= \mathbf{Q}_n \\
\mathbf{U}_{n*}\mathbf{D}\mathbf{Z}^T\Delta^{-1}\Delta_n^2\Delta_n^{-1}\mathbf{Z}_n &= \mathbf{Q}_n \qquad (3.10) \\
\mathbf{U}_{n*}\mathbf{D} = \mathbf{Q}_n\mathbf{Z}_n^T\Delta_n^{-1}\Delta\mathbf{Z} &= \boldsymbol{X}_{n*}\boldsymbol{W} \qquad (3.11)
\end{aligned}$$

where $\boldsymbol{W} := \Delta^{-1}\mathbf{Z}\mathbf{D}$. By contrast for the single graph, $\mathbf{U}_n\mathbf{D}_n = \mathbf{Q}_n = \boldsymbol{X}_n\Delta_n^{-1}\mathbf{Z}_n\mathbf{D}_n$ (Eq.(3.3)). If we have the middle term in Eq. (3.11) from data, we can solve for joint community assignments:

$$\left(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{W}}\right) = \underset{\boldsymbol{X}\in\mathcal{M}_{|V|,K}, \boldsymbol{W}\in\mathbb{R}^{K\times K}}{\arg\min} \sum_n^N \|\boldsymbol{X}_n\boldsymbol{W} - \mathbf{Q}_n\boldsymbol{Z}_n^T\Delta_n^{-1}\Delta\mathbf{Z}\|_F^2 \tag{3.12}$$

While we can estimate $\mathbf{Q}_n$ from the data, $\mathbf{Q}_n = \boldsymbol{U}_n\boldsymbol{D}_n$, we cannot estimate $\mathbf{Z}_n$ or $\mathbf{Z}$ trivially. Instead, we will optimize an upper bound of a transformation of Eq.(3.12).

Lemma 3.2.1 *Eq.* (3.12) *is equivalent to*

$$\underset{\substack{\boldsymbol{X}\in\mathcal{M}_{|V|,K}\\ \boldsymbol{W}\in\mathbb{R}^{K\times K}}}{\arg\min}\sum_{n=1}^{N}\left\|a_n(\boldsymbol{X}_n,\boldsymbol{W})+b_n(\boldsymbol{X}_n)\right\|_F^2, \textit{where} \tag{3.13}$$

$$\begin{aligned} a_n(\boldsymbol{X}_n,\boldsymbol{W}) &:= \left(\boldsymbol{X}_n\mathbf{W}-\mathbf{Q}_n^*\right)\boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n \\ b_n(\boldsymbol{X}_n) &:= \mathbf{Q}_n^*\left(\boldsymbol{Z}^T\Delta^{-1}\Delta_n\mathbf{Z}_n-\sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K\right) \\ \mathbf{Q}_n^* &:= \mathbf{Q}_n\sqrt{|V||V_n|}. \end{aligned}$$

Next we derive a bound for Eq. (3.13) using the triangle inequality and sub-multiplicative norm property. Let $|\boldsymbol{M}|$ be the element-wise absolute values of matrix $\boldsymbol{M}$. Then

Lemma 3.2.2 *With*

$$\gamma_n=\left\|\boldsymbol{Z}\Delta^{-1}\Delta_n\mathbf{Z}_n^T\right\|_F^2=tr\left(\Delta_n^2\Delta^{-2}\right)=\sum_{m=1}^{K}\frac{|G_{nm}|}{|G_{\cdot m}|},$$

$$\begin{aligned} &\frac{1}{2}\left\|a_n(\boldsymbol{X}_n,\boldsymbol{W})+b_n(\boldsymbol{X}_n)\right\|_F^2\leq \\ &\qquad\leq\left\|\boldsymbol{X}_n\mathbf{W}-\mathbf{Q}_n^*\right\|_F^2\gamma_n+\left\||\mathbf{Q}_n^*|\left(\Delta^{-1}\Delta_n+\sqrt{|V_n|/|V|}\right)\right\|_F^2 \\ &\qquad:=\widetilde{a}_n(\boldsymbol{X}_n,\boldsymbol{W})+\widetilde{b}_n(\boldsymbol{X}_n):=\eta_n(\boldsymbol{X}_n,\boldsymbol{W}) \end{aligned} \tag{3.14}$$

We can optimize the bound on Eq. (3.13):

$$\left(\widehat{\boldsymbol{X}},\widehat{\boldsymbol{W}}\right)=\underset{\boldsymbol{X}\in\mathcal{M}_{|V|,K},\boldsymbol{W}\in\mathbb{R}^{K\times K}}{\arg\min}\sum_{n}^{N}\eta_n(\boldsymbol{X}_n,\boldsymbol{W}) \tag{3.15}$$

The terms $\widetilde{a}_n(\cdot)$ and $\widetilde{b}_n(\cdot)$ are weighted sums of squares of $\boldsymbol{Q}_n$. However, we center each $\boldsymbol{Q}_n$ at the global weighted mean $\boldsymbol{W}$ in $\widetilde{a}_n(\cdot)$. The term γ_n controls the importance of the

global parameter $\boldsymbol{W}$ in each graph. Thus, γ_n downweights the effect of $\boldsymbol{W}$ in small graphs and in graphs with highly underrepresented communities. Intuitively, the term $\widetilde{a}_n(\cdot)$ is assigning nodes to clusters assuming $\boldsymbol{Z}_n^T \Delta_n^{-1} \Delta \mathbf{Z} = \sqrt{|V|/|V_n|}\mathbb{I}_K$, and $\widetilde{b}_n(\cdot)$ accounts for the distance between a given graph and the global distribution of nodes over clusters.

Optimization: We optimize Eq.(3.15) using a heuristic inspired by Lloyd's algorithm for k-means. This involves iterating two steps: (1) compute the means $\boldsymbol{W}$ given observations in each cluster $\boldsymbol{X}$; (2) assign observations to clusters $\boldsymbol{X}$ given means $\boldsymbol{W}$:

1. **Compute the means:** We update $\boldsymbol{W}$ by minimizing $\eta_n(\boldsymbol{X_n}, \boldsymbol{W})$ for a given $\boldsymbol{X}$, i.e. $\sum_n^N \nabla_{\boldsymbol{W}} \eta_n(\boldsymbol{X}_n, \boldsymbol{W}) = 0$. Note that the $\widetilde{b}_n$ does not involve $\boldsymbol{W}$ so it is dropped:

$$\begin{aligned} &\sum_n^N 2\boldsymbol{X}_n^T(\boldsymbol{X}_n\widehat{\boldsymbol{W}} - \boldsymbol{Q}_n^*)\gamma_n = 0 \\ &\implies \widehat{\boldsymbol{W}} = [\sum_n^N \boldsymbol{X}_n^T\boldsymbol{X}_n\gamma_n]^{-1} \sum_n^N \boldsymbol{X}_n^T\boldsymbol{Q}_n^*\gamma_n \end{aligned} \tag{3.16}$$

2. **Assign nodes to communities:** We assign each node to the cluster that minimizes Eq. (3.15). Accordingly, define $\omega_{ni}(k)$ as the distance of node i to cluster k:

$$\begin{aligned} \omega_{ni}(k) = &\|\mathbf{W}_k - \mathbf{Q}_{ni}^*\|^2 \operatorname{tr}\left(\widetilde{\Delta}_{ni}^2(k)\right) \\ &+ \left\| |\mathbf{Q}_{ni}^*| \left(\widetilde{\Delta}_{ni}(k) + \sqrt{|V_n|/|V|}\mathbb{I}_K\right)\right\|^2 \end{aligned} \tag{3.17}$$

$\boldsymbol{W}_k$ is the k-th row of $\boldsymbol{W}$ and $\widetilde{\Delta}_{ni}(k))$ is the value of $\Delta^{-1}\Delta_n$ if node i is placed in cluster k. Precisely, say node i is currently in cluster l, then we have

$$\begin{aligned} \widetilde{\Delta}_{ni}(k) = &\left[(\Delta^2 - \operatorname{diag}(H_l) + \operatorname{diag}(H_k))\right]^{-1/2} \times \\ &\times \left[(\Delta_n^2 - \operatorname{diag}(H_l) + \operatorname{diag}(H_k))\right]^{1/2} \end{aligned} \tag{3.18}$$

where H_l is a size K one-hot vector at position l. Then $\boldsymbol{X}_{ni} = \arg\min_k \omega_{ni}(k)$.

Algorithm: The complexity is $\mathcal{O}(N\phi + |V|K)$. Recall that ϕ refers to the complexity of eigen decomposition on a single graph. Note that our derived objective does not require decomposition of the full graph $\mathbf{A}$, instead decomposing each individual graph and then using the results to jointly estimate $\mathbf{X}$ and $\mathbf{W}$. In addition, the extra $\mathcal{O}(NK!)$ for alignment in the *Isolated SBM* is not needed here. Given the cluster assignments $\boldsymbol{X}$, we can easily estimate the cluster edge probabilities $\boldsymbol{\Theta}$ (see Eq.(3.5)). Algorithm 1 outlines the overall procedure for learning the *Joint SBM* . Table 3.1 compares its computational complexity to various baselines. We also include results on asymptotic properties of the estimates for the global parameter, $\widehat{\boldsymbol{\Theta}}$, and the membership matrix, $\widehat{\mathbf{X}}$ in B.5.

3.3 Comparing Joint SBM to Isolated SBM

Both *Joint SBM* and *Isolated SBM* use the eigendecompositions of the individual N graphs, and are closely related. The biggest difference is the $\boldsymbol{Z}_n^T\Delta_n^{-1}\Delta\mathbf{Z}$ term in the definition of $\widetilde{a}_n(\mathbf{X}_n, \mathbf{W})$ (Eq. (3.14)), which, intuitively, normalizes each individual graph eigenvector based on the distribution of nodes over the clusters in the graph. If the graphs are balanced, i.e., they have roughly the same proportion of nodes over clusters, then $\boldsymbol{Z}_n^T\Delta_n^{-1}\Delta\mathbf{Z} \approx \sqrt{|V_n|^{-1}|V|}$ which does not depend on the cluster sizes. Lemma 3.3.1 formalizes this.

Lemma 3.3.1 *Let the pair $(\boldsymbol{X}, \boldsymbol{\Theta})$ represent a joint-SBM with K communities for N graphs, where $\boldsymbol{X}$ is the stacked membership matrix over the N graphs and $\boldsymbol{\Theta}$ is full rank. Assume the graphs are balanced in expectation in terms of communities, i.e., assume the same distribution of cluster membership for all graphs: $\boldsymbol{X}_{ni} \overset{iid}{\sim}$ Mult$(\boldsymbol{\zeta})$ for all $n \in [1, ..., N]$ and $i \in [1, ..., |V_n|]$. Then, $\mathbb{E}\left[\boldsymbol{Z}_n^T\Delta_n^{-1}\Delta\mathbf{Z}\right] = \sqrt{|V_n|^{-1}|V|}$.*

When Lemma 3.3.1 is true, we have: $\mathbb{E}\left[\eta_n(\boldsymbol{X}_n, \boldsymbol{W})\right] = \|\boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^*\|_F^2 \frac{|V_n|K}{|V|} + \left\|2\left|\mathbf{Q}_n^*\right|\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2$. Now, in expectation, the RHS of Eq.(3.15) depends only on $\widetilde{a}_n(\boldsymbol{X}_n, \boldsymbol{W})$, since $\widetilde{b}_n(\cdot)$ no longer depends on $\boldsymbol{X}_n$. If the graphs are also of the same size, then for each graph, the objective function of *Joint SBM* is equivalent to that of *Isolated SBM*. Lemma 3.3.2 formalizes this.

Lemma 3.3.2 *Let the pair* $(\boldsymbol{X}, \boldsymbol{\Theta})$ *represent a joint-SBM with* K *communities for* N *graphs. If the sizes of the graphs are equal, i.e.,* $|V_n| = |V|/N$ *and the graphs are balanced in expectation in terms of communities, i.e., they have the same distribution of cluster membership* $\boldsymbol{X}_{ni} \overset{iid}{\sim} Mult(\boldsymbol{\zeta})$ *for all* $n \in [1, ..., N]$ *and* $i \in [1, ..., |V_n|]$. *Then,* $\mathbb{E}[\eta_n(\boldsymbol{X}_n, \boldsymbol{W})] \propto \|\boldsymbol{X}_n\boldsymbol{W}_n - \widehat{\boldsymbol{U}}_n\|_F^2$.

Lemma 3.3.2 illustrates the setting when clustering the nodes of each graph individually (Isolated) and jointly have the same solution (i.e., based on optimizing $\mathbf{Q}_n$). However, this is only true with respect to the node assignments for each individual graph. If we look to the global assignments, the Isolated and the Joint model are expected to be the same only up to permutations of the community labels. Thus, for a good global clustering result, the Isolated model needs an extra $\mathcal{O}(NK!)$ step to realign the estimated $\boldsymbol{X}_n$s, which can also introduce additional error. If the data does not have graphs of the same size or the distribution of clusters varies across graphs, then the Isolated model will likely miss some blocks on each graph. Our joint method avoids these issues by pooling information across the graphs to improve estimation. Next, we illustrate this effect using a toy example.

3.3.1 Toy data example

As an example to illustrate the effect of $\boldsymbol{Z}_n^T\Delta_n^{-1}\Delta\boldsymbol{Z}$ on the individual eigendecomposition, consider three graphs, where each graph is a village, nodes represent individuals and edges represent relationships between them. Assume that individuals are clustered in four different blocks based on their personalities, which reflects how they form relationships. Figure 3.1(left) shows the connectivity matrix based on those personalities. Also, consider that each village has its own distribution of people over the clusters, as shown, for instance, in Figure 3.1 (right).

Now, using the adjacency matrix eigendecomposition for each village $\mathbf{A}_n = \widehat{\mathbf{U}}_n\widehat{\mathbf{D}}_n\widehat{\mathbf{U}}_n^T$, we have $\widehat{\mathbf{Q}}_n = \widehat{\mathbf{U}}_n\widehat{\mathbf{D}}_n$ as a proxy for $\mathbf{Q}_n$. Since $\widehat{\mathbf{U}}_n\widehat{\mathbf{D}}_n\widehat{\mathbf{U}}_n^T = \widehat{\mathbf{T}}_n\widehat{\mathbf{D}}_n\widehat{\mathbf{T}}_n^T$ for $\boldsymbol{T}_n = -\boldsymbol{U}_n$, we consider $|\widehat{\mathbf{Q}}_n|$ instead. Figure 3.2 (left) shows the two largest absolute eigenvectors of each adjacency matrix. Each block has a different center of mass depending on which village

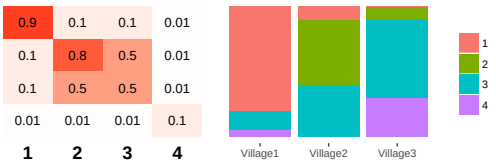

Figure 3.1.: Toy data connectivity matrix (left) and distribution of blocks per village (right).

Figure 3.2.: Original and transformed eigenvectors

(graph) it is in. This is due the fact that villages have completely different distribution of nodes over the blocks. Therefore, sharing information across villages is fundamental not only to assign underrepresented nodes to the correct block, but also to map the blocks across villages. Using our proposed transformation given in Equation (3.11), we obtain the results shown on Figure 3.2 (right). We re-scale and rotate the eigenvectors in order to have an embedding of the nodes that is closer to the global eigendecomposition. Therefore, using any clustering algorithm one can correctly recover the membership for nodes across the three villages.

3.4 Related work

Community detection in graphs has seen a lot of recent attention. We focus on extensions to multiple graphs, emphasizing two relevant directions: heterogeneity across communities and nonaligned graphs. For the former, [66] propose a degree corrected SBM to account for heterogeneity inside a community, though [70] showed that this fails to retrieve true communities in a high heterogeneous setting. [71] introduce a normalized Laplacian form that account for high heterogeneous scenarios, however this comes at a high computational cost.

There is also a large literature that focus on clustering multi-layered (also called multiplex) networks [28, 30–33]. Our work is in a different domain, specifically, we are interested in multiple exchangeable graphs, with no mapping (or alignment) either known or

possible, among nodes across graphs. Multi-layered methods explore observed dependencies (e.g., shared nodes) across graphs.

More generally, methodology for multiple graphs can be divided into geometric [3, 8, 46] and model-based [2, 4] approaches. The first approach seeks to characterize graphs topologically and explore hyperspace measures. [3] introduces a geometric characterization of the network using the so-called Fréchet mean while their approaces are mathematically elegant, they are substantially less flexible than our work. The second approach aims to embed graph to a lower dimension space without oversimplifying the problem by making use of latent models. [1, 2] proposed a mixture of latent space model to perform hypothesis testing on population of binary and weighted aligned graphs, respectively. On the other hand, [4] worked with non-aligned graphs, $\mathbf{X}_{ni} \neq \mathbf{X}_{n'i}$, modeling node features conditioned on its neighbors. They provided a convolutional neural network approach which is invariant to node permutation. However, unlike us, their method assumes knowledge of node features X_{ni}. Overall, the closest work to our model is from [72] where a Bayesian nonparametric inference method (IRM) for the scenario specified in Lemma 3.3.1 is used to make cross domain recommendation in bipartite graphs. Aside from being a sampling-based inference method, their method assumes balanced communities, and we show in our experiments that as heteregeneity increases the accuracy of their method decreases.

3.5 Experiments

3.5.1 Synthetic data

We generate data using the following generative process:

$$\begin{aligned} \boldsymbol{\pi}_n|\alpha &\sim \text{Dir}\left(K, 1/\alpha K\right), \\ X_{ni}|\boldsymbol{\pi}_n &\sim \text{Mult}(\boldsymbol{\pi}_n), \\ a_{nij}|\boldsymbol{X}_{ni}, \boldsymbol{X}_{nj}, (\theta_{kl})_{k=1,l=1}^{K,K} &\overset{ind}{\sim} \text{Bern}(\boldsymbol{X}_{ni}\boldsymbol{\Theta}.\boldsymbol{X}_{nj}^T). \end{aligned} \tag{3.19}$$

Unless specified, we use $K = 6$. We vary hyperparameters including N (number of graphs), $|V_n|$ (individual graph sizes), and α (cluster-size heterogeneity). Recall $\alpha \approx 0$ corresponds to a homogeneous setting (similar $\boldsymbol{\pi}$ across graphs), and that α increases the cluster-heterogeneity.

Experiments design: We assess community retrieval performance, and global $\boldsymbol{\Theta}$ estimation for each model. We design two sets of experiments, one for each assessment objective:

1. Community retrieval $\widehat{\boldsymbol{X}}_n$:
   (a) Fixed $N = 1000$ and $\alpha = 1$. We generate datasets for varying $|V_n| = [25, 50, 100, 200]$.
   (b) Fixed $N = 100$ and $|V_n| = 200$. We generate datasets for varying values of $\alpha \in [0.1, 2]$;
   (c) Fixed $N = 200$, $|V_n| = 500$ and $\alpha = 1$. We assess performance as runtime increases.
   (d) Fixed $K = 2$, $\boldsymbol{\pi} = [1/2, 1/2]$, $N = 2$ and $|V_1| = 500$. We assess performance for varying $|V_2| = [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]$.
2. Global $\widehat{\boldsymbol{\Theta}}$:
   (a) Fixed $\alpha = 1$. $N = [50, 200, 400, 600, 800, 1000]$. For each N, $|V_n| = [25, 50, 200, 500]$.

Appendix B.2.1 includes additional experiments evaluating cluster retrieval for varying graph sizes.

Baselines: We evaluate our joint spectral clustering algorithm (*JointSpec*), and compare against:

- *IsoSpec*: separately running spectral SBM on the individual graphs and then aligning. We used the `Blockmodels` R package [73].
- `node2vec` [74]: embeds the nodes into a low-dimensional vector space, clusters the embeddings, and then aligns. We used a Python implementation [74].

- `ReMatch` [72]: an IRM-based approach to do cross domain recommendation using bipartite graphs. This is a sampling version of Lemma 3.3.1.

For *IsoSpec* and `node2vec`, which are nonaligned, we consider two re-alignment procedures:

- `perm`: (1) fix a pivot connectivity, $\mathbf{\Theta}_{\text{pivot}}$, of the graph with the largest the number of nodes in each community, (2) search a permutation of $\mathbf{\Theta}_n$ that best approximates $\mathbf{\Theta}_{\text{pivot}}$ for each graph, then (3) re-order the connectivity matrix and membership accordingly.
- `km`: (1) cluster the centers W_n across graphs, then (2) re-order the connectivity matrix and membership accordingly.

Evaluation: We measured performance using cluster retrieval performance measures (CRPMs) and Standardized Square Error (SSE). For the CRPMs, we used Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and $1-$ misclustering rate (MCR). We measure individual graph CRPMs as well as the overall CRPMs across all graphs, in each case, comparing the estimated membership $\widehat{\mathbf{X}}_n$ with the ground truth $\mathbf{X}_n$. To measure the quality of the estimated connectivity matrix $\mathbf{\Theta}$, we used the standardized square error, the square error normalized by the true variance. Thus, $SSE = \sum_k^K \sum_l^K \frac{(\widehat{\theta}_{kl}-\theta_{kl})^2}{\theta_{kl}(1-\theta_{kl})}$. We normalize the square error by the true variance because very high (or very low) edge probabilities have lower variances and need to be up-weighted accordingly. For good $\mathbf{\Theta}$ estimates, we expect this to be close to zero.

Results for community retrieval $\widehat{\mathbf{X}}_n$, fixed $\alpha = 1$: Figure 3.3(left) shows the CRPMs curves for increasing number of nodes (top row: individual CRPMs, and bottom row: overall CRPMs). For individual CRPMs, each data point records the median over the graphs and the shaded region shows the interquartile range. We see that *JointSpec* outperformed *IsoSpec*, `node2vec` and `ReMatch` for both overall and individual CRPMs. That the overall CRPMs for *IsoSpec* and `node2vec` is poor is unsurprising, given the two-stage alignment procedure involved. Interestingly however, they perform poorly on the individual CRPMs as well. This indicates that the use of *only* local information is not enough to

accurately assign nodes to clusters, and that it is important to pool statistical information across graphs. `ReMatch` is a Bayesian nonparametric model that creates more communities as the graphs become more heterogeneous. Unlike NMI, ARI and $1-$MCR capture the fact that `ReMatch` is assigning nodes to very large number of communities (> 50 on average). Fig. B.1 in Appendix B.2.1 shows results for datasets with $N = [50, 200, 600]$. Overall, the results are similar to Fig. 3.3(left). See Appendix B.2.2 for a qualitative assessment of the connectivity matrix.

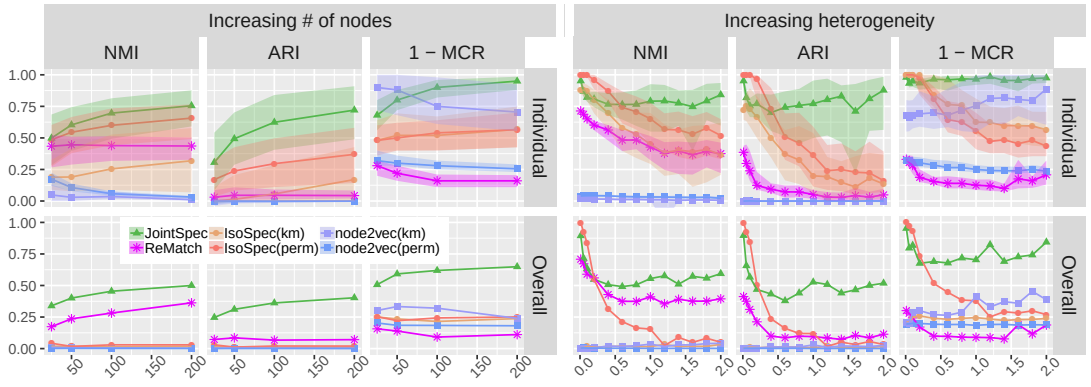

Figure 3.3.: **(Left)** Fixed $N = 1000$ and α: CRPMs for increasing number of nodes (25, 50, 100 and 200). **(Right)** Fixed $N = 100$ and $|V_n| = 200$: CRPMs curves for increasing heterogeneity (α). Top row: median and the interquartile range curves of individual CRPMs. Bottom row: overall CRPMs.

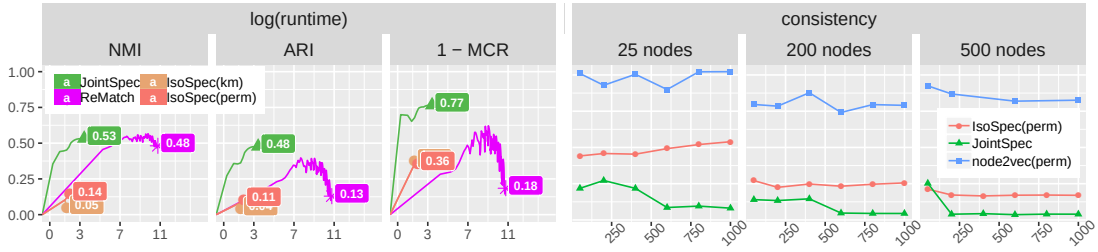

Figure 3.4.: **(Left)** Fixed $N = 200$, $|V_n| = 500$ and $\alpha = 1$: CRPMs for increasing log(runtime) of each method. **(Right)** Standardized square error of $\mathbf{\Theta}$ (SSE) for increasing number of graphs N, for $|V_n| = 25, 50, 200, 500$.

Table 3.1.: Asymptotic computational complexity for each method

Inference method	Algorithm step		
	Initialization	Community assignment	Re-alignment
IsoPerm	$\mathcal{O}\left(K\sum_n^N \lvert V_n\rvert^2\right)$	$\mathcal{O}\left(\lvert V\rvert K\right)^*$	$\mathcal{O}\left(NK!\right)$
IsoKM	$\mathcal{O}\left(K\sum_n^N \lvert V_n\rvert^2\right)$	$\mathcal{O}\left(\lvert V\rvert K\right)^*$	$\mathcal{O}\left(NK^2\right)^*$
JointSpec	$\mathcal{O}\left(K\sum_n^N \lvert V_n\rvert^2\right)$	$\mathcal{O}\left(\lvert V\rvert K\right)^*$	0
`ReMatch`	$\mathcal{O}\left(K\sum_n^N \lvert V_n\rvert^2\right)$	$\mathcal{O}\left(K\sum_n^N \lvert V_n\rvert^2\right)^*$	0

* per iteration/sample.

Runtime analysis: Figure 3.4(left) shows $\log$-performance against runtime for each method and each measure. All runs were on a Macbook Pro 2.3 GHz Intel Core i7, 8gb 1600 MHz DDR3. *JointSpec* clearly outperforms the baselines, taking much less time to converge to a good solution. Moreover, we compare the asymptotic complexity of our algorithm with some baselines in Table 3.1. Overall, *JointSpec* has the best scalability for increasing K, N and $|V_n|$. See Appendix B.2.1 for experiments using $\alpha = .1, 2$.

Results for community retrieval $\widehat{\mathbf{X}}_n$, fixed $N = 100$ and $|V_n| = 200$: Here, we evaluate cluster retrieval performance as the cluster sizes became more heterogeneous (e.g. α increases). Figure 3.3(right) shows the CRPMs curves. The results suggest that *IsoSpec*(`perm`) and *JointSpec* have similar performance when the graphs are balanced in terms of distribution of nodes over communities (low values of α). However, the CRPMs curves diverge as α increases and for more heterogeneous settings (unbalanced graphs), the Joint model outperforms *IsoSpec* `node2vec` and `ReMatch` by a large amount, both for overall and individual cluster retrieval performance. `ReMatch` increases the number of communities as the graphs becomes more heterogeneous, ARI and $1-$MCR capture this.

Results for community retrieval $\widehat{\mathbf{X}}_n$, fixed $N = 2$: We consider a simplified connectivity matrix $\mathbf{\Theta} = [[.9, .1], [.1, .5]]$ in order to focus the assessment of varying graphs sizes. The results in Fig.3.5 shows *JointSpec* and *IsoSpec* (overlapping in Fig 3.5) performances are not affected by varying graph sizes when community proportions are the same, as oppose

to `ReMatch`. We also investigated more complex settings where the individual graph sizes are sampled from an overdispersed negative binomial distribution. The results are in Appendix B.2.1 and it coincides with what is shown in Fig.3.5. Overall, we found that varying α had more impact on cluster retrieval performance than varying the spread of graph sizes.

Results for global $\widehat{\mathbf{\Theta}}$: Figure 3.4(Right) shows the standardized square error (SSE) for the $\mathbf{\Theta}$ estimates (lower values mean better estimates of the true $\mathbf{\Theta}$). Again, *JointSpec* outperforms *IsoSpec* and `node2vec` in all scenarios. Even when each graph does not have many nodes (e.g. $|V_n| = 25$), statistical pooling allows *JointSpec* to achieve good performance that is comparable with settings with larger nodes. These results also illustrate the consistency of our estimation scheme, with decreasing error as the number of samples increases. We do not compare with `ReMatch`, which produces a connectivity matrix significantly larger than K.

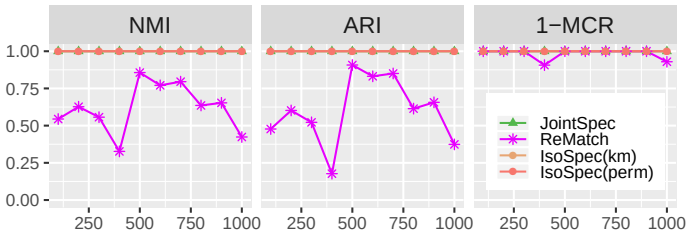

Figure 3.5.: Fixed $N = 2$, $|V_1| = 500$, $K = 2$ and $\boldsymbol{\pi} = [1/2, 1/2]$: CRPMs for increasing $|V_2|$ of each method.

3.5.2 Real world data

Karnataka village dataset Available at `https://goo.gl/Vw66H4`, this consists of a household census of 75 villages in Karnataka, India. Each village is a graph, each person is a node, and edges represent if one person went to another's house or vice versa. Overall, we have 75 graphs with size from 354 to 1773 nodes. As ground truth is unknown, we cannot assess the performance of the models precisely. To quantitatively measure performance,

we cluster religion, caste and mother-tongue and compare with the communities assigned using each model. We expect these demographic variables are related to the connectivity behavior of individuals. We consider a conservative setting of six communities ($K = 6$) for all models. Table 3.2 shows the NMIs and ARI computed in this setting, we can see that *JointSpec* has a larger NMI and ARI than the baselines.

MovieLens dataset This data consists of a bipartite graph of 943 users and 1682 movies, with 100k ratings. We aim to assess cross-domain recommendation using the learned models. While MovieLens is a single domain, we use it to simulate multiple domains (e.g., Netflix, Amazon Prime, and Hulu) by sampling a collection of graphs. Specifically, we sample a set of users and movies to create a single subgraph. Each subgraph is sampled from the full set of data so a user may appear in multiple subgraphs. We fix $N = 5$ and learn the models from the set of sampled subgraphs. We consider two scenarios: 1) homogeneous where each graph has the same number of users (171) and movies (841), and 2) heterogeneous where the size of graphs vary (users: 30-460, movies: 32-820). Any user/movie pair that is not added to a subgraph sample is held out for the test set. For each model, we estimate the connectivity matrix from the sampled subgraphs and use it to compute the probability of ratings in the test set (i.e., whether a user has rated a movie or not) as a proxy for cross-domain recommendations. Table 3.2 reports prediction performance in terms of area under the ROC curve (AUC) and precision-recall (PR). The results show that *JointSpec* is more precise and outperforms the baselines for both the homogeneous and heterogeneous scenarios. Note that AUC and PR performance for the *Isolated SBM* and `node2vec` is equivalent to random guessing.

Twitter We collected tweets from April 6th to May 31st 2016 from hashtags of 7382 users from both sides of the political crisis in Brazil, where one side was for the impeachment of the former president, Dilma Rousseff, and the other described the process as a coup. We constructed word co-occurrence graphs by forming edge-links between words which were co-tweeted more than 20% of the time for a given user. Thus, networks represent users and nodes represent words, with sizes varying from 25 to 1039 nodes. Here,

Table 3.2.: Overall CRPMs (NMI and ARI) for village dataset and cross domain recommendation performance for MovieLens dataset.

Model	Village dataset		MovieLens dataset			
			Homogeneous		Heterogeneous	
	NMI	ARI	AUC	PR	AUC	PR
JointSpec	**0.043**	**0.034**	**0.867**	**0.781**	**0.608**	**0.594**
ReMatch	0.00	0.00	0.832	0.760	0.556	0.546
IsoSpec(km)	0.018	−0.007	0.469	0.458	0.469	0.458
IsoSpec(perm)	0.018	0.016	0.469	0.458	0.469	0.458
node2vec(km)	0.004	0.003	0.469	0.458	0.469	0.458
node2vec(perm)	0.008	0.003	0.469	0.458	0.469	0.458

Table 3.3.: Top 3 words assigned to communities by each model (pro and against government)

K	JointSpec		IsoSpec	
	Pro	Against	Pro	Against
1	naovaitergolpe golpe dilmafica	foradilma forapt dilma	naovaitergolpe golpe foradilma	foradilma forapt dilma
2	hora galera democralica	janaiva compartilhar faltam	turno venceu anavilarino	arte objetos apropriou
3	sociais coxinha naonaors	mito elite compartilhar	rua april continuar	rua coxinha elite
4	vai brasil povo	lulanacadeia lula vai	dilmafica dilma forapt	impeachment brasil lulanacadeia

we used four communities ($K = 4$), and we show in Table 3.3 the top three words assigned to communities by the *JointSpec* and *IsoSpec* models differ significantly. We color words based on whether it is more pro government (red), against (blue) or neutral (black). For *JointSpec*, we see that community 1 has important key arguments per side such as "naovaitergolpe" (no coup) and "foradilma"(resign dilma). We also see that community 2 seems to have some stopwords such as "hora"(time) and "compartilhar" (share), and community

3 consist of aggressive and pejorative terms each side uses against the other like “elite” and “coxinha”. For *IsoSpec*, the words do not seem to reflect a clear pattern.

3.6 Conclusion

We consider the problem of multiple graph community detection, and proposed a novel spectral clustering algorithm to solve this task. Our results show that we need to jointly perform stochastic block model decompositions in order to be able to estimate a reliable global structure. We compared our method with *Isolated SBM*, `node2vec`, and a Bayesian community detection method for multiple bipartite graphs called `ReMatch`. Our method outperformed the baselines on global measures (overall CRPMs and SSE of the connectivity matrices), but interestingly also on local measures (individual CRPMs). This demonstrates that our method is more accurately able to assign nodes to clusters regardless of the choice of re-alignment procedure. Overall, the other methods do not pool global information in the inference step which indicates that they can only be used in homogeneous scenarios.

4. ZERO INFLATED GRAPHS

4.1 Introduction

Community detection and collaborative filtering on networks has received a lot of attention in the last several years. In terms of highly sparse graphs, the main perspectives are: preferential attachment [75], degree corrected stochastic blockmodel (DCSBM) [66, 76], exchangeable sparse graphs [12, 38]. In this chapter, we explore a different angle, focusing on graph settings where there is a 'missingness' mechanism obfuscating potential interaction between nodes. This perspective has recent received attention specially in [41–43] where the general generative process of interactions between nodes i and j for all $i, j \in V \times V$, where V is the set of nodes, is given by

$$\begin{aligned} X_i \sim \text{Multi}(\boldsymbol{\pi}) \qquad & \mathcal{L}_{ij}|\phi_{ij} \sim \text{Bern}(\phi_{ij}) \\ \boldsymbol{B}[i,j] := b_{ij}|X_i, X_j, \boldsymbol{\mu} \sim \eta\left(X_i \boldsymbol{\mu} X_j'\right) & \\ \boldsymbol{A}[i,j] := a_{ij} = \mathcal{L}_{ij} b_{ij} & \end{aligned} \tag{4.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, where $n = |V|$, is the observed adjacency matrix with n nodes, $\boldsymbol{\mu}$ is the $K \times K$ connectivity matrix, X_i and X_j are the one-hot vector $(1 \times K)$ representing the cluster membership of nodes i and j, respectively, and η represents some distribution (e.g. Poisson, Normal, Bernoulli, Zipf, and so on).

This process can lead to very challenging inference schemes. Spectral clustering, for instance, assumes that the observed adjacency matrix $\boldsymbol{A}$ is a good proxy for the mean connectivity $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ (i.e., $\mathbb{E}[\boldsymbol{A}] = \boldsymbol{P}$). This means in cases where there is missing values in the observed adjacency matrix the overall bias between $\boldsymbol{A}$ and $\boldsymbol{P}$ can be very large leading the great inaccuracies. Proposed solutions to this issue use either k-means regularization ([41]) or a Bayesian framework [42, 43]. In both cases, there are specific distribution assumptions and structures on the missing mechanism in order to facilitate the

inference scheme. [41] proved consistency and of the connectivity estimation for when $\phi_{ij} = \phi$ for all i and j. In this work, we aim to efficiently and accurately assign nodes to cluster, and, also, relax the assumption of structured ϕ_{ij}. We also aim to tackle a more general problem: the Zero inflated case *de facto*. Precisely, integrating out $\boldsymbol{\mathcal{L}}$ from Eq. (4.1), we have

$$a_{ij}|X_i, X_j, \boldsymbol{\mu}, \phi_{ij} \sim \text{ZeroInf}_\eta \left(X_i \boldsymbol{\mu} X_j', \phi_{ij}\right) \tag{4.2}$$

This problem has been studied in a Bayesian perspective at [43]. They used a variational inference for binary data, their inference scheme included some structure assumptions on ϕ_{ij}. We focus on spectral clustering methods which have been proven to be very intuitive, efficient and accurate [63]. Most importantly, spectral clustering requires less parametric assumptions. In this work, we propose two inference schemes based on spectral clustering to deal with Zero inflated settings: self-similar and ego-nets. The former focus on settings where the missing mechanism is known $\boldsymbol{\mathcal{L}}$ and the later is used when there is no knowledge about $\boldsymbol{\mathcal{L}}$.

Our proposed methods outperforms the baselines in terms of cluster retrieval and connectivity matrix estimation in synthetic data settings. We also compare our inference schemes with the baselines in two real-world datasets: U.S. and France Political blogs. U.S. political blogs dataset is commonly used in DCSBM works, we included the France version introduced at [77]. We also included in Appendix C.7 a clicks on news articles experiment. This was introduced at [78] and it is based on clicks on news article of the largest Brazilian news portal, and we show that without using specific knowledge of the news articles our inference scheme was able to correctly predict users clicking behavior, comparable to more advance recommender methods that uses additional covariates.

4.2 Background

We first review the complete case SBM, and how inference is carried-out via spectral clustering.

4.2.1 Complete data blockmodel (SBM)

Consider an interaction data $\boldsymbol{A}$ (adjacency matrix) with n nodes, and an underlying community-based model that governs connectivity intensity between nodes i and j. Precisely, we define $\mu_{ij} := \mu_{kl}$ as the interaction intensity rate between nodes i and j in clusters $k, l \in [0, ..., K]$, respectively. We also define $\boldsymbol{\pi} = [\pi_1, ..., \pi_K]$ as the K dimension vector where the k-th element represents the probability of node i to be in cluster k. Write $\boldsymbol{X}$ as the $n \times K$ matrix of stacked membership vectors where X_i and X_j are the rows i and j representing the one-hot vector $(1 \times K)$ of the cluster memberships, respectively. Thus, $\boldsymbol{A} \sim \text{SBM}(\boldsymbol{\mu}, \boldsymbol{\pi})$ is given by

$$\begin{aligned} X_i &\sim \text{Multi}(\boldsymbol{\pi}) \\ \boldsymbol{A}[i,j] := a_{ij} | X_i, X_j, \boldsymbol{\mu} &\sim \eta\left(X_i \boldsymbol{\mu} X_j'\right) \end{aligned} \tag{4.3}$$

where η is some random variable (e.g., Gaussian, Bernoulli, Poisson, Binomial, Zipf, so on). Next, we describe how in order to estimate $\boldsymbol{X}$ and $\boldsymbol{\mu}$. Inference on Eq.(4.3) can be carried-out with different perspectives and assumptions, the most common ones are: likelihood [66, 79, 80], Bayesian [28, 81], and spectral clustering [63, 71]. Both likelihood and Bayesian approaches needs a full distributional assumption on the data (e.g., $\boldsymbol{\pi}$, $\boldsymbol{\mu}$), estimation of the parameters are done either using EM or MCMC. In this work we focus on spectral clustering since it is the family of models with the least number of conditions where the main assumption is $\mathbb{E}[\boldsymbol{A}] = \boldsymbol{X}\boldsymbol{\mu}\boldsymbol{X}^T - \text{diag}(\boldsymbol{X}\boldsymbol{\mu}\boldsymbol{X}^T)$. For more details, see Section 3.2.2.

4.2.2 Degree corrected SBM (DCSBM)

The degree Corrected Stochastic blockmodel (DCSBM) [66] is an extension of the regular SBM which allows nodes to have heterogeneous degree within community, i.e., it

allows nodes to have a specific heterogeneity parameter that also affects the connectivity. Formally, we generate $\boldsymbol{A} \sim \text{DCSBM}\left(\boldsymbol{\mu}, \boldsymbol{\pi}, \{\phi_{ij}\}_{(ij)\in V\times V}\right)$ as

$$\begin{aligned} X_i &\sim \text{Multi}(\boldsymbol{\pi}) \\ \boldsymbol{A}[i,j] := a_{ij} | X_i, X_j, \boldsymbol{\mu}, \phi_{ij} &\sim \eta\left(\phi_{ij} X_i \boldsymbol{\mu} X_j'\right) \end{aligned} \tag{4.4}$$

This is a highly flexible approach and it has been proposed many different inference schemes. The main difficulty is related to identifiability, thus some additional constraints are needed in order to perform inference.

4.2.3 Spectral clustering

Now, we recall the spectral clustering method for SBMs [63, 67]. Let $\boldsymbol{P}$ refer to the edge probability matrix of the graph under an SBM, where $\boldsymbol{P} = \boldsymbol{X}\boldsymbol{\Theta}\boldsymbol{X}^T$. Let $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ be the eigendecomposition of $\boldsymbol{P}$, we have

$$\begin{aligned} \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = \boldsymbol{P} = \boldsymbol{X}\boldsymbol{\mu}\boldsymbol{X}^T = \boldsymbol{X}\Delta^{-1}\Delta\boldsymbol{\mu}\Delta\Delta^{-1}\boldsymbol{X}^T \\ = \boldsymbol{X}\Delta^{-1}\boldsymbol{H}\tilde{\boldsymbol{D}}\boldsymbol{H}^T\Delta^{-1}\boldsymbol{X}^T \end{aligned} \tag{4.5}$$

where $\Delta = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{1/2}$ define a $K \times K$ diagonal matrix with entries $\sqrt{|G_k|}$ and G_k is the number of nodes in cluster k. Since $\boldsymbol{D}$ and $\tilde{\boldsymbol{D}}$ are both diagonal, and $\boldsymbol{X}\Delta^{-1}$ and $\boldsymbol{H}$ are both orthonormal,

$$\boldsymbol{D} = \tilde{\boldsymbol{D}}, \qquad \boldsymbol{U} = \boldsymbol{X}\Delta^{-1}\boldsymbol{H}. \tag{4.6}$$

In summary, we use the observed adjacency matrix $\boldsymbol{A}$ as a proxy for $\boldsymbol{P}$, and replace $\boldsymbol{U}$ in Eqs. (4.5) and (4.6) with $\widehat{\boldsymbol{U}}$ calculated from the eigendecomposition $\boldsymbol{A} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{D}}\widehat{\boldsymbol{U}}^T$. The memberships are given by solving for the following optimization problem:

$$\left(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{W}}\right) = \underset{\substack{\boldsymbol{X}\in\mathcal{M}_{n,K} \\ \boldsymbol{W}\in\mathbb{R}^{K\times K}}}{\arg\min} \|\boldsymbol{X}\boldsymbol{W} - \widehat{\boldsymbol{U}}\|_F^2 \tag{4.7}$$

For the case of DCSBM, it is clear that as within community heterogeneity increases, using $\boldsymbol{A}$ as a proxy for $\boldsymbol{P}$ becomes a problem. In fact, [63] proposed an efficient algorithm to perform community detection on Degree Corrected SBM with few assumptions (and constraints). They assumed $\phi_{ij} = \nu_i\nu_j$ and $\mathbb{E}[\boldsymbol{A}] = \boldsymbol{P} - diag(\boldsymbol{P})$ where $\boldsymbol{P} = diag(\boldsymbol{\nu})\mathbf{X}\mu\mathbf{X}^T diag(\boldsymbol{\nu}) = \tilde{\mathbf{X}}\mathbf{H}D\mathbf{H}^T\tilde{\mathbf{X}}^T$, and $\mathbf{H}D\mathbf{H}^T$ is the eigendecomposition of $\tilde{\boldsymbol{\nu}}\boldsymbol{\mu}\tilde{\boldsymbol{\nu}}$ and $\tilde{\nu}_i := \sum_{k=1}^{K} \frac{\nu_i}{\sum_{m=1}^{K}\nu_m^2}$. Under their assumptions, the rows have K distintic directions, instead of having K distinct values. Using this fact, they proposed a spherical (normalize rows) K-median algorithm to cluster the rows of the K largest eigenvectors of $\boldsymbol{P}$. Their method assumes a rank one on the structure of the node heterogeneity $\phi_{ij} = \nu_i\nu_j$, and they also indirectly assumes that the connectivity matrix $\boldsymbol{\mu}$ is modular (i.e., high connectivity within community and very low across communities) since the matrix $\tilde{\mathbf{X}}\mathbf{H}$ is orthogonal (i.e., equals the eigenvector matrix of $\boldsymbol{P}$) only if heterogeneity $\boldsymbol{\nu}$ is within community.

Next, we describe a pre-processing step that can further reduce the variance in $\boldsymbol{A}$ and improve cluster retrieval.

4.3 Self-similar spectral clustering

The use of $\boldsymbol{A}$ as a proxy for $\boldsymbol{P}$ is usually an issue when $\boldsymbol{A}$ becomes highly heterogeneous [70]. The heterogeneity may be due withing community heterogeneity (studied in DCSBM), large variance on the elements of $\boldsymbol{A}$, small sample size, large inflation of zeros and so on. In real-world settings, we usually have a combination of different sources of heterogeneity. We aim to reduce the overall noise by performing spectral clustering on a pre-estimator of $\boldsymbol{P}$ where each element has lower variance than the same element in $\boldsymbol{A}$.

In genetic statistics there is large literature focusing on improving genetic models by using self-similarity matrices [82–85]. Those are two-stage approaches by first computing similarities , and then perform clustering or other analysis on the data. In our case, we focus on SBMs and we define a two-step approach that reduces the overall variance by using the

data to estimate $\boldsymbol{P}$ (where $P_{ij} = \mathbb{E}[a_{ij}]$). Our proposed estimator for the interactions of nodes i and j is given by

$$\widehat{P}_{ij} := \left(\mathbf{1}^T \boldsymbol{W}_j^{\setminus i} \mathbf{1}\right)^{-1} \mathbf{1}^T \boldsymbol{W}_j^{\setminus i} a_{i*} \tag{4.8}$$

where $\boldsymbol{W}_j^{\setminus i} = \text{diag}\left(\{\omega_{aj}\}_{a \in V \setminus i}\right)$ and ω_{aj} indicates similarity strength of nodes a and j. In words, each element of our estimator $\widehat{\boldsymbol{P}}$ is a weighted mean of the rows in the adjacency matrix $\boldsymbol{A}$ where we scale the interactions by a similarity measure.

Intuitively, for a given interaction a_{ij} there are (plenty) of similar interactions that node i (or j) has with other nodes. Thus, we aim to estimate the expected value of a_{ij} by averaging nodes that behave similarly to node j (or i). Lemma 4.3.1 shows that if we choose the similarity measure to be the correlation of the connectivity then $\widehat{P}_{ij}$ is unbiased and converges to $\boldsymbol{P}$ almost surely.

Lemma 4.3.1 *Let $\boldsymbol{A}$ be a SBM generated from Eq.* (4.3)*. Write V as the set of nodes and $|V| = n$. For convenience, let η represents the normal distribution where $\boldsymbol{\mu}$ is a fixed $K \times K$ matrix of the means, and $\boldsymbol{\Sigma}$ is a fixed $K \times K$ matrix of variances. Write μ_{kl} and σ_{kl}^2 as the elements k and l in the matrices $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. Let $\boldsymbol{\pi}$ be the distribution of nodes over clusters where $\boldsymbol{\pi}[k] = \pi_k > 0 \; \forall k = 1, ..., K$, and X_i and X_j be the one-hot vector $(1 \times K)$ of the cluster memberships of nodes i and j, respectively. Moreover, write G_k as the set of nodes in cluster k, and assume $i \in G_k$ and $j \in G_l$. Thus, we re-write Eq.* (4.8) *as*

$$\widehat{P}_{ij} := \left(\mathbf{1}^T \boldsymbol{W}_j^{\setminus i} \mathbf{1}\right)^{-1} \mathbf{1}^T \boldsymbol{W}_j^{\setminus i} a_{i*} = \frac{\sum_{u \in V \setminus i} \rho_{ju} a_{iu}}{\sum_{u \in V \setminus i} \rho_{ju}} \tag{4.9}$$

where ρ_{ab} is the correlation of the connectivity between nodes a and b for any $a, b \in V$. Hence,

$$\mathbb{E}\left[\widehat{P}_{ij}\right] = \mu_{kl} \tag{4.10}$$

$$\widehat{P}_{ij} \underset{n \to \infty}{\overset{a.s.}{\to}} \mu_{kl} = P_{ij} \tag{4.11}$$

where $\xrightarrow{a.s.}$ *denotes strong convergence.*

See proof in Appendix C.1. While we do not observe $\boldsymbol{X}$ and consequently we cannot compute $\boldsymbol{W}_j^{\setminus i}$, we can efficiently estimate it. Generally, the choice of $\widehat{\boldsymbol{W}}_j^{\setminus i}$ depends on specific goals and available information. Here, we propose to estimate the correlation of nodes j and u, ρ_{ju} in Eq. (4.9) using the correlation coefficient of the connectivity information.

$$\widehat{P}_{ij}^{*} := \widehat{a}_{i\cdot}(j) := \left(\mathbf{1}^T \widehat{\boldsymbol{W}}_j^{\setminus i} \mathbf{1}\right)^{-1} \mathbf{1}^T \widehat{\boldsymbol{W}}_j^{\setminus i} a_{i*} = \frac{\sum_{u \in V \setminus i} \widehat{\rho}_{ju} a_{iu}}{\sum_{u \in V \setminus i} \widehat{\rho}_{ju}} \tag{4.12}$$

Lemma 4.3.2 shows that, as oppose to a_{ij}, $\widehat{a}_{i\cdot}(j)$ is (weakly)consistent to P_{ij}.

Lemma 4.3.2 *Let* $\boldsymbol{A}$ *be a* SBM *generated from Eq.* (4.3) *where the conditions of Lemma 4.3.1 hold. Hence, define*

$$\widehat{\boldsymbol{W}}_j^{\setminus i} := \textit{diag}\left(\{\widehat{\rho}_{ju}\}_{a \in V \setminus i}\}\right) \tag{4.13}$$

$$\textit{where } \widehat{\rho}_{ju} = \frac{\widehat{Cov}\left(a_{j*}, a_{u*}\right)}{\sqrt{\widehat{Var}(a_{j*})}\sqrt{\widehat{Var}(a_{u*})}} \tag{4.14}$$

Thus, for $\widehat{a}_{i\cdot}(j)$ *defined in Eq.* (4.12)*, we have*

$$\widehat{a}_{i\cdot}(j) \underset{n \to \infty}{\xrightarrow{P}} \mu_{kl} = P_{ij} \tag{4.15}$$

where $\xrightarrow{P}$ *denotes convergence in probability.*

See appendix C.2 for proof. Write $\widehat{\rho}_{aj}^{*}$ as a general estimator of $\rho_{ju} = \mathbb{I}_{X_u = X_j}$ in $\boldsymbol{W}_j^{\setminus i} = \text{diag}\left(\{\rho_{ju}\}_{a \in V \setminus i}\}\right)$. Thus, if one proves that $\widehat{\rho}_{ju}^{*}$ converges in probability to ρ_{ju}, then the result of Lemma 4.3.2 follows. We emphasize that the correlation coefficient estimator $\widehat{\boldsymbol{W}}_j^{\setminus i}$ in Eq. (4.13) is *one* out of many options to use for the similarity measure. However, not all the options for similarity guarantee convergence in probability of $\widehat{a}_{i\cdot}(j)$ to the true P_{ij}. For instance, one can estimate ρ_{ju} as $\widehat{\rho}_{ju}^{\text{stacked}} = \mathbb{I}_{\widehat{X}_u = \widehat{X}_j}$ where $\widehat{X}_u$ and $\widehat{X}_j$ are the estimated cluster memberships of nodes a and j using spectral clustering. This is

a two-stage clustering process, but in this case $\widehat{\rho}_{ju}^{\text{stacked}}$ is not consistent to the true ρ_{ju} as n increases. Lemma 4.3.3 formalizes this statement.

Lemma 4.3.3 *Let $\boldsymbol{A}$ be a SBM generated from Eq.* (4.3) *where the conditions of Lemma 4.3.1 hold. Define*

$$\widehat{\rho}_{ju}^{\textit{stacked}} = \mathbb{I}_{\widehat{X}_u = \widehat{X}_j} \tag{4.16}$$

where $\widehat{X}_u$ and $\widehat{X}_j$ are the estimated cluster memberships of nodes a and j using spectral clustering described in Section 4.2.3. Thus,

$$\widehat{\rho}_{ju}^{\textit{stacked}} \underset{n\to\infty}{\overset{P}{\not\to}} \rho_{ju}. \tag{4.17}$$

See appendix C.3 for proof. In practice, the use of external covariates to estimate the similarities can further reduce the noise in $\boldsymbol{A}$.

We define the self-similar estimator of $\boldsymbol{P}$ as $\widehat{\boldsymbol{P}}^*$ where $\widehat{\boldsymbol{P}}^*[i,j] = \widehat{a}_{i\cdot}(j)$, and we illustrate its behavior by performing a simple experiment. We compared the regular spectral clustering (`reg_spec`), the self-similar using Eq. (4.16) (`self_stacked`), the DCSBM (`dc_spec`) and the proposed self-similar using Eq. (4.14) (`self_similar_spec`). See Appendix C.4 for setup details on this experiment. Figure 4.1 shows the cluster retrieval performance (ARI) for increasing sample size N in low and high $Var(a_{ij})$ settings. Each data point is the mean and the band represents the 95% confidence interval of 30 samples. In high variance settings the self-similar is significantly better than the traditional spectral clustering regardless of the sample size. Nonetheless, for low variance settings, the performances are not statistically different as graph size increases.

4.3.1 Limitations

While $\widehat{\boldsymbol{P}}^*$ (weakly)consistent $\boldsymbol{P}$ and $\boldsymbol{A}$ it is not, it is not advised to use it in some settings. Mainly, estimating the correlation coefficient of all pair of nodes is computationally costly (at least $\mathcal{O}(n^{2.73})$ in speed up procedures for matrix multiplications). Thus, the

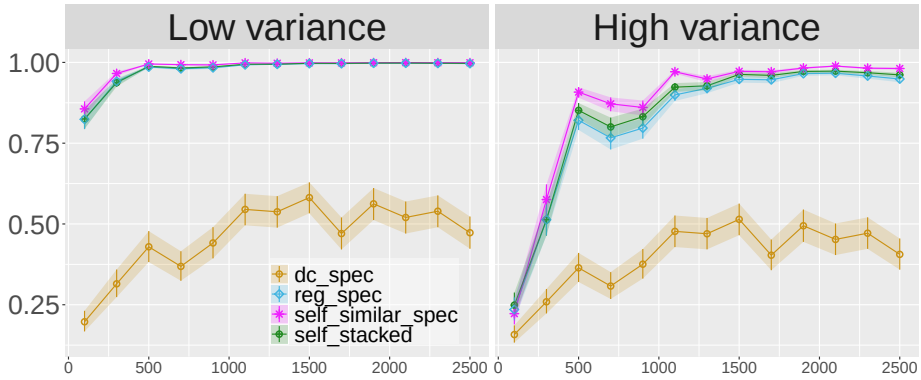

Figure 4.1.: Restuls for cluster retrieval performance (ARI) for complete data in low variance (left) and high variance (right) settings. X-axis represents increasing graph size.

self-similar *should not* be used in settings where communities connectivities are highly separable and there is not much heterogeneity of any source. Nonetheless, in zero inflated settings (described next), one can further reduce the computational cost by utilizing the fact that the matrix is sparse. Moreover, we discuss on the next section how this self-similar procedure compare to other approaches when there is a missing mechanism obfuscating potential interaction between nodes.

4.4 Zero inflated blockmodel (ZinfSBM)

Graph sparsity has been studied in different perspectives, e.g. exchangeable sparsity [12, 38], preferential attachment [75], and most common degree corrected SBM (DCSBM) [63, 66]. In these works, the sparsity is mostly due to naturally caused phenomena (random sparsity), such as time, geographic locations, and so on. Here, we consider a broader view of this problem by seen as a zero inflated setting where there is missing mechanism governing the unseen interactions. The generative process shown in Eq. (4.1) might be Missing-at-Random (MAR) (i.e., $\boldsymbol{\mathcal{L}}_{ij} \perp\!\!\!\perp (X_i, X_j)$), or it can be targeted Missing-Not-at-Random (MNAR) (i.e., $\boldsymbol{\mathcal{L}}_{ij} \not\perp\!\!\!\perp (X_i, X_j)$) such as in recommender systems where some users cannot see other users or products based on their connectivity behavior. In this work, we focus on the former. The missing mechanism, $\boldsymbol{\mathcal{L}}$, can be seen as an opportunity indicator

matrix which means that for $\mathcal{L}_{ij} = 1$ nodes i and j had the opportunity to connect. We called the whole process, zero inflated stochastic blockmodel (ZinfSBM).

4.4.1 ZinfSBM vs DCSBM

The ZinfSBM is fundamentally different than degree corrected SBM. Precisely, the DCSBM allows nodes to have heterogeneous degree given their community. The main issue is that DCSBM does not consider a missing mechanism governing the lack of interaction, i.e., it assumes that statistical inference based only on the complete data gives unbiased estimates of the true full inference. That being said, ZinfSBM and DCSBM coincides for the case when η is Bernoulli, but this is not the case for any other distributions. For instance, when η is the Poisson distribution, we have the following likelihood function for each model

$$f_{\text{ZinfSBM}}\left(\boldsymbol{A}\right) = \prod_{i \neq j}(1 - \phi_{ij}^{t}) + \phi_{ij}^{t}\frac{e^{-\mu_{ij}}\mu_{ij}^{a_{ij}(t)}}{a_{ij}(t)!} \tag{4.18}$$

$$f_{\text{DCSBM}}\left(\boldsymbol{A}\right) = \prod_{i \neq j}\frac{e^{-\mu_{ij}\phi_{ij}^{t}}(\mu_{ij}\phi_{ij}^{t})^{a_{ij}(t)}}{a_{ij}(t)!} \tag{4.19}$$

Equations (4.18) and (4.19) shows the likelihood difference between models which the difference is clear. All in all, DCSBM incorporates an overdispersion correction, but not necessary deals with overall sparsity (e.g., missing inflation). We need to use a different approach, specially for non binary data.

4.4.2 Observed missing mechanism Z

From Eq.(4.1), we have

$$\begin{aligned} \boldsymbol{A} = \boldsymbol{\mathcal{L}} \odot \boldsymbol{B} \Rightarrow \boldsymbol{B} - \boldsymbol{B} \odot (\boldsymbol{J} - \boldsymbol{\mathcal{L}}) = \boldsymbol{A} \\ \boldsymbol{B} = \boldsymbol{A} + \boldsymbol{B} \odot (\boldsymbol{J} - \boldsymbol{\mathcal{L}}) \end{aligned} \tag{4.20}$$

where $\odot$ represents element-wise multiplication between matrices, $\mathbb{E}[\boldsymbol{B}] = \boldsymbol{P} - \text{diag}(\boldsymbol{P})$, and $\boldsymbol{J}$ is a $n \times n$ matrix of ones. Recall that we aim to perform spectral clustering on $\boldsymbol{P}$, but the opportunity matrix $\boldsymbol{\mathcal{L}}$ masks values in $\boldsymbol{B}$ which makes the eigenvectors matrix of $\boldsymbol{A}$ (used in spectral clustering) very noisy.

Write $\mathbb{E}[a_{ij}] = \mu_{kl}$ and $Var(a_{ij}) = \sigma_{kl}$ for a given pair of nodes i and j where $i \in G_k$, $j \in G_l$ and $\mathcal{L}_{ij} = 0$. Thus, one could impute a_{ij} using the membership matrix, i.e.,

$$\bar{a}^G_{\cdot\cdot} = \frac{\sum_{a\in V}\sum_{b\in V} a_{ab}\mathcal{L}_{ab}\mathbb{I}_{a\in G_k}\mathbb{I}_{b\in G_l}}{\sum_{a\in V}\sum_{b\in V} \mathcal{L}_{ab}\mathbb{I}_{a\in G_k}\mathbb{I}_{b\in G_l}} \tag{4.21}$$

$$\mathbb{E}\left[\bar{a}^G_{\cdot\cdot}\right] = \mu_{kl} \tag{4.22}$$

This is an unbiased estimator, but not useful in practice since we assumes knowledge of community membership of the nodes. A two-step approach where we estimate a membership matrix $\widehat{\boldsymbol{X}}$ and apply Eq. (4.21) might be consistent (despite not being unbiased), but defeats the purpose of performing spectral clustering.

Now, consider the mean square error (MSE) measure which accounts for both variance and bias. If we do not impute any value in a_{ij}, we have the following MSE:

$$\text{MSE}_{\mu_{kl}}(a_{ij}) = \text{Var}(a_{ij}) + \text{Bias}(a_{ij}, \mu_{kl})^2 = 0 + \mu^2_{kl} \tag{4.23}$$

While the variance is 0 (since $\mathcal{L}_{ij} = 0$), the square bias is as large as the expected connectivity strength which can make the overall MSE very large. We can reduce the MSE by imputing values in $\boldsymbol{A}$ systematically. Notice that, however, even with the knowledge of the missing mechanism $\boldsymbol{\mathcal{L}}$, it is highly challenging to perform the spectral clustering on a imputed $\boldsymbol{A}$. Next, we describe potential imputation methods and their limitations.

Global mean: Impute the overall interaction mean of the complete data, i.e.

$$\bar{a}_{..} = \frac{\sum_{a\in V}\sum_{b\in V} a_{ab}\mathcal{L}_{ab}}{\sum_{a\in V}\sum_{b\in V}\mathcal{L}_{ab}} \tag{4.24}$$

$$\mathbb{E}\left[\bar{a}_{..}\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{\sum_{a\in V}\sum_{b\in V} a_{ab}\mathcal{L}_{ab}}{\sum_{a\in V}\sum_{b\in V}\mathcal{L}_{ab}}\right]\Big|\boldsymbol{X}\right] \tag{4.25}$$

$$= \sum_{k}^{K}\sum_{l}^{K}\pi_k\pi_l\mu_{kl} = \bar{\mu}_{..} \tag{4.26}$$

where π_k and π_l are the population proportion of nodes in cluster k and l, respectively. Moreover,

$$Var(\bar{a}_{..}) = \sum_{k}\sum_{l}\pi_k\pi_l(\sigma^2_{kl}+\mu_{kl})^2 - \bar{\mu}^2_{..} = \sigma^2 \tag{4.27}$$

Hence, the MSE of $\bar{a}_{..}$ is given by:

$$\text{MSE}_{\mu_{kl}}(\bar{a}_{..}) = \sigma^2 + (\bar{\mu}_{..} - \mu_{kl})^2 \tag{4.28}$$

As oppose to Eq. (4.28), the MSE defined here is not sensible for large values of $|\mu_{kl}|$. However, Eq. (4.36) is highly dependent of the distribution of nodes over clusters $\boldsymbol{\pi}$, i.e. if π_l and π_k are under represented in $\boldsymbol{A}$ then the bias term can be very large. One alternative is to focus on mean to rows related to nodes i and j.

Mean of means: Impute the mean of rows in $\boldsymbol{A}$ related to nodes i and j , i.e.

$$\bar{a}^{ij}_{\cdot\cdot} = \frac{\bar{a}_{i\cdot} + \bar{a}_{j\cdot}}{2} \tag{4.29}$$

$$\text{where } \bar{a}^{ij}_{i\cdot} = \frac{\sum_{b\in V} a_{ib}\mathcal{L}_{ib}}{\sum_{b\in V}\mathcal{L}_{ib}} \tag{4.30}$$

$$\tag{4.31}$$

Thus,

$$\mathbb{E}\left[\bar{a}_{\cdot\cdot}^{ij}\right] = \sum_{l}^{K} \pi_l \mu_{kl} = \bar{\mu}_{k\cdot} \tag{4.32}$$

$$Var\left(\bar{a}_{\cdot\cdot}^{ij}\right) = \frac{Var\left(\bar{a}_{i\cdot}\right) + Var\left(\bar{a}_{j\cdot}\right)}{2} = \frac{\sigma_k^2 + \sigma_l^2}{2} \tag{4.33}$$

$$\text{where } \sigma_k^2 = \sum_{l}^{K} \pi_l (\sigma_{kl} + \mu_{kl})^2 - \bar{\mu}_{k\cdot}^2 \tag{4.34}$$

$$\tag{4.35}$$

Hence,

$$\text{MSE}_{\mu_{kl}}\left(\bar{a}_{\cdot\cdot}^{ij}\right) = \frac{\sigma_k^2 + \sigma_l^2}{2} + \left(\frac{\bar{\mu}_{k\cdot} + \bar{\mu}_{l\cdot}}{2} - \mu_{kl}\right)^2 \tag{4.36}$$

Using only nodes i and j to compute the means, $\bar{a}_{\cdot\cdot}^{ij}$, it reduces the effect of under represented communities in $\boldsymbol{A}$ comparing to $\bar{a}_{\cdot\cdot}$ then the bias term can be very large. Next, we show how we can reduce the bias term.

Self-similar: This estimator was described in section 4.3. It is similar to the mean of means, but instead of giving every interaction in each row the same weight, we aim to up-weight interactions which are closer to the one that we want to estimate. Also, the self-similar approach is not an actual imputation method, since we also 'impute' seen values aiming to reduce overall noise. Precisely,

$$\bar{a}_{\cdot\cdot}^{w} = \frac{\widehat{a}_{i\cdot}(j) + \widehat{a}_{j\cdot}(i)}{2} \tag{4.37}$$

where $\widehat{a}_{i\cdot}(j)$ and $\widehat{a}_{i\cdot}(j)$ are defined in Eq (4.12). This estimator can significantly reduces the MSE. For $n \to \infty$, (4.21) and (4.37) coincides. Consequently ,

$$\mathbb{E}\left[\bar{a}_{i\cdot}(j)\right] = \mathbb{E}\left[\bar{a}_{j\cdot}(i)\right] = \mu_{kl} \tag{4.38}$$

$$\text{MSE}_{\mu_{kl}}\left(\bar{a}_{\cdot\cdot}^{w}\right) = \sigma_{kl}^2 \tag{4.39}$$

We called this procedure self-similar spectral clustering since $\boldsymbol{W}_j$ is capturing global dependencies without having to estimate the memberships.

[41] method: [41] has an interesting approach which they proposed to scale the observed values by a factor $\hat{p}$, i.e. perform spectral clustering on $\boldsymbol{A}^* := \boldsymbol{A}/\hat{p}$ where $\hat{p} = \frac{|\boldsymbol{\mathcal{L}}|_0}{n^2}$ and $|\boldsymbol{\mathcal{L}}|_0$ counts the number of non zero elements in $\boldsymbol{\mathcal{L}}$. In this case, the MSEs are given by:

$$\mathbb{E}\left[\widehat{a}_{ij}^{\text{Gao}} | \mathcal{L}_{ij} = 1\right] = \mu_{kl}\left(\frac{2n^2}{\|\boldsymbol{\mathcal{L}}\|_0}\right) \tag{4.40}$$

$$\mathbb{E}\left[\widehat{a}_{ij}^{\text{Gao}} | \mathcal{L}_{ij} = 0\right] = 0 \tag{4.41}$$

$$\begin{aligned}\text{MSE}_{\mu_{kl}}\left(\widehat{a}_{ij}^{\text{Gao}} | \mathcal{L}_{ij} = 1\right) = \\ = \sigma_{kl}^2\left(\frac{2n^2}{\|\boldsymbol{\mathcal{L}}\|_0}\right)^2 + \mu_{kl}^2\left[1 - \left(\frac{2n^2}{\|\boldsymbol{\mathcal{L}}\|_0}\right)\right]^2\end{aligned} \tag{4.42}$$

$$\text{MSE}_{\mu_{kl}}\left(\widehat{a}_{ij}^{\text{Gao}} | \mathcal{L}_{ij} = 0\right) = 0 + \mu_{kl}^2 \tag{4.43}$$

This method has the potential to reduce the overall MSE, however it assumes all the interactions have the same probability of being missing. As we have in Eq. (4.1), we aim to relax this constraint.

We included a summary table (Table C.1) of each method and their main limitation in Appendix C.5 . Overall, the MSE values does not guarantee good (or bad) cluster retrieval performance in spectral clustering settings. The MSE is measuring the distance between the matrix used in the spectral clustering (e.g. $\boldsymbol{A}$, imputed $\boldsymbol{A}$, self-similar $\boldsymbol{A}$) and the true $\boldsymbol{P}$. That being said, it our experiments section we showed that most of these methods outperform regular spectral clustering for the various different Zero inflated settings. Next, we described our second proposed method for retrieving communities focus for cases when the missing mechanism is unknown.

4.4.3 Unobserved missing mechanism Z

When the missing mechanism $\boldsymbol{\mathcal{L}}$ is unknown, and depending on the connectivity distribution, there is an additional layer of complexity since there are two types of zeros in

the data. In this scenario, [43] proposed a variational inference method to tackle the Binary case. In their work, they propose different sampling schemes depending on assumed structure of the missing mechanism. Here, we aim to relax specific assumptions of the structure of $\boldsymbol{\mathcal{L}}$, and we focus on areas of the graph where it is highly likely that $\boldsymbol{\mathcal{L}}_{ij} = 1$, i.e. nodes 'know' the existence of each other. Using ideas from Chapter 3, define the Ego-net $\boldsymbol{A}_i := \boldsymbol{A}_{\mathcal{N}(i)*}$ to be the induced adjacency matrix generated by one hop of node i where $\mathcal{N}(i)$ is the set of nodes on the neighborhood of i. Now, define a supra-adjacency matrix $\boldsymbol{A}^*$

$$\boldsymbol{A}^* := \begin{bmatrix} \boldsymbol{A}_1 & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{A_n} \end{bmatrix} \boldsymbol{P^*} = \begin{bmatrix} \boldsymbol{P}_1 & \dots & \boldsymbol{P}_{1n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{P}_{1n}^T & \dots & \boldsymbol{P_n} \end{bmatrix} \tag{4.44}$$

where $\boldsymbol{A}^*$ and $\boldsymbol{P}^*$ are $(|V| \times |V|)$ matrices and $|V|$ is the total number of nodes. In this setting, we know that all interactions a_{ab} in each $\boldsymbol{A}_i$ are observed $\mathcal{L}_{ab} = 1$. Using Chapter 3 derivations, we can estimate the membership matrix $\boldsymbol{X}^*$ by the following optimization problem

$$\left(\widehat{\boldsymbol{X}^*}, \widehat{\boldsymbol{W}}\right) = \underset{\substack{\boldsymbol{X}^* \in \mathcal{M}_{|V|,K} \\ \boldsymbol{W} \in \mathbb{R}^{K \times K}}}{\arg\min} \sum_{i}^{n} \|\boldsymbol{X}_i^* \boldsymbol{W} - \boldsymbol{Q}_i \boldsymbol{H}_i^T \Delta_i^{-1} \Delta \boldsymbol{H}\|_F^2 \tag{4.45}$$

Here, we will use the fact that each node in $\boldsymbol{A}$ is represented by multiple nodes in $\boldsymbol{A}^*$, thus we will average the $\boldsymbol{Q}$ values for each node, and use that to cluster the nodes. Write $\boldsymbol{\Gamma}$ as a $(|V| \times n)$ design matrix where each column represent the incidence of each node on each graph, thus we define $\bar{\boldsymbol{Q}} := (\boldsymbol{\Gamma}^T \boldsymbol{\Gamma})^{-1} \boldsymbol{\Gamma}^T \boldsymbol{Q}$. Therefore, we have the following optimization

$$\left(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{W}}\right) = \underset{\substack{\boldsymbol{X}^* \in \mathcal{M}_{n,K} \\ \boldsymbol{W} \in \mathbb{R}^{K \times K}}}{\arg\min} \|\boldsymbol{X}\boldsymbol{W} - \bar{\boldsymbol{Q}}\|_F^2 \tag{4.46}$$

In real-world settings, there are many sources of heterogeneity and one method cannot deal with all possible problems. However, as we show in our experiment section for over-dispersed, high sparsity and/or high variance and unknown missing mechanism, the Ego-nets spectral clustering is the best option.

4.5 Related work

Community detection in graphs has seen a lot of recent attention. We focus on extensions to zero inflation settings, emphasizing two relevant directions: heterogeneity across communities and zero inflated environments. For the former, [66] propose a degree corrected SBM to account for heterogeneity inside a community, though [70] showed that this fails to retrieve true communities in a high heterogeneous setting. [71] introduce a normalized Laplacian form that account for high heterogeneous scenarios, however this comes at a high computational cost. For zero inflated environments, the works [41, 43] have focus on specifying structure for the missing mechanism $\boldsymbol{\mathcal{L}}$ understanding the missing mechanism. [41] proposed a spectral clustering framework to when $\phi_{ij} = \phi$ for all i and j and $\boldsymbol{\mathcal{L}}_{ij}$ is known. More recent, [43] relaxed assumption of $\boldsymbol{\mathcal{L}}$ knowledge in a variational inference framework, however their framework is limited to binary data. Here, we used spectral clustering which reduces the required assumptions with the main goal of retrieving communities. In the same domain of Zero inflated settings, there has been some interesting work on biased environment (Missing-not-at-random), but mostly on supervised learning tasks (e.g. rating-based recommender systems), [86, 87] proposed some procedures to reduce the effect of the missing mechanism, but it requires the knowledge of ϕ_{ij}. We focus on (Missing-at-random) settings.

4.6 Experiments

The experiment section is divided in synthetic data where the community ground-truth is known, and real-world data (political blogs interaction) where the ground-truth is assumed to be the political affiliation. In Appendix C.7, we included another real-world experiment in which the community detection procedure is used a middle step to a higher level statistical task (predict click in news articles). Our proposed method outperform community detection baselines, and it had comparable performance with a recommender system model that uses additional covariates (our method only uses clicking behavior).

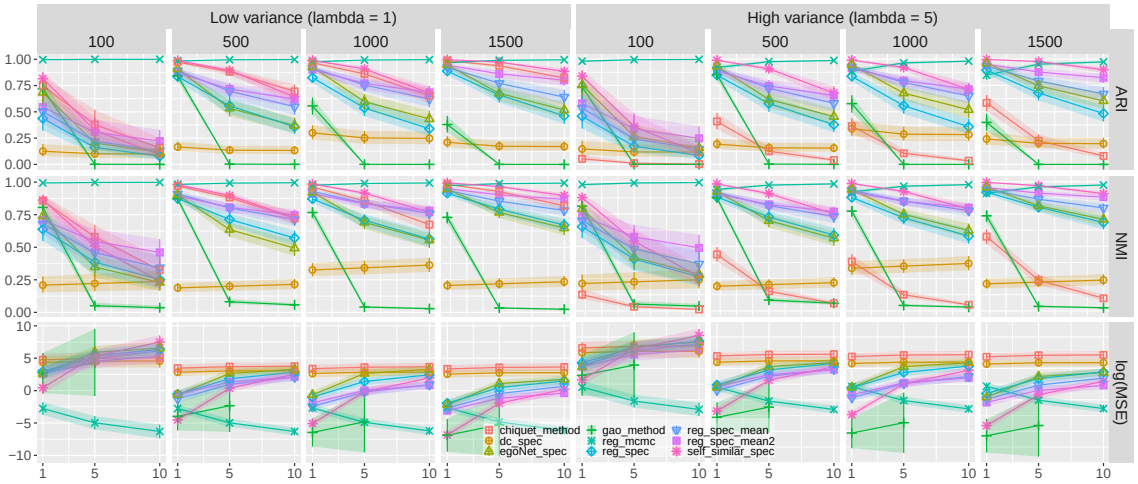

Figure 4.2.: Synthetic data results for increasing graph size (100, 500, 1000 and 1500) in two settings low variance (left) and high variance (right). Each row represent a performance metric: ARI (top) and NMI (middle) for cluster retrieval assessment, and MSE (bottom) for $\widehat{\boldsymbol{P}}$ assessment). X-axis represents increasing level of zero inflation.

4.6.1 Synthetic data

Cluster and community detection assessment requires synthetic data evaluation given real-world ground-truth is never known. In this section, we are mainly interested in evaluating the cluster retrieval performance against other inference schemes in different settings. In this sense, we generate the interactions of the adjacency matrix $\boldsymbol{A}$ as

$$\boldsymbol{\phi}[i,j] := \phi_{ab} \sim \text{Beta}(1,\beta), \quad \mathcal{L}_{ij}|\boldsymbol{\phi} \sim \text{Bern}(\phi_{ij}) \tag{4.47}$$

$$\boldsymbol{\mu}[a,b] := \mu_{ab} \sim \text{Pareto}(1), \quad X_i|\boldsymbol{\pi} \sim \text{Multi}(\boldsymbol{\pi}) \tag{4.48}$$

$$b_{ij}|X_i, X_j\lambda, \boldsymbol{\mu} \sim \text{Poi}\left(\lambda X_i \boldsymbol{\mu} X_j^T\right) \tag{4.49}$$

where $\boldsymbol{\pi} \sim \text{Dir}(\alpha)$ and $\boldsymbol{A}[i,j] = a_{ij} := \mathcal{L}_{ij} b_{ij}$. We included experiments using Gaussian in the Appendix C.6. Overall, the results are very similar to the Poisson.

Setup: We fixed $K = 4$, and we generated graphs using the generative process described in (4.47)- (4.49). We generated 30 graphs for each tuple (β, λ, N), where

- $\beta = \{1, 5, 10\}$: increasing sparsity

- $\lambda = \{1, 5\}$: increasing connectivity variability
- $N = \{100, 500, 1000, 1500\}$: increasing graph size.

Baselines: We compare our proposed self-similar (`self_similar_spec`) and ego-net (`egoNet_spec`) spectral clustering methods described earlier with the following baselines:

- **Spectral clustering:**
  - **Regular (`reg_spec`):** spectral clustering on the observed $\boldsymbol{A}$, no imputation in this case
  - **Degree corrected [63] (`dc_spec`):** degree corrected spectral clustering on the observed $\boldsymbol{A}$, no imputation is performed
  - **Global mean (`reg_spec_mean`):** spectral clustering on $\boldsymbol{A}$, but imputing the missing values with the overall mean
  - **Mean of means (`reg_spec_mean2`):** spectral clustering on $\boldsymbol{A}$, but imputing the missing values using the means of means.
- **Other methods:**
  - **[41] method (`gao_method`):** this was described in Section 4.4. It assumes $\phi_{ij} = \phi$ for all $i, j \in V \times V$.
  - **[43] method (`chiquet_method`):** variational inference scheme for binary data where there some structure assumptions on ϕ_{ij}. In order to perform inference, we threshold the adjacency matrices.
  - **MCMC (`reg_mcmc`):** this is (semi)oracle-based baseline since it uses Eqs. (4.47)-(4.49) as priors. This gives an unfair advantage in comparison to the other methods, but it helps to keep track on the difficulty of each scenario. Inference is carried-out via gibbs sampling .

Evaluation: Our assessment was based on cluster retrieval performance measures (ARI and NMI) and connectivity quality estimation using Mean Square Error (MSE), $MSE = V^{-2} \sum_{i \neq j \in V} (\widehat{\boldsymbol{P}_{ij}} - \boldsymbol{P}_{ij})^2$.

Results

Figure 4.2 shows the results for increasing graph size and different performance measures (ARI, NMI and log(MSE)). Each data point represent the mean performance for each setting and the ribbon around represents the estimated 95% confidence intervals. The x-axis represents values β which indicates increasing sparsity.

Cluster retrieval performance (ARI, NMI): Overall, we see that the degree corrected has the worst overall performance for cluster retrieval. This was expected since it is solving a different problem. The `chiquet_method` has a good performance on a low variance settings. However, since `chiquet_method` deals specifically with the Bernoulli case, it does not perform well on over-dispersed Poisson settings since as the problem becomes more heterogeneous, thresholding the data becomes very complex. While the overall performance of the imputation methods (`reg_spec_mean1-2`) is good, our proposed self-similar (`self_similar_spec`) outperform them in almost every setting. Moreover, our second proposed method, the egoNet, has also a very good overall performance comparable with the mean imputation methods without 'knowing' which of the values are missing. For cases where the regular spectral clustering performed well (low variance settings), the egoNet method was not statistically better in terms of performance. However, for high variance cases, the egoNet outperformed `reg_spec`.

Connectivity estimation (MSE): Here, we aim to have an overall idea of the quality of the estimated $\boldsymbol{P}$, however it does not guarantee good community retrieval performance. For instance, while `gao_method` shows the lowest MSE of the estimated $\boldsymbol{P}$, it had a poor performance on cluster retrieval. Recall that `gao_method` method scales the interactions of adjacency matrix based on the amount of missing values, thus the estimated connectivity matrix $\boldsymbol{P}$ is close to 0. The second lowest MSE was for the self-similar, and `chiquet_method` had the highest.

4.6.2 Real-world data: political blogs (U.S. and France)

These are two very similar datasets from presidential elections in U.S. [88] and France [77]. In each network, nodes represent blogs, and an edge between blogs a and b represents that either a or b had referenced the other blog in a post. We aim to retrieve the political affiliation (community) each blog belongs. The U.S. graph has a collection of 1222 blogs, and the France has 196 blogs. Moreover, the U.S. is expected to have 2 communities (conservatives and liberals), and the France dataset has more granularities with 9 communities. We assumed $\boldsymbol{\mathcal{L}}$ is a $N \times N$ matrix of ones (where N is the number of nodes) since $\boldsymbol{\mathcal{L}}$ is unknown.

Results

Table 4.1 shows the ARI and NMI for each method. While the self-similar was outperformed by the `dc_spec` in the US dataset and by `chiquet_method` in France dataset, it had the best performance across datasets. The egoNets method had a poor performance overall since in both settings there is very low between community connectivity.

Table 4.1.: Cluster retrieval for the political blog datasets

Model	US		France	
	ARI	**NMI**	**ARI**	**NMI**
`chiquet_method`	0.001	0.000	0.459	**0.677**
`reg_spec`	0.080	0.187	0.328	0.549
`reg_spec_mean`	0.080	0.187	0.017	0.175
`reg_spec_mean2`	0.080	0.187	0.013	0.148
`gao_method`	0.065	0.178	0.317	0.514
`dc_spec`	**0.813**	**0.723**	−0.008	0.092
`reg_mcmc`	0.001	0.000	**0.515**	0.641
`self_similar_spec`	**0.772**	**0.686**	**0.403**	**0.642**
`egoNets`	0.01	0.190	0.092	0.195

4.7 Conclusion

In this work, we dealt the problem of missing data in stochastic blockmodel. Specifically, we proposed two inference schemes based of spectral clustering: self-similar (requires knowledge of $\boldsymbol{Z}$) and ego-nets (does not use $\boldsymbol{Z}$). Our methods requires very low parametric assumptions, and works specially well in highly heterogeneous settings.

5. CONCLUSION AND FUTURE WORK

In this dissertation, we dealt with the problem of missing data where only *local* regions of the graph are 'allowed' to connect, this is defined precisely in Eq.(1.1) where the local mechanism $\mathcal{L}$ characterizes the types of problems. We showed that this process can be used to define population of networks (multiple graphs) scenarios when there is a a block structure on the local mechanism (i.e. $\mathcal{L} = \mathbb{I}_{Z_i,Z_j}$), and also zero inflated graphs problems where the local term is a random variable parametrized by ϕ_{ij} (i.e. $\mathcal{L} \sim \text{Bern}(\phi_{ij})$). In terms of multiple graphs, we worked with two main problems: hypothesis testing on weighted aligned graphs and community detection on non-aligned heterogeneous graphs. In terms of zero inflated graphs, we focus on the task of detecting communities using spectral clustering.

5.1 Summary of the contributions

Aligned graphs: We devised a Bayesian hypothesis testing framework for weighted networks. We used our framework to investigate whether brain connectivity is statistically different across some pre-defined groups (e.g., creative versus non creative people). Our framework is highly flexible and powerful. It is capable of dealing with time varying networks. The broad scope of our methodology was shown in case studies for social media datasets (Twitter and Instagram) where each user is a network, nodes are words, and edges are representative of co-occurrences between words.

Non-aligned graphs: We developed a spectral clustering algorithm to assign nodes to communities based on the way these nodes connect. We showed that using pooled information across graphs and jointly estimating local and global structures is crucial to gain a precise understanding of heterogeneous networks. Our model outperforms current two-

step approaches where the first step individually models each graph, and where the second step estimates the global structure. Our algorithm is fast, efficient, robust, and has very few probabilistic assumptions.

Zero inflated graphs: We devised a self-similar pre-processing step which reduces the overall noise and improves inference using spectral clustering. Our method requires very low parametric assumptions, and works specially well in highly heterogeneous settings.

5.2 Future work

Natural extensions of our work would consider more complicated structures on $\mathcal{L}$ in Eq.(1.1). Some examples are block structure with partial alignment and inference on biased environments. Next, we describe these potential directions:

5.2.1 Community detection on partially aligned dynamic networks

Given a graphs observed for T time points, $\mathcal{G}$, where each graph is defined as a snapshot $\mathcal{G}_t = (V_t, E_t, L_t)$ with V_t being the set of nodes at time t, E_t the set of edges, and L_t is the set of node labels at time t. In this case, $L_t \neq L_{t'}$, but $L_t \bigcap L_{t'} \neq \emptyset$ for any $t, t' \in \{1, ..., T\}$. This is a special case of multiple graph (block structure on the local mechanism) which we have partial alignment information. Thus, newcomers might also have an effect on cluster membership of nodes that were already present on previous time points. There multiple ways to approach this, e.g., evolutionary spectral clustering (ESC) [35] and hidden markov model (HMM) [36,37]. For graphs evolving over time, it is common to fix the connectivity matrix and let the membership matrix vary over time, i.e. nodes are allowed to change cluster membership. We aim to allow graphs to vary size over time as well. [36, 37] have sampling approach based on HMM in which they assume an underlying stationary Markov chain process associated with the community membership. Formally, $\boldsymbol{X}_t$ is a Markov chain subject to a transition probability matrix $\boldsymbol{M}$, thus $P(\boldsymbol{X}_t = k|\boldsymbol{X}_t = l) = \boldsymbol{M}_{kl}$. [36] propose inference using Bayesian non-parametric (IRM) and [37] using variational EM. In

one hand, ESC presents an easier, somewhat flexible, way of retrieving memberships in a time-dependent scenario, on the other hand the HMM seems to be more suitable to include an adversary strategy in the modeling framework. A relevant disadvantage of the HMM approach is related to its constraint of fixed size graphs, one may suggest to only consider the nodes that overlap across, but this will most likely lead to a less powerful analysis. More recently, [38] proposed a Hawkes based approach using completely random measures (CRM) to model sparsity in dynamic networks. In their work, community detection is treated as a by-product and it has not been assessed in terms of cluster retrieval performance. We propose using a joint community detection method based on a version of Eq. 3.12 which we aim to incorporate a time structure to control serial correlation of the nodes connectivity.

5.2.2 Collaborative filtering on biased environments

In this case, the local mechanism is a function (or a random funcion) of the latent variables (e.g. $\mathcal{L}_{ij} \not\perp\!\!\!\perp (\boldsymbol{X}_i, \boldsymbol{X_j})$). For instance, consider an observed temporal user-item *bipartite* interaction data (e.g. buyer-product, reader-article, listener-podcast), $\mathcal{D} = (t, i, j)_{k\geq 1}$ where $(t, i, j) \in \mathbb{R}_+ \times \mathcal{I} \times \mathcal{J}$ and $\mathcal{I} \cap \mathcal{J} = \emptyset$ and $\mathcal{I}, \mathcal{J} \in \mathbb{N}_*$. Also, assume the interactions are subject to an environment $\mathcal{E}$ (e.g., amazon, google news,itunes podcasts) which means that user i can only connect to exposed products (i.e., users only connect to products they know exist). Let $(t_k)_{k\geq 1}$ be a sequence of event times with $t_k \geq 0$, and write $N_{ij}(t) = \sum_{k\geq 1} 1_{t_k \leq t}$ as the number of events between times 0 and t from node i to node j. Moreover, assume that $\mathcal{E}$ controls the parameter ϕ_{ij}^t (e.g., exposure/recommendation probability of product j to user i), and its main goal is to maximize the number of overall interactions (up to time T) by choosing the best $\boldsymbol{\phi}^t$, i.e.,

$$\max_{\boldsymbol{\phi}^t} \sum_{i,j} N_{ij}(T) = \max_{\boldsymbol{\phi}^t} \sum_{k} \mathbb{I}_{\{t_k < T\}} \tag{5.1}$$

where ϕ_{ij} is controlled by the recommender system and it is a function of the connectivity of the nodes. Using ideas of the self-similar spectral clustering presented on Chapter

4, we aim to build a precise inference process of the true connectivity of nodes which is robust to this highly biased environment.

REFERENCES

REFERENCES

[1] Guilherme Gomes, Vinayak Rao, and Jennifer Neville. Multi-level hypothesis testing for populations of heterogeneous networks. In *Data Mining (ICDM), 2018 IEEE 18th International Conference on Data Mining*. IEEE, 2018.

[2] Daniele Durante, David B Dunson, et al. Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis*, 13(1):29–58, 2018.

[3] Cedric E Ginestet, Jun Li, Prakash Balachandran, Steven Rosenberg, Eric D Kolaczyk, et al. Hypothesis testing for network data in functional neuroimaging. *The Annals of Applied Statistics*, 11(2):725–750, 2017.

[4] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[5] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, pages 6410–6421, 2018.

[6] Onur Varol, Emilio Ferrara, Christine L. Ogan, Filippo Menczer, and Alessandro Flammini. Evolution of online user behavior during a social upheaval. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 81–90, New York, NY, USA, 2014. ACM.

[7] Sebastian Moreno and Jennifer Neville. Network hypothesis testing using mixed kronecker product graph models. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1163–1168. IEEE, 2013.

[8] Dena Marie Asta and Cosma Rohilla Shalizi. Geometric network comparisons. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 102–110. AUAI Press, 2015.

[9] Daniele Durante, David B Dunson, and Joshua T Vogelstein. Nonparametric bayes modeling of populations of networks. *Journal of the American Statistical Association*, 2015.

[10] David B Dunson and Chuanhua Xing. Nonparametric bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.

[11] Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):437–461, 2015.

[12] François Caron and Emily B Fox. Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1295–1366, 2017.

[13] Harry Crane and Walter Dempsey. A framework for statistical network modeling. *arXiv preprint arXiv:1509.08185*, 2015.

[14] G de S SOUZA. *Introdução aos modelos de regressão linear e não-linear*. EMBRAPA-SPI Brasília, 1998.

[15] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[16] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

[17] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[18] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[19] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 77–118. Springer, 2015.

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

[21] Christopher Aicher, Abigail Z Jacobs, and Aaron Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248, 2014.

[22] MEJ Newman. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *arXiv preprint arXiv:1606.02319*, 2016.

[23] Tiago P Peixoto. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Physical Review E*, 92(4):042807, 2015.

[24] Purnamrita Sarkar, Deepayan Chakrabarti, and Michael I Jordan. Nonparametric link prediction in dynamic networks. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1897–1904. Omnipress, 2012.

[25] Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 155–162. ACM, 2015.

[26] Daniele Durante and David B Dunson. Nonparametric bayes dynamic modelling of relational data. *Biometrika*, pages 1–16, 2014.

[27] Emanuele Cozzo, Mikko Kivelä, Manlio De Domenico, Albert Solé, Alex Arenas, Sergio Gómez, Mason A Porter, and Yamir Moreno. Clustering coefficients in multiplex networks. *arXiv preprint arXiv:1307.6780*, 2013.

[28] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010.

[29] HongFang Zhou, Jin Li, JunHuai Li, FaCun Zhang, and YingAn Cui. A graph clustering method for community detection in complex networks. *Physica A: Statistical Mechanics and Its Applications*, 469:551–562, 2017.

[30] Pedro Mercado, Antoine Gautier, Francesco Tudisco, and Matthias Hein. The power mean laplacian for multilayer graph clustering. *arXiv preprint arXiv:1803.00491*, 2018.

[31] James D Wilson, John Palowitch, Shankar Bhamidi, and Andrew B Nobel. Community extraction in multilayer networks with heterogeneous community structure. *The Journal of Machine Learning Research*, 18(1):5458–5506, 2017.

[32] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[33] Caterina De Bacco, Eleanor A Power, Daniel B Larremore, and Cristopher Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 95(4):042317, 2017.

[34] Tatiana Von Landesberger, Felix Brodkorb, Philipp Roskosch, Natalia Andrienko, Gennady Andrienko, and Andreas Kerren. Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE transactions on visualization and computer graphics*, 22(1):11–20, 2016.

[35] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(4):17, 2009.

[36] Katsuhiko Ishiguro, Tomoharu Iwata, Naonori Ueda, and Joshua B Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *Advances in Neural Information Processing Systems*, pages 919–927, 2010.

[37] Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1119–1141, 2017.

[38] Xenia Miscouridou, François Caron, and Yee Whye Teh. Modelling sparsity, heterogeneity, reciprocity and community structure in temporal interaction data. In *Advances in Neural Information Processing Systems*, pages 2343–2352, 2018.

[39] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: Some first steps. *Social Networks*, 5:109–137, 1983.

[40] Stanley Wasserman and Carolyn Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social networks*, 9(1):1–36, 1987.

[41] Chao Gao, Yu Lu, Zongming Ma, and Harrison H Zhou. Optimal estimation and completion of matrices with biclustering structures. *The Journal of Machine Learning Research*, 17(1):5602–5630, 2016.

[42] Satoshi Goto, Mariko Takagishi, and Hiroshi Yadohisa. Bi-clustering for time-varying relational count data analysis. *arXiv preprint arXiv:1812.09481*, 2018.

[43] Timothée Tabouy, Pierre Barbillon, and Julien Chiquet. Variational inference for stochastic block models from sampled data. *JASA*, pages 1–23, 2019.

[44] Geng Li, Murat Semerci, Bulent Yener, and Mohammed J Zaki. Graph classification via topological and label attributes. In *Proceedings of the 9th international workshop on mining and learning with graphs (MLG), San Diego, USA*, volume 2, 2011.

[45] Dena Marie Asta and Cosma Rohilla Shalizi. Geometric network comparisons. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 102–110. AUAI Press, 2015.

[46] Soumendu Sundar Mukherjee, Purnamrita Sarkar, and Lizhen Lin. On clustering network-valued data. pages 7074–7084, 2017.

[47] Francesca Arrigo, Peter Grindrod, Desmond J Higham, and Vanni Noferini. On the exponential generating function for non-backtracking walks. 2017.

[48] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015.

[49] S. Moreno and J. Neville. Network hypothesis testing using mixed kronecker product graph models. *International Conference on Data Mining*, 2013.

[50] Rion Brattig Correia, Lang Li, and Luis M Rocha. Monitoring potential drug interactions and reactions via network analysis of instagram user timelines. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 21, page 492. NIH Public Access, 2016.

[51] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[52] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

[53] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[54] Iain Murray, Ryan Prescott Adams, and David J.C. MacKay. Elliptical slice sampling. *J. Mach. Learn. Res. W&CP*, 9, 2010.

[55] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

[56] M. Mongiovı, P. Bogdanov, R. Ranca, E. Papalexakis, C. Faloutsos, and A. Singh. Netspot: Spotting significant anomalous regions on dynamic networks. In *SIAM International Conference on Data Mining*, 2013.

[57] Hsiaw-Chan Yeh. Six multivariate zipf distributions and their related properties. *Statistics & probability letters*, 56(2):131–141, 2002.

[58] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[59] Tiago Simas and Luis M Rocha. Distance closures on complex networks. *Network Science*, 3(2):227–268, 2015.

[60] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[61] Drew Schmidt and Christian Heckendorf. ngram: Fast n-gram tokenization, 2017. R package version 3.0.4.

[62] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, pages 1878–1915, 2011.

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

[64] Purnamrita Sarkar, Peter J Bickel, et al. Role of normalization in spectral clustering for stochastic blockmodels. *The Annals of Statistics*, 43(3):962–990, 2015.

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

[66] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

[67] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[68] Victor Y Pan and Zhao Q Chen. The complexity of the matrix eigenproblem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 507–516. ACM, 1999.

[69] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[70] Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. A spectral method for community detection in moderately sparse degree-corrected stochastic block models. *Advances in Applied Probability*, 49(3):686–721, 2017.

[71] Hafiz Tiomoko Ali and Romain Couillet. Improved spectral community detection in large heterogeneous networks. *The Journal of Machine Learning Research*, 18(1):8344–8392, 2017.

[72] Tomoharu Iwata, James Robert Lloyd, and Zoubin Ghahramani. Unsupervised many-to-many object matching for relational data. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):607–617, 2015.

[73] Jean-Benoist Leger. Blockmodels: A r-package for estimating in latent block model and stochastic block model, with various probability functions, with or without covariates. *arXiv preprint arXiv:1602.07587*, 2016.

[74] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[75] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[76] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[77] Hugo Zanghi, Christophe Ambroise, and Vincent Miele. Fast online graph clustering via erdős–rényi mixture. *Pattern recognition*, 41(12):3592–3599, 2008.

[78] Gabriel de Souza Pereira Moreira, Dietmar Jannach, and Adilson Marques da Cunha. Contextual hybrid session-based news recommendation with recurrent neural networks. *arXiv preprint arXiv:1904.10367*, 2019.

[79] David S Choi, Patrick J Wolfe, and Edoardo M Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, 99(2):273–284, 2012.

[80] Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

[81] Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel. Infinite hidden relational models. *arXiv preprint arXiv:1206.6864*, 2012.

[82] Takahiro Nakamura, Akira Shoji, Hironori Fujisawa, and Naoyuki Kamatani. Cluster analysis and association study of structured multilocus genotype data. *Journal of human genetics*, 50(2):53, 2005.

[83] Nianjun Liu and Hongyu Zhao. A non-parametric approach to population structure inference using multilocus genotypes. *Human genomics*, 2(6):353, 2006.

[84] Shameek Biswas, Laura B Scheinfeldt, and Joshua M Akey. Genome-wide insights into the patterns and determinants of fine-scale population structure in humans. *The American Journal of Human Genetics*, 84(5):641–650, 2009.

[85] Daniel John Lawson, Garrett Hellenthal, Simon Myers, and Daniel Falush. Inference of population structure using dense haplotype data. *PLoS genetics*, 8(1):e1002453, 2012.

[86] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352*, 2016.

[87] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 279–287. ACM, 2018.

[88] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[89] George Kingsley Zipf. Human behavior and the principle of least effort. 1949.

[90] Nasrollah Etemadi. Convergence of weighted averages of random variables revisited. *Proceedings of the American Mathematical Society*, 134(9):2739–2744, 2006.

[91] Albert W Marshall and Ingram Olkin. Multivariate chebyshev inequalities. *The Annals of Mathematical Statistics*, pages 1001–1014, 1960.

[92] Bartolomeo Stellato, Bart PG Van Parys, and Paul J Goulart. Multivariate chebyshev inequality with estimated mean and variance. *The American Statistician*, 71(2):123–127, 2017.

[93] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[94] Alan Stuart and J Keith Ord. *Kendall's Advanced Theory of Statistics: Distribution theory; Vol. 2, Classical inference and relationship; Vol. 3, Design and analysis, and time-series*. Charles Griffin, 1987.

APPENDICES

A. APPENDIX TO CHAPTER 2

A.1 Synthetic data details

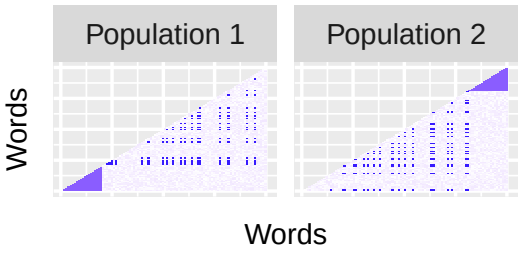

Figure A.1.: Structure used to simulate homogeneous data

$$
\begin{aligned}
p &\sim \text{Beta}(1, \lambda) \\
\boldsymbol{X}_n | p &\overset{ind}{\sim} \text{Geo}(p) \qquad \text{(A.1)} \\
\Rightarrow \boldsymbol{X}_n &\sim \text{M}^{(m)}\text{Zipf}(IV)(0, \lambda, 1, 1)
\end{aligned}
$$

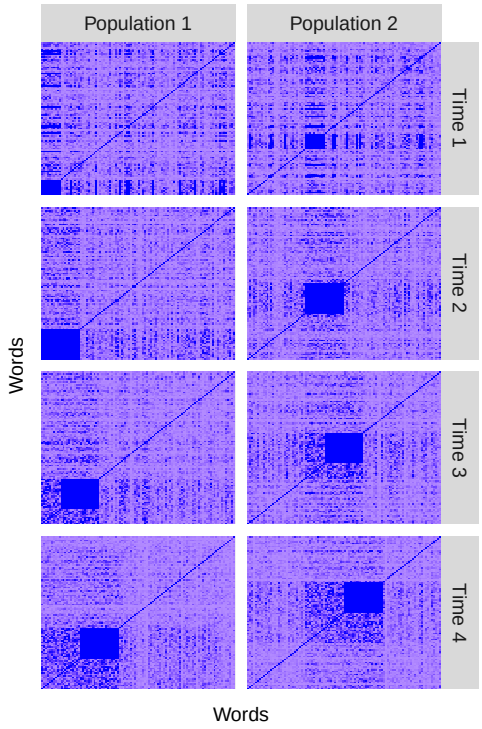

Figure A.2.: Structure used to simulate heterogeneous data

The synthetic data was generated to have the same construction of the Twitter dataset set, i.e. a set word co-occurrences networks. In this sense, each node is a word and each edge is a co-occurrence of two words. Moreover, each pair of words can co-occur at most the minimum occurrence of each individual word. In other words, say words i and j occurred x and y times, respectively, then the co-occurrence of words i and j is at most $\min(x, y)$. Hence, in order to generate graphs in this setting, we need to have the individual occurrences of all the words and probability structure for the co-occurrences. Since Zipf's law [89] , or discrete Pareto, is almost always used to describe words frequencies,

we used a multivariate Zipf generating process [57] to generate the individual counts. Equation A.1 shows how to generate multivariate Zipf's values, $\boldsymbol{X}_n \in \mathbb{R}^V$ is the vector with all the individual counts of V words for entity n. In our case, we considered the standard Zipf where $\lambda = 1$.

The structures were arbitrarily chosen to have a clear difference across populations. Figure A.1 shows the structure used to simulate the homogeneous data used on the experiments section and Figure A.2 shows the structure used to simulate for the heterogeneous data. Given the individual counts and the probability of each edge, we simulate edge count using Binomial distribution. Formally, $A_{nij} \sim Bin\left(\min\left(X_{ni}, X_{nj}\right), \boldsymbol{\theta}_{ij}\right)$, where A_{nij} is the co-occurrence of words i and j, X_{ni} and X_{nj} are their individual counts and $\boldsymbol{\theta}_{ij}$ is the probability of i and j co-occur once.

B. APPENDIX TO CHAPTER 3

B.1 *JointSpec* algorithm

B.2 Additional experiments

B.2.1 Additional synthetic data experiments

Cluster retrieval performance experiments for increasing number of graphs Here, we assess the performance of the inference models for increasing number of graphs ($N = (50, 200, 600, 1000)$) and nodes ($|V_n| = (25, 50, 100, 200)$). The results in Figure B.1 are in accordance with the ones shown in Figure 3.3(left) where *JointSpec* outperform all baselines, both on individual graph level performance and overall global performance.

Varying graphs sizes Here, we aim to assess cluster retrieval performance in settings where the graphs have different sizes. We used $|V_n| \overset{iid}{\sim} NB(\mu, r)$ to sample the size of each graph, where μ is the mean and r the dispersion parameter. We fixed $\mu = 200$ and we vary $r \in [1, 10]$. Lower values of r mean more variability in graph size distribution. We also consider two main scenario: 1) homogeneous $\alpha = \frac{1}{K}$; and 2) heterogeneous $\alpha = 1$. Figure B.2 shows the curves for each model in each scenario. For the heterogeneous scenario it is clear that the JointSpec outperform the baselines. Also, NMI curves look flat for increasing r on both scenarios (homogeneous and heterogeneous) which suggests that the distribution of nodes over clusters (controlled by α) is more detrimental for cluster retrieval than the size of the graphs.

Computational complexity We also performed experiments to assess cluster retrieval performance over computational runtime. We simulated data using Eq. (3.19) for $K = 6, 9$, $\alpha = .1, 1, 2$ and $|V_n| = 500, 800$. For `ReMatch`, we sampled 200 instances of the model. All runs were on a Macbook Pro 2.3 GHz Intel Core i7, 8gb 1600 MHz DDR3. Figure B.3

Algorithm 1 JointSpec SBM

Input N adjacency matrices $\boldsymbol{A}_n \in \{0,1\}^{|V_n|\times|V_n|}$, number of communities K and tolerance ε

for $n \in [1, ..., N]$ **do**

Compute $\widehat{\mathbf{U}}_n \in \mathbb{R}^{|V_n|\times K}$, $\widehat{\mathbf{D}}_n \in \mathbb{R}^{K\times K}$ as the leading K eigenvectors and eigenvalues of $\mathbf{A}_n$.

set $\widehat{\mathbf{Q}}_n^* \leftarrow |\widehat{\mathbf{U}}_n\widehat{\mathbf{D}}_n|\sqrt{\frac{|V|}{|V_n|}}$

Initialize $\widehat{\boldsymbol{X}}_n$ randomly

set $\widehat{\Delta}_n^2 \leftarrow \widehat{\boldsymbol{X}}_n^T\widehat{\boldsymbol{X}}_n$

end for

compute $\widehat{\Delta}^2 \leftarrow \sum_n^N \widehat{\Delta}_n^2$

set $\text{loss}_0 \leftarrow 0$, $\text{loss}_1 \leftarrow 1$ and $t \leftarrow 1$

while $|\text{loss}_t - \text{loss}_{t-1}| > \varepsilon$ **do**

update $t \leftarrow t + 1$

for $n \in [1, ..., N]$ **do**

compute $\gamma_n \leftarrow \text{tr}(\widehat{\Delta}_n^2\widehat{\Delta}^{-2})$ ▷ Eq. (3.14)

end for

update $\widehat{\mathbf{W}}$ ▷ Eq. (3.16)

for $n \in [1, ..., N]$ **do**

for $i \in [1, ..., |V_n|]$ **do**

for $k \in [1, ..., K]$ **do**

compute $\tilde{\Delta}_{ni}(k)$ ▷ Eq. (3.18)

compute $\omega_{ni}(k)$ ▷ Eq. (3.17)

end for

update $\widehat{\mathbf{X}}_{ni} \leftarrow \text{onehot}(\arg\min_k \omega_{ni}(k))$

end for

compute $\widehat{\Delta}_n^2 \leftarrow \widehat{\boldsymbol{X}}_n^T\widehat{\boldsymbol{X}}_n$

update $\eta_n(\widehat{\boldsymbol{X}}_n, \widehat{\boldsymbol{W}})$ ▷ Eq. (3.14)

end for

compute $\widehat{\Delta}^2 \leftarrow \sum_n^N \widehat{\Delta}_n^2$

compute $\text{loss}_t \leftarrow \sum_n^N \eta_n(\widehat{\boldsymbol{X}}_n, \widehat{\boldsymbol{W}})$

end while

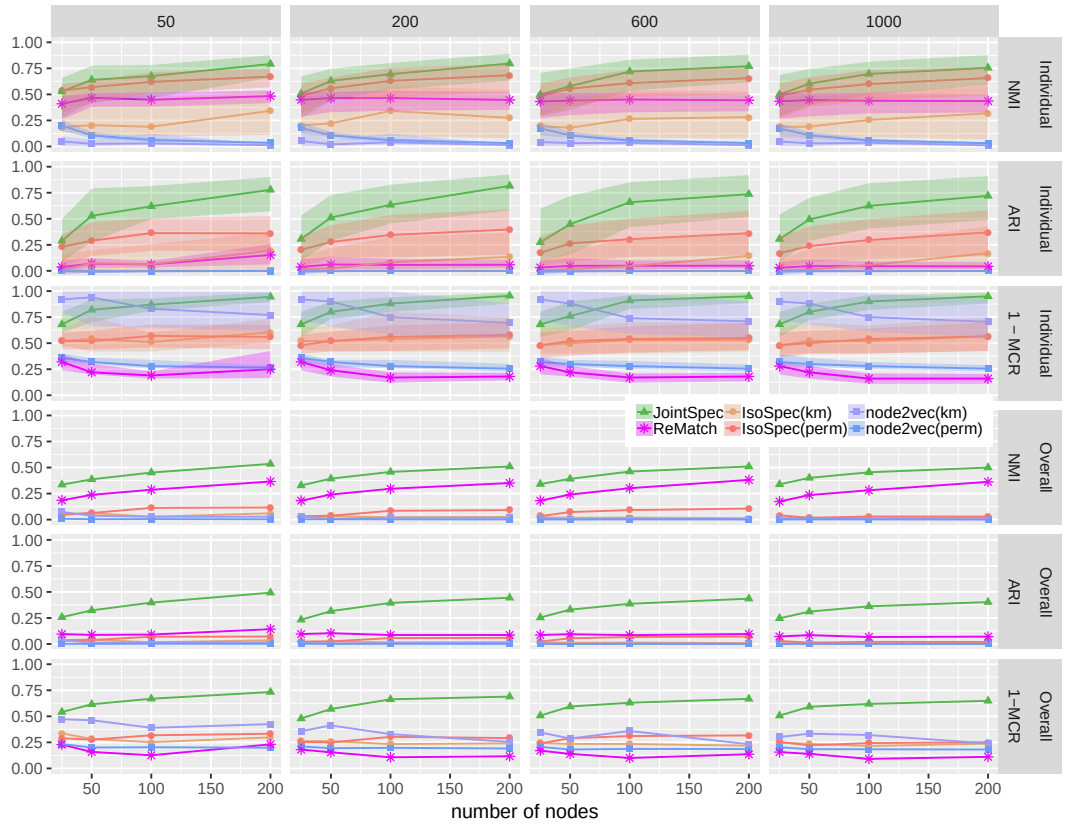

Figure B.1.: Fixed $\alpha = 10$: Cluster retrieval performance curves for each measure (ARI,MCR and NMI) for each model for increasing number of nodes and number of graphs. Top row: median and the interquartile range curves of the individual graphs performance. Bottom row: overall curves.

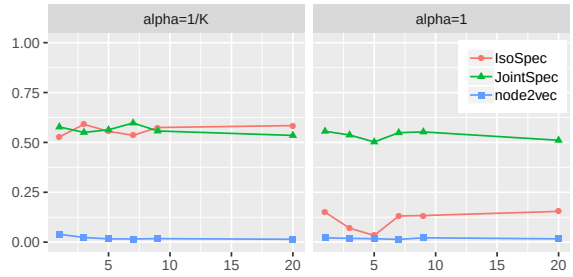

Figure B.2.: NMI curves over r where r is the dispersion parameter in $|V_n| \overset{iid}{\sim} NB(\mu, r)$ for homogeneous ($\alpha = 1/K$) and heterogeneous ($\alpha = 1$) scenarios

shows the log runtime for each experiments, we measured cluster retrieval performance using NMI, ARI and $1-$misclustering rate. Looking at *Joint SBM*, JointSpec takes significantly less time to converge to a good measured performance than `ReMatch`.

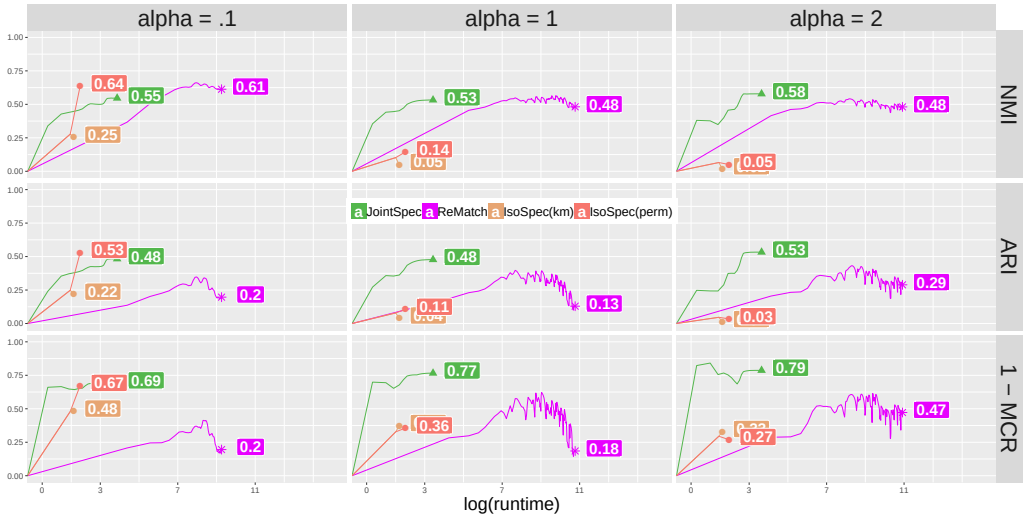

Figure B.3.: Cluster retrieval performance measures (NMI, ARI, 1 - MCR) over number of operations for different inference methods, for $K = 6$, and different heterogeneity scenario $\alpha = .1, 1, 2$. Showing only the first 50 samples of `ReMatch`.

B.2.2 Qualitative connectivity assessment

Figure B.4 shows the true connectivity used to generate the synthetic data. Figure B.5 (top row) shows the estimated connectivity matrix for each approach for the case of 200 graphs, 200 nodes per graph. The *JointSpec* estimates are the most similar to the true connectivity and `node2vec` performs worst.

0.9	0.8	0.1	0.1	0.3	0.01
0.8	0.9	0.6	0.1	0.3	0.01
0.1	0.6	0.5	0.3	0.3	0.01
0.1	0.1	0.3	0.4	0.01	0.01
0.3	0.3	0.3	0.01	0.2	0.01
0.01	0.01	0.01	0.01	0.01	0.1
1	**2**	**3**	**4**	**5**	**6**

Figure B.4.: True connectivity matrix $\mathbf{\Theta}$ used for synthetic data.

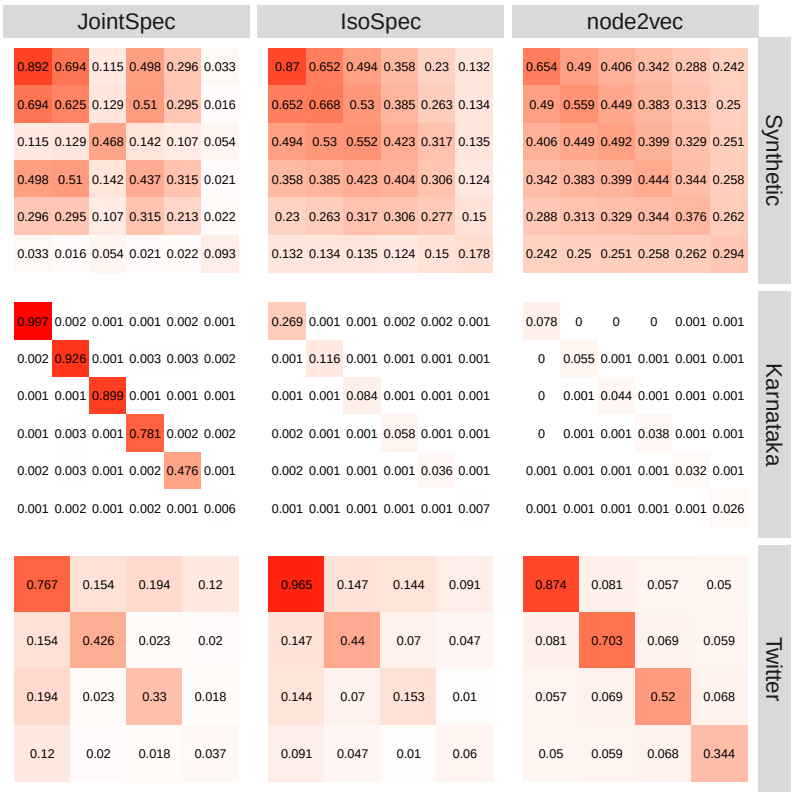

Figure B.5.: Connectivity matrix estimated ($\widehat{\mathbf{\Theta}}$) by each approach (columns), for each dataset (rows).

B.2.3 Additional assessment of Twitter experiments

We also compute the entropy of the community assignment per words across graphs. We expect the community of the words to be consistent across graphs, therefore a lower entropy. We found that the Joint model had the lowest entropy overall, Figure B.6 shows the results. Also, we include a pairwise distribution of the difference of the entropies which shows that JointSpec had the lowest entropy for the majority of the words.

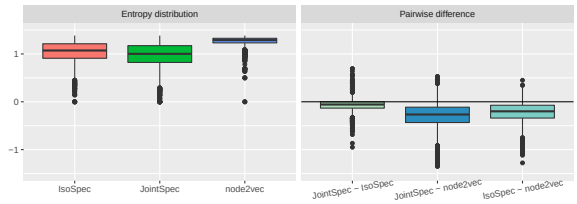

Figure B.6.: Entropy distribution of the words per model (left) and pairwise entropy difference between models for each word (right).

B.3 Spectral clustering (single graph): Derivation of Eq.(3.5) and unbiasedness

Recall that the connectivity matrix $\boldsymbol{\Theta}$ consists of edge probability for within and between communities. Now, say θ_{kl} is the element of $\boldsymbol{\Theta}$ on the kth row l-th column. Thus, one can estimate θ_{kl} by counting the number of edges between communities k and l and dividing by the total number of possible edges. For $k \neq l$, the total number of possible edges is the number of nodes in k multiplied by the number of nodes in l. For a adjacency matrix $\boldsymbol{A}_n$, we can generalize the estimation of the connectivity matrix using matrix notation as

$$\widehat{\boldsymbol{S}}_n = [\widehat{\boldsymbol{X}}_n^T \widehat{\boldsymbol{X}}_n]^{-1} \widehat{\boldsymbol{X}}_n^T \boldsymbol{A}_n \widehat{\boldsymbol{X}}_n [\widehat{\boldsymbol{X}}_n^T \widehat{\boldsymbol{X}}_n]^{-1} \tag{B.1}$$

If assume we know the true membership matrix (i.e. $\widehat{\boldsymbol{X}} = \boldsymbol{X}_n$), the off-diagonal elements of $\widehat{\boldsymbol{S}}_n$ in Eq. (B.1) above have unbiased estimates, however the diagonal elements (i.e. the within community probability) are biased. More specifically, the total possible number of edges for nodes in the same communities is being assumed to have self loops which is incorrect in this setting, and also the term $\widehat{\boldsymbol{X}}_n^T \boldsymbol{A}_n \widehat{\boldsymbol{X}}_n$ is double counting the edges within communities. Formally,

$$\begin{aligned} \mathbb{E}\left[\widehat{\boldsymbol{S}}_n\right] &= \Delta_n^{-2} \left(\boldsymbol{X}_n^T \boldsymbol{P}_n \boldsymbol{X}_n - \boldsymbol{X}_n^T \text{diag}\left(\boldsymbol{P}_n\right) \boldsymbol{X}_n \right) \Delta_n^{-2} \\ &= \boldsymbol{\Theta} - \Delta_n^{-2} \text{diag}\left(\boldsymbol{\Theta}\right) \end{aligned} \tag{B.2}$$

Here, we are using the fact that $\mathbb{E}[\boldsymbol{A}_n] = \boldsymbol{P}_n - \text{diag}\left(\boldsymbol{P}_n\right)$. Furthermore, we can construct an unbiased estimator for $\boldsymbol{\Theta}$ by adding the following term to each diagonal element of $\widehat{\boldsymbol{S}}_n$:

$$\frac{\text{number of edges in cluster } k}{|G_{nk}| \binom{|G_{nk}|}{2}} \tag{B.3}$$

where $|G_{nk}|$ is the number of nodes in community k. For all k, we have Eq.(B.3) in matrix notation as

$$\Delta_n^{-2} \left[\mathbb{I}_k - \Delta_n^{-2}\right]^{-1} \text{diag}\left(\widehat{\boldsymbol{S}}_n\right) \tag{B.4}$$

Now, it follows from Eq. (B.2) that

$$\mathbb{E}\left[\text{diag}\left(\widehat{\boldsymbol{S}}_n\right)\right] = \left[\mathbb{I} - \Delta_n^{-2}\right] \text{diag}(\boldsymbol{\Theta}) \tag{B.5}$$

Using Eqs.(B.1) and (B.4), we get the expression in Eq.(3.5). And using Eq. (B.2) and (B.5), we have

$$\mathbb{E}\left[\widehat{\boldsymbol{\Theta}}_n\right] = \boldsymbol{\Theta} \tag{B.6}$$

We also have that

$$\text{Var}\left(\widehat{\boldsymbol{\Theta}}_n\right) = \begin{cases} \frac{\theta_{kl}(1-\theta_{kl})}{|G_{nk}||G_{nl}|}, & \text{off-diagonal elements} \\ \frac{\theta_{kk}(1-\theta_{kk})}{\binom{|G_{nk}|}{2}}, & \text{diagonal elements} \end{cases} \tag{B.7}$$

where $|G_{nk}|$ is the number of nodes of graph n in cluster k.

B.4 Joint spectral clustering

B.4.1 Proof of Lemma 3.2.1

Proof From Equation (3.11), and since $\mathbf{X}_n = \mathbf{X}_{n*}$, we have:

$$\begin{aligned}
0 &= \boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n\boldsymbol{Z}_n^T\Delta_n^{-1}\Delta\mathbf{Z} \\
&= \boldsymbol{X}_n\mathbf{W}\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n \\
&= \boldsymbol{X}_n\mathbf{W}\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n\sqrt{\frac{|V|}{|V_n|}}\sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K \\
&= \boldsymbol{X}_n\mathbf{W}\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K, \text{where } \mathbf{Q}_n^* = \mathbf{Q}_n\sqrt{\frac{|V|}{|V_n|}} \\
&= \boldsymbol{X}_n\mathbf{W}\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n + \mathbf{Q}_n^*\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K \\
&= \left(\boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^*\right)\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n + \mathbf{Q}_n^*\left(\mathbf{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K\right)
\end{aligned}$$

Recall that $\mathbf{Q}_n$ corresponds to the data from a single graph. $\mathbf{Q}_n^*$ is then a weighted version, based on the relative number of nodes in the graph.

From this we can transform Eq. (3.12) to

$$\underset{\substack{\boldsymbol{X}\in\mathcal{M}_{|V|,K}\\ \boldsymbol{W}\in\mathbb{R}^{K\times K}}}{\arg\min} \sum_{n=1}^{N} \left\| a_n(\boldsymbol{X}_n, \boldsymbol{W}) + b_n(\boldsymbol{X}_n) \right\|_F^2 .$$

where

$$a_n(\boldsymbol{X}_n, \boldsymbol{W}) := \left(\boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^*\right) \boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n$$

$$b_n(\boldsymbol{X}_n) := \mathbf{Q}_n^* \left(\boldsymbol{Z}^T\Delta^{-1}\Delta_n\mathbf{Z}_n - \sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K \right)$$

□

B.4.2 Proof of Lemma 3.2.2

Proof

$$\begin{aligned}
\frac{1}{2}\left\| a_n(\boldsymbol{X}_n, \boldsymbol{W}) + b_n(\boldsymbol{X}_n) \right\|_F^2 &\leq \left\| a_n(\boldsymbol{X}_n, \boldsymbol{W}) \right\|_F^2 + \left\| b_n(\boldsymbol{X}_n) \right\|_F^2 \\
&\leq \left\| \boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^* \right\|_F^2 \left\| \boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n \right\|_F^2 + \left\| b_n(\boldsymbol{X}_n) \right\|_F^2 \\
&= \left\| \boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^* \right\|_F^2 \gamma_n + \left\| \mathbf{Q}_n^*\boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\sqrt{\frac{|V_n|}{|V|}} \right\|_F^2 \\
\text{(See Lemma B.4.1)} \qquad &\leq \left\| \boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^* \right\|_F^2 \gamma_n + \left\| |\mathbf{Q}_n^*|\,\Delta^{-1}\Delta_n + |\mathbf{Q}_n^*| \sqrt{\frac{|V_n|}{|V|}} \right\|_F^2 \qquad \text{(B.8)} \\
&= \left\| \boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^* \right\|_F^2 \gamma_n + \left\| |\mathbf{Q}_n^*| \left(\Delta^{-1}\Delta_n + \sqrt{\frac{|V_n|}{|V|}} \right) \right\|_F^2 \qquad \text{(B.9)} \\
&:= \widetilde{a}_n(\boldsymbol{X}_n, \boldsymbol{W}) + \widetilde{b}_n(\boldsymbol{X}_n) := \eta_n(\boldsymbol{X}_n, \boldsymbol{W})
\end{aligned}$$

where $|\boldsymbol{M}|$ is element-wise absolute value of elements of matrix $\boldsymbol{M}$, and

$$\gamma_n = \left\| \boldsymbol{Z}\Delta^{-1}\Delta_n \mathbf{Z}_n^T \right\|_F^2 = \text{tr}\left(\boldsymbol{Z}_n\Delta_n\Delta^{-2}\Delta_n\boldsymbol{Z}_n^T\right)$$
$$= \text{tr}\left(\Delta_n^2\Delta^{-2}\right) = \sum_{m=1}^{K}\frac{|G_{nm}|}{|G_{\cdot m}|}. \tag{B.10}$$

Recall that G_{nk} is the set of nodes from n that are in cluster k, and $G_{\cdot k}$ is the set of nodes from all graphs in cluster k.

Lemma B.4.1 *The following inequality holds:*

$$\|\boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^*\|_F^2\,\gamma_n + \left\|\mathbf{Q}_n^*\boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2 \leq$$
$$\|\boldsymbol{X}_n\mathbf{W} - \mathbf{Q}_n^*\|_F^2\,\gamma_n + \left\||\mathbf{Q}_n^*|\,\Delta^{-1}\Delta_n + |\mathbf{Q}_n^*|\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2 \tag{B.11}$$

Proof Eq. B.11 is equivalent to

$$\left\|\mathbf{Q}_n^*\boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2 \leq \left\||\mathbf{Q}_n^*|\,\Delta^{-1}\Delta_n + |\mathbf{Q}_n^*|\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2 \tag{B.12}$$

Rewriting LHS of Eq. (B.12) using trace operator, we have

$$\left\|\mathbf{Q}_n^*\boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n - \mathbf{Q}_n^*\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2 = tr\left(\Delta_n^2\Delta^{-2}\boldsymbol{Z}\mathbf{Q}_n^{*T}\mathbf{Q}_n^*\boldsymbol{Z}^T\right) - \sqrt{\frac{|V_n|}{|V|}}tr\left(\boldsymbol{Z}_n^T\Delta_n\Delta^{-1}\boldsymbol{Z}\mathbf{Q}_n^{*T}\mathbf{Q}_n^*\right)$$
$$- \sqrt{\frac{|V_n|}{|V|}}tr\left(\mathbf{Q}_n^{*T}\mathbf{Q}_n^*\boldsymbol{Z}^T\Delta^{-1}\Delta_n\boldsymbol{Z}_n\right) + \frac{|V_n|}{|V|}tr\left(\mathbf{Q}_n^{*T}\mathbf{Q}_n^*\right) \tag{B.13}$$

The RHS is given by

$$\left\||\mathbf{Q}_n^*|\,\Delta^{-1}\Delta_n + |\mathbf{Q}_n^*|\sqrt{\frac{|V_n|}{|V|}}\right\|_F^2 = tr\left(\Delta_n^2\Delta^{-2}|\mathbf{Q}_n^{*T}||\mathbf{Q}_n^*|\right) +$$

$$+ 2\sqrt{\frac{|V_n|}{|V|}} tr\left(\Delta_n \Delta^{-1} |\mathbf{Q}_n^{*T}||\mathbf{Q}_n^*|\right) + \frac{|V_n|}{|V|} tr\left(|\mathbf{Q}_n^{*T}||\mathbf{Q}_n^*|\right) \tag{B.14}$$

Notice that,

$$tr\left(\Delta_n^2 \Delta^{-2} \boldsymbol{Z} \mathbf{Q}_n^{*T} \mathbf{Q}_n^* \boldsymbol{Z}^T\right) \leq tr\left(\Delta_n^2 \Delta^{-2} |\mathbf{Q}_n^{*T}||\mathbf{Q}_n^*|\right) \text{ and} \tag{B.15}$$

$$\frac{|V_n|}{|V|} tr\left(\mathbf{Q}_n^{*T} \mathbf{Q}_n^*\right) \leq \frac{|V_n|}{|V|} tr\left(|\mathbf{Q}_n^{*T}||\mathbf{Q}_n^*|\right) \tag{B.16}$$

The result thus follows. ■

B.5 Estimation Consistency

B.5.1 Consistency of connectivity matrix

The global parameter $\boldsymbol{\Theta}$ is central in order to multi-graph settings, not only summarizing how communities interact, but also allowing information to be pooled across graphs. Here, we discuss the asymptotic behavior of $\boldsymbol{\Theta}$ in multi-graph joint SBM as $N \to \infty$. We show that the estimator $\widehat{\boldsymbol{\Theta}} = N^{-1} \sum_n \widehat{\boldsymbol{\Theta}}_n$ in Eq. (3.5) converge to the global $\boldsymbol{\Theta}$ almost surely when we know the true membership. Lemma B.5.1 (below) formalizes these statements.

Lemma B.5.1 *Let the pair $(\boldsymbol{X}, \boldsymbol{\Theta})$ parametrize an SBM with K communities for N graphs where $\boldsymbol{X}$ contains the membership matrix of all graphs stacked and $\boldsymbol{\Theta}$ is full rank. Write $\nu \leq \min_n |V_n|$. Now if assume $(\widehat{\boldsymbol{W}}, \widehat{\boldsymbol{X}})$ is the solution of Eq. 3.12 then $\widehat{\boldsymbol{\Theta}}$ converges to $\boldsymbol{\Theta}$ in probability Eq.* (B.17)*. If we also assume $\widehat{\boldsymbol{X}}_n = \boldsymbol{X}_n$ then $\widehat{\boldsymbol{\Theta}}$ converge to $\boldsymbol{\Theta}$ almost surely Eq.* (B.18)*.*

$$\lim_{\nu \to \infty; N \to \infty} \widehat{\boldsymbol{\Theta}} \xrightarrow{P} \boldsymbol{\Theta} \tag{B.17}$$

$$\lim_{N \to \infty} \widehat{\boldsymbol{\Theta}} \xrightarrow{a.s.} \boldsymbol{\Theta} \tag{B.18}$$

Proof Eq.(B.17) follows directly from the fact that $\boldsymbol{A}_n$ converge in probability to $\boldsymbol{P}_n$ for large $|V_n|$, Theorem 5.2 [63]. Eq.(B.18) follows from Eq. (B.6), we know that $\mathbb{E}[\widehat{\boldsymbol{\Theta}}_n] = \boldsymbol{\Theta}$. Thus, using Kolmogorov-Khintchine strong law of large numbers, $\widehat{\boldsymbol{\Theta}} \to \boldsymbol{\Theta}$ almost surely for large N.

□

Eq. (B.18) is only true in the multi-graph joint case. In the isolated setting, we need to *re-align* the memberships across graphs which adds an extra layer of complexity. For instance, assume the re-alignment procedure consists on (1) rank each community on each graph based on $diag(\boldsymbol{\Theta}_n)$, then (2) re-order the connectivity matrix and membership accordingly. In this case, $Var(\widehat{\boldsymbol{\Theta}}) = \sum_n^N Var(\boldsymbol{\Theta}_n) \to \infty$, unless we assume graph size to be large, i.e. $\nu \to \infty$ where $\nu \leq \min_n |V_n|$. Nevertheless, this gives weak consistency at most. In fact, this is true for any realignment procedure whose performance is a function of graph size. Figure 3.4(Right) in the Synthetic experiments shows that the Joint model estimates $\boldsymbol{\Theta}$ well even for small graph settings which is not true for Isolated models.

B.5.2 Consistency of membership assignments

We sketch a proof of weak consistency for the membership assignments as follows: 1) we assume that we observe the $\boldsymbol{P}_n$s instead of the $\boldsymbol{A}_n$s, and 2) then we include the expected error of using $\boldsymbol{A}_n$s as a proxy for $\boldsymbol{P}_n$s using Theorem 5.2 in [63].

Notice that node assignment depends only on $\widetilde{a}_n(\boldsymbol{X}_n, \boldsymbol{W})$ for $N \to \infty$ in algorithm 1 since $\widetilde{\Delta}_{ni}(k) \approx C$ for any k. Therefore, $\boldsymbol{Z}^T \Delta^{-1} \Delta_n \mathbf{Z}_n \approx \sqrt{|V_n|/|V|}\mathbb{I}_K$. Now, we incorrectly place node $ni \in G_k$ in some cluster l if $\omega_{ni}(k) > \omega_{ni}(l)$. Formally, define the set of misplaced nodes as $S_k^p := \left\{ ni \in G_k : \left\| \widehat{\boldsymbol{W}}_k - \boldsymbol{Q}_{ni}^* \right\| \geq \min_{l \neq k} \left\| \widehat{\boldsymbol{W}}_k - \widehat{\boldsymbol{W}}_l \right\| /2 \right\}$

where $(\widehat{\boldsymbol{W}}, \widehat{\boldsymbol{X}})$ is the optimal solution of Eq. 3.15 using $\boldsymbol{P}_n$s. Thus, the probability of node $ni \in G_k$ to be misclustered is given by

$$P\left(ni \in S_k^p\right) = P\left(\left\|\widehat{\boldsymbol{W}}_k - \boldsymbol{Q}_{ni}^*\right\|^2 \geq \frac{\min_{l\neq k}\left\|\widehat{\boldsymbol{W}}_k - \widehat{\boldsymbol{W}}_l\right\|^2}{4}\right) \tag{B.19}$$

Now, using Equation (3.16), we have

$$\widehat{\boldsymbol{W}} = \left[\sum_n^N \boldsymbol{X}_n^T \boldsymbol{X}_n \gamma_n\right]^{-1} \sum_n^N \boldsymbol{X}_n^T \boldsymbol{Q}_n^* \gamma_n \tag{B.20}$$

Given γ_n is a monotonically decreasing sequence, and $\sum_n^N \boldsymbol{X}_n^T \boldsymbol{X}_n \gamma_n \to \infty$, we can use the law of large numbers for weighted averages (Theorem 1 of [90]). As $N \to \infty$, we have:

$$\widehat{\boldsymbol{W}} \xrightarrow{P} \left[\boldsymbol{X}^T \boldsymbol{X}\right]^{-1} \sum_n^N \boldsymbol{X}_n^T \boldsymbol{Q}_n^* \to \left[\boldsymbol{X}^T \boldsymbol{X}\right]^{-1} \sum_n^N \boldsymbol{X}_n^T \mathbb{E}\left[\boldsymbol{Q}_n^*\right] \tag{B.21}$$

$$= \left[\boldsymbol{X}^T \boldsymbol{X}\right]^{-1} \sum_n^N \sqrt{|V|/|V_n|} \boldsymbol{X}_n^T \mathbb{E}\left[\boldsymbol{X}_n \boldsymbol{W} \boldsymbol{Z}^T \Delta^{-1} \Delta_n \mathbf{Z}_n\right] = \boldsymbol{W} \tag{B.22}$$

Eq. (B.22) arises from the fact that, for a given data of N graphs, $|V|$ nodes and K communities, the overall number of nodes over communities is $\Delta^2 = \text{diag}(|G_1|, ..., |G_K|)$. Thus,

$$\mathbb{E}[\Delta_n^2 \boldsymbol{\Theta} \Delta_n^2 | \Delta^2] = \frac{|V_n|}{|V|} \Delta^2 \boldsymbol{\Theta} \Delta^2 \frac{|V_n|}{|V|} \tag{B.23}$$

$$\mathbb{E}[\Delta_n \boldsymbol{Z}_n \boldsymbol{D}_n \boldsymbol{Z}_n' \Delta_n | \Delta^2] = \frac{|V_n|}{|V|} \Delta \boldsymbol{Z} \boldsymbol{D} \boldsymbol{Z}' \Delta \frac{|V_n|}{|V|} \tag{B.24}$$

$$\mathbb{E}\left[\sqrt{\frac{|V_n|}{|V|}} \boldsymbol{Z}' \Delta^{-1} \Delta_n \boldsymbol{Z}_n \boldsymbol{D}_n \boldsymbol{Z}_n' \Delta_n \Delta^{-1} \boldsymbol{Z} \sqrt{\frac{|V_n|}{|V|}} \middle| \Delta^2\right] = \frac{|V_n|}{|V|} \boldsymbol{D} \tag{B.25}$$

Since $\boldsymbol{D}_n$ and $\boldsymbol{D}$ are diagonal matrices, B.25 holds if, and only if, the term $\sqrt{\frac{|V_n|}{|V|}}\boldsymbol{Z}'\Delta^{-1}\Delta_n\boldsymbol{Z}_n$ is ortogonal and equals $\mathbb{I}_K$. In other words,

$$\mathbb{E}\left[\sqrt{\frac{|V_n|}{|V|}}\boldsymbol{Z}'\Delta^{-1}\Delta_n\boldsymbol{Z}_n\boldsymbol{Z}_n'\Delta_n\Delta^{-1}\boldsymbol{Z}\sqrt{\frac{|V_n|}{|V|}}\middle|\Delta^2\right] = \mathbb{I}_K$$

Thus,

$$\mathbb{E}\left[\boldsymbol{Z}'\Delta^{-1}\Delta_n\boldsymbol{Z}_n\right] = \mathbb{E}\left[\mathbb{E}\left[\boldsymbol{Z}'\Delta^{-1}\Delta_n\boldsymbol{Z}_n|\Delta^2\right]\right] = \sqrt{\frac{|V_n|}{|V|}}\mathbb{I}_K \tag{B.26}$$

Going back to Eq. B.19 and using the multivariate version of Chebyshev Inequality [91, 92], we have

$$\begin{aligned}\lim_{N\to\infty} P&\left(\left\|\left(\widehat{\boldsymbol{W}}_k - \boldsymbol{Q}^*_{ni}\right)\right\|^2 \geq \frac{\min_{l\neq k}\left\|\left(\widehat{\boldsymbol{W}}_k - \widehat{\boldsymbol{W}}_l\right)\right\|^2 \|\Sigma_k\|_F^2}{4\lambda_1\|\Sigma_k\|_F^2}\right)\\ &= P\left(\|(\boldsymbol{W}_k - \boldsymbol{Q}^*_{ni})\|^2 \geq \frac{\delta_k^2\|\Sigma_k\|_F^2}{4\|\Sigma_k\|_F^2}\right) \leq \min\left(1, \frac{4K}{\delta_k^2\|\Sigma_k\|_F^2}\right)\end{aligned} \tag{B.27}$$

where $\delta_k := \min_{l\neq k}\|\boldsymbol{W}_k - \boldsymbol{W}_l\|$ and $\boldsymbol{\Sigma}_k$ is the covariance matrix of cluster k. Also, we show that

$$\|\boldsymbol{W}_k - \boldsymbol{W}_l\|^2 = \sum_m^K |G_m|(\theta_{km} - \theta_{lm})^2, \ \ \forall l \neq k \tag{B.28}$$

Thus, expected number of misclustered nodes is, in the worst case,

$$\mathbb{E}\left[\sum_{k=1}^K |S_k^p|\right] = \sum_{k=1}^K |G_k| \min\left(1, \frac{4K}{\|\Sigma_k\|_F^2 \min_{l\neq k}\sum_m^K |G_m|(\theta_{km} - \theta_{lm})^2}\right) \tag{B.29}$$

For $N \to \infty$, Eq. (B.29) is not finite, however the misclustered rate is, i.e., $|V|^{-1}\mathbb{E}\left[\sum_{k=1}^K |S_k^p|\right] < \infty$. Thus, using Markov inequality, we have

$$P\left(\sum_{k=1}^K |S_k^p| \geq |V|\varepsilon\right) \leq (\varepsilon|V|)^{-1}\mathbb{E}\left[\sum_{k=1}^K |S_k^p|\right] \tag{B.30}$$

$$\lim_{N\to\infty} P\left(\sum_{k=1}^{K} |S_k^p| \geq |V|\varepsilon\right) = 0 \tag{B.31}$$

Now, using $\boldsymbol{A}_n$s instead of $\boldsymbol{P}_n$s, we have from Theorem 5.2 [63] that, for any $r > 0$,

$$P\left(\|\boldsymbol{A}_n - \boldsymbol{P}_n\| \leq C^0(r, c_0)\sqrt{d}\right) \geq 1 - |V_n|^{-r} \tag{B.32}$$

$$P\left(\max_n \|\boldsymbol{A}_n - \boldsymbol{P}_n\| \leq C^0(r, c_0)\sqrt{d}\right) \geq (1 - \nu^{-r})^N \tag{B.33}$$

where $d \geq \nu \max_{kl} \theta_{kl}$, $d \geq c_0 \log \nu$ and $c_0 > 0$, and $\nu = \min_n |V_n|$.

Also, define $\psi := \max_n \|\boldsymbol{A}_n - \boldsymbol{P}_n\|$. Thus,

$$P\left(\sum_{k=1}^{K} |S_k| \leq |V|\varepsilon\right) \geq P\left(\left(\sum_{k=1}^{K} |S_k| \leq |V|\varepsilon\right) \bigcap \left(\psi \leq C^0(r, c_0)\sqrt{d}\right)\right) \tag{B.34}$$

If we further assume $\nu \to \infty$ and

$$\lim_{\nu\to\infty; N\to\infty} N/\nu = 0$$

We have

$$\lim_{\nu\to\infty; N\to\infty} (1 - \nu^{-r})^N = 1 \tag{B.35}$$

Therefore,

$$|V|^{-1} \sum_{k=1}^{K} |S_k| \xrightarrow[\nu\to\infty; N\to\infty]{P} 0 \tag{B.36}$$

□

B.6 Comparing *Joint SBM* and *Isolated SBM*

B.6.1 Proof of Lemma 3.3.1

Proof For graph n, the vector of counts of nodes in each cluster has expected value given by $\mathbb{E}\left[\sum_i^{|V_n|} \boldsymbol{X}_{ni}\right] = |V_n|\boldsymbol{\zeta}$. Assuming the same distribution of the nodes over cluster for all graphs, $\mathbb{E}\left[\sum_n^N \sum_i^{|V_n|} \boldsymbol{X}_{ni}\right] = |V|\boldsymbol{\zeta}$. We know that $\sum_i^{|V_n|} \boldsymbol{X}_{ni} = \text{diag}(\boldsymbol{X}_n^T \boldsymbol{X}_n) = \text{diag}(\Delta_n^2)$ and $\sum_i^{|V|} \boldsymbol{X}_{ni} = \text{diag}(\boldsymbol{X}^T \boldsymbol{X}) = \text{diag}(\Delta^2)$. Defining $\alpha_n = |V_n|/|V|$ for all $n \in [1, ..., N]$, we have

$$\mathbb{E}\left[\Delta_n \Theta \Delta_n\right] = \sqrt{\alpha_n}\mathbb{E}\left[\Delta \Theta \Delta\right]\sqrt{\alpha_n}$$

Note that if all graphs have the same size, $|V_n|$, then $\alpha_n = N^{-1}$. Furthermore, using the eigendecomposition on both sides, we have

$$\mathbb{E}\left[\boldsymbol{Z}_n \boldsymbol{D}_n \boldsymbol{Z}_n^T\right] = \sqrt{\alpha_n}\mathbb{E}\left[\boldsymbol{Z}\boldsymbol{D}\boldsymbol{Z}^T\right]\sqrt{\alpha_n}$$

Thus, $\boldsymbol{Z}_n = \boldsymbol{Z} \iff \boldsymbol{D}_n = \alpha_n \boldsymbol{D}$.

Finally,

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{Z}_n^T \Delta_n^{-1} \Delta \mathbf{Z}\right] &= \mathbb{E}\left[\boldsymbol{Z}^T\left(\alpha_n^{-1/2}\Delta^{-1}\right)\Delta \mathbf{Z}\right] \\ &= \mathbb{E}\left[\alpha_n^{-1/2}\boldsymbol{Z}^T\boldsymbol{Z}\right] = \alpha_n^{-1/2}\end{aligned}$$

□

B.6.2 Proof of Lemma 3.3.2

Proof By Lemma 3.3.1,

$$\mathbb{E}[\eta_n(\boldsymbol{X}_n, \boldsymbol{W})] \propto \|\boldsymbol{X}_n \mathbf{W} - \mathbf{Q}_n^*\|_F^2 \, \frac{|V_n|K}{|V|}$$

where $\mathbf{Q}_n^* = \mathbf{Q}_n \frac{|V|}{|V_n|} = \boldsymbol{U}_n \boldsymbol{D} \frac{|V_n|}{|V|}\frac{|V|}{|V_n|} = \boldsymbol{U}_n \boldsymbol{D}$. Given $\frac{|V_n|}{|V|} = \frac{1}{N}$ and dropping all constants across graphs, we have $\mathbb{E}[\eta_n(\boldsymbol{X}_n, \boldsymbol{W})] \propto \|\boldsymbol{X}_n \boldsymbol{W}_n - \widehat{\boldsymbol{U}}_n\|_F^2$ □

C. APPENDIX TO CHAPTER 4

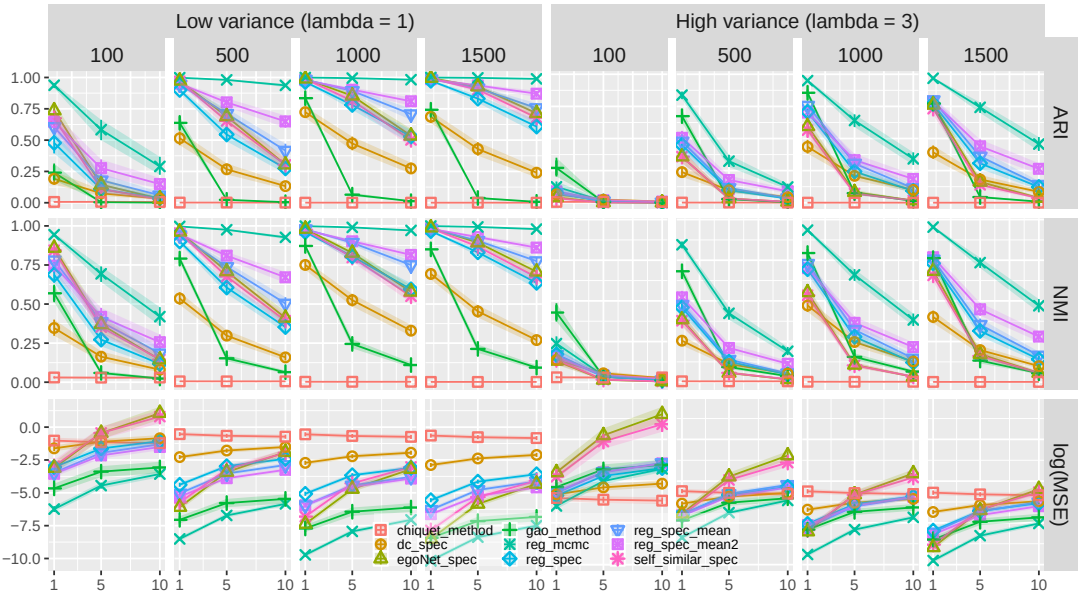

Figure C.1.: Synthetic data results using Gaussian distribution for increasing graph size (100, 500, 1000 and 1500) in two settings low variance (left) and high variance (right). Each row represent a performance metric: ARI (top) and NMI (middle) for cluster retrieval assessment, and MSE (bottom) for $\widehat{\boldsymbol{P}}$ assessment). X-axis represents increasing level of zero inflation.

C.1 Proof of Lemma 4.3.1

Proof The interaction between nodes i and j in $\boldsymbol{A}$ is generated as

$$a_{ij}|X_i, X_j, \boldsymbol{\mu}, \boldsymbol{\Sigma} \overset{ind}{\sim} N(X_i \boldsymbol{\mu} X_j^T, X_i \boldsymbol{\Sigma} X_j^T) \tag{C.1}$$

Part 1. Show that Eq. (4.9) is unbiased.

For node i where $X_i[k] = 1$, we marginalize out the membership X_j in (C.1):

$$a_{ij}|X_i, \boldsymbol{\mu} = \sum_m \pi_m \phi\left(a_{i*} - \mu_{km}\right) \tag{C.2}$$

where $\phi\left(\cdot\right)$ is the normal distribution probability density function. Thus,

$$\mathbb{E}\left[a_{ij}|X_i\right] = \sum_m \pi_m \mu_{km} = \mu^*_{k\cdot} \tag{C.3}$$

$$Var\left(a_{ij}|X_i\right) = \sum_m \pi_m(\sigma^2_{km} + \mu^2_{km}) - \left(\sum_m \pi_m \mu_{km}\right)^2 = \sigma^2_k \tag{C.4}$$

And the covariance between a_{ij} and a_{iu} is given by

$$\begin{aligned} Cov\left(a_{ij}, a_{iu}|X_i\right) &\overset{def}{=} \mathbb{E}\left[a_{ij}, a_{iu}|X_i\right] - \mathbb{E}\left[a_{ij}|X_i\right]\mathbb{E}\left[a_{iu}|X_i\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[a_{ij}, a_{iu}|X_j[m] = 1, X_u[m'] = 1\right]|X_i\right] - (\mu^*_{k\cdot})^2 \\ &= \mathbb{E}\left[\mu_{km}\mu_{km'}|X_i\right] - (\mu^*_{k\cdot})^2 = \sum_{m=1}^K \sum_{m'=1}^K \mu_{km}\mu_{km'}\pi_m\pi_{m'} - (\mu^*_{k\cdot})^2 \end{aligned}$$

Hence,

$$Cov(a_{ij}, a_{iu}) = \begin{cases} \sum_{m=1}^K \mu^2_{km}\pi^2_m - (\mu^*_{k\cdot})^2\sigma^2_{X_j} = \sigma^2_k, & \text{if } m = m' \\ \sum_{m=1}^K \mu_{km}\pi_m \sum_{m'=1}^K \mu_{km'}\pi_{m'} - (\mu^*_{k\cdot})^2 = 0, & \text{otherwise} \end{cases} \tag{C.5}$$

The correlation between nodes j and u is given by $\rho_{ju} \overset{def}{=} Cov(a_{ij}, a_{iu})/\sigma^2_k = \mathbb{I}_{X_j = X_u}$. Assume $X_j[l] = 1$, and using the correlation ρ_{ju}, we have an unbiased estimator for μ_{kl} as

$$\widehat{P}_{ij} = \left(\mathbf{1}^T \boldsymbol{W}_j^{\setminus i}\mathbf{1}\right)^{-1}\mathbf{1}^T\boldsymbol{W}_j^{\setminus i}a_{i*} \quad \text{where } \boldsymbol{W}_j^{\setminus i} = \text{diag}\left(\{\rho_{aj}\}_{a\in V\setminus i}\}\right) \tag{C.6}$$

$$= \frac{\sum_{j\in V} a^*_{ij}}{|G_l|} = \frac{\sum_{j\in G_l} a_{ij}}{|G_l|} \tag{C.7}$$

where $a^*_{iu} := \rho_{ju}a_{iu}$.

Thus,

$$\mathbb{E}\left[\widehat{P}_{ij}\right] = \frac{\sum_{j\in G_l}\mathbb{E}\left[a_{ij}\right]}{|G_l|} = \mu_{kl} \tag{C.8}$$

Part 2. Show that the transformed a^*_{iu}'s are independent $\forall u \in V \setminus i, j$.

$$\begin{aligned} Cov\left(a^*_{iw}, a^*_{iu}|X_i\right) &\stackrel{def}{=} \mathbb{E}\left[a^*_{ij}, a^*_{iu}|X_i\right] - \mathbb{E}\left[a^*_{ij}|X_i\right]\mathbb{E}\left[a^*_{iu}|X_i\right] \\ &= \mathbb{E}\left[a_{ij}\mathbb{I}_{X_j=X_w}, a_{iu}\mathbb{I}_{X_j=X_u}|X_i\right] - \mathbb{E}\left[a_{ij}\mathbb{I}_{X_j=X_w}|X_i\right]\mathbb{E}\left[a_{iu}\mathbb{I}_{X_j=X_u}|X_i\right] \end{aligned}$$

If either $X_j \neq X_w$ or $X_j \neq X_u$ then $Cov\left(a^*_{iw}, a^*_{iu}|X_i\right) = 0$. When $X_j = X_w = X_u$,

$$\begin{aligned} Cov\left(a^*_{iw}, a^*_{iu}|X_i\right) &= \mathbb{E}\left[\mathbb{E}\left[a_{ij}\mathbb{I}_{X_j=X_w}, a_{iu}\mathbb{I}_{X_j=X_u}|X_w, X_u\right]|X_i\right] - \mu^2_{kl} \\ &= \mathbb{E}\left[a_{ij}\mathbb{I}_{X_j=X_w}|X_i\right]\mathbb{E}\left[a_{iu}\mathbb{I}_{X_j=X_u}|X_i\right] - \mu^2_{kl} = 0 \end{aligned}$$

Hence, a^*_{iw}, a^*_{iu} are independent since $Cov\left(a^*_{iw}, a^*_{iu}|X_i\right) = 0\ \forall w, u \in V \setminus i, j$.

Part 3. Show $\widehat{P}_{ij}$ converges to P_{ij} almost surely.

Notice that for $n \to \infty$ then $|G_l| \to \infty$ since $\pi_l > 0$. Thus, since $\widehat{P}_{ij}$ is unbiased, and a^*_{ij}s are independent and identically distributed for any $j \in V \setminus i$, it follows from Khintchin-Kolmogorov convergence theorem.

$$\lim_{n\to\infty} \widehat{P}_{ij} \stackrel{a.s.}{\to} P_{ij} \tag{C.9}$$

■

C.2 Proof of Lemma 4.3.2

Proof This proof is divided in three parts. For parts 1 and 2, we used the Markov law to prove consistency in probability, and in part 3 we used the Slutsky's theorem. For details on these results, see [14, 93, 94].

Part 1. Show $\widehat{\rho}_{ju}$ converges in probability to ρ_{ju}.

We considered the result from [94] where they showed $\widehat{\rho}_{ju}$ is unbiased for the degenerated cases when $\rho_{ju} = 0$ and $|\rho_{ju}| = 1$. [94] also showed in Example 10.6 that the variance of $\widehat{\rho}_{ju}$ is given by

$$Var\left(\widehat{\rho}_{ju}\right) = \frac{\rho_{ju}}{n} C \tag{C.10}$$

where C is a function independent of n. Thus, since $\widehat{\rho}_{ju}$ is unbiased and $Var\left(\widehat{\rho}_{ju}\right) \overset{n\to\infty}{\to} 0$, it follows from the Markov law that $\widehat{\rho}_{ju}$ converges in probability to ρ_{ju}.

Part 2. Show $\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}}{n-1} \underset{n\to\infty}{\overset{P}{\to}} \pi_l$ and $\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{iu}a_{iu}}{n-1} \underset{n\to\infty}{\overset{P}{\to}} \pi_l\mu_{kl}$.

We know that

$$\mathbb{E}\left[\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}}{n-1}\right] = \frac{\mathbb{E}\left[\sum_{u\in V\setminus i}\mathbb{E}\left[\widehat{\rho}_{ju}\right]|\boldsymbol{X}\right]}{n-1} = \pi_l \tag{C.11}$$

$$\begin{aligned} Var\left(\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}}{n-1}\right) &= \frac{Var\left(\sum_{u\in V\setminus i}\mathbb{E}\left[\widehat{\rho}_{ju}\right]|\boldsymbol{X}\right) + \mathbb{E}\left[\sum_{u\in V\setminus i}Var\left(\widehat{\rho}_{ju}\right)|\boldsymbol{X}\right]}{(n-1)^2} \\ &= \frac{Var\left(\sum_{u\in V\setminus i}\rho_{ju}|\boldsymbol{X}\right) + \mathbb{E}\left[\sum_{u\in V\setminus i}\frac{\rho_{ju}}{n}C|\boldsymbol{X}\right]}{(n-1)^2} \\ &< \frac{\pi_l(1-\pi_l) + \left(1-\frac{1}{n}\right)\mathbb{E}\left[\rho_{ju}C_{max}|\boldsymbol{X}\right]}{(n-1)^2} \end{aligned} \tag{C.12}$$

where $C_{max} = \max_{u\in V\setminus i} C$. Thus, it follows from the Markov law that $\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}}{n-1}$ converges to π_l in probability since Eq. (C.11) is finite and Eq. (C.12) converges to 0 as n increases. Moreover,

$$\mathbb{E}\left[\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}a_{iu}}{n-1}\right] = \frac{\mathbb{E}\left[\sum_{u\in V\setminus i}\mathbb{E}\left[\widehat{\rho}_{ju}\right]a_{iu}|\boldsymbol{X}\right]}{n-1} = \pi_l\mu_{kl} \tag{C.13}$$

$$Var\left(\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}a_{ij}}{n-1}\right) = \frac{Var\left(\sum_{u\in V\setminus i}\mathbb{E}\left[\widehat{\rho}_{ju}\right]a_{ij}|\boldsymbol{X}\right) + \mathbb{E}\left[\sum_{u\in V\setminus i}Var\left(\widehat{\rho}_{ju}\right)a_{ij}|\boldsymbol{X}\right]}{(n-1)^2}$$

$$= \frac{Var\left(\sum_{u\in V\setminus i}\rho_{ju}a_{ij}|\boldsymbol{X}\right) + \mathbb{E}\left[\sum_{u\in V\setminus i}\frac{\rho_{ju}a_{ij}}{n}C|\boldsymbol{X}\right]}{(n-1)^2}$$
$$< \frac{n\sigma_{kl} + \left(1-\frac{1}{n}\right)\mathbb{E}\left[\rho_{ju}a_{ij}C_{max}|\boldsymbol{X}\right]}{(n-1)^2} \tag{C.14}$$

Again, it follows from the Markov law that $\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{iu}a_{iu}}{n-1} \underset{n\to\infty}{\overset{P}{\to}} \pi_l\mu_{kl}$ since Eq. (C.13) is finite and Eq. (C.14) converges to 0 as n increases.

Part 3. $\widehat{P}^*_{ij}$ converges in probability to P_{ij}.

Using **Part 2** results and applying the Slutsky's theorem, we have

$$\widehat{P}^*_{ij} = \left(\frac{n-1}{n-1}\right)\frac{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}a_{iu}}{\sum_{u\in V\setminus i}\widehat{\rho}_{ju}} \underset{n\to\infty}{\overset{D}{\to}} \mu_{kl} \implies \widehat{P}^*_{ij} \underset{n\to\infty}{\overset{P}{\to}} \mu_{kl} = P_{ij} \tag{C.15}$$

where $\overset{D}{\to}$ denotes convergence in distribution. Convergence in probability is implied from convergence in distribution in Eq. (C.15) since μ_{kl} is a constant.

■

C.3 Proof of Lemma 4.3.3

Proof From Section 4.2.3, define $\boldsymbol{W}_{a*}$ as the centroid of community a and the eigenvector $\boldsymbol{U}_j$ the low dimension representation of the connectivity of node j. While [63] showed the spectral clustering misclustering rate is bounded as n increases, the individual misclustering probability is not null. Hence,

$$\text{Prob}\left(\|\boldsymbol{W}_{a*} - \boldsymbol{U}_j\| < \|\boldsymbol{W}_{b*} - \boldsymbol{U}_j\| | j \in G_b\right) > 0 \qquad \text{(miscluster)} \tag{C.16}$$

Equivalently,

$$\text{Prob}\left(\mathbb{I}_{\widehat{X}_j=\widehat{X}_u} = \mathbb{I}_{X_j=X_u}|X_j \neq X_u\right) > 0 \tag{C.17}$$

Thus, it exists an ϵ where $0 < \epsilon <= 1$

$$\text{Prob}\left(\left|\mathbb{I}_{\widehat{X}_j=\widehat{X}_u} - \mathbb{I}_{X_j=X_u}\right| > \epsilon | X_j \neq X_u\right) > 0 \qquad \forall n \in \mathbb{N} \tag{C.18}$$

In other words, with probability larger than 0 the estimator $\mathbb{I}_{\widehat{X}_j=\widehat{X}_u}$ does not converge to the true $\mathbb{I}_{X_j=X_u}$ when nodes j and u do not belong to the same cluster. Using the same argument, we have

$$\text{Prob}\left(\mathbb{I}_{\widehat{X}_j=\widehat{X}_u} \neq \mathbb{I}_{X_j=X_u} | X_j = X_u\right) > 0 \tag{C.19}$$

$$\text{Prob}\left(\left|\mathbb{I}_{\widehat{X}_j=\widehat{X}_u} - \mathbb{I}_{X_j=X_u}\right| > \epsilon | X_j \neq X_u\right) > 0 \qquad \forall n \in \mathbb{N} \tag{C.20}$$

In this case, with probability larger than 0 the estimator $\mathbb{I}_{\widehat{X}_j=\widehat{X}_u}$ does not converge to the true $\mathbb{I}_{X_j=X_u}$ when nodes j and u belong to the same cluster. Results presented in Eqs.(C.18) and (C.20) imply

$$\widehat{\rho}_{ju}^{\text{stacked}} \underset{n\to\infty}{\overset{P}{\not\to}} \rho_{ju} \tag{C.21}$$

■

C.4 Complete data experiment setup

We generated graphs using the following generative process

$$\boldsymbol{\pi} \sim \text{Dir}(\alpha), \quad X_i | \boldsymbol{\pi} \sim \text{Multi}(\boldsymbol{\pi}) \tag{C.22}$$

$$\boldsymbol{\mu}[a, b] := \mu_{ab} \sim \text{Pareto}(0.5) \tag{C.23}$$

$$\boldsymbol{A}[i, j] := a_{ij} | X_i, X_j \sim \text{N}\left(X_i \boldsymbol{\mu} X_j^T, \lambda X_i \boldsymbol{\Sigma} X_j^T\right) \tag{C.24}$$

where $\boldsymbol{\Sigma}$ is a $K \times K$ matrix of variances. We fixed $K = 4$, $\alpha = 10$ (somewhat balanced nodes over clusters), and $\boldsymbol{\Sigma} = \boldsymbol{J}_K$ is a $K \times K$ matrix of ones. We generated 30 graphs for each tuple (λ, N):

- $\lambda = \{1, 5\}$: 1 is low variance and 5 is a high variance
- $N = 100, 300, ..., 2300, 2500$

C.5 Summary table of imputation methods and their limitations

Table C.1 contains the summary of each imputation method and their main limitation.

Table C.1.: Imputation methods. Assume a_{ij} where $i \in G_k$ and $j \in G_l$.

Method	$\mathcal{L}_{\mathbf{ij}} = \mathbf{1}$		$\mathcal{L}_{\mathbf{ij}} = \mathbf{0}$		**Problems when**
	$\mathbf{a_{ij}}$	**MSE**	$\mathbf{a_{ij}}$	**MSE**	
No imputation	a_{ij}	σ^2_{kl}	0	$0 + \mu^2_{kl}$	$\lvert\mu_{kl}\rvert \gg 0$
Global mean	a_{ij}	σ^2_{kl}	Eq. (4.24)	Eq. (4.28)	1. $\boldsymbol{\pi}$ is unbalanced, i.e. $\pi_k \not\approx \pi_l \forall k, l = 1, ..., K$ 2. Heterogeneous $\boldsymbol{\mu}$
Mean of means	a_{ij}	σ^2_{kl}	Eq. (4.29)	Eq. (4.36)	1. Large number of missing interactions, i.e. $\lVert\boldsymbol{\mathcal{L}}_{i*}\rVert_0 \ll n$ 2. Heterogeneous $\boldsymbol{\mu}$
Self-similar (ours)	Eq. (4.37)	$\sigma^2_{kl} + 0$	Eq. (4.37)	$\sigma^2_{kl} + 0$	Large number of missing interactions, i.e. $\lVert\boldsymbol{\mathcal{L}}_{i*}\rVert_0 \ll n$
Gao's [41]	$a_{ij}\left(\frac{2n^2}{\lVert\boldsymbol{\mathcal{L}}\rVert_0}\right)$	Eq. (4.42)	0	$0 + \mu^2_{kl}$	$\phi_{ij} \neq \phi \forall i, j = 1, ..., n$

C.6 Synthetic data experiments using Gaussian distribution

Figure C.1 shows the results for increasing graph size and different performance measures (ARI, NMI and log(MSE)). The results are very similar to the ones shown in 4.2, the main difference is that for smaller size graphs $N = 100$ the performance across methods is very poor. Our two proposed methods `self_similar_spec` and `egoNet_spec` performs very similar using Gaussian distribution. In fact, for most cases they are not statistically different.

C.7 Additional real-world experiment: clicks on news articles

We also consider the data presented in [78] where there are 384 hours of clicks on a news from a news portal in Brazil called G1. Since community ground-truth is unknown, the task is to predict future clicks based solely on past clicking behavior. For each model, we used a 5-hour training window and we test the predictions on the next hour. For each

user, we recommend 50 news articles with the highest predicted connectivity Λ, and evaluate the Hit Rate (HR@10) and the Mean Reciprocal Rank (MRR@10) for the next hour. In order to have a more precise evaluation of each model performance, we also included a recommender model CHAMELEON as a baseline. This model uses additional information such as news article embeddings, all the other models only considers user's clicking behavior.

C.7.1 Results

Table C.2 shows that the recommender system model, `CHAMELEON`, outperforms the competitors in terms of HR@10 and MRR@10, however `self_similar_spec` and `egoNet_spec` are not very behind using only clicking behavior to make the predictions.

Table C.2.: Prediction results for real-world data

Model	**HR@10**	**MRR@10**
`CHAMELEON`	**0.6738**	**0.3458**
`reg_spec`	0.3718	0.1886
`reg_mcmc`	0.3998	0.2559
`dc_spec`	0.2432	0.1498
`self_similar_spec`	**0.5593**	**0.2739**
`egoNet_spec`	0.4816	0.2643