Informative structures in complex networks

Ruizhong Miao

M.S., University of Illinois at Urbana-Champaign, United States, 2017

B.S., Fudan University, P.R.China, 2015

A Dissertation Presented to the Graduate Faculty

of the University of Virginia in Candidacy for the Degree of

Doctor of Philosophy

Department of Statistics

University of Virginia

April 2022

Abstract

Networks or graphs represent the relationships or interactions among entities and provide valuable information about the underlying data generating systems. Network data can be observed alone or accompanied by other forms of data, and in both cases, network data can be effectively leveraged to learn the underlying structures of the data. In this thesis, we consider both cases. First, when only the network data alone are observed, an unanswered question in statistical network analysis is how researchers should identify the informative component of the network data and filter out the noises. We address this problem in Chapter 2. Second, we consider the problem of integrating network data with other data modalities in the context of topic modeling in Chapter 3.

In statistical network analysis, an important task is using statistical models to describe the underlying structures. However, in practice, the structure of modeling interest is usually hidden in a larger network in which most structures are not informative. The noise and bias introduced by the non-informative component in networks can obscure the salient structure and limit many network modeling procedures' effectiveness. In Chapter 2, we introduce a novel core-periphery model for the non-informative periphery structure of networks without imposing a specific form for the informative core structure. Based on the model, we propose spectral algorithms for core identification as a data preprocessing step for general downstream network

analysis tasks. The algorithm enjoys a strong theoretical guarantee of accuracy and is scalable for large networks. We evaluate the proposed method by extensive simulation studies demonstrating various advantages over many traditional core-periphery methods. The method is applied to extract the informative core structure from a citation network and give more informative results in the downstream hierarchical community detection.

Next, in Chapter 3, we consider the problem of incorporating network data into topic models. We develop a topic model that incorporates document-level features and citation networks. To the best of our knowledge, compared with existing topic models that also incorporate the document-level features, our model takes into account two different types of causal relations between the document-level features and the topic distributions. In addition, no existing topic models were able to incorporate both network data and document-level features. We compare our proposed model to existing topic models on the same data set in terms of several automated topic model evaluation metrics. We showed that our proposed model could simultaneously achieve high held-out likelihood, coherence, and stability. Specifically, the inclusion of network data offers an improvement in topic stability.

Acknowledgements

I would like to express my sincerest gratitude to my advisors, Dr. Tianxi Li and Dr. Jordan Rodu, for their continued support and guidance. I was impressed by their immense knowledge and brilliant ideas. During our weekly meetings, they are able to show me the bigger pictures of my research, as well as dive deep into details with me. I also sincerely appreciate their willingness to listen to my perspectives. I will forever benefit from the experience of working with them.

I would also like to express special thanks to my committee members, Dr. João Sedoc and Dr. Worthy Martin, for serving on my dissertation committee, and their insightful feedback and suggestions.

I want to thank all members of the department. They admitted me into the graduate program five years ago, which started this whole exciting journey for me. Throughout my Ph.D. studies, I learned not only statistical knowledge but also how to become a better person. The courses I have taken and the discussions I have had with other fellow students will always be an asset and cherished memory for me.

I want to thank all my friends, who are always there with me whenever I need support. Lastly, I want to thank my parents for their unconditional love and support throughout all my endeavors.

Contents

Chapter 1

Introduction

Network data, representing interactions and relationships between units, have become ubiquitous with the rapid development of science and technology. Analyzing such complex and structurally novel data has resulted in a rich body of new ideas and tools in physics, mathematics, statistics, computer science, and social sciences (Strogatz, 2001; Albert and Barabási, 2002; Newman, 2003). In particular, given that complex network structures are typically noisy and complicated, treating the network as a random instantiation of a probabilistic model has been widely used to learn the structural properties while ignoring unnecessary noisy details. This approach can be traced back as early as the work of Erdös (1959). Later work of Aldous (1981); Hoover (1979) further set up foundations and frameworks for more flexible random network modeling. More recently, significant progress has been achieved to make network analysis more computationally efficient, and scientifically interpretable with theoretical guarantees (Albert and Barabási, 2002; Hoff et al., 2002; Bickel and Chen, 2009; Zhao et al., 2012; Newman, 2016a; Gao et al., 2017; Athreya et al., 2017; Mukherjee et al., 2018).

A network consists of nodes, or vertices, that are interconnected by a set of edges.

The nodes represent entities we want to model, and edges represent these entities' relations or interactions. A network can be directed or undirected depending on whether the edges have directions. We can also assign weights to the edges. These weights can represent the edges' cost, length, or importance. Examples include the World Wide Web (Broder et al., 2000), in which nodes are web pages and edges are hyperlinks that point from one page to another; the social network (Newman, 2016b), in which the nodes represent individuals, and the edges represent social connections; the coauthorship or citation network (Ji and Jin, 2016), in which the nodes are authors or academic papers, and the edges represent the coauthorship or citations.

One important task in network analysis is to accurately characterize the structure of a given network. As pointed out by Strogatz (2001), the structure of a network always affects its function. For instance, the topology of social networks affects the spread of information and disease. The popular notion of "six degrees of separation" refers to the finding that the mean geodesic distance between node pairs in a social network is small (Milgram, 1967). Another example is the transportation network. Nowadays, as a result of economic considerations and political relations between different regions, most airlines employ a hub-and-spoke philosophy, in which passengers are routed through a few hub airports (Verma et al., 2016). Perhaps the most well-studied network structure is the community structure (Fortunato, 2010), in which nodes in a graph are partitioned into clusters that are densely interconnected. In contrast, connections between different clusters are relatively sparser. A variety of models and algorithms have been proposed to address the problem of community detection in networks (Girvan and Newman, 2002; Karrer and Newman, 2011; Qin and Rohe, 2013; Jin, 2015).

Furthermore, in many statistical analysis tasks, network data are available, and incorporating such network data may improve the quality of the analysis. Various ap-

proaches have been proposed to utilize network data to improve topic modeling (Liu et al., 2009; Lim and Buntine, 2015; Lim et al., 2016). For example, in prediction models where network cohesion is present, Li et al. (2019) proposed a network-based penalty to encourage similarities between linked data points and showed that it leads to improved performance both theoretically and empirically. Another example is topic modeling: topic model is a machine learning technique for modeling a collection of text documents, and in many topic modeling tasks, in addition to the text documents, metadata is also available. These metadata include the authors, publication year, publication venue of the documents, and a network linking pairs of documents together, such as a citation network or a Twitter network.

In this thesis, we first focus on the problem of identifying the informative structures in a given network. Specifically, we assume that only part of the network has a non-trivial structure and is therefore of modeling interest. In contrast, the remaining part consists of pure noises. We use a core-periphery structure to represent this type of network, where the core component is the informative part, and the periphery component is the non-informative part. Then, we proposed an algorithm for finding the core component in a given network. We also conducted a theoretical analysis to establish performance guarantees for our proposed methods under mild assumptions. Our proposed method is then validated through extensive simulation studies and an application to a real-world data set, the Statistics Paper Citation Network (Ji and Jin, 2016; Wang et al., 2016).

Next, we focus on incorporating network data into standard statistical modeling tasks. Specifically, we focus on topic modeling of the same statistics paper data set. We developed a topic model for this data set that incorporates document-level features, such as the papers' publication time and venues and the citation network among these academic papers. We applied our topic model to analyze the abstracts of

the papers in the data set. We employed Laplacian approximation and stochastic EM algorithm for model fitting. Our estimation algorithm converges to meaningful parameter estimates. We considered several automated topic model evaluation metrics to evaluate our model and compare it to other existing topic models.

Statistics Papers with Citation Network Data Set

In this thesis, we will primarily focus on analyzing the statistics papers with citation network data set collected by Ji and Jin (2016). This data set contains all statistics papers published in four of the top journals in statistics from 2003 to the first half of 2012.

There are in total 3248 papers in this data set. For each paper in the data set, we have the abstract, several document-level features, such as authors, keywords, DOI, publication year, and publication journal. The citation network among these papers is also available. Each node of the network is a paper, and two nodes are connected if one paper cited the other. We ignored the citation direction, so the citation network is symmetrized. The average node degree in this network is 3.52. There are also 778 isolated nodes in this network, which are papers that neither cite nor receive citations from other papers in the data set.

Notations

We use capital boldface letters such as $\boldsymbol{M}$ to denote matrices. Given a matrix $\boldsymbol{M}$, $\boldsymbol{M}_{i,*}$, $\boldsymbol{M}_{*,j}$, and $\boldsymbol{M}_{i,j}$ are the i-th row, j-th column, and (i,j)-th entry, respectively. Let $\|\boldsymbol{M}\|_F$, $\|\boldsymbol{M}\|_2$, $\|\boldsymbol{M}\|_{2,\infty}$ be the Frobenius norm, the spectral norm, the two-to-infinity norm (maximum Euclidean norm of rows) of $\boldsymbol{M}$, respectively. In particular, we use $\boldsymbol{I}_d$ to denote the $d \times d$ identity matrix, and $\mathbf{1}_d$ to denote the $d \times 1$ vector whose entries are all 1. Let $\text{rank}(\boldsymbol{M})$ be the rank of $\boldsymbol{M}$, and $\boldsymbol{M}^t$ be the transpose of

$\boldsymbol{M}$. Let $[l]$ be the index set $\{1, 2, ..., l\}$. Let $\mathbb{O}_{p_1,p_2}$ be the set of $p_1 \times p_2$ matrices with orthonormal columns, and let $\mathbb{O}_p$ be the shorthand for $\mathbb{O}_{p,p}$. For any two positive sequences $\{a_n\}$ and $\{b_n\}$, we say $a_n \preceq b_n$ if there exists a positive constant C such that $a_n \leq Cb_n$ for sufficiently large n; $a_n \succeq b_n$ if $-a_n \preceq b_n$; $a_n \simeq b_n$ if $a_n \succeq b_n$ and $a_n \preceq b_n$; $a_n \succ b_n$ if for an arbitrarily large $C > 0$, $a_n > Cb_n$ for sufficiently large n.

Chapter 2

Identifying Informative Components in Complex Networks

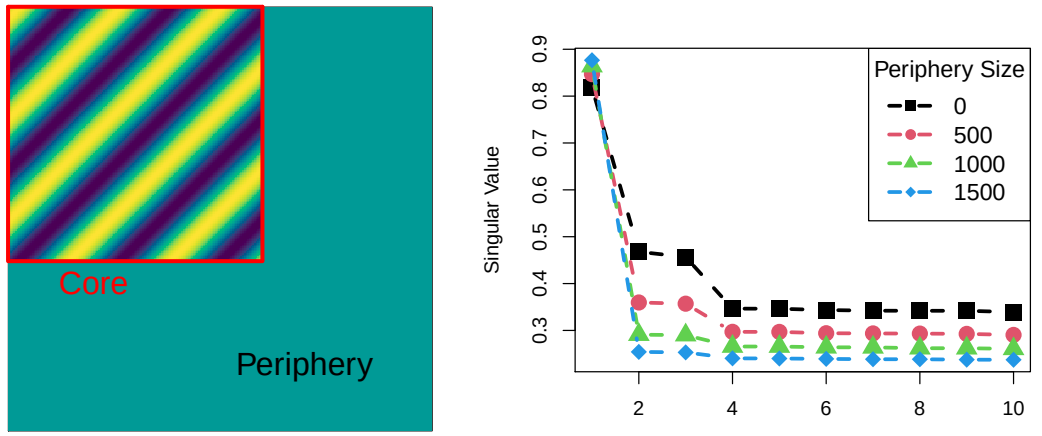
2.1 Introduction

Though many existing network analysis methods have been used to solve significant problems in different fields, empirically, they sometimes fail to learn structural information effectively. This is because most network models assume a particular type of structure of interest. However, one issue that complicates matters in practice is the scarcity of interesting or informative structures in large-scale networks. In other words, the presumed structure of interest may only be valid for a subnetwork, while the rest of the network may be noninformative. For example, it is observed by Ugander et al. (2013) that the first few moments in 100 Facebook subnetworks are very similar to the Erdös-Renyi model. Moreover, Gao and Lafferty (2017) tested these networks, observing that most of them show no evident difference from purely random connections and admit no interesting structure. For another example, preprocessing was applied in Wang et al. (2016); Li et al. (2020c,a) to remove a subset of nodes

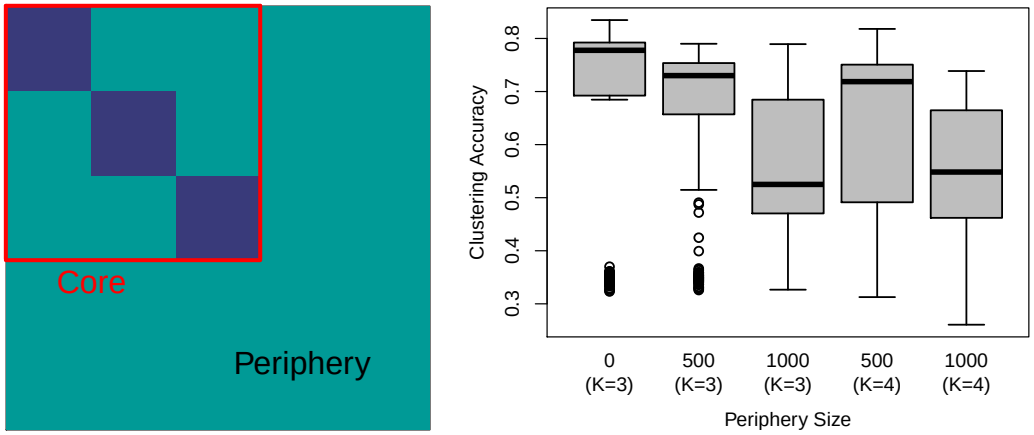
before applying community detection algorithms. Such preprocessing is reported as a crucial step for successful community analysis. In these analyses, the networks under study were assumed to have a core-periphery structure, and the k-core pruning algorithm (Seidman, 1983) was applied to the networks, which effectively removes low-degree nodes, to separate the core from the periphery. Subsequent analysis was then focused only on the core component. The motivation is that only the core component contains the structure of modeling interest, while the periphery consists of only noises. In addition, the presence of the periphery can undermine the performance of standard statistical analysis tools.

To illustrate the effect of including the periphery in the network, in Figure 2.1, we consider two examples in which the core components contain non-trivial network structures, while the periphery is generated from Erdös–Rényi model. In the first example, we plot the top eigenvalues of the random network. The core network has rank three. Hence, when the signal-to-noise ratio is manageable, we would observe a large eigengap between the 3rd and the 4th eigenvalues. As we increase the number of periphery nodes, however, the eigengap vanishes, and the model looks like rank 1, obscuring the informative structure. In the second example, the core network has a community structure, and each community has 500 nodes. We then use adjacency spectral embedding (Sussman et al., 2012) to classify its nodes into clusters while increasing the periphery size and compare the clustering results to the ground truth. The clustering accuracy decreases as the periphery size increases. Changing the number of clusters from 3 to 4 is still not an effective solution. These observations necessitate an effective preprocessing method to identify the core correctly.

The core-periphery structure has been studied in network literature for long. For example, Borgatti and Everett (2000) define the structure as a special case of the stochastic block model (Holland et al., 1983). This definition of core-periphery is used



(a) Impacts on eigengaps



(b) Impacts on community detection accuracy

Figure 2.1: Illustrations of the impacts of including the periphery component in the analysis. (a) Example 1: The impact on the eigengap of the model by including periphery nodes. The core model has rank 3, but including too many periphery nodes would overwhelm the signal, so all eigenvalues except the largest one become negligible. (b) Example 2: The impact on the accuracy of community detection. As the periphery size increases, the clustering accuracy decreases. Changing the number of clusters is not an effective solution.

by Zhang et al. (2015); Priebe et al. (2019) as well as a related problem called "planted clique problem" (Alon et al., 1998; Dekel et al., 2014). Under this definition, the network core is a densely connected Erdös-Rényi network, which is too restrictive to be interesting settings for any downstream analysis. Meanwhile, this definition heavily relies on the density gap between the core and the periphery (Zhang et al., 2015; Kojaku and Masuda, 2018) which may not be true in many applications. Naik et al. (2021) recently propose another core-periphery model. The core structure is more general than the Erdös-Rényi but still follows a restrictive parametric form. Moreover, the model can only generate networks with node degrees at least as dense as the square root of the network size, which is too dense to model most real-world networks. On the other hand, algorithm-based methods (Lee et al., 2014; Della Rossa et al., 2013; Barucca et al., 2016; Cucuringu et al., 2016; Rombach et al., 2017) typically assign a "coreness" score to each node based on certain topological assumptions. This class of methods is not well-understood in their statistical properties. Another related research problem is the submatrix localization problem (Butucea et al., 2015; Deshpande and Montanari, 2015; Hajek et al., 2017; Cai et al., 2017). The objective is to find K densely connected subgraphs planted in a large Erdös-Rényi graph in this type of problem, and the K subgraphs are usually assumed to be Erdös-Rényi graphs, which is again too restrictive in practice.

We aim to bridge the gap between the theoretically predicted effectiveness of network modeling and the empirical expectation in data analysis by proposing a principled and computationally efficient preprocessing method of extracting the informative structure from the non-informative background noise. We introduce a core-periphery model for informative and non-informative structures. The novelty of our model comes in two folds. Firstly, unlike traditional definitions, our distinction between the core and periphery components is whether the component has informative

connection patterns. Secondly, our model does not assume a specific model for the core component. These two substantive distinctions highlight the advantages of our method. Since we do not constrain our core structure to a specific network model, our framework admits the generality needed as a preprocessing step for any downstream network analysis. Meanwhile, our core-periphery definition emphasizes what we care about the most – the informative structure for network modeling. Therefore, our assumption can be phrased as an "informative-core-noninformative-periphery" structure.

Under the proposed model, we develop spectral algorithms to identify the core structure with theoretically provable guarantees. In particular, we will show that our algorithms can exactly identify the core component even on sparse networks – the so-called "strong consistency" guarantee. The strong consistency is crucial in our context (compared with its "weak consistency" cousin). This is because we design our method to be a general preprocessing step both in practice and theory. With strong consistency, the theoretical analysis for any downstream modeling of the core component remains valid by conditioning on the success of our method. On the contrary, such a seamless transition would not be available when only weak consistency is achieved.

The rest of this chapter is organized as follows. In Section 2.2, we review existing research on the core-periphery structure and the methods for finding the core component. Next, we propose our core-periphery model in Section 2.3.1 and then introduce the spectral methods for core identification under the proposed model in Section 2.3.2. Section 2.4 focuses on the theoretical properties of the algorithms concerning the accuracy of core identification. Extensive evaluations are included in Section 2.5, where we demonstrate the advantage of our method against several benchmark methods for this problem. In Section 2.6, we demonstrate our method by extracting informative

core structure from a citation network to improve downstream hierarchical community detection. We give a brief discussion of our results in Section 2.7. All the proofs of theoretical results and additional simulation examples for this chapter are included in Section 4.1.

2.2 Review of Core-Periphery Structure and Existing Methods

In its traditional definition, the core-periphery consists of two components: A network core, which is a group of central, cohesive, and densely connected nodes within the network, and a network periphery, which consists of the remaining nodes that form a sparsely connected halo or periphery surrounding the core. Although the concept of network core and periphery appeared as early as in the 1970s (Mullins et al., 1977; Alba and Moore, 1978), its formal definition is first given in Borgatti and Everett (2000). In their definitions, the core-periphery network can be defined in terms of a stochastic block model (Holland et al., 1983) consisting of a core block and a periphery block. Denote by P_{cc}, P_{cp}, P_{pp} the edge probabilities between core-core, core-periphery, and periphery-periphery. This type of definition requires $P_{cc} > P_{cp} \geq P_{pp}$ or $P_{cc} \geq P_{cp} > P_{pp}$, so the core part is a group of cohesive and densely connected nodes and the periphery is relatively sparsely connected. This definition of core-periphery structure has been widely adopted in many subsequent works and serves as the basis for their proposed methods (Zhang et al., 2015; Barucca et al., 2016; Cucuringu et al., 2016; Rombach et al., 2017).

Since the seminal paper of Borgatti and Everett (2000), various other notions of core-periphery structures have been proposed. The notion of the core-periphery

structure defined through block models is based on edge densities among different sets of nodes. Della Rossa et al. (2013) derived a core-periphery profile from a random walk. The idea is that a random walker tends to escape from the periphery, so the periphery is a subnetwork with low persistence probability. Lee et al. (2014) developed a transport-based core-periphery structure. Their idea is that core network components are used more frequently than periphery components in a transportation network, as is quantified by betweenness centrality or other similar metrics. Kojaku and Masuda (2018) considered the configuration model as the null model, against which we assess the significance of the discovered structure and showed that we need at least three blocks for the core-periphery structure to exist relative to the configuration model. Jia and Benson (2019) developed a generative core-periphery model that incorporates spatial information, as well as a nearly linear-time approximate algorithm for efficient inference and data generation. Naik et al. (2021) also developed a notion of core-periphery structure based on edge densities, not through the block model, but the scaling properties with the network size.

Our objective is to distinguish the network core from its periphery which can be a helpful step in many applications. For example, core nodes and periphery nodes can play different roles in the same network (Guimera and Amaral, 2005). The periphery may have a different community structure than the core component, and we are interested only in the structure of the core (Wang et al., 2016; Li et al., 2020a). Meanwhile, since core nodes are generally considered more "important", identifying network cores can tell us which part of the network is more important. Gu et al. (2020) applied the weighted stochastic block model (Aicher et al., 2015) to functional magnetic resonance imaging data (fMRI), and used the core score defined in Rombach et al. (2017) to examine the existence of core-periphery structure in the imaging data, which are modeled as networks. In this network, each node represents 1 of 333 cortical

areas, and each network edge is defined as the Pearson correlation coefficient between the mean BOLD time series of two nodes, followed by a Fisher's r-to-z transformation. Kojaku et al. (2019) applied an extension of the KM algorithm in Kojaku and Masuda (2017, 2018) to a global liner shipping network, and identified multiple core-periphery pairs at different resolution scales. Their original network data is bipartite. In this network, a node is either a port or a shipping route, and port i is adjacent to shipping route r if and only if port i is a calling port of route r. Then, Kojaku et al. (2019) projected this bipartite network to a one-mode network consisting of only ports. The resulting network is weighted and undirected.

For the problem of core-periphery detection, existing methods follow two general approaches: First is the algorithmic approach (Barucca et al., 2016; Cucuringu et al., 2016; Rombach et al., 2017). A "coreness" score is assigned to each network node, measuring how likely the node belongs to the core. The derivation of the coreness score is application-driven. The other is the model-based approach (Zhang et al., 2015; Jia and Benson, 2019; Naik et al., 2021), in which a parametric model is fitted to the observed network. This approach is generally more computationally intensive.

Among the algorithmic approaches, spectral methods are a family of algorithms that uses the top eigenvectors derived from the affinity, adjacency, or Laplacian matrix of the network (Ng et al., 2002; Rohe et al., 2011; Sussman et al., 2012; Qin and Rohe, 2013; Jin, 2015; Li et al., 2020a). These methods are easy to implement, and only require the top eigenvectors, which reduces the computation burden (Boutsidis et al., 2015). Spectral methods have been widely applied to graph partition and community detection problems. For the task of core-periphery detection, several spectral methods have been proposed. For the two-block core-periphery structure, Cucuringu et al. (2016) proposed a core score based on the low-rank projection of the adjacency matrix, as well as a spectral method based on the random-walk graph

Laplacian. Priebe et al. (2019) compared the performance of adjacency spectral embedding and Laplacian spectral embedding in the presence of both affinity and core-periphery structure.

2.2.1 Traditional Definitions of Core-Periphery Structure

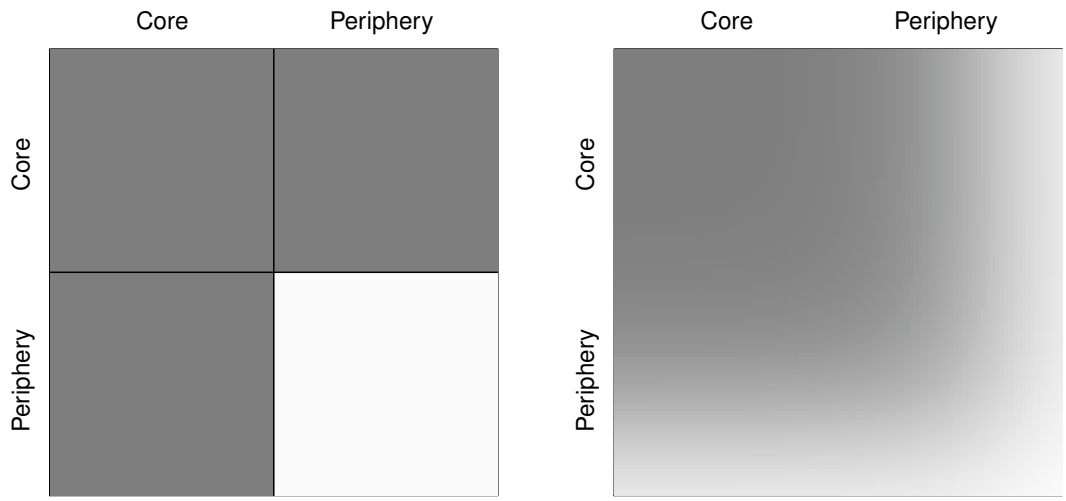
For an undirected, unweighted network with N nodes, we define its adjacency matrix $\boldsymbol{A}$ to be

$$\boldsymbol{A}_{i,j} = \begin{cases} 1, \text{ if there is an edge between node } i \text{ and node } j, \\ 0, \text{ otherwise.} \end{cases}$$

In a directed network, $A_{i,j}$ and $A_{j,i}$ represent edges pointing in the opposite direction, and therefore they do not have to be equal. In a weighted network, the elements of $\boldsymbol{A}$ take continuous values which represent edge weights. For our task, we focus on undirected, unweighted networks only.

Borgatti and Everett (2000) proposed two models for the core-periphery network, namely the discrete model and the continuous model. Figure 2.2 shows the adjacency matrices of the two models. In the discrete model (Figure 2.2a), the entries of the adjacency matrix are either 0 or 1. In an ideal core-periphery network, core nodes are fully connected to other core nodes, and periphery nodes are fully connected to the core nodes, but there are no connections between any two periphery nodes. In block modeling terms, the core-core region is a 1-block, the core-periphery regions are (possibly imperfect) 1-blocks, and the periphery-periphery region is a 0-block. It is claimed in Borgatti and Everett (2000) that this pattern is characteristic of core-periphery structures.

Formally, let $\boldsymbol{\Delta}$ denote the adjacency matrix of the ideal core-periphery network.



(a) Discrete model (b) Continuous Model

Figure 2.2: Core-periphery network structure. Darker color indicates stronger strength of the connections.

Let $\mathcal{C}$ and $\mathcal{P}$ denote the vertex sets of core vertices and periphery vertices, respectively. Then,

$$\boldsymbol{\Delta}_{i,j} = \begin{cases} 1, \text{ if } i \in \mathcal{C}, \text{ or } j \in \mathcal{C}, \\ 0, \text{ otherwise.} \end{cases} \tag{2.1}$$

In Borgatti and Everett (2000), $\boldsymbol{\Delta}$ is called the pattern matrix. Suppose $\boldsymbol{A}$ is the adjacency matrix of the observed network. Then under the discrete model, a simple measure of how close the observed network is to the ideal core-periphery structure is

$$\rho = \sum_{i,j} \boldsymbol{A}_{i,j} \boldsymbol{\Delta}_{i,j} \tag{2.2}$$

For any partition of the nodes into core and periphery, Equation (2.2) gives a score of the partition. On the basis of this score, a simple algorithm for detecting core-periphery structure can be constructed. That is, using any combinatorial optimization

technique, such as simulated annealing or genetic algorithm, to find a partition such that Equation (2.2) is maximized. Borgatti and Everett (2000) applied such algorithm on a network of co-citations among social work journals (Baker, 1992), and got a correlation of 0.54, indicating "strong but far from perfect fit with the ideal".

One limitation of the discrete model is that sometimes the dichotomy between the core and periphery is overly simplified. To remedy this, Borgatti and Everett (2000) also proposed a continuous model, in which each node is assigned a measure of "coreness". Let $\boldsymbol{c}$ be a vector of nonnegative values, whose entries, $\boldsymbol{c}_i$, indicates the coreness of node i for $i = 1, 2, ..., N$. Then, the pattern matrix is defined as

$$\boldsymbol{\Delta}_{i,j} = \boldsymbol{c}_i \boldsymbol{c}_j.$$

Under this definition, the pattern matrix will have large entries for pairs of nodes that are both in the core, intermediate entries for pairs of nodes in which one is in the core. The other is in the periphery, and small entries for pairs of nodes that are both peripheral (Figure 2.2b). We can estimate the coreness $\boldsymbol{c}$ empirically by finding a set of values $\boldsymbol{c}_i$ that maximize Equation (2.2).

We can also view the continuous model as a convex relaxation of the discrete model. In the discrete model, finding the global optimum of Equation (2.2) requires enumeration of all possible node membership assignments, which is computationally prohibitive. In the continuous model, with the appropriate constraint, Maximizing Equation (2.2) can be formulated as a convex optimization problem, which avoids the enumeration process and can be solved more computationally efficiently.

The coreness measure $\boldsymbol{c}$ can be considered as a type of centrality measure. There are many other centrality measures that are similar to the coreness. For example, if we assume the adjacency matrix is symmetric, which is the case for an undirected

network, and instead of maximizing Equation (2.2), we minimize the sum of squared difference $\sum_{i,j}(\boldsymbol{A}_{i,j} - \boldsymbol{c}_i\boldsymbol{c}_j)^2$. Then the resulting vector $\boldsymbol{c}$ will be the principal eigenvector of $\boldsymbol{A}$. If we further assume that the diagonal entries of $\boldsymbol{A}$ are not meaningful, which is the case in networks without self links, the vector $\boldsymbol{c}$ that minimizes the sum of squared differences $\sum_{i\neq j}(\boldsymbol{A}_{i,j} - \boldsymbol{c}_i\boldsymbol{c}_j)^2$ is exactly the MINRES centrality (Comrey, 1962).

We have introduced two existing core-periphery structures: The discrete model gives a clear cut between core and periphery; The continuous model assigns a continuous score to each network node to indicate how "core-like" this node is. In recent years, other notions of core-periphery structures have also been developed. We introduce two examples in the following.

Elliott et al. (2020) noted that the definition in Equation (2.1) is for undirected networks, and extended the definition in Equation (2.1) to directed networks. In their definition, the core and periphery sets depend on the edge directions. Specifically, they split each vertex set $\mathcal{C}$ and $\mathcal{P}$ into two sets. This yields four sets in total, which are denoted by $\mathcal{C}_{in}$, $\mathcal{C}_{out}$, $\mathcal{P}_{in}$, and $\mathcal{P}_{out}$. Then, the adjacency matrix of the ideal core-periphery structure for directed networks is given by the following:

$$\begin{array}{c} \\ \mathcal{P}_{out} \\ \mathcal{C}_{in} \\ \mathcal{C}_{out} \\ \mathcal{P}_{in} \end{array} \begin{array}{c} \begin{array}{cccc} \mathcal{P}_{out} & \mathcal{C}_{in} & \mathcal{C}_{out} & \mathcal{P}_{in} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The interpretation is that the two core sets are internally fully connected, while the two periphery sets have no internal connection. $\mathcal{P}_{out}$ has outgoing edges to $\mathcal{C}_{in}$,

and $\mathcal{C}_{out}$ has outgoing edges to all but $\mathcal{P}_{out}$. Elliott et al. (2020) also proposed four methods to detect such core-periphery structure. One of them is based on low-rank approximation, two of them are based on the HITS algorithm (Kleinberg, 1999), and the last one is based on likelihood-maximization.

Naik et al. (2021) noted that the block model definition in Equation (2.1) generates dense graphs, whereas real-world networks are usually sparse, so they defined the notion of core-periphery structure based on the sparsity properties of the subgraphs of core and periphery nodes.

Let $G = (G_\alpha)_{\alpha \geq 0}$ be a family of undirected graphs, where α is the size parameter. Let $G_\alpha = (V_\alpha, E_\alpha)$ where V_α and E_α are the set of vertices and edges respectively, and let $N_\alpha = |V_\alpha|$ and $N_\alpha^{(e)} = |E_\alpha|$ be the number of vertices and edges respectively. Assume $N_\alpha, N_\alpha^{(e)} \to \infty$ almost surely as $\alpha \to \infty$. Then, $(G_\alpha)_{\alpha \geq 0}$ is said to be dense if

$$N_\alpha^{(e)} = \Omega(N_\alpha^2)$$

almost surely as $\alpha \to \infty$, and it is said to be sparse if

$$N_\alpha^{(e)} = o(N_\alpha^2)$$

almost surely as $\alpha \to \infty$.

Let $(V_{\alpha,\mathcal{C}})_{\alpha \geq 0}$ be a growing sequence of vertex sets containing only the core vertices. Let $N_{\alpha,\mathcal{C}} = |V_{\alpha,\mathcal{C}}|$ be the number of core vertices, and $N_{\alpha,\mathcal{C}}^{(e)}$ be the number of edges between core vertices. Assume $N_{\alpha,\mathcal{C}}, N_{\alpha,\mathcal{C}}^{(e)} \to \infty$ as $\alpha \to \infty$. Then, Naik et al. (2021) defined that $(G_\alpha)_{\alpha \geq 0}$ is sparse with core-periphery structure if

$$N_{\alpha,\mathcal{C}}^{(e)} = \Omega(N_{\alpha,\mathcal{C}}^2)$$

and

$$N_{\alpha}^{(e)} = o(N_{\alpha,\mathcal{C}}^{2}).$$

The interpretation of the above definition is that, given a core-periphery network G, the total number of edges scales sub-quadratically with the number of vertices. Meanwhile, there exists a subgraph, namely the core subgraph, within which the number of edges scales quadratically with the number of vertices. Naik et al. (2021) also proposed a parametric model based on Poisson point process to simulate and perform posterior inference for this family of graphs.

2.2.2 Network Centralities

Existing General-Purpose Centrality Measures

In the previous section, the coreness measure can be considered as a centrality measure. There are also many other centrality measures. In this section, we review some well-known centrality measures, and how they are related to the problem of detecting core-periphery structures. We first introduce some existing general-purpose centrality measures.

The *degree centrality* of each network node is simply the degree of that node, which is defined as the number of edges incident on the node. Let $\boldsymbol{c}$ denote the vector of centralities. Then given an undirected adjacency matrix $\boldsymbol{A}$, the degree centrality of node i is $\boldsymbol{c}_i = \sum_j \boldsymbol{A}_{i,j}$.

The *eigenvector centrality* is the eigenvector corresponding to the greatest eigenvalue (Bonacich, 1987). The idea is that the centrality of each node is proportional to the sum of centralities of all its neighbors, which can be expressed as $\lambda \boldsymbol{c} = \boldsymbol{A}\boldsymbol{c}$. We can see that $\boldsymbol{c}$ is an eigenvector of the adjacency matrix $\boldsymbol{A}$. With the additional requirement that the entries of $\boldsymbol{c}$ are nonnegative, by the Perron–Frobenius theorem, $\boldsymbol{c}$

can only be the eigenvector corresponding to the largest eigenvalue. For a symmetric adjacency matrix $\boldsymbol{A}$, another view of the eigenvector centrality is that it minimizes the sum of squared differences $\sum_{i,j}(\boldsymbol{A}_{i,j} - \boldsymbol{c}_i\boldsymbol{c}_j)^2$. In this sense, it is closely related to the *MINRES centrality* which minimizes $\sum_{i\neq j}(\boldsymbol{A}_{i,j} - \boldsymbol{c}_i\boldsymbol{c}_j)^2$.

The *PageRank centrality* is a variant of the eigenvector centrality, which Google uses to rank web pages in their search results (Page et al., 1999). The PageRank algorithm outputs a probability distribution over network nodes. This probability distribution can be interpreted as the likelihood of arriving at any particular node when we travel randomly along network edges.

The *closeness centrality* measures how "close" a node is to all other nodes in the network (Sabidussi, 1966). It is defined as $\boldsymbol{c}_i = \frac{1}{\sum_j d(i,j)}$, where $d(i,j)$ is the distance between node i and node j, which is usually calculated as the length of the shortest paths between the two nodes.

The *betweenness centrality* measures how important a node is in controlling information flow through a network (Freeman, 1977). It is defined as $\boldsymbol{c}_i = \sum_{j,k\neq i}\frac{\sigma_{jk}(i)}{\sigma_{jk}}$, where σ_{jk} is the total number of shortest paths from node j to k, and $\sigma_{jk}(i)$ is the number of shortest paths from node j to k that pass through node i.

The above centrality measures are defined for individual nodes, but we can also generalize the centrality measures to network edges. Girvan and Newman (2002) extended the betweenness centrality to edges. They defined edge-betweenness as the number of shortest paths between pairs of nodes that run through it. The motivation in their paper is community detection. The intuition is that different communities are only loosely connected by a small number of inter-community edges, so the shortest paths between different communities must go through one of these few edges. Thus, the inter-community edges will have a high edge betweenness. By recursively removing these edges with high betweenness, the resulting disconnected components are

communities of the network.

Many of these centrality measures existed before the notion of core-periphery network structure appeared. Therefore, having a high centrality and being in a network core are two different concepts. By its definition, core nodes are necessarily central nodes. However, the converse is not true, as not every set of central actors forms a core. Borgatti and Everett (2000) pointed out that the key difference between a centrality measure and a coreness measure is that coreness carries with it a model of the pattern of ties in the network as a whole. The coreness measure is only interpretable to the extent that the model fits. In contrast, a centrality measure is always interpretable no matter the network's structure.

These existing centrality measures can serve as benchmarks for different core-periphery detection algorithms. Barucca et al. (2016) compared the ability of different centrality measures to identify core nodes in core-periphery networks generated by the stochastic block model with and without degree correction. The methods they investigated include membership probability marginals obtained with belief propagation, degree centrality, eigenvector centrality, PageRank, and non-backtracking centrality. They found that on the stochastic block model without degree correction, the belief propagation marginals have the best performance, and PageRank and degree centrality perform only slightly worse. When strong degree heterogeneity is present, the performance of PageRank and degree centrality surpass belief propagation marginals. Rombach et al. (2017) compared these centrality measures to the method they proposed in the paper and demonstrated that their method has better performance than these existing centrality measures. The centralities they compared include closeness, betweenness, MINRES, degree, PageRank, and aggregate core score.

Centrality measures developed for core-periphery networks

In addition to these existing centrality measures, many centrality measures explicitly devised for core-periphery detection. In the following, we list some of these centrality measures.

Aggregate core score: Rombach et al. (2017) proposed a centrality measure of coreness called *aggregate core score*. They defined the core quality as

$$R_\gamma = \sum_{i,j} \boldsymbol{A}_{i,j} \boldsymbol{C}_{i,j}, \tag{2.3}$$

where γ is a vector of parameters for the transition function, which we will introduce in the next paragraph, $\boldsymbol{C}_{i,j} = f(\boldsymbol{C}_i, \boldsymbol{C}_j)$ for some function f, and $\boldsymbol{C}_i \geq 0$ is the local core value of the ith node. In practice, they employed a product form

$$\boldsymbol{C}_{i,j} = \boldsymbol{C}_i \boldsymbol{C}_j. \tag{2.4}$$

These core values form a core vector $\boldsymbol{C}$. We seek a core vector $\boldsymbol{C}$ that maximizes Equation (2.3), with the requirement that $\boldsymbol{C}$ is normalized (so its entries add up to 1) and is a shuffle of the vector $\boldsymbol{C}^*$, whose entries $\boldsymbol{C}^*_i = g_\gamma(i)$ are determined using a transition function g_γ.

The transition function is chosen by the users. For example, Rombach et al. (2017) used the sharp function, which has the form

$$\boldsymbol{C}^*_i(\alpha, \beta) = g_{\alpha,\beta}(i) = \begin{cases} \frac{i(1-\alpha)}{2\lfloor \beta N \rfloor}, \ i \in \{1, ..., \lfloor \beta N \rfloor\}, \\ \frac{(i-\lfloor \beta N \rfloor)(1-\alpha)}{2(N-\lfloor \beta N \rfloor)} + \frac{1+\alpha}{2}, \ i \in \{\lfloor \beta N \rfloor + 1, ..., N\}. \end{cases} \tag{2.5}$$

In Equation (2.5), the parameter vector $\gamma = (\alpha, \beta)$, where $\alpha \in [0, 1]$ determines the

sharpness of the core-periphery division, and $\beta \in [0, 1]$ determines the fraction of nodes in the core. With the transition function (Equation (2.5)) and the product form (Equation (2.4)), the core quality in Equation (2.3) becomes

$$R(\alpha, \beta) = \sum_{i,j} \boldsymbol{A}_{i,j} \boldsymbol{C}_i(\alpha, \beta) \boldsymbol{C}_j(\alpha, \beta), \tag{2.6}$$

Then, we find a shuffle $\boldsymbol{C}(\alpha, \beta)$ of $\boldsymbol{C}^*(\alpha, \beta)$ that maximizes $R(\alpha, \beta)$. Beside the sharp function above, Rombach et al. (2017) also discussed some alternative transition functions to the one in Equation (2.5).

For any choice of f and transition function g_γ, the aggregate core score of each node i is defined as

$$CS(i) = Z \sum_{\gamma} \boldsymbol{C}_i(\gamma) \times R_\gamma, \tag{2.7}$$

where Z is a normalizing constant to ensure that $\max_k\{CS(k)\} = 1$.

The aggregate core score $CS(i)$ produces a continuous ranking of nodes in terms of their coreness. In their simulation study about core-periphery networks generated by the stochastic block model, Rombach et al. (2017) compared the ranking produced by the aggregated core score to that of closeness, betweenness, MINRES, Degree, and PageRank. They found that assuming we know the size of the core, the aggregate core score outperforms other measures by having a higher classification accuracy on a range of model parameters on the stochastic block model.

Path-Core: In a transportation system, some locations and routes are much more important and are used more frequently than others. These locations and routes can be considered as the core components in a transportation network, while others form the network periphery. Cucuringu et al. (2016) defined a transport-based coreness measure called *Path-Core*, which is a modification of the betweenness centrality.

Denote by E the edge set for a given network. The Path-Core score is defined as

$$PS(i) = \sum_{\substack{(j,k)\in E \\ j,k\neq i}} \frac{\sigma_{jk}(i)|_{G\setminus(j,k)}}{\sigma_{jk}|_{G\setminus(j,k)}}, \tag{2.8}$$

where $\sigma_{jk}|_{G\setminus(j,k)}$ counts the number of shortest paths between node j and k in network G after the edge (j,k) itself is removed, and $\sigma_{jk}(i)|_{G\setminus(j,k)}$ counts, among those paths, how many of them pass through node i.

From its definition, we can see that Path-Core is closely related to the betweenness centrality. We can define the Path-Core of a node i as the betweenness of this node when considering paths only between pairs of adjacent node j and k, but for which the edge (j,k) is removed. The emphasis here is that we consider "back up" paths when the direct connection between pairs of nodes is removed to amplify the usage of connections from arbitrary parts of a network to core parts.

Lee et al. (2014) compared the Path-Core to the aggregate core score and the betweenness centrality on a variety of empirical networks, including social, financial, and transportation networks. Specifically, they calculated the correlation between these scores and their correlations to other properties of the networks, and they showed that these correlations could be very different in different types of networks. In addition, they also extended the Path-Core score from nodes to edges, which allows us to assign a coreness measure to edges. This extension is important for transportation networks, since there we may want to focus on "routes" instead of "locations".

Global measures of core-periphery structures

Apart from defining centrality measures for individual network nodes, we can also define quantities for the entire network to measure how pronouncedly a network exhibits core-periphery structures. These measures are closely related to the centrality measures introduced above, as these quantities' calculations usually depend on certain centrality measures on individual nodes. We introduce two examples in the following.

Core coefficient: Intuitively, we expect the network core to be well connected to other parts of the network. Da Silva et al. (2008) introduced a parameter called *network capacity* as a measure of connectivity of a network. The network capacity is defined as

$$K = \sum_{i=1}^{m} \frac{1}{PL_i}, \tag{2.9}$$

where m is the total number of connected pairs in the network, and PL_i is the length of the shortest path between each pair. Networks with more connected pairs will have higher network capacity, and if the shortest paths between those pairs are shorter, the network capacity will also be higher.

Then, based on the network capacity and the concept of closeness centrality, they further defined a parameter called *core coefficient* (cc) to quantitatively evaluate the core-periphery structure of a network. The core coefficient is defined as

$$cc = \frac{n}{N}, \tag{2.10}$$

where N is the total number of nodes in the network, and n satisfies the equation

$$\sum_{i=1}^{n} K_i = 0.9 \sum_{j=1}^{N} K_j,$$

where K_i is the capacity of the network after removal of i nodes. The nodes are removed in order of closeness centrality.

The intuition behind this definition is that the removal of core nodes, which are assumed to be well connected and therefore have high closeness centrality, will decrease the network capacity more significantly than periphery nodes. In a network with a core-periphery structure, the first few removals of the nodes with high closeness will result in a large decrease in K, so the first few K_is will be relatively small. On the contrary, in a network not presenting core-periphery structure, the first few K_is will be relatively large, and the sum $\sum_{j=1}^{n} K_j$ accumulates faster, so in a core-periphery network, it takes a larger number of n for the sum $\sum_{i=1}^{n} K_i$ to get to $0.9\sum_{j=1}^{N} K_j$, resulting in a larger core coefficient.

Da Silva et al. (2008) calculated the core coefficient for several metabolic networks and artificial networks, and found that networks with a core-periphery structure have a generally higher core coefficient than the artificial networks without a core-periphery structure.

Core-periphery coefficient: Similar to core coefficient, the *core-periphery coefficient* proposed by Holme (2005) is another parameter to measure if the network has a clear-cut core-periphery structure. The core-periphery coefficient extends the notion of closeness centrality from one node to a subgraph. Suppose G is the given network, and $V(G)$ is the set of its vertices. For a given subgraph $U \subset V(G)$, the closeness centrality of U is defined as

$$C_C(U) = \frac{1}{\langle d(i,j)_{j \in V \setminus \{i\}} \rangle_{i \in U}}, \tag{2.11}$$

where $d(i,j)$ is the graph distance between node i and j. The angle brackets mean taking the average over $i \in U$. Then, let $V_{core}(G)$ be the core nodes of the graph

G. In Holme (2005), $V_{core}(G)$ is chosen to be the k-core of G (Seidman, 1983). The definition of core-periphery coefficient is

$$C_p(G) = \frac{C_C[V_{core}(G)]}{C_C[V(G)]} - \left\langle \frac{C_C[V_{core}(G')]}{C_C[V(G')]} \right\rangle_{G' \in \mathcal{G}(G)}. \tag{2.12}$$

Again, angle brackets mean taking the average over $G' \in \mathcal{G}(G)$, and $\mathcal{G}(G)$ is the set of graphs with the same degree distribution as G.

Computation complexities

In this subsection, we briefly discuss the computation complexities of different centrality measures. In the following, let $\boldsymbol{A}$ be the $N \times N$ adjacency matrix of the network, where N is the number of nodes, and let M be the number of edges.

For degree centrality, if the network is stored as an adjacency matrix, the computation complexity is $\mathcal{O}(N^2)$. If we use an adjacency table to store the network, the computation complexity is $\mathcal{O}(M)$.

To calculate the first eigenvector, MINRES centrality, or PageRank centrality, we can employ an iterative algorithm, in which each iteration requires one vector-matrix multiplication that requires $\mathcal{O}(N^2)$ time complexity. The number of iterations is chosen by the user, and this number determines the precision of the final estimates.

Both betweenness centrality and closeness centrality require calculating the shortest paths between all pairs of nodes in a network, for which we can use the Floyd–Warshall algorithm (Floyd, 1962), modified to not only find the length of but also the number of shortest paths between two nodes. The time complexity of this algorithm is $\Theta(N^3)$. Brandes (2001) introduced an algorithm for the shortest paths between pairs of nodes. Given a source node, both the length and the number of shortest paths to other nodes can be determined in $\mathcal{O}(M)$ using breadth-first searches (BFS) on

unweighted networks or in $\mathcal{O}(M + N\log N)$ using Dijkstra's algorithm on weighted networks. Consequently, we can iterate this algorithm over all nodes to get the lengths and numbers of shortest paths between all pairs of nodes, which requires $\mathcal{O}(NM)$ and $\mathcal{O}(NM + N^2\log N)$ on unweighted and weighted networks, respectively. When the network is sparse, this algorithm may be more efficient.

For calculating the Path-Core centrality, Cucuringu et al. (2016) also included an algorithm that runs BFSs on unweighted networks and Dijkstra's algorithm on weighted networks. The difference in betweenness centrality is that the iteration is over the edges. The total time complexity is, therefore, $\mathcal{O}(M^2)$ for unweighted networks, and $\mathcal{O}(M^2 + MN\log N)$ for weighted networks.

Calculating the aggregate core score $CS(i)$ requires averaging over the parameter γ, and for each value of γ, shuffling the vector $\boldsymbol{C}^*$ to maximize R_γ, so the aggregate core score is instead a computationally intensive approach.

Table 2.1 summarizes these results.

Table 2.1: Computation complexities of different centrality measures.

Degree	$\mathcal{O}(N^2)$ with adjacency matrix and $\mathcal{O}(M)$ with adjacency table.
Eigenvector, MINRES, PageRank	$\mathcal{O}(N^2)$ per iteration.
Betweenness, Closeness	$\Theta(N^3)$, or $\mathcal{O}(NM)$ on unweighted networks and $\mathcal{O}(NM + N^2\log N)$ on weighted networks.
Path-Core	$\mathcal{O}(M^2)$ on unweighted networks and $\mathcal{O}(M^2 + MN\log N)$ on weighted networks.

2.2.3 The Stochastic Block Model

A *stochastic block model* (SBM) is a generative model for blocks, groups, or communities in networks (Holland et al., 1983). The discrete model of the core-periphery

structure in Figure 2.2a can be formulated as an SBM, in which there are two blocks, namely the core and the periphery. Each network node belongs to one of the two groups. The SBM can be used to generate networks with a core-periphery structure. Meanwhile, we can also fit an SBM to a given network and use the fitted membership labels to classify network nodes as either core or periphery. In the following, we first introduce the definition of SBM. Then, we talk about its application to core-periphery networks.

General form and some variants

The definition of an SBM is the following:

- The network contains N nodes.
- The network contains K groups or blocks. We denote these blocks by $C_1, C_2, ..., C_K$.
- With probability γ_k, a node is assigned to C_k, for $k = 1, 2, ..., K$. This creates a partition of the N nodes into the K groups.
- For any two node i and j, suppose $i \in C_\mu$ and $j \in C_\nu$, then with probability $B_{\mu\nu}$ there is an edge between i and j. In addition, all edges are independent.

The edge probabilities $B_{\mu\nu}$ are parameters of the model, and they form a probability matrix. In traditional community structures, we assume assortativity, which means that the diagonal entries of this probability matrix are larger than the off-diagonal entries, so the nodes within each block (community) are more densely connected than nodes in different blocks. Meanwhile, if all entries of the probability matrix are the same, the SBM degenerates into an Erdös–Rényi model.

Many variants of the SBM have also been proposed. For example, one limitation of SBM is that within each block, all nodes have the same expected degree. As a

result, it does not work well in many applications to real-world networks. Especially in networks with substantial degree heterogeneity, SBM tends to categorize nodes into different groups largely based on their degrees, and therefore is unable to find the true group structures. To overcome this problem, Karrer and Newman (2011) proposed the *degree corrected stochastic block model* (DC-SBM), which incorporates heterogeneous degree distribution. Their approach is to add an additional set of parameters, $\theta_i > 0$ for each node i, that control the node degrees. Suppose $i \in C_\mu$ and $j \in C_\nu$, the probability of an edge between node i and j now becomes $\theta_i B_{\mu\nu} \theta_j$. For identifiability, a constraint is imposed:

$$\sum_{i=1}^{N} \theta_i \mathbf{1}(i \in C_k), \text{ for } k = 1, 2, ..., K.$$

$\mathbf{1}(i \in C_k)$ is the indicator function, and it equals 1 if $i \in C_k$ and 0 otherwise. This constraint means that, within each block, the summation of θ_is is 1. There are also other forms of constraints that can make the model identifiable. Compared with traditional SBM, the DC-SBM is not affected by divisions based solely on degree, and is more sensitive to the true underlying structure. Karrer and Newman (2011) showed that the DC-SBM can infer group structure better than SBM on both synthetic and real-world networks.

Newman and Peixoto (2015) proposed another generalization of the SBM. Their idea is that edge probabilities are arbitrary functions of continuous node parameters. Instead of assigning each node to a group, we assign each node to a position in a "latent space". For example, $x_i \in [0, 1]$ for node i. Then, Newman and Peixoto (2015) defined an edge function $\omega(x_i, x_j)$. The edge probability between node i and j is determined by

$$p_{ij} = \frac{d_i d_j}{2m} \omega(x_i, x_j),$$

where d_i and d_j are the degrees of the node i and j respectively, and $m = \frac{1}{2}\sum_{i=1}^{N} d_i$ is the total number of edges in the network. The inclusion of the degrees allows us to match the expected degree distribution of the model network to the distribution for the observed network. The edge function $\omega(x_i, x_j)$ can take arbitrary forms. In the paper, $\omega(x_i, x_j)$ is expressed in terms of a set of Bernstein polynomial basis functions. Newman and Peixoto (2015) also gave a method for fitting it to empirical data using Bayesian inference, and found that it successfully uncovers nontrivial structural information about both artificial and real networks.

Application to Core-Periphery Networks

We go back to the basic SBM. In the context of core-periphery network structure, there are two blocks, namely core block and periphery block. We denote the edge probabilities by p_{cc}, p_{cp}, and p_{pp}, where letter c indicates core and p indicates periphery. We typically assume that $p_{cc} \geq p_{cp} > p_{pp}$ or $p_{cc} > p_{cp} \geq p_{pp}$, so the core block is relatively more densely connected than other parts of the network. The definition of the ideal core-periphery structure, as shown in Figure 2.2a, can be expressed in terms of SBM by setting $p_{cc} = p_{cp} = 1$ and $p_{pp} = 0$.

The core-periphery networks generated by SBM have been used in many research as the benchmark networks to compare and evaluate the performance of the proposed methods (Barucca et al., 2016; Cucuringu et al., 2016; Rombach et al., 2017).

Meanwhile, on average, the best way to detect structures in a data set generated by a model is to fit the same model to the data set through the maximum likelihood. Zhang et al. (2015) proposed to perform such maximum-likelihood estimation to fit a stochastic block model to a given observed network. The maximum likelihood is implemented using expectation maximization (EM) and belief propagation. The resulting parameter estimates are edge probabilities $\{p_{cc}, p_{cp}, p_{pp}\}$, membership

probabilities $\{\gamma_c, \gamma_p\}$, and one-vertex marginal probabilities $\{q_c^i, q_p^i \text{ for } i = 1, 2, ..., N\}$, where q_k^i is the marginal probability that node i belongs to group k. Then after the parameters converge, we assign each node to either the core or the periphery, whichever has the higher marginal probability q_k^i.

Detectability

Zhang et al. (2015) also investigated the detectability problem in a core-periphery structure. The detectability problem originally arose in community detection problems. In a network with community structures, the within-group connections are usually denser than between-group connections, and if there is a strong difference between the density of the two types of connections, the community structure is easy to detect, and a variety of algorithms can do a good job. However, if the community structure is sufficiently weak, the structure can be undetectable. It is provable that under SBM, there is a threshold for the difference between within-group and between-group edge probabilities, and under this threshold, no algorithm can assign nodes to communities better than random coin toss (Mossel et al., 2012, 2018).

Since there is a connection between community detection and core-periphery detection, it is natural to ask if such detectability threshold also exists for core-periphery detection problem? As is discussed in Zhang et al. (2015), in a core-periphery network generated by SBM, core nodes have more degrees than periphery nodes on average (We can validate this by looking at Figure 2.2a), so a division based solely on degree can perform better than chance on average. So, instead of finding the detectability threshold, Zhang et al. (2015) tried to answer the question: can we do any better than simply dividing nodes according to their degrees? As is shown in the paper, there are circumstances in which division based on degree is optimal. We first reparameterize

the edge probabilities as

$$p_{cc} = \frac{c_{cc}}{N},\ p_{cp} = \frac{c_{cp}}{N},\ p_{pp} = \frac{c_{cc}}{N},$$

Then, we assume

$$c_{cc} = c + \alpha_1\delta,\ c_{cp} = c,\ c_{pp} = c - \alpha_2\delta,$$

where α_1, α_2, and c are constant, and δ is a small quantity. The case $\delta \to 0$ corresponds to the weak core-periphery structure. We consider the odds ratio q_c^i/q_p^i. Substituting these values into the maximum likelihood estimation procedure, assuming $\gamma_1 = \gamma_2 = 0.5$, and approximating terms in first order of δ, we get

$$\frac{q_c^i}{q_p^i} = 1 + \frac{1}{2}(\alpha_1 + \alpha_2)\frac{d_i - c}{c}\delta, \tag{2.13}$$

where d_i is the degree of node i. We can see that Equation (2.13) depends only on the degree of node i.

We know that the maximum likelihood estimation of the SBM is optimal, since it fits the "correct" model to the data. Also, as we have shown above, when the core-periphery structure is weak, the maximum likelihood estimate degenerates to a division based only on the degrees. This means, in a weak core-periphery structure, a division based on degree is optimal. This is analogous to the situation of weak community structure, in which we cannot do any better than random guessing. In this case, we cannot do any better than division based solely on the degrees.

In addition to the weak core-periphery structure, in the three dimensional parameter space defined by c_{cc}, c_{cp}, and c_{pp}, we can find a plane

$$c_{cc} = \theta r,\ c_{cp} = \theta,\ c_{pp} = \frac{\theta}{r}, \tag{2.14}$$

where $\theta > 0$ and $r > 1$. With these parameters, the odds ratio of the marginal probabilities becomes

$$\frac{q_c^i}{q_p^i} = \frac{\gamma_1}{\gamma_2} e^{\bar{d}_p - \bar{d}_c} r^{d_i}, \tag{2.15}$$

where $\bar{d}_c$ and $\bar{d}_p$ are average degrees in the core and periphery. Equation (2.15) also only depends on the degree of node i.

Another situation where degree-based division can perform well is when the core-periphery structure is extremely strong. In this case, the degree distributions of the core nodes and periphery nodes will be far away from each other, and we can easily distinguish between the two groups based on degree.

For the core-periphery structure of intermediate strength, also away from the plane defined by Equation (2.14), Zhang et al. (2015) showed empirically that their maximum likelihood approach can do better than simply looking at the degrees.

2.2.4 k-core Structure and k-core Pruning Algorithm

Network centralities and SBM revolve around the continuous model and the discrete model respectively. In this section, we introduce the concept of k-core, which is not based on the two models, but is defined for any type of network.

The k-core of a network G is a maximal subgraph of G in which all nodes have degree at least k (Seidman, 1983). The k-core can be obtained through the following algorithm: we recursively remove nodes whose degrees are less than k at each iteration, until all remaining nodes have degrees greater than k or the whole network disappears. If the former is the case, the remaining component is the k-core of the network. This is an efficient algorithm with linear complexity on the number of edges and nodes in the original network. In the following, we use the term k-core to refer to the maximal subgraph of G, and the term k-core pruning algorithm to refer to the

recursive algorithm.

It is easy to see that the $(k+1)$-core is a subgraph of the k-core of the same network. By setting k to different values, we can create a hierarchical structure of the network. Larger values of k correspond to more central and connected components of the network, which are surrounded by less central and sparsely connected parts corresponding to smaller values of k. The notion of k-shell of a network is defined as the set of nodes that belong to the k-core, but do not belong to the $(k+1)$-core, of the network.

The hierarchical structure unveiled by k-core algorithm can be used to visualize large networks (Alvarez-Hamelin et al., 2006). Also, k-core has been used to model the dynamics of social networks. For example, in a social network, a person's behavior is influenced by his or her connections. The k-core pruning process can be used to model the iterated withdrawals from the engagement in certain social events or from the tendency to buy products from certain brands. Bhawalkar et al. (2015) introduced the anchored k-core problem, in which a subset of nodes are "anchored". The "anchored" nodes will remain in the network no matter what their current degrees are. The "anchored" nodes are chosen so as to maximize the resulting subgraph after k-core pruning, or in the language of social science, to keep the maximum number of people engaged.

Despite its wide application, the investigation of the k-core pruning algorithm in random network setting is very limited. We introduce two examples of analysis on k-core pruning algorithm in the following.

Emergence of k-core

Dorogovtsev et al. (2006) derived exact equations describing the size and organization of k-core in a randomly damaged uncorrelated network with arbitrary degree distri-

bution (the configuration model). In a given network, suppose a fraction $Q = 1 - p$ of nodes are removed at random. Dorogovtsev et al. (2006) considered the treelike structure of the infinite sparse configuration model, in which the k-core coincides with the infinite $(k-1)$-ary subtree. (The m-ary tree is a tree, in which all nodes have branching at least m) Let R be the probability that a given end of an edge of a network is not the root of an infinite $(k-1)$-ary subtree. Then, a node belongs to the k-core if at least k of its neighbors are roots of infinite $(k-1)$-ary subtrees. Then, the probability of a node being in the k-core is

$$M(k) = p \sum_{q \geq k} P(q) \sum_{n=k}^{q} \binom{q}{n} R^{q-n}(1-R)^n. \tag{2.16}$$

The interpretation of Equation (2.16) is that, in order for a node to be in the k-core, the node must survive the random removal (times p), have degree at least k (the summation $\sum_{q \geq k} P(q)$), and among its $q > k$ edges at least k of them lead to roots of infinite $(k-1)$-ary subtrees. The quantities p, $P(q)$, and k are pre-specified, the only thing left is the value of R.

Dorogovtsev et al. (2006) also derived an equation for R based on the following intuition: an end of an edge is not a root of an infinite $(k-1)$-ary subtree if at most $(k-2)$ of its children are roots of infinite $(k-1)$-ary subtrees. This translates to

$$R = 1 - p + p \sum_{n=0}^{k-2} \left[\sum_{i=n}^{\infty} \frac{(i+1)P(i+1)}{z_1} \binom{i}{n} R^{i-n}(1-R)^n \right]. \tag{2.17}$$

The interpretation of Equation (2.17) is the following: The first term $1-p$ means the end of the edge is removed, so that end is not a root of infinite $(k-1)$-ary subtree. The second term accounts for the case when the end of the edge is present (times p). We explain this term from inside out. $z_1 = \sum_q qP(q)$ is the mean number of nearest

neighbors of a node in the network, and $\frac{(i+1)P(i+1)}{z_1}$ is the probability that a randomly chosen edge leads to a node with branching i. $\binom{i}{n}R^{i-n}(1-R)^n$ is the probability that exactly n among those i children are roots of infinite $(k-1)$-ary subtrees. So, the quantity inside the square brackets is the probability of having exactly n children that are roots of infinite $(k-1)$-ary subtrees. Then, we sum this quantity from $n=0$ to $n=k-2$, which means the end of the edge can have at most $(k-2)$ children that are roots of infinite $(k-1)$-ary subtrees, so that end itself is not a root of $(k-1)$-ary subtree.

Equation (2.17) has the trivial solution $R=1$, which means a given end of an edge is always not the root of an infinite $(k-1)$-ary subtree. Since k-core coincides with infinite $(k-1)$-ary subtree, this in turn means the k-core does not exist. The lowest nontrivial solution $R<1$ of Equation (2.17) corresponds to the emergence of the k-core.

Dorogovtsev et al. (2006) applied their results to two types of networks: First, the Erdös–Rényi network: they found as the random damage Q becomes larger, the k-cores disappear consecutively, starting from the highest core; Second, the scale free network with $P(q) \propto (q+c)^{-\gamma}$: when $\gamma > 3$, the existence of k-cores is determined by the complete form of the degree distribution including its low degree region; when $2 < \gamma \leq 3$, which is realized in most important real-world networks, there is an infinite sequence of successively enclosed k-cores. One has to remove at random almost all nodes in order to destroy any of these cores, which indicates the robustness of the entire k-core architecture in this type of networks.

Dynamics of k-core Pruning

Baxter et al. (2015) derived exact equations describing the k-core pruning process and the evolution of the network structure. They considered infinite uncorrelated sparse

random networks. Being uncorrelated means the formations of edges are independent from each other. The network is completely defined by its degree distribution $P(q)$.

Let $P(q,t)$ be the proportion of nodes having degree q at time t. It has the initial condition $P(q,0) = P(q)$. Let r_t be the probability that, we randomly follow an edge within the network at time t, and arrive at a node with degree less than k. r_t satisfies the equation

$$r_t = \frac{1}{\langle q \rangle_t} \sum_{q<k} qP(q,t), \tag{2.18}$$

Such nodes with less than k degree will be removed at time $(t+1)$, along with the edges connecting to them. $\langle q \rangle_t$ is the mean degree of the surviving nodes at time t, which satisfies the equation

$$\langle q \rangle_t = \sum_{q} qP(q,t). \tag{2.19}$$

For a node having degree $q' \geq k$ at time t. The probability of the node having degree $q > 0$ at time $(t+1)$ is $\binom{q'}{q}(1-r_t)^q r_t^{q'-q}$. This is the probability that, among its q' edges, $q'-q$ of them lead to nodes with degree less than k, and therefore will be pruned. Summing over all q', we get

$$P(q,t+1) = \sum_{q' \geq \max\{q,k\}} P(q',t) \binom{q'}{q} (1-r_t)^q r_t^{q'-q}, \tag{2.20}$$

for $q > 0$. The fraction of pruned nodes is described by the following equation

$$P(0,t+1) = \sum_{0 \leq q' < k} P(q',t) + \sum_{q' \geq k} P(q',t) r_t^{q'}. \tag{2.21}$$

Since the network we consider here is uncorrelated, Equations (2.18) to (2.21) completely describe the evolution of the network at all time. These equations can be solved numerically. When the probability r_t is very small, the pruning can then be

considered as a branching process. The probability that a vertex loses two neighbors in a single step is negligible. Under this condition, Baxter et al. (2015) developed an approximation method that allows analytical analysis.

Baxter et al. (2015) solved Equations (2.18) to (2.21) numerically for Erdös–Rényi networks (Poisson degree distributions) near the critical regime, and they found that for any $k \geq 3$, the evolution of the pruning process exhibits three different behaviors depending on whether the mean degree $\langle q \rangle_0$ of the initial network is above, equal, or below a certain threshold $\langle q \rangle_c$, whose value depends on k. $\langle q \rangle_c$ determines the existence of k-core. When $\langle q \rangle_0 > \langle q \rangle_c$, the network relaxes exponentially to the k-core; When $\langle q \rangle_0 < \langle q \rangle_c$, the network first experiences a transient process (a "plateau" stage), during which the pruning is slow. After this transient process, the network collapse in which the entire network disappears; When $\langle q \rangle_0 = \langle q \rangle_c$, the dynamics become critical, characterized by a power-law relaxation time ($\propto 1/t^2$).

2.2.5 Planted Clique Problem

In graph theory, a clique is a subgraph of an undirected graph such that every pair of its vertices are connected. A planted clique can be formed in the following way: Generated a graph of size n from Erdös–Rényi model with parameter $\frac{1}{2}$; randomly select k vertices in the generated graph, and place an edge between every pair of selected vertices, so the selected k vertices form a clique. Denote the generated graph by $G(n, \frac{1}{2}, k, 1)$. This graph can be viewed as a stochastic block model with parameter $B_{11} = 1$, and $B_{12} = B_{21} = B_{22} = 0.5$. It also has the core-periphery structure, and the core is a fully connected graph.

The *planted clique problem*, or *hidden clique problem*, is to find a clique, whose size is at least k, in $G(n, \frac{1}{2}, k, 1)$. The difficulty of the problem depends on k. Intuitively,

the larger the value of k, the easier it is to detect the planted clique. Kučera (1995) showed that, when $k \geq c\sqrt{n \log n}$ for a large enough constant c, the vertices in the planted clique have higher degrees than those outside the clique almost surely. In this case, the planted clique can be discovered effectively by a degree-based division.

Alon et al. (1998) proposed an algorithm that almost surely finds the planted clique when $k \geq c\sqrt{n}$ for a large enough constant c. Their algorithm is based on the spectral property of the adjacency matrix of the graph. Specifically, their algorithm consists of the following steps:

1. Calculate the second eigenvector of the adjacency matrix.

2. Sort the vertices in decreasing order of the absolute values of their coordinates in the second eigenvector. Let $\mathcal{W}$ be the first k vertices in this order.

3. Return all vertices with at least $3k/4$ neighbors in $\mathcal{W}$.

Dekel et al. (2014) extends the planted clique problem to the *planted dense graph problem*. Namely, let $G(n, p, k, q)$ be a family of graphs of size n and $0 < p < q \leq 1$. Let V be the set of its vertices. Let K be a subset of V and $|K| = k$. For every pair of vertices $i, j \in V$, if both i and j are in K, we place an edge between i and j with probability q; otherwise place an edge between i and j with probability p. Notably, this family of graphs is exactly a stochastic block model with parameter $B_{11} = q$, and $B_{12} = B_{21} = B_{22} = p$. The problem is to find the dense subgraph induced by K.

Let $G = (V, E)$ be the input graph, $0 < \alpha < 1$, and $\beta, \eta > 0$. The algorithm of Dekel et al. (2014) consists of three phases, which we list in the following:

1. (First phase) Iteratively find a decreasing sequence of subgraphs of G, denoted the sequence by $G = G_0 \supset G_1 \supset G_2 \supset ... \supset G_t$, with vertex set $V = V_0 \supset V_1 \supset V_2 \supset ... \supset V_t$. To find the sequence of subgraphs, we first pick a random

subset of vertices $S_i \subseteq V_{i-1}$ by including each vertex in S_i independently with probability α. Then define

$$\tilde{S}_i = \{v \in S_i : d_{S_i}(v) \geq p|S_i| + \eta\sqrt{p(1-p)|S_i|}\},$$

where $d_S(v) = |\{u \in S : \{u, v\} \in E\}|$. V_i is defined as

$$V_i = \{v \in V_{i-1} \setminus S_i : d_{\tilde{S}_i}(v) \geq p|\tilde{S}_i| + \beta\sqrt{p(1-p)|\tilde{S}_i|}\}.$$

G_i is then defined as the subgraph of G_{i-1} induced by V_i.

2. (Second phase) Let $\tilde{K}$ be the set of vertices in G_t whose degree is at least $p|V_t| + \frac{1}{2}(p+q)k_t$.

3. (Third phase) Let K' be the set of vertices containing $\tilde{K}$ and the vertices in G that have least $\frac{1}{2}(p+q)|\tilde{K}|$ neighbors in $\tilde{K}$. Let K^* be the set of vertices in G that have at least $\frac{1}{2}(p+q)k$ neighbors in K'.

4. Return K^* as the candidate for the planted dense graph.

With the correct tuning parameters, the above algorithm finds the planted dense graph in $G(n, p, k, q)$ with success probability converging to 1, for any $0 < p < q \leq 1$, and $k \geq c\sqrt{n}$, where c is a large enough constant. The running time of this algorithm is $\mathcal{O}(n^2)$.

The above algorithms were developed under the regime $k = \Omega(\sqrt{n})$. For the regime $k = \mathcal{O}(\sqrt{n})$, there exists quasi-polynomial time ($n^{\mathcal{O}(\log n)}$) algorithms to find the planted clique (Hazan and Krauthgamer, 2011; Feldman et al., 2017). First, for $k = \mathcal{O}(\log n)$, it is easy to see that the planted clique can be found in quasi-polynomial time $n^{\mathcal{O}(\log n)}$ via exhaustive search. Then, for any $k \geq 2\log n$, we can enumerate all

subsets of size $2 \log n$; for each subset that forms a clique, which we denote by S, find the set of vertices in G that are adjacent to all vertices in S, denote this set by T; return T as the candidate for the planted clique. The running time of this method is also quasi-polynomial ($n^{\mathcal{O}(\log n)}$). Up to today, it is still widely believed that there is no polynomial-time solution to the planted clique problem for any k. Such belief is called planted clique conjecture, and has been used as a computational hardness assumption.

2.3 Proposed Methodology

2.3.1 Core-Periphery Models Based on Informative Component

Assume the network size to be n. We will focus on undirected and unweighted networks without self-loops. Such a network can be represented by an $n \times n$ symmetric binary adjacency matrix $\boldsymbol{A}$ such that $\boldsymbol{A}_{i,j}$ is 1 if and only if node i and j are connected. We will embed our discussion in the following probabilistic framework for $\boldsymbol{A}$, which can be seen as a conditional version of the Aldous-Hoover representation when the network nodes are exchangeable (Aldous, 1981; Hoover, 1979). Specifically, we assume that there exists an underlying $n \times n$ probability matrix $\boldsymbol{P}$ such that $\boldsymbol{A}_{i,j} \sim \text{Bernoulli}(\boldsymbol{P}_{i,j})$, for $1 \leq i < j \leq n$ independently. We denote by $\boldsymbol{E}$ the difference between $\boldsymbol{A}$ and $\boldsymbol{P}$, i.e. $\boldsymbol{A} = \boldsymbol{P} + \boldsymbol{E}$. The elements $\{\boldsymbol{P}_{i,j}\}$ are called edge probabilities or connection probabilities. The matrix $\boldsymbol{P}$ fully specifies the structural information of the network.

In our context, the periphery component should not admit structures that may be interesting for modeling. Though whether a particular type of structure is interesting

may depend on specific applications, we believe the widely regarded *uninteresting* pattern is relatively easy to define. The following core-periphery structure is defined according to one such pattern for the periphery.

Model 1 (The ER-type core-periphery structure)**.** *The nodes in the network can be partitioned into a core set* $\mathcal{C}$ *and a periphery set* $\mathcal{P}$*, where*

$$\mathcal{P} = \{i \in [n] | \boldsymbol{P}_{i,j} = \boldsymbol{P}_{i,k}, \text{ for all } j, k \in [n], j \neq i, k \neq i\}.$$

and $\mathcal{C} = [n]/\mathcal{P}$*.*

Note that due to symmetry of $\boldsymbol{P}$, Model 1 indicates that all edges involving periphery nodes are generated randomly with the same probability resembling the Erdös-Rényi (ER) model (Erdös, 1959). The subnetwork of the core, in contrast, can follow any connection pattern as long as it is different from the periphery. Such generality in the core structure renders the flexibility to use our model as a data preprocessing step for any downstream analysis. In the special case when the core subnetwork is also an ER model but with a different density from the periphery part, the model reduces to the block model core-periphery structure used in Borgatti and Everett (2000), Zhang et al. (2015), and Priebe et al. (2019). Figure 2.3 shows one example of the core-periphery structure following Model 1.

The ER-type periphery is arguably the most basic form of non-informative structure. It also indicates that the periphery nodes should have similar degrees. Even if the nodes have heterogeneous degrees in many settings, their connection patterns may not be interesting either. One way to define such variation of the uninteresting connection only depends on two nodes separably, as defined next.

Model 2 (The configuration-type core-periphery structure)**.** *Let* d_i *be the expected*

degree of node i. The nodes in the network can be partitioned into a core set $\mathcal{C}$ and a periphery set $\mathcal{P}$, where

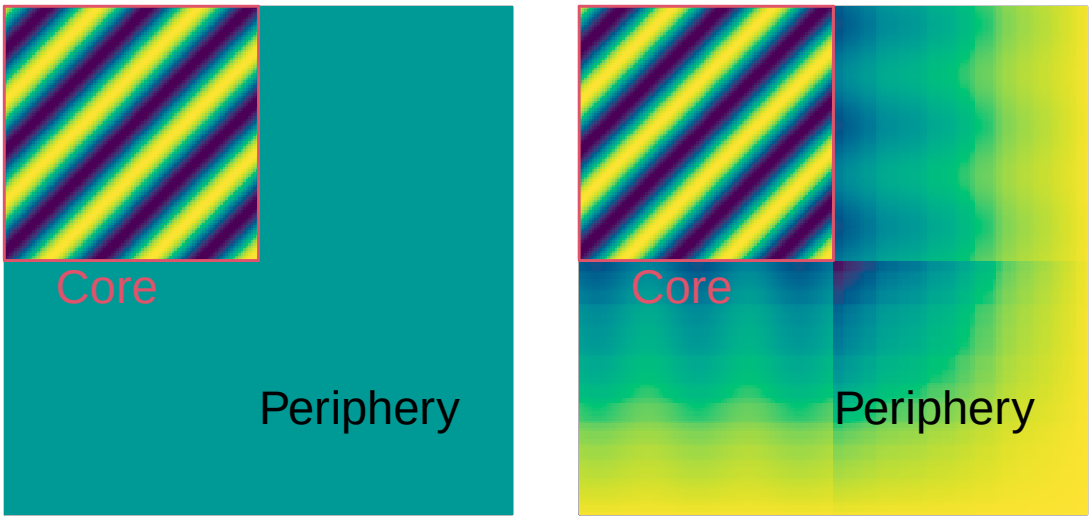
$$\mathcal{P} = \{i \in [n] | \boldsymbol{P}_{i,j} = \frac{d_i d_j}{\sum_{k=1}^{n} d_k}, \text{ for all } j \in \mathcal{V}, j \neq i\}. \tag{2.22}$$

and $\mathcal{C} = [n]/\mathcal{P}$.

The periphery connection pattern under Model 2 essentially assumes $\boldsymbol{P}_{i,j} \propto d_i d_j$ for any pair involving at least one periphery node. Such a pattern resembles the configuration model (Bollobás, 1980; Chung and Lu, 2002; Newman, 2018), where the connection probability between two nodes is based on the degree of the two nodes. Figure 2.3 illustrates this definition. Compared with the ER-type periphery, the periphery also exhibits a heterogeneous connection pattern. This model can adopt arbitrary degree distributions for the periphery nodes.

2.3.2 Spectral Algorithms for Core Identification

We proceed to introduce our algorithms to identify the core (and periphery) components under the models of Model 1 and Model 2. The likelihood-based procedures will not be applicable in the current context because we do not assume any specific model for the core subnetwork. Instead, we will resort to spectral methods for our purpose. Spectral methods have been used extensively in fitting various network models (Rohe et al., 2011; Sussman et al., 2012; Jin, 2015; Lei and Rinaldo, 2015; Qin and Rohe, 2013; Ma et al., 2020; Lei et al., 2020; Li et al., 2020b; Wang et al., 2020), which also has the advantage of computational efficiency easy implementation. The crucial step in designing such an algorithm is to find the desired spectral properties to leverage. Next, we will describe our algorithms for the ER-type model and configuration-type model separately.



(a) ER-type (b) Configuration-type

Figure 2.3: Illustrations of our core-periphery models (a) The ER-type core-periphery model, where the expected degrees of the periphery nodes are constant. (b) The configuration-type core-periphery structure, where the expected degrees of the periphery nodes are randomly sampled from a uniform distribution.

Under the ER-type model (Model 1), for any periphery node i, $\boldsymbol{P}_{i,*}$ is a vector of the same value except for the diagonal entry; for any core node i, the entries in $\boldsymbol{P}_{i,*}$ exhibit a large variation. Therefore, the core and periphery may be split according to the variation of entries in $\boldsymbol{P}_{i,*}$. Define the centering matrix $\boldsymbol{H}$ to be $\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^t$. Then $||\boldsymbol{P}_{i,*}\boldsymbol{H}||_2^2$ is the squared total variation of the entries in $\boldsymbol{P}_{i,*}$. In particular, the norm $||\boldsymbol{P}_{i,*}\boldsymbol{H}||_2$ is almost zero for $i \in \mathcal{P}$, since $\boldsymbol{P}_{i,*}$ is a constant vector except on the ith coordinate. The periphery nodes can thus be identified for small $||\boldsymbol{P}_{i,*}\boldsymbol{H}||_2$ values.

In practice, when we only observe $\boldsymbol{A}$ instead of $\boldsymbol{P}$, the above strategy would not work due to the large perturbation of $\boldsymbol{A}$ from $\boldsymbol{P}$. The solution to this difficulty is denoising $\boldsymbol{A}$ by an estimator $\hat{\boldsymbol{P}}$ and applying the above procedure to $\hat{\boldsymbol{P}}$. Notice that $\text{rank}(\boldsymbol{P}) \leq \text{rank}(\boldsymbol{P}^{\mathcal{C}}) + 1$ where $\boldsymbol{P}^{\mathcal{C}}$ is the model for the core subnetwork. Similar

Algorithm 1 Spectral algorithm for core identification from the ER-type periphery

Input: The adjacency matrix $\boldsymbol{A}$, the core size $N_{\mathcal{C}}$ and approximating rank r.

1. Find the low-rank approximation of $\boldsymbol{A}$ through rank r truncated SVD. Denote the resulting matrix by $\hat{\boldsymbol{P}}$.
2. Compute the score $S_i = ||\hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}||_2$, for $i \in [n]$.
3. Sort the scores $S_1, S_2, ..., S_n$.
4. For each $i \in [n]$, classify node i as a core node if S_i is among the top-$N_{\mathcal{C}}$ scores; otherwise classify node i as a periphery node.

properties can be obtained for many reasonable definitions of stable rank. On the other hand, as studied in Chatterjee (2015), almost all interesting network models give approximately low-rank structure. These motivate us to consider $\boldsymbol{P}$ as *approximately low-rank* (to be formally defined in our theory) and use some singular value truncating/thresholding estimator as $\hat{\boldsymbol{P}}$. The simplest estimator would be the universal singular value thresholding method of Chatterjee (2015). However, theoretically and empirically, using an adaptive way to cut off the singular values of $\boldsymbol{A}$ to a certain rank turns out to be more effective. Specifically, given a positive integer r, we use the rank-r truncated SVD of $\boldsymbol{A}$ as $\hat{\boldsymbol{P}}$. Our algorithm for Erdös-Renyi periphery defined in Model 1 is summarized in Algorithm 1. In the algorithm, we treat the approximating rank r as given. In practice, The r will be selected according to data-driven methods. In our analysis, we always use the cross-validation method of Li et al. (2020c) to select a proper r, which can be seen as a procedure the select the best low-rank approximation for link predictions.

Under the configuration-type core-periphery model (Model 2), a similar strategy can be applied with an additional modification. The key ingredient is a degree-correction step to neutralize the impacts of heterogeneous degrees. According to the

Algorithm 2 Spectral algorithm for core identification from the configuration-type periphery

Input: The adjacency matrix $\boldsymbol{A}$, the core size $N_{\mathcal{C}}$ and approximating rank r.

1. Find the low-rank approximation of $\boldsymbol{A}$ through rank r truncated SVD. Denote the resulting matrix by $\hat{\boldsymbol{P}}$.
2. Compute $\hat{d}_i = \sum_{j=1}^{n} \boldsymbol{A}_{ij}$, and let $\hat{\boldsymbol{D}} = \text{diag}\{\hat{d}_1, \hat{d}_2, ..., \hat{d}_n\}$.
3. Compute $S'_i = ||\hat{\boldsymbol{P}}_{i,*}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}||_2$, for $i \in [n]$.
4. Sort scores $S'_1, S'_2, ..., S'_n$.
5. For each $i \in [n]$, classify node i as a core node if S'_i is among the top-$N_{\mathcal{C}}$ scores; otherwise classify node i as a periphery node.

periphery connection probabilities in (2.22), for any $i \in \mathcal{P}$, we have

$$\boldsymbol{P}_{i,j}/d_j = \frac{d_i}{\sum_k d_k}, \quad \text{for any} \quad j \neq i.$$

Hence, normalizing the columns by the corresponding degrees would result in a the matrix in which the row for each periphery node is a constant, except for the diagonal entry. Define $\boldsymbol{D} = \text{diag}(d_1, \cdots, d_n)$. The column correction step can be written as $\boldsymbol{P}\boldsymbol{D}^{-1}$. After this degree-correction step, the same idea in Algorithm 1 can be applied here and we will use $||\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}\boldsymbol{H}||_2$ to separate the core nodes from the periphery nodes. In practice, $\boldsymbol{P}$ is again substituted by its estimate $\hat{\boldsymbol{P}}$, and $\boldsymbol{D}$ is replaced by its sample version $\hat{\boldsymbol{D}}$. The details are summarized in Algorithm 2.

As can be seen, the major computational burden of Algorithm 1 and 2 is on the SVD of $\boldsymbol{A}$, which is highly efficient. Thus both of the algorithms are scalable to large networks. Moreover, in the next section, we will show that these algorithms can accurately identify the core nodes even on sparse networks.

2.4 Theoretical Properties of Our Proposed Algorithms

This section will introduce a few theoretical results about the accuracy of core identification by our spectral algorithms. We will start from the ER-type model, and then the same set of theoretical properties will be extended to the configuration-type model.

2.4.1 Theory under the ER-type Model

The success of Algorithm 1 depends on the magnitude of $\|\boldsymbol{P}_{i,*}\boldsymbol{H}\|_2$ for core nodes. To quantify this magnitude, additional notations have to be introduced. First, define

$$h(n) = \min_{i\in\mathcal{C}} \|\boldsymbol{P}_{i,*}\boldsymbol{H}\|_2, \quad \text{and} \quad p^* = \max_{1\le i,j\le n} \boldsymbol{P}_{i,j}.$$

For $i \in \mathcal{P}$, it is not difficult to show that $\|\boldsymbol{P}_{i,*}\boldsymbol{H}\|_2 < p^*$, since $\boldsymbol{P}_{i,*}$ is essentially a constant vector. Therefore, a larger gap between $h(n)$ and p^* leads to a better separation between the core and periphery.

Our algorithms also relies on a good estimate of the probability matrix $\hat{\boldsymbol{P}}$. As mentioned in the previous section, we will use the rank-r truncated SVD of the observed adjacency matrix $\boldsymbol{A}$ as $\hat{\boldsymbol{P}}$. Suppose $\boldsymbol{P}$ and $\boldsymbol{A}$ admit the following eigen-decompositions:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{U}_\perp \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_\perp \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^t \\ \boldsymbol{U}_\perp^t \end{bmatrix} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t + \boldsymbol{U}_\perp\boldsymbol{\Lambda}_\perp\boldsymbol{U}_\perp^t, \tag{2.23}$$

$$\boldsymbol{A} = \begin{bmatrix} \hat{\boldsymbol{U}} & \hat{\boldsymbol{U}}_\perp \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\Lambda}} & \mathbf{0} \\ \mathbf{0} & \hat{\boldsymbol{\Lambda}}_\perp \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{U}}^t \\ \hat{\boldsymbol{U}}_\perp^t \end{bmatrix} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t + \hat{\boldsymbol{U}}_\perp\hat{\boldsymbol{\Lambda}}_\perp\hat{\boldsymbol{U}}_\perp^t, \tag{2.24}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, ..., \lambda_r\}$ and $\boldsymbol{\Lambda}_\perp = \mathrm{diag}\{\lambda_{r+1}, \lambda_{r+2}, ..., \lambda_n\}$ consist of the eigenvalues of $\boldsymbol{P}$ sorted in decreasing order. $\boldsymbol{U} \in \mathbb{O}_{n,r}$ and $\boldsymbol{U}_\perp \in \mathbb{O}_{n,n-r}$ contain corresponding eigenvectors as columns. The matrices $\hat{\boldsymbol{\Lambda}}$, $\hat{\boldsymbol{\Lambda}}_\perp$, $\hat{\boldsymbol{U}}_\perp$ and $\hat{\boldsymbol{U}}_\perp$ are similarly defined for $\boldsymbol{A}$. Our estimator of $\boldsymbol{P}$ is $\hat{\boldsymbol{P}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t$. For such a low-rank approximation to work well, we will impose the following assumptions:

Assumption 1 (Approximate low-rankness)**.** $|\lambda_r| \succeq \frac{np^*}{\sqrt{r}}$*, and* $|\lambda_{r+1}| \preceq \sqrt{p^* \log n}$*.*

Assumption 2 (Incoherence)**.** $\|\boldsymbol{U}\|_{2,\infty} \leq \mu_0\sqrt{\frac{r}{n}}$*, for a scalar* μ_0 *that may depend on* n*.*

Assumption 1 above is about the gap between the rth and $(r+1)$th eigenvalues, which is needed for low-rank approximation to be reasonable. Notice that the condition implicitly requires that $p^* \succeq \frac{r \log n}{n^2}$, which eliminates extremely sparse network models such as bounded-degree networks. However, as can be seen later on, such a requirement is trivial and will be overwritten by a stronger density requirement for a valid network concentration. The incoherence condition ensures that the entries of $\boldsymbol{P}$ spread out evenly across all nodes. Such an assumption is widely used in matrix completion and random matrix literature (Candès and Recht, 2009; Chen, 2015; Fan et al., 2018; Cape et al., 2019; Abbe et al., 2020), and is generally considered necessary for highly accurate entrywise or row/column-wise recovery of random matrices.

Theorem 1. *Assume the network* $\boldsymbol{A}$ *is generated from the ER-type model in Model 1, under Assumption 1 and Assumption 2. Algorithm 1 is used to identify the core nodes with the correct* N_C *and* r*. Furthermore, suppose* $p^* \succeq \max\left\{\frac{\mu_0^2 r \log n}{n}, \frac{\mu_0^2 r^2}{n}\right\}$,

and $|\lambda_1/\lambda_r|$ *are bounded. If*

$$h(n) \succeq \mu_0 \sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}, \tag{2.25}$$

then, for sufficiently large n*, Algorithm 1 exactly identifies the core and periphery nodes with probability at least* $1 - (B(r) + 2)n^{-\gamma}$ *for some positive constant* γ*, where* $B(r) = 10 \min\{r, 1 + \log_2(|\lambda_1/\lambda_r|)\}$.

We present Theorem 1 under the approximately low-rank condition of Assumption 1 for conciseness. The assumption can be further relaxed. The more general version of the theorem is included in Section 4.1.1. Notice that we do not assume that the density of the core subnetwork is denser than the periphery. Nor do we have to assume that the core size is in the same order as the periphery size, though the sizes' impacts are implicitly considered in $h(n)$. Such generality gives our method significant advantages in practice, as demonstrated later in Section 2.5 and 2.6.

To illustrate condition (2.25), consider the stochastic block model (SBM) as an example for the core structure, and the periphery is the ER-type model. Specifically, we consider the following balanced assortative SBM:

$$\boldsymbol{B} = (a - b)\boldsymbol{I} + b\mathbf{1}_K\mathbf{1}_K^t,$$

where $a > b > 0$, and K is the number of blocks. The edge probability matrix is $\boldsymbol{P}_{\mathcal{C}} = \rho \boldsymbol{Z}\boldsymbol{B}\boldsymbol{Z}^t$, where $\boldsymbol{Z}$ is the membership matrix, and $\boldsymbol{Z}_{i,k} = 1$ if and only if node i belongs to block k. All blocks have the same size. Then, for the core-periphery structure, suppose the core is such a balanced SBM, and the periphery is an ER network. For simplicity, we assume core and periphery have the same size. We also let the edge probability of the periphery be ρb, which is actually the worse case

scenario for Algorithm 1. In this case, it can be shown that the conditions in (2.25) becomes

$$\rho \succeq \frac{K^2 \log n}{n} + \frac{K^3}{n}. \tag{2.26}$$

If we compare (2.26) to the requirement of a network clustering algorithm, then, for balanced assortative SBM, the best requirement of a computationally feasible approach is given by the semidefinite programming (SDP) approach (Fei and Chen, 2018), which requires $\rho \succeq \frac{K \log n}{n} + \frac{K^2}{n}$. This is slightly better than Equation (2.26). However, the SDP approach is a model-based approach, which has a very specific model assumption, whereas our approach does not need a model assumption for the core. In addition, for spectral clustering, the best requirement we are aware of is from Lei (2019), which is $\rho \succeq \frac{K^3 \log n}{n}$, and this is worse than Equation (2.26).

In practice, the number of core nodes, $N_{\mathcal{C}}$, is often unknown. However, under a slightly stronger condition than Theorem 1, we can calculate a threshold such that the correct $N_{\mathcal{C}}$ can be recovered by cutting off the scores in Algorithm 1. In particular, define $\hat{p} = \frac{2}{n^2-n} \sum_{i<j} \boldsymbol{A}_{i,j}$ and replace the $N_{\mathcal{C}}$ in Step 4 of Algorithm 1 by

$$\hat{N}_{\mathcal{C}} = |\{i : S_i > \sqrt{\hat{p}^{1-\epsilon} \log n}\}| \tag{2.27}$$

for some small constant ϵ. In all of our experiments, we use $\epsilon = 0.01$. The same type of performance as (2.25) can still be theoretically guaranteed by this thresholding strategy.

Corollary 1. *Under the conditions of Theorem 1, suppose μ_0 and r are bounded. Furthermore, assume*

$$\min_{1 \le i,j \le n} \boldsymbol{P}_{ij} \simeq \max_{1 \le i,j \le n} \boldsymbol{P}_{i,j} = p^*,$$

and

$$h(n) \succ \sqrt{p^{*(1-\epsilon)} \log n}$$

for the constant ϵ *in* (2.27)*. If the* $\hat{N}_{\mathcal{C}}$ *defined by* (2.27) *is used in Algorithm 1, with sufficiently large* n*, the core and periphery can be exactly identified with probability at least* $1 - (B(r) + 4)n^{-\gamma}$ *for some positive constant* γ*.*

We conclude this section by providing an upper bound for the number of misidentified core nodes under weaker assumptions.

Theorem 2. *Assume the network* $\boldsymbol{A}$ *is generated from the ER-type model in Model 1, and Algorithm 1 is used to identify the core nodes with the correct* $N_{\mathcal{C}}$*. Suppose* $h(n) > p^*$*. Denote the number of misclassified core nodes by* M*. For a sufficiently large* n*, we have*

$$M \preceq \max\{r, \operatorname{rank}(\boldsymbol{P})\} \cdot \frac{\left(\max\{\sqrt{np^*}, \sqrt{\log n}\} + |\lambda_{r+1}|\right)^2}{\left(h(n) - p^*\right)^2} \tag{2.28}$$

with probability at least $1 - n^{-\gamma}$ *for some positive constant* γ*.*

For illustration, consider the SBM example after Theorem 1 again with $p^* \geq \log n/n$. In this case, (2.28) indicates that the misidentified number is upper bounded by $K/p^* \leq Kn/\log n = o(n)$. Such a vanishing proportion of misidentified core nodes is also called the "weak consistency". However, compared with the strong consistency of Theorem 1, the weak consistency is less useful in our scenario. This is because, as a general data preprocessing step, having strong consistency in our method ensures that the downstream theoretical analysis can still go through as if the core is already given. The weak consistency, in contrast, loses this possibility, and the downstream analysis has to consider the potential errors of the core identification and the potential dependence introduced by this preprocessing step.

2.4.2 Theory under the Configuration-type Model

Next, we consider the configuration-type model following Model 2. Recall that for a periphery node i, $\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}$ is a constant vector except for the diagonal entry. Therefore, the proof can be done by applying the same strategy of last section on the degree corrected version of $\boldsymbol{P}$. Define

$$h'(n) = \min_{i \in \mathcal{C}} ||\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}\boldsymbol{H}||_2.$$

Under the configuration-type model, the quantity $h'(n)$ has a similar role to the $h(n)$ for the ER-type model.

Theorem 3. *Assume the network $\boldsymbol{A}$ is generated from the configuration-type model in Model 2, under Assumption 1 and Assumption 2. Algorithm 2 is used to identify the core nodes with the correct $N_{\mathcal{C}}$ and r. Let $d_{\min} = \min_{1\le i\le n}\sum_{j=1}^{n}\boldsymbol{P}_{i,j}$, and suppose $d_{\min} \succ \log n$, $p^* \succ \max\left\{\frac{\mu_0^2 r\log n}{n}, \frac{\mu_0^2 r^2}{n}\right\}$, and $|\lambda_1/\lambda_r|$ is bounded. If*

$$h'(n) \succ \frac{1}{d_{\min}}\left(\mu_0\sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}\right) + \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\sqrt{\frac{\log n}{d_{\min}}}, \tag{2.29}$$

then, for sufficiently large n, Algorithm 2 exactly identifies the core and periphery nodes with probability at least $1 - (B(r)+4)n^{-\gamma}$, where $B(r) = 10\min\{r, 1 + \log_2(|\lambda_1/\lambda_r|)\}$.

Again, a more general version of the theorem is provided in the Section 4.1.1. To illustrate the condition (2.29), we consider the example when the degree-corrected stochastic block model (DC-SBM) (Karrer and Newman, 2011) is true core model. Specifically, assume that the whole network follows the DC-SBM with the first $K-1$ clusters being the core while the last cluster being the periphery. Suppose all clusters

have equal size, and K is fixed. Let $z_i \in \{1, \cdots, K\}$ be the cluster label of node i. The model can be parametrized by a sequence of node popularity parameters $\theta_i, 1 \leq i \leq n$ and a $K \times K$ matrix $\rho \boldsymbol{B}$ where $\boldsymbol{B}$ is a fixed symmetric matrix with the last row and column containing only 1's and ρ depends on n. The connection probability of this DC-SBM is given by $\boldsymbol{P}_{i,j} = \theta_i \theta_j \rho \boldsymbol{B}_{z_i, z_j}$. To ensure the identifiability of the model, we use the constraint of Zhao et al. (2012): $\sum_{z_i=k} \theta_i = n/K$. Furthermore, assume that $\boldsymbol{B}$ satisfies $\sum_{k'} \boldsymbol{B}_{k,k'} = K, 1 \leq k \leq K-1$, it can be verified that this model satisfies Model 2. Under this model, in the simplified setting such that μ_0 is bounded, $r = K$, and $\theta_i \simeq 1$ for all i, the condition (2.29) reduces to the degree requirement of $d_{\min} \succ \log n$.

Similar to the case of the ER-type model, when $N_{\mathcal{C}}$ is unknown, a threshold to cut off scores can be used to determine the core-periphery separation under slightly stronger conditions. Recall that $\hat{p} = \frac{2}{n^2 - n} \sum_{i<j} \boldsymbol{A}_{i,j}$. We can replace the $N_{\mathcal{C}}$ in Step 5 of Algorithm 2 by

$$\hat{N}'_{\mathcal{C}} = |\{i : S'_i > \frac{\sqrt{\log n}}{n\sqrt{\hat{p}^{1+\epsilon}}}\}| \tag{2.30}$$

for some small constant ϵ. In all of our experiments, we use $\epsilon = 0.01$.

Corollary 2. *Under the conditions of Theorem 3, suppose μ_0 and r are bounded. Furthermore, assume*

$$\min_{1 \leq i,j \leq n} \boldsymbol{P}_{i,j} \simeq \max_{1 \leq i,j \leq n} \boldsymbol{P}_{i,j} = p^*,$$

and

$$h'(n) \succ \frac{\sqrt{\log n}}{n\sqrt{p^{*1+\epsilon}}}$$

for the constant ϵ in (2.30)*. If the $\hat{N}'_{\mathcal{C}}$ defined by* (2.30) *is used in Algorithm 2, with a sufficiently large n, the core and periphery nodes can be exactly identified with probability at least $1 - (B(r) + 6)n^{-\gamma}$ for some positive constant γ.*

Finally, the following result is still available under weaker conditions.

Theorem 4. *Assume the network $\boldsymbol{A}$ is generated from the configuration-type model in Model 2, and Algorithm 2 is used to identify the core nodes with the correct $N_{\mathcal{C}}$. Suppose $d_{\min} \succ \log n$, and $h'(n) > \frac{d_{\max}}{(n-1)d_{\min}}$. Denote the number of misclassified core nodes by M'. Then,*

$$M' \preceq \max\{r, \operatorname{rank}(\boldsymbol{P})\} \cdot \frac{np^* + \lambda_{r+1}^2 + \|\boldsymbol{P}\boldsymbol{D}^{-1}\|_2^2 \cdot d_{\min} \cdot \log n}{d_{\min}^2 \left[h'(n) - \frac{d_{\max}}{(n-1)d_{\min}}\right]^2}$$

with probability at least $1 - \frac{3}{n^\gamma}$ for some positive constant γ.

2.5 Simulation Studies

In this section, we evaluate the performance of our proposed algorithm on finite-size synthetic networks. We will demonstrate the effectiveness and the advantage of our method under a few different core models and density gaps between the core the periphery.

In generating our networks, we always set the first $N_{\mathcal{C}}$ nodes to core. To demonstrate the flexibility with respect to the core structure, we set the core component according to the graphon models (Aldous, 1981). Specifically, the core submatrix $\boldsymbol{P}^{\mathcal{C}}$ is generated in the following way. Given a graphon function $g : [0,1] \times [0,1] \to [0,1]$, we first generate $N_{\mathcal{C}}$ i.i.d. random variables $\xi_i \sim \text{Uniform}[0,1], i = 1, \cdots, N_{\mathcal{C}}$, and then $\boldsymbol{P}^{\mathcal{C}}$ is set as

$$\boldsymbol{P}^{\mathcal{C}}_{i,j} = g(\xi_i, \xi_j), 1 \le i, j \le N_{\mathcal{C}} \tag{2.31}$$

We use three graphon functions defined in Zhang et al. (2017) as our simulation examples. The first one gives the simplest SBM for $\boldsymbol{P}^{\mathcal{C}}$ with blockwise constant structure;

Table 2.2: Graphons for simulating network cores.

Graphon function $g(\mu, \nu)$	Rank
$k/7$, if $\mu, \nu \in ((k-1)/6, k/6)$; $0.3/7$ otherwise.	6
$\sin[5\pi(\mu+\nu-1)+1]/2+0.5$	3
$1/\{1+\exp[15(0.8\lvert\mu-\nu\rvert)^{4/5}-0.1]\}$	Full

The second one still has a low-rank $\boldsymbol{P}^{\mathcal{C}}$, but does not have the nice block structure; The third model is even more complicated and generates a full-rank $\boldsymbol{P}^{\mathcal{C}}$ – this is a setting to verify the validity of our low-rank approximation strategy when the model is full-rank. The three models are summarized in Table 2.2 and the heatmaps of the $\boldsymbol{P}^{\mathcal{C}}$ in the three models are shown in Figures 2.4 and 2.5. Given $\boldsymbol{P}^{\mathcal{C}}$, we fill in the other positions of $\boldsymbol{P}$ by periphery probabilities. For the ER-type model, we simply fill in a constant value. For the configuration-type model, the construction involves multiple steps. Let $\theta_i^{\mathcal{C}} = \sum_{j=1}^{N_{\mathcal{C}}} \boldsymbol{P}_{i,j}^{\mathcal{C}}$, and sample $\theta_i^{\mathcal{P}}, i = 1, 2, ..., N_{\mathcal{P}}$ from a uniform distribution between $0.5 \min_{i\in\mathcal{C}} \theta_i$ and $1.5 \max_{i\in\mathcal{C}} \theta_i$. Then, let $\boldsymbol{\theta} = \{\theta_1^{\mathcal{C}}, \theta_2^{\mathcal{C}}, ..., \theta_{N_{\mathcal{C}}}^{\mathcal{C}}, \theta_1^{\mathcal{P}}, \theta_2^{\mathcal{P}}, ..., \theta_{N_{\mathcal{P}}}^{\mathcal{P}}\}$. The edge probability involving periphery node is set as $\boldsymbol{P}_{i,j} = \frac{\boldsymbol{\theta}_i \boldsymbol{\theta}_j}{\sum_{k=1}^{N_{\mathcal{C}}} \theta_k^{\mathcal{C}}}$. It is not difficult to see that from this procedure, $d_i = \sum_{j=1}^{n} \boldsymbol{P}_{i,j} = \frac{\boldsymbol{\theta}_i \sum_{j=1}^{n} \boldsymbol{\theta}_j}{\sum_{k=1}^{N_{\mathcal{C}}} \theta_k^{\mathcal{C}}}$, and $\boldsymbol{P}_{i,j} = \frac{d_i d_j}{\sum_{k_1}^{n} d_k}$ for $i \in \mathcal{P}$, matching Model 2.

We then rescale the generated probability matrix, so the average edge density is around 0.02. In different configurations, we vary the average degrees of core and periphery nodes to demonstrate the effects of varying density ratios between the two components. We focus on the settings where the core has an equal or higher density than the periphery [1]. The core size and periphery size are both 1000 in this section. In Section 4.1.2, we also include simulation results for imbalanced core-periphery sizes.

Several benchmark core-periphery identification methods are included in the evaluation. The first two methods are degree thresholding (Degree) and PageRank (Page

[1] Our methods perform well even if the core is sparser than the periphery. However, such a setting may be less realistic, so it is not included.

et al., 1999) thresholding (PageRank). These two centrality measures are shown to be competitive for identifying the core component in the study of (Barucca et al., 2016; Rombach et al., 2017). Theoretically, it is shown by Zhang et al. (2015) that under the SBM core-periphery model, the degree thresholding is optimal in favorable configurations. Another commonly used method is thresholding by the local clustering coefficient (Watts and Strogatz, 1998) (Local CC). The k-core pruning (k-core) algorithm (Seidman, 1983) is also included in our evaluation. It can be seen as a more adaptive version than the degree thresholding and is shown to effectively extract meaningful subnetworks in Wang et al. (2016); Li et al. (2020c,a). The final method is from Priebe et al. (2019), where the Adjacency Spectral Embedding (ASE) Sussman et al. (2012) is used to capture the core-periphery structure when both affinity and core-periphery structures are present.

To fully characterize the core identification performance, we consider the tradeoff between the true positive rate (TPR) and the false positive rate (FPR), define as

$$\text{TPR} = \frac{\#\{\text{Correctly identified nodes}\}}{\#\{\text{Identified nodes}\}} \text{ and } \text{TPR} = \frac{\#\{\text{Incorrectly identified nodes}\}}{\#\{\text{Identified nodes}\}}.$$

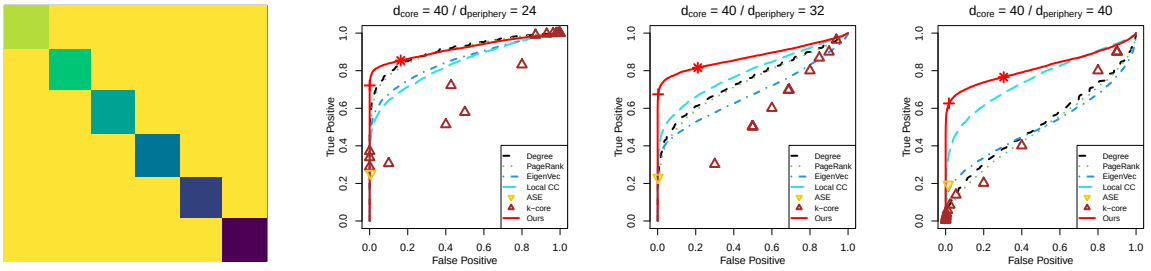
These two metrics can be shown by the receiver operating characteristic (ROC). For each thresholding-based method, the full ROC curve is obtained if by varying the threshold. The k-core pruning is applied with k increasing from 0 to the large integer, producing a sequence of points in the ROC space. The ASE, however, only gives a single point in the ROC space. For our method, we also include the single points based on our recommended threshold selection methods in Corollary 1 and 2, denoted by "$*$". Empirically, we also found that applying k-means algorithm with $k = 2$ to the log-transformed scores works well in our simulation, and we mark the point obtained this way by "+" on the ROC curves.

Figure 2.4 shows the results under the ER-type model. As can be seen, the easiest setting is when the core is much denser than the periphery. In this setting, most of the methods are reasonably good, and though our method is the most effective one, the advantage not moderate. As the density between the core and the periphery becomes more similar, the problem becomes more difficult, and some of the benchmarks become close to random guesses. However, our method still maintains good performance, and the advantage over other methods becomes more significant. This is expected since many of the benchmarks rely on the density gap between the two components while our method does not. By comparing the results across different core models, one can see that the benchmark methods may perform well under one model but fails under another. In contrast, our method remains the best one in all settings, thanks to our model's generality. Finally, the thresholds given by our theory ($*$) and k-means clustering ($+$) render good model selections in the ROC space.

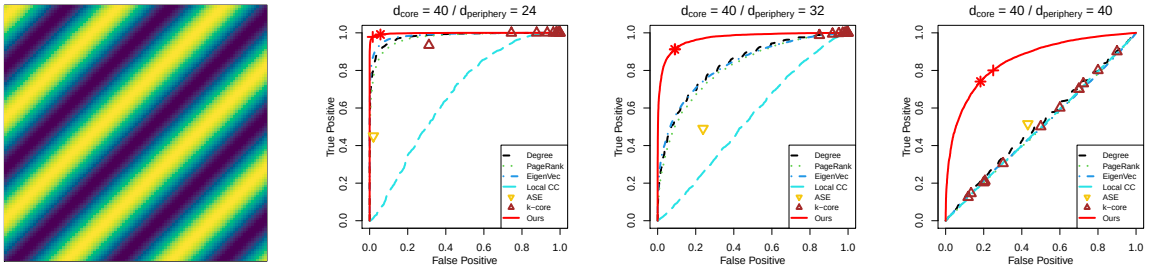
Figure 2.5 shows the results under the configuration-type model. The pattern is very similar to that of Figure 2.4. Overall, the simulation examples show that our methods outperform the benchmark methods in the core identification accuracy across various core models and varying core-periphery degree gaps.

2.6 Core Extraction in the Statistics Papers Citation Network

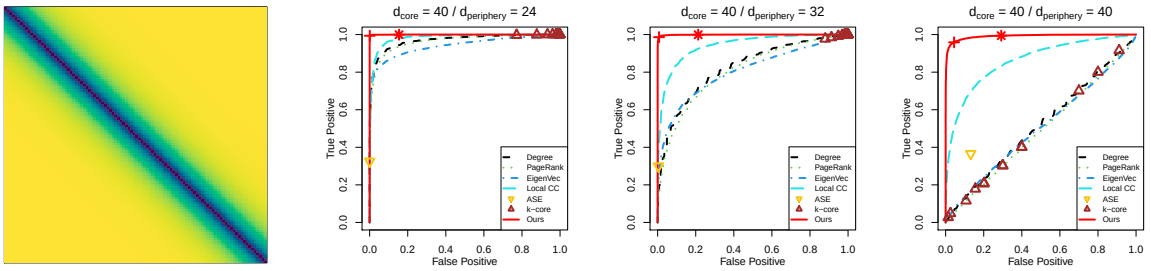
We illustrate the impact of our core extraction method in downstream community analysis for the statistics papers citation network collected by Ji and Jin (2016). We focus on the largest connected component of the network. This network has 2248 nodes and the average node degree is 4.95. In Figure 2.6, we plot the whole citation



(a) Graphon 1

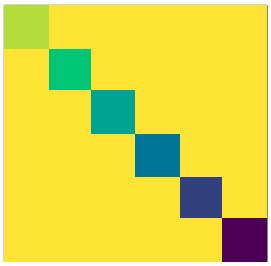
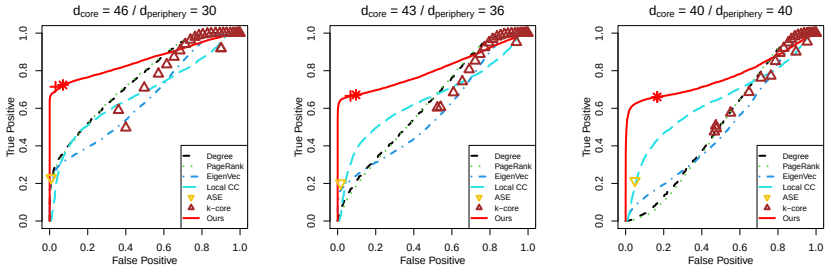


(b) Graphon 2

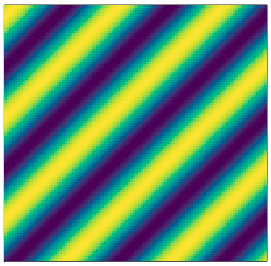
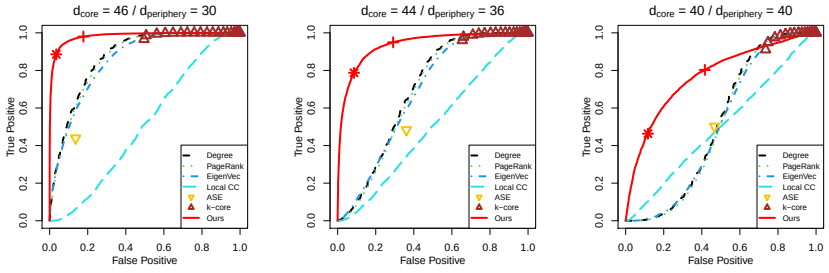


(c) Graphon 3

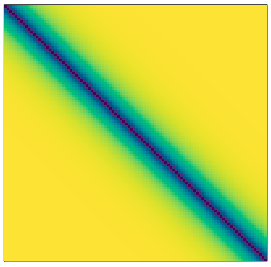
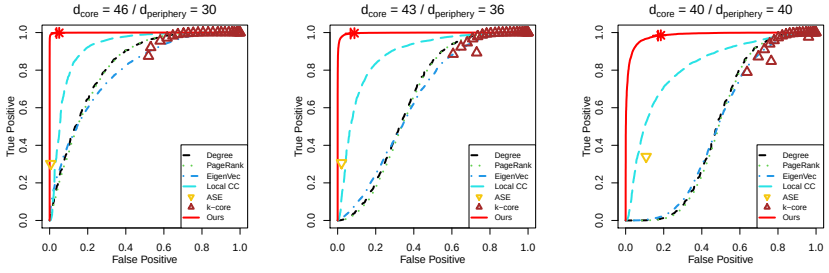
Figure 2.4: Simulation results under ER-type core-periphery model where $N_{\mathcal{C}} = N_{\mathcal{P}} = 1000$. The left figures are the core graphon functions, and the corresponding ROC curves are shown on the right, under different degree-gaps between core and periphery. The point "$*$" gives the model selection based on Corollary 1, and "$+$" indicates the model selection by k-means clustering with $k = 2$.

(a) Graphon 1

(b) Graphon 2

(c) Graphon 3

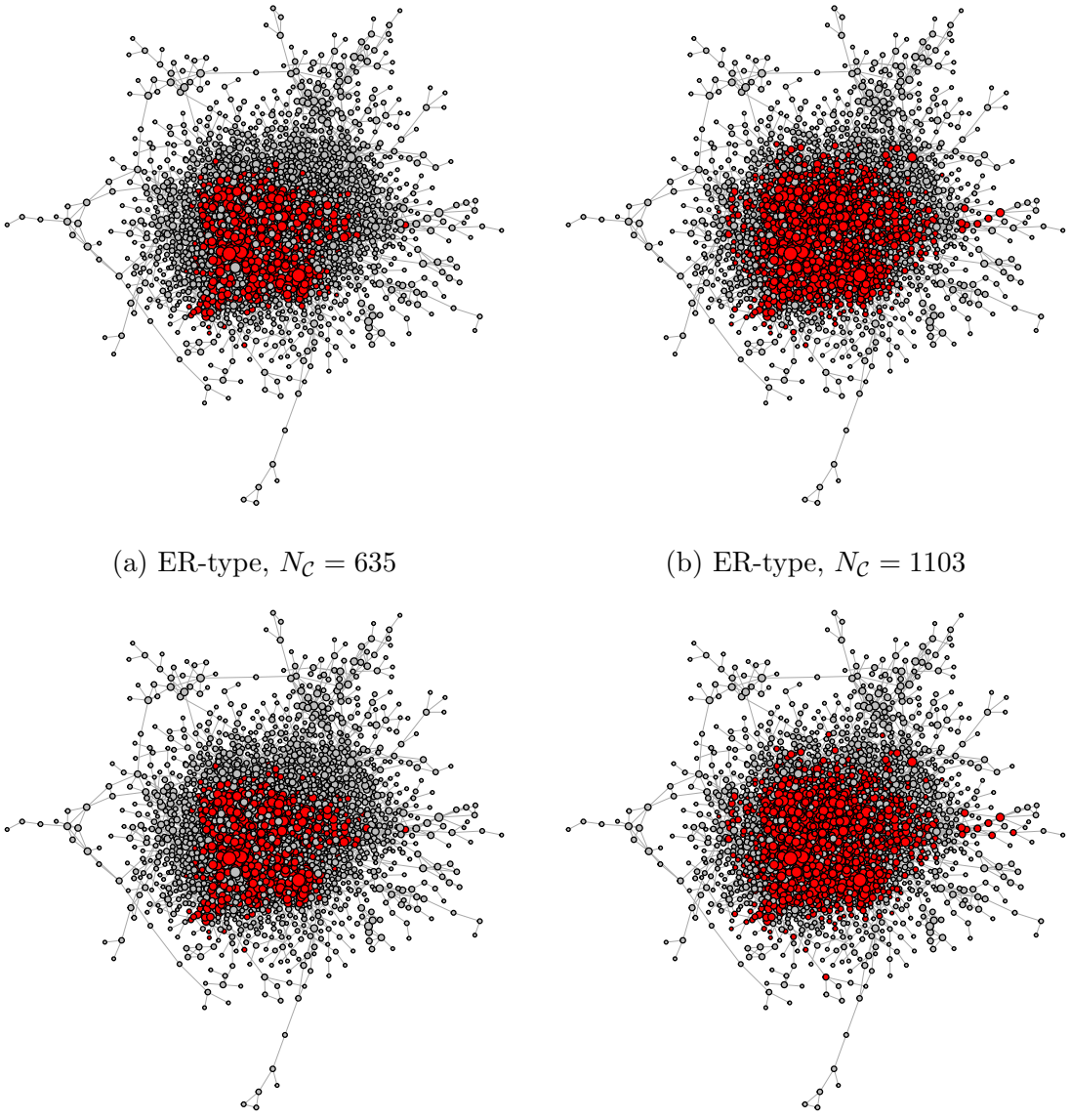
Figure 2.5: Simulation results under configuration-type core-periphery model where $N_{\mathcal{C}} = N_{\mathcal{P}} = 1000$. The left figures are the core graphon functions, and the corresponding ROC curves are shown on the right, under different degree-gaps between core and periphery. The point "$*$" gives the model selection based on Corollary 2, and "$+$" indicates the model selection by k-means clustering with $k = 2$.

network, and the core component extracted by Algorithm 1 and Algorithm 2, with two different core sizes. The core sizes are selected to match that of the k-core algorithm, for easy comparison between the two approaches.

In the analysis of Wang et al. (2016), the 4-core pruning is applied to the network, resulting in a core of 635 nodes for their downstream analysis. In this example, we compare several methods in Section 2.5 and evaluate the performance by comparing the validity of the hierarchical community detection results on the extracted cores. For fair comparisons, we follow Wang et al. (2016) to use either 3-core and 4-core pruning algorithms to obtain cores of size 1103 and 635, respectively. We then use other algorithms to extract cores of the same sizes. In addition to our methods, the other benchmark methods applicable for this task include degree centrality, eigenvector centrality, PageRank centrality, and local clustering coefficient.

The hierarchical community detection (HCD) algorithm from Li et al. (2020a) is then applied to the extracted cores. The HCD simultaneously detects the community membership and the hierarchical relation between the communities in the form of a binary tree. According to Li et al. (2020a), this hierarchical relationship can be transformed into a similarity matrix $\boldsymbol{S}$ where $\boldsymbol{S}_{k,k'}$ measures the similarity between community k and k' along the hierarchy. The tuning parameter $n.min$ in HCD, which determines the leaf node size, is set to $N_{\mathcal{C}}/r$, the core size divided by the estimated rank, so the trees across different core sizes will have similar depths.

We want to evaluate the meaningfulness of the hierarchical relationships in a quantitative way by comparing the hierarchical similarity $\boldsymbol{S}$ (based on the citation network) with the content similarity based on text data. In particular, the abstracts of all papers are available from Wang et al. (2016). We represent each abstract as a term-frequency vector and apply the standard text mining processing such as



(a) ER-type, $N_{\mathcal{C}} = 635$

(b) ER-type, $N_{\mathcal{C}} = 1103$

(c) Configuration-type, $N_{\mathcal{C}} = 635$

(d) Configuration-type, $N_{\mathcal{C}} = 1103$

Figure 2.6: Plots of the citation network, and the core components are highlighted in red.

Table 2.3: Correlation between $\boldsymbol{S}$ and $\boldsymbol{T}$.

Methods	Correlation	
	$N_{\mathcal{C}} = 635$	$N_{\mathcal{C}} = 1103$
Degree	0.099	0.089
k-core	0.167	0.108
PageRank	0.013	0.106
EigenVec	0.143	0.050
Local CC	0.058	0.045
Ours (ER)	**0.340**	**0.164**
Ours (Config)	**0.350**	**0.155**

stemming and stopwords (including punctuations and numbers) removal[2]. The term frequency-inverse document frequency (TF- IDF) weighting (Rajaraman and Ullman, 2011) is then applied to each word. We remove words that appear in less than 1% of the papers, and 966 words remain after processing. The correlation similarity between each pair of papers is calculated, and a community level similarity matrix $\boldsymbol{T}$ is constructed where $\boldsymbol{T}_{k,k'}$ is the average correlation similarity between papers from community k and community k'. We then calculate the Spearman correlation between $\boldsymbol{S}$ and $\boldsymbol{T}$ as a metric to measure how well the hierarchical structure discovered by HCD from the network matches the similarity derived from the abstracts. The results for cores extracted by different methods are summarized in Table 2.3.

It can be seen that the cores extracted by both of our two models render significantly more meaningful hierarchies than the other benchmarks. The difference between the ER-type model and the configuration-type model is negligible. Also, applying HCD to the two cores from the ER-type model and the configuration-type model leads to the same hierarchical structure, with some marginal differences.

Figure 2.7 shows the extracted core by the configuration-type model with $N_{\mathcal{C}} = 635$, and the corresponding hierarchical structure given by the HCD algorithm. It

[2]We use the SMART information retrieval system. The list can be found in the `stopwords` R package

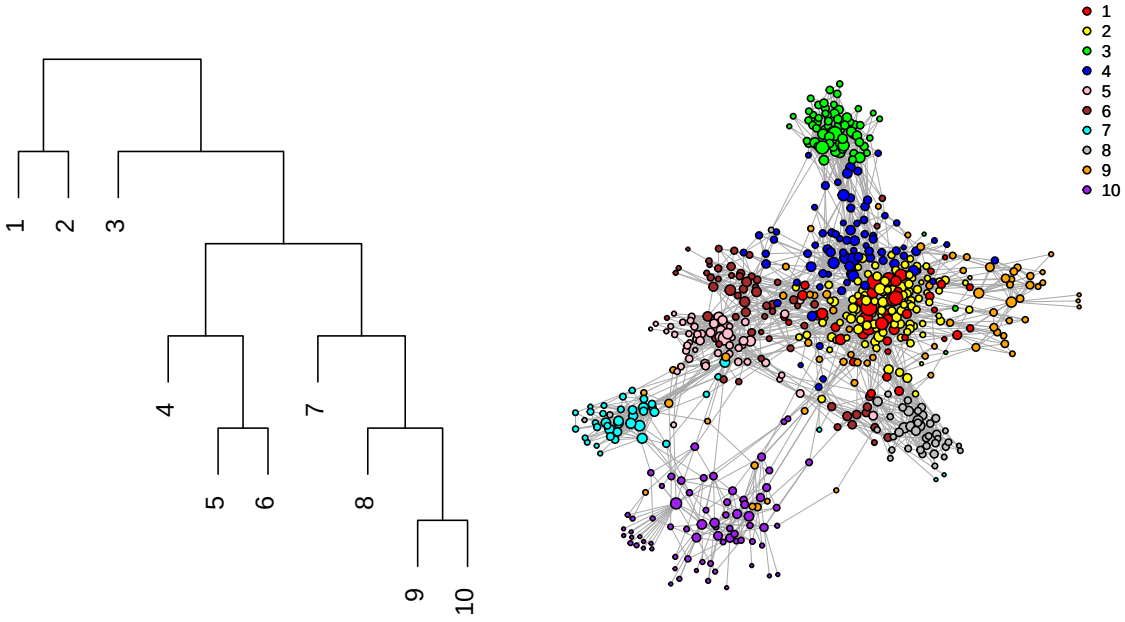


Figure 2.7: Hierarchical community structure of the core. The core has $N_{\mathcal{C}} = 635$, and the configuration-type model is used.

turns out that the community labels are also very interpretable. Since each cluster is a group of papers, we list the most frequent keywords of the papers in each cluster in Table 2.4. The keywords in each group are highly coherent.

2.7 Conclusion and Discussion

We have proposed a core-periphery model for extracting informative structures from networks and proposed two efficient algorithms for core identification under the model. Our model does not assume a specific form for the core component, so it can be used for preprocessing for downstream network modeling in general. The proposed algorithms have theoretical guarantees of correctly identifying the core component under mild conditions. The strong consistency property is advantageous for our model

Table 2.4: The most frequent keywords for each cluster in the hierarchy.

Cluster	Most frequent keywords
1	lasso, variable selection, smoothly clipped absolute deviation, model selection, asymptotic normality, sparsity
2	lasso, variable selection, oracle property, sparsity, regularization, model selection, smoothly clipped absolute deviation
3	false discovery rate, multiple testing, multiple comparisons, familywise error rate, p-value, stepdown procedure
4	sparsity, lasso, regularization, covariance matrix, high dimensional data, model selection, thresholding
5	functional data, smoothing, principal component, eigenfunction, eigenvalue, functional regression
6	nonparametric regression, generalized estimating equation, functional data, longitudinal data, partially linear model, semiparametric model
7	mixture model, nonparametric bayes, dirichlet process, hierarchical model, stick breaking
8	sliced inverse regression, central subspace, sliced average variance estimation, dimension reduction, nonparametric regression
9	classification, model selection, oracle inequality, support vector machine, aggregation, sparsity, statistical learning
10	markov chain monte carlo, bayesian inference, gaussian markov random field, gaussian process, generalized linear mixed model, kriging, spatial statistics

since conditioning on the core extract success, any downstream network theoretical analyses will remain valid on the core part. Our algorithms only require the first few eigenvectors of the adjacency matrix and are therefore computationally efficient.

There are several possible extensions to pursue following the proposed framework. For example, what are the other generally uninteresting structures in network model cases, and would they be incorporated into the same framework? Another interesting question is how to generalize the current framework to more complicated data structures for network modeling settings such as multiplex networks and dynamic networks. Such extensions may require delicate definitions of uninteresting structures in the new scenarios and potentially new model fitting tools.

The implementation of our algorithms and the data example used in this chapter can be found on `https://github.com/tianxili/Core-Periphery`.

Chapter 3

Incorporating Network into Topic Model

3.1 Introduction

In the previous chapter, we have seen that through the network data alone, we can discover meaningful structures of the underlying data generating system. On the other hand, network data can also play a complementary role for many standard statistical modeling tools. Examples include, but not limit to, linear regression and survival analysis (Li et al., 2019; Le and Li, 2020), and topic modeling (Liu et al., 2009; Zhu et al., 2013; Lim and Buntine, 2015; Lim et al., 2016). In this chapter, we focus on one specific modeling task: topic modeling, and incorporate network data into the topic modeling.

Topic models are machine learning techniques for discovering latent "topics" in a collection of text documents. In topic models, each topic is modeled as a probability distribution over distinct words, and each document in the collection is modeled as a probability distribution over the topics. We can think of topic modeling as a matrix

factorization task: Suppose we have D documents in our data set, there are W distinct word tokens, and K topics in the corpus. Then, using the bag-of-words representation, the entire corpus can be represented as a $D \times W$ word frequency matrix, which we denote as $\boldsymbol{N}$. Our objective is to find a factorization of $\boldsymbol{N}$ such that $\boldsymbol{N} \approx \boldsymbol{\theta}\boldsymbol{\phi}$, where $\boldsymbol{\theta}$ is the $D \times K$ document-topic matrix, and $\boldsymbol{\phi}$ is the $K \times W$ topic-word matrix. We also require that the entries of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are nonnegative. The number of topics K is usually small compared with the number of documents and the number of words W. Therefore, $\boldsymbol{\theta}$ provides an efficient summary of individual documents, and $\boldsymbol{\phi}$ captures the topics present in the entire corpus. It is worth noting that, for many recent topic models, the central ideas still resemble the matrix factorization.

The nonnegative matrix factorization (NMF) has been widely applied in text mining tasks (Lee and Seung, 1999; Shahnaz et al., 2006). NMF finds $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ through an optimization task, which minimizes the Frobenius norm of the difference $\boldsymbol{N} - \boldsymbol{\theta}\boldsymbol{\phi}$, while subject to the constraint that the entries of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are nonnegative. The nonnegative constraint differentiates NMF from traditional dimension reduction techniques, such as principal component analysis (PCA), and allows a part-based interpretation, i.e., the entity being modeled (e.g. a document) is an additive combination of different parts (e.g. topics). The probabilistic latent semantic analysis (pLSA) (Hofmann, 1999) assumes a probability model, in which $\boldsymbol{N}$ follows a multinomial distribution parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. In pLSA, the entries of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ have a probability interpretation. However, pLSA has two main shortcomings: First, it is unable to generalize to previously unseen documents; Second, the number of parameters is $D \cdot K + K \cdot W$, which grows linearly as the number of documents. As a result, it tends to overfit the data. To overcome these shortcomings, Blei et al. (2003) proposed the Latent Dirichlet Allocation (LDA), which is perhaps the most widely known topic model to date. The general idea of LDA is still the same: each word in

a document is generated by first sampling a topic according to the topic distribution of the document, and then sampling a word in the selected topic according to the word distribution of that topic. Compared with pLSA, LDA treats topic distribution parameter $\boldsymbol{\theta}$ as a hidden variable, and assigns a Dirichlet prior to it. This greatly reduced the number of parameters in the model, and improved its generalization performance on the previously unseen data.

In many topic modeling tasks, in addition to the written text documents, metadata is also available. These metadata can include the document-level features, such as the authors, timestamps, and publication venues, as well as the relations between the documents, such as hyperlinks or citations. By exploiting the correlation between these metadata and the topics, we can expect to improve the quality of the discovered topics. For example, the author information of the document can carry significant signals about the topics of the document: each author often has a unique topical interest, which is usually their research interests or domain of expertise. Relational features, such as networks, are sometimes available in addition to the text data. For example, online blogs often contain links to other blogs, and academic papers have a citation network. These network data contain information about the underlying topic structures. Therefore, it is of interest to incorporate these network data into the topic model when such data are available.

On the other hand, investigating how these metadata influences the formation and the evolution of topics can be an interesting subject in itself. For example, if the timestamps of the documents are available, then by studying the relationship between time and topic distributions, we can identify those trending topics and know how people's interests change over time.

In order to better leverage the available metadata, researchers have developed various extensions of LDA. They are designed to incorporate specific characteristics

of the data and the task at hand. In the author topic model (Rosen-Zvi et al., 2012), a multinomial distribution is associated with each author, instead of each document. Then, each document is modeled as a mixture of topic distributions of its authors. The word generating process is the following: for each word in the document, we first sample an author from the author list of the document, then we sample a topic from the topic distribution of the selected author, and then we sample a word from the word distribution of the selected topic. The author topic model allows direct modeling of each individual author, which enables author-level analysis, such as discovering the topical interests of each author and calculating similarities between authors. The dynamic topic models (Blei and Lafferty, 2006; Wang et al., 2008) are able to incorporate time effect and estimate topic evolutions. They use the state-space model or Brownian motion to model the evolution of the multinomial parameters for the topic distributions at different time points. Then, at each time point, an LDA is fitted. Zhu et al. (2013) proposed a topic model that incorporated network data. Specifically, they assumed that the probability of two documents being linked together is a function of the weighted inner product between the topic distributions of the two documents. Therefore, documents sharing similar topic distributions are more likely to be connected. They also proposed a scalable algorithm for model estimation. In the topic-link LDA (Liu et al., 2009), the authors also considered including network data. They proposed that the link formation between documents is not only due to the content similarity between documents but also affected by the community ties between the authors. In their approach, each author is associated with a community membership $\boldsymbol{\mu}$, which is different from the topic distribution $\boldsymbol{\theta}$ associated with each document, and both $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ play a role in network link formation. Roberts et al. (2016) developed a topic model that can incorporate an arbitrary number of document-level features, such as news sources and time of release. The proposed

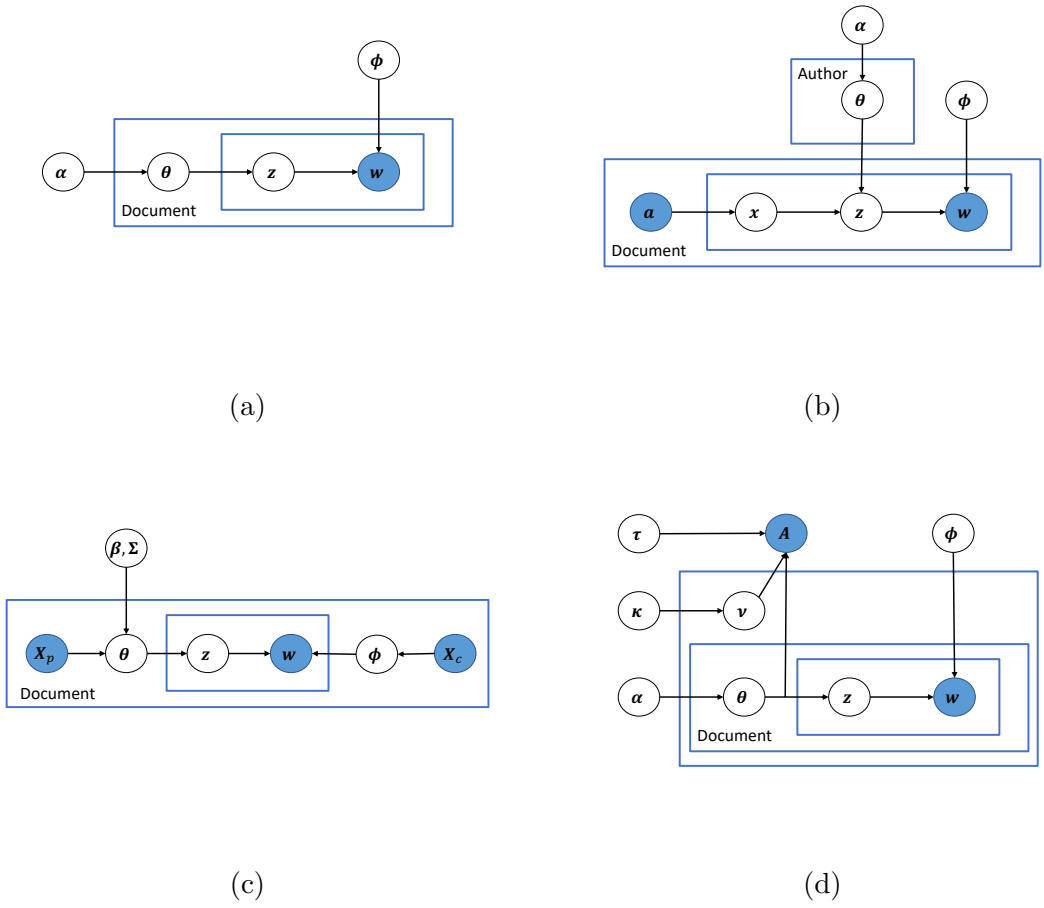


Figure 3.1: Graphical representations of some existing topic models: (a) Latent Dirichlet Allocation (Blei et al., 2003), (b) Author-Topic Model (Rosen-Zvi et al., 2012), (c) Structural Topic Model (Roberts et al., 2016), (d) Topic-link LDA (Liu et al., 2009).

model is called a structural topic model. In the proposed model, both the document-topic distribution $\boldsymbol{\theta}$ and the topic-word distribution $\boldsymbol{\phi}$ are parameterized as functions of document-level features through generalized linear models.

One implicit assumption made in the structural topic model is that the causal direction is from document-level features to topic distribution, but not the other way around, as is shown in Figure 3.1, which means, for an individual document, we first have those document-level features, and then the topic distribution is formed

based on the document-level features. However, in many cases, this may lead to counterintuitive interpretations. For example, for academic paper collections, one document-level feature we can observe is the journal in which the paper was published. Using the above causal interpretation, we will conclude that, when composing an academic paper, we will first determine the journal in which it will be published, and then we will determine the topics of the paper, which is unlikely.

In practice, some document-level features indeed influence the formation of the topics, and the topics of the document influence some document-level features. This idea has appeared in the research for network community detection problems: Zare et al. (2019) proposed a probabilistic graphical model for network community detection. In their work, they considered incorporating nodal features into the community detection algorithm, and they categorize nodal features into two types based on their causal relationships with the community memberships, namely, *assortative features*, which influence the formation of communities, and *generative features*, which are influenced by the community structures. Then, an efficient estimation algorithm was proposed to infer the community memberships of the network nodes. In the context of topic modeling, a natural adaptation of their approach is simply replacing the community membership of each node with the topic distribution of each document. However, to the best of our knowledge, no such model has been proposed for topic modeling.

In this chapter, we proposed a topic model that jointly models the text, the document-level features, and the links between documents. For the text data, we use an LDA-type text generation model. For the document-level features, we employ the same idea from Zare et al. (2019), which accounts for the distinction between assortative features and generative features. For the network links, we also assume an assortative link formation process, which means similar documents are more likely

to connect to each other. Since the proposed model has a complicated likelihood, we propose an estimation algorithm that is based on Laplacian approximation and stochastic expectation-maximization (stochastic EM). We apply our model to the statistics paper citation network data set. We showed that our estimation algorithm converges efficiently, and the proposed model is able to find both interpretable and stable latent topics. The rest of the chapter is organized as follows: In Section 3.2, we introduce our proposed topic model. In Section 3.3, we describe our model estimation procedure. In Section 3.4, we introduce several automated topic model evaluation metrics. In Section 3.5, we show the results of applying our model to the statistics paper with citation network data set, and compare it to several existing topic models. We then give a brief discussion in Section 3.6. Additional tables and plots of the model fitting results with different numbers of topics are included in Section 4.2.

3.2 Our Proposed Model

Suppose we have D documents and W unique words in those documents. Let $\boldsymbol{N}$ be the document-word matrix, where $\boldsymbol{N}_{d,w}$ is the number of occurrences of word w in document d. In addition, each document also has P assortative features and L generative features. Let $\boldsymbol{X}$ be the $D \times P$ assortative feature matrix. Let $\boldsymbol{Y}$ be the $D \times L$ generative feature matrix. There can also be a network linking pairs of documents together, which we denote as $\boldsymbol{A}$. Let $\boldsymbol{\theta}$ be the $D \times K$ document-topic distribution matrix, where K is the number of topics, and let $\boldsymbol{\phi}$ be the $K \times W$ topic-word distribution matrix.

3.2.1 Components of Our Model

Our proposed model consists of four main components:

Assortative features → Topic distribution:

The assortative features influence the formation of topic distributions. Typically these are the features that exist before the authors compose a document, such as the year in which the document is published, the authors of the document, the research area of the authors, etc. Following Roberts et al. (2016), we use a logistic normal distribution to model $\boldsymbol{\theta}$, and the mean vector of the distribution is parameterized by the assortative features $\boldsymbol{X}$. Let $\boldsymbol{\beta}$ be a coefficient matrix. We first define the variable $\boldsymbol{\mu}$, which is a $D \times K$ matrix. Its d-th row, $\boldsymbol{\mu}_{d,*}$, is generated from a multivariate normal distribution:

$$\boldsymbol{\mu}_{d,*} \sim N(\boldsymbol{X}_{d,*}\boldsymbol{\beta}, \boldsymbol{\Sigma}), \tag{3.1}$$

where the mean vector is a linear function of $\boldsymbol{X}$, and $\boldsymbol{\Sigma}$ is the covariance matrix. $\boldsymbol{\mu}$ is defined on the entire real domain. Next, $\boldsymbol{\theta}$ can be obtained by applying the softmax function on $\boldsymbol{\mu}$:

$$\boldsymbol{\theta}_{d,k} = \frac{e^{\boldsymbol{\mu}_{d,k}}}{\sum_{k'=1}^{K} e^{\boldsymbol{\mu}_{d,k'}}} = \frac{e^{\sum_{p=1}^{P} \boldsymbol{X}_{d,p}\boldsymbol{\beta}_{p,k}}}{\sum_{k'=1}^{K} e^{\sum_{p=1}^{P} \boldsymbol{X}_{d,p}\boldsymbol{\beta}_{p,k'}}}. \tag{3.2}$$

The softmax transformation automatically ensures that elements of $\boldsymbol{\theta}$ is within [0,1]. Therefore, no additional constraint is required. The log-likelihood of generating topic distributions given the assortative features is the following,

$$l_a(\boldsymbol{\theta}(\boldsymbol{\mu})|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{D}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{d=1}^{D}(\boldsymbol{\mu}_{d,*} - \boldsymbol{X}_{d,*}\boldsymbol{\beta})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_{d,*} - \boldsymbol{X}_{d,*}\boldsymbol{\beta})^t. \tag{3.3}$$

The formulation of the multivariate logit function in Equations (3.1) and (3.2) is symmetric, meaning that there is no reference class. The advantage of this formulation is that the coefficient $\boldsymbol{\beta}$ will be more interpretable. The identifiability issue can be solved by including a regularization term on the coefficient (Friedman et al., 2010).

Topic distribution → Generative features:

In contrast to the assortative features, the generative features are dependent on the topic distributions. Examples of this type of feature include the journal in which a research paper is published, and the number of clicks/likes/replies an online post received. We also note that the citation network is also "generative", since citation will only happen after a paper has been composed, and it depends on the topic distributions. However, to account for the bilateral nature of the network, we use a different component to model it.

In the current model, we only consider cases where the generative features are categorical variables. Given the topic distribution $\boldsymbol{\theta}$, we assume the generative features $\boldsymbol{Y}$ follow a logistic regression model:

$$P(\boldsymbol{Y}_{d,l} = C_{l,m} | \boldsymbol{\theta}_{d,*}, \boldsymbol{\alpha}) = \frac{e^{\sum_{k=1}^{K} \boldsymbol{\theta}_{d,k} \boldsymbol{\alpha}_{k,l,m}}}{\sum_{m'} (e^{\sum_{k=1}^{K} \boldsymbol{\theta}_{d,k} \boldsymbol{\alpha}_{k,l,m'}})}, \tag{3.4}$$

where $C_{l,m}$ denotes the m-th category for variable $\boldsymbol{Y}_{d,l}$, and $\boldsymbol{\alpha}$ contains the regression coefficients. Equation (3.4) is also symmetric. To resolve identifiability issue, we can also add a regularization on $\boldsymbol{\alpha}$.

Let $\boldsymbol{R}_{d,l,m}$ denote $P(\boldsymbol{Y}_{d,l} = C_{l,m} | \boldsymbol{\theta}_{d,*}, \boldsymbol{\alpha})$. The log-likelihood of this component can be written as

$$l_g(\boldsymbol{Y} | \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{d=1}^{D} \sum_{l=1}^{L} \sum_{m=1}^{M} 1(\boldsymbol{Y}_{d,l} = C_{l,m}) \cdot \log \boldsymbol{R}_{d,l,m}. \tag{3.5}$$

Topic distribution → Text:

This component describes how words are generated in each document, given the topic distribution $\boldsymbol{\theta}_{d,*}$ of document d. For each word position in document d, the probability

of observing word w is given by

$$P(w|\boldsymbol{\theta}_{d,*}, \boldsymbol{\phi}) = \sum_{k=1}^{K} \boldsymbol{\theta}_{d,k}\boldsymbol{\phi}_{k,w}. \tag{3.6}$$

Recall that $\boldsymbol{N}$ is the document-word co-occurrence matrix, and $\boldsymbol{N}_{d,w}$ is the number times word w appears in document d. Then, the log-likelihood of this component is

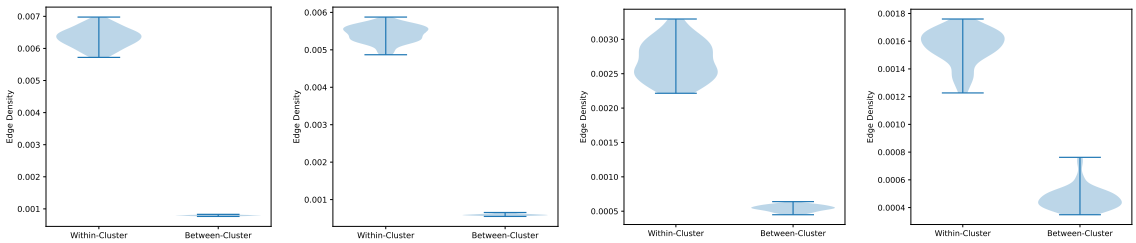
$$l_t(\boldsymbol{N}|\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{d=1}^{D}\sum_{w=1}^{W} \boldsymbol{N}_{d,w} \cdot \log\left(\sum_{k=1}^{K} \boldsymbol{\theta}_{d,k}\boldsymbol{\phi}_{k,w}\right). \tag{3.7}$$

Topic distribution → Network:

When network data are also present, we can also incorporate this information. In this model, we consider assortative link formations, which means documents sharing similar topic distributions are more likely to be linked together. Specifically, for two documents d_1 and d_2, given their topic distributions $\boldsymbol{\theta}_{d_1,*}$ and $\boldsymbol{\theta}_{d_2,*}$, we assume the link $\boldsymbol{A}_{d_1,d_2}$ is distributed as a Poisson variable:

$$\boldsymbol{A}_{d_1,d_2} \sim \text{Pois}\left(\sum_{k=1}^{K} \boldsymbol{\theta}_{d_1,k}\boldsymbol{\eta}_k\boldsymbol{\theta}_{d_2,k}\right). \tag{3.8}$$

The Poisson parameter is a weighted inner product between the two topic distributions, and $\boldsymbol{\eta}$ denotes the weight. We note that in the cases that $\boldsymbol{A}$ is unweighted, $\boldsymbol{A}_{d_1,d_2}$ can only be 0 or 1, whereas Equation (3.8) allows $\boldsymbol{A}_{d_1,d_2} > 1$. The idea of using Poisson distribution to approximate the link probability comes from Zhu et al. (2013). The motivation is that, in many real-world applications, $\boldsymbol{A}$ will be sparse, and Poisson distribution with small parameter values is a good enough approximation for a Bernoulli distribution with small success probabilities. More importantly, the mathematical form in Equation (3.8) makes the derivatives of the likelihood with



(a) 20 clusters (b) 10 clusters (c) 5 clusters (d) 2 clusters

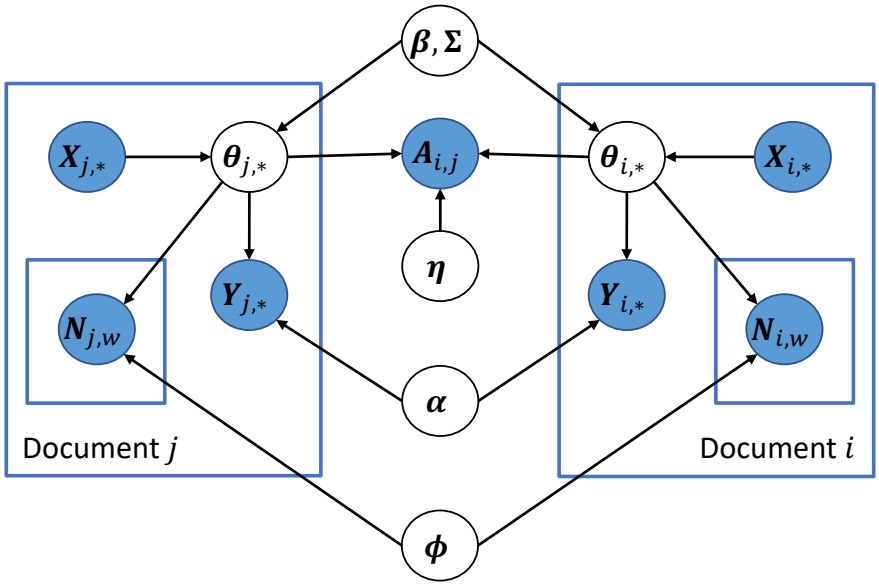
Figure 3.2: Violin plots of within-cluster and between-cluster edge densities.

respect to $\boldsymbol{\theta}$ take a simple form. The log-likelihood of this component has the form

$$l_n(\boldsymbol{A}|\boldsymbol{\theta},\boldsymbol{\eta}) = \sum_{d_1<d_2}\left[\boldsymbol{A}_{d_1,d_2}\cdot\log\left(\sum_{k=1}^{K}\boldsymbol{\theta}_{d_1,k}\boldsymbol{\eta}_k\boldsymbol{\theta}_{d_2,k}\right)-\sum_{k=1}^{K}\boldsymbol{\theta}_{d_1,k}\boldsymbol{\eta}_k\boldsymbol{\theta}_{d_2,k}\right]. \tag{3.9}$$

The key assumption we made when including the network data into the model is the assortative link assumption. To check the validity of this assumption, we can use our model-fitting result without the network. We obtain the estimated document-topic distribution $\hat{\boldsymbol{\theta}}$, and apply K-means clustering to its rows. This procedure will put documents with similar topic distributions into clusters. Then, we can look at the actual citation network, and find out the number of citations within and between the clusters. In Figure 3.2, we plot the within-cluster and between-cluster edge densities. We tried a few numbers of clusters for the K-means. The within-cluster densities are much higher than the between-cluster densities for all different numbers of clusters. Note that we did not include the network data when estimating these $\hat{\boldsymbol{\theta}}$s. Therefore, this suggests that there is indeed strong assortativity in the link formation process.

In summary, the full log-likelihood of the entire model can be obtained by adding



Node	Type	Description
$\boldsymbol{\theta}_{d,*}$	Hidden Variable	Topic distribution of document d
$\boldsymbol{\beta}$	Weight Factor	Correlation level between topic distribution and assortative features
$\boldsymbol{\alpha}$	Weight Factor	Correlation level between topic distribution and generative features
$\boldsymbol{\eta}$	Weight Factor	Level of interactions between topics
$\boldsymbol{A}$	Observation	Citation network
$\boldsymbol{Y}_{d,*}$	Observation	Generative features of document d
$\boldsymbol{X}_{d,*}$	Observation	Assortative features of document d
$\boldsymbol{N}_{d,w}$	Observation	Frequency of word w in document d
$\boldsymbol{\phi}$	Weight Factor	Topic-word distribution

Figure 3.3: Graphical representation of our proposed model.

up Equations (3.3), (3.5), (3.7) and (3.9):

$$l(\boldsymbol{\theta}(\boldsymbol{\mu}), \boldsymbol{Y}, \boldsymbol{N}, \boldsymbol{A}|\boldsymbol{X}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\eta}) = -\frac{D}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{d=1}^{D}(\boldsymbol{\mu}_{d,*} - \boldsymbol{X}_{d,*}\boldsymbol{\beta})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_{d,*} - \boldsymbol{X}_{d,*}\boldsymbol{\beta})^t$$
$$+ \sum_{d=1}^{D}\sum_{l=1}^{L}\sum_{m=1}^{M} 1(\boldsymbol{Y}_{d,l} = C_{l,m}) \cdot \log \boldsymbol{R}_{d,l,m} + \sum_{d=1}^{D}\sum_{w=1}^{W} \boldsymbol{N}_{d,w} \cdot \log\left(\sum_{k=1}^{K} \boldsymbol{\theta}_{d,k}\boldsymbol{\phi}_{k,w}\right)$$
$$+ \sum_{d_1<d_2}\left[\boldsymbol{A}_{d_1,d_2} \cdot \log\left(\sum_{k=1}^{K} \boldsymbol{\theta}_{d_1,k}\boldsymbol{\eta}_k\boldsymbol{\theta}_{d_2,k}\right) - \sum_{k=1}^{K} \boldsymbol{\theta}_{d_1,k}\boldsymbol{\eta}_k\boldsymbol{\theta}_{d_2,k}\right]. \quad (3.10)$$

In Figure 3.3, we present a graphical representation of our proposed model.

3.3 Model Estimation

3.3.1 E-step

The general approach to model estimation is the EM algorithm. At the E-step, we need to calculate the expectation of the full log-likelihood, which is the following:

$$\int l(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{N}, \boldsymbol{A}|\boldsymbol{X}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\eta}) \cdot P(\boldsymbol{\theta})\ d\boldsymbol{\theta} + l^{\text{prior}}(\boldsymbol{\alpha}) + l^{\text{prior}}(\boldsymbol{\beta}) + l^{\text{prior}}(\boldsymbol{\Sigma}), \quad (3.11)$$

where $P(\boldsymbol{\theta})$ is the posterior distribution of $\boldsymbol{\theta}$ given the observed data and other model parameters, and $l^{\text{prior}}(\boldsymbol{\alpha})$, $l^{\text{prior}}(\boldsymbol{\beta})$, $l^{\text{prior}}(\boldsymbol{\Sigma})$ are logarithm of the prior distributions for these parameters. Since the full log-likelihood in Equation (3.10) is complicated, integrating the log-likelihood analytically with respect to the hidden variable $\boldsymbol{\theta}$ is infeasible. To solve this problem, we employ Laplacian approximation (Wang and Blei, 2013) and stochastic EM (Nielsen, 2000) to estimate the model parameters.

Laplacian Approximation

The basic idea of Laplacian approximation is that, we use Taylor expansion to obtain an approximation for the posterior distribution for $\boldsymbol{\mu}$. Note that $\boldsymbol{\mu}$ is equivalent to $\boldsymbol{\theta}$ by Equation (3.2). In addition, $\boldsymbol{\mu}$ has the desirable property that its elements are defined on the entire real axis, while elements of $\boldsymbol{\theta}$ must be within $[0, 1]$. Let $P(\boldsymbol{\mu})$ denote the posterior distribution of $\boldsymbol{\mu}$ (we omitted its dependence on other variables for simplicity), and let $\hat{\boldsymbol{\mu}}$ denote the maximum a posteriori (MAP) estimate that maximizes $P(\boldsymbol{\mu})$. Then, we do Taylor expansion of $\log P(\boldsymbol{\mu})$ at $\hat{\boldsymbol{\mu}}$:

$$\log P(\boldsymbol{\mu}) \approx \log P(\hat{\boldsymbol{\mu}}) + (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^t [\log P(\hat{\boldsymbol{\mu}})]' + \frac{1}{2}(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^t H(\hat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}), \tag{3.12}$$

where $H(\hat{\boldsymbol{\mu}})$ is the Hessian matrix evaluated at $\hat{\boldsymbol{\mu}}$. We note that the value of $H(\hat{\boldsymbol{\mu}})$ depends on other parameters in the model as well as the observed data. Since $\hat{\boldsymbol{\mu}}$ maximizes the posterior, $[\log P(\hat{\boldsymbol{\mu}})]' = 0$, and we have

$$\log P(\boldsymbol{\mu}) \approx \log P(\hat{\boldsymbol{\mu}}) + \frac{1}{2}(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^t H(\hat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}), \tag{3.13}$$

and therefore,

$$P(\boldsymbol{\mu}) \propto \exp\left\{\frac{1}{2}(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^t H(\hat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})\right\}. \tag{3.14}$$

This means we can use a normal distribution with mean $\hat{\boldsymbol{\mu}}$ and covariance matrix $-[H(\hat{\boldsymbol{\mu}})]^{-1}$ to approximate $P(\boldsymbol{\mu})$.

In practice, we can find the MAP estimate $\hat{\boldsymbol{\mu}}$ through gradient descent. For the Hessian matrix $H(\hat{\boldsymbol{\mu}})$, we ignore its off-diagonal entries. The motivation is that, since $H(\hat{\boldsymbol{\mu}})$ is $(D * K) \times (D * K)$, if we include the off-diagonal entries, the number of parameters to be estimated grows quadratically, and inverting such a large matrix

is computationally infeasible, so we only keep the diagonal entries, which reduce the number of parameters to $D * K$, and inverting a diagonal matrix is easy.

Stochastic EM

Given we can approximate the likelihood $l(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{N}, \boldsymbol{A}|\boldsymbol{X}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\eta})$ with a normal distribution, the next step is to replace the integration in Equation (3.11) with an summation. Specifically, in each iteration of the stochastic EM algorithm, we first find the MAP $\hat{\boldsymbol{\mu}}$ while fixing all other parameters. This can be done through gradient descent. Then, we calculate the Hessian $H(\hat{\boldsymbol{\mu}})$, which is evaluated at $\hat{\boldsymbol{\mu}}$. Then, we sample $\tilde{\boldsymbol{\mu}}^{(b)}$ from $N\left(\hat{\boldsymbol{\mu}}, -[H(\hat{\boldsymbol{\mu}})]^{-1}\right)$, for $b = 1, 2, ..., B$. We next maximize the following sum with respect to the model parameters $\boldsymbol{\phi}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\eta}$:

$$\frac{1}{B}\sum_{b=1}^{B} l(\boldsymbol{\theta}(\tilde{\boldsymbol{\mu}}^{(b)}), \boldsymbol{Y}, \boldsymbol{N}, \boldsymbol{A}|\boldsymbol{X}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\eta}) + l^{\text{prior}}(\boldsymbol{\alpha}) + l^{\text{prior}}(\boldsymbol{\beta}) + l^{\text{prior}}(\boldsymbol{\Sigma}) \quad (3.15)$$

3.3.2 M-step

During each iteration of the stochastic EM algorithm, after obtaining the summation in Equation (3.15), we update the model parameters. Specifically, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be obtained through gradient descent.

For the covariance $\boldsymbol{\Sigma}$, we assign an inverse-Wishart prior with identity matrix as the scale matrix and degree of freedom K, $\mathcal{W}^{-1}(\boldsymbol{I}, K)$. The inverse-Wishart distribution is a conjugate prior for the covariance matrix of a multivariate normal distribution, so the posterior distribution is also inverse-Wishart, and there is a closed-form solution for $\boldsymbol{\Sigma}$ that maximizes the posterior. The updating equation is

$$\hat{\boldsymbol{\Sigma}} = \left(\boldsymbol{I} + \frac{1}{B}\sum_{b=1}^{B}(\tilde{\boldsymbol{\mu}}^{(b)} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^t(\tilde{\boldsymbol{\mu}}^{(b)} - \boldsymbol{X}\hat{\boldsymbol{\beta}})\right) / (D + 2K + 1)\,. \quad (3.16)$$

For $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$, we use the same EM algorithm as in Zhu et al. (2013) to update the parameters. For $\boldsymbol{\phi}$, during iteration i, the updating equation is

$$\tilde{\boldsymbol{\phi}}_{k,w}^{(i+1)} = \sum_{b=1}^{B}\sum_{d=1}^{D} \boldsymbol{N}_{d,w} \frac{\tilde{\boldsymbol{\theta}}_{d,w}^{(b)} \hat{\boldsymbol{\phi}}_{k,w}^{(i)}}{\sum_{k'=1}^{K} \tilde{\boldsymbol{\theta}}_{d,k'}^{(b)} \hat{\boldsymbol{\phi}}_{k',w}^{(i)}}, \tag{3.17}$$

$$\hat{\boldsymbol{\phi}}_{k,w}^{(i+1)} = \frac{\tilde{\boldsymbol{\phi}}_{k,w}^{(i+1)}}{\sum_{w'=1}^{W} \tilde{\boldsymbol{\phi}}_{k,w'}^{(i+1)}}. \tag{3.18}$$

For $\boldsymbol{\eta}$, the updating equation is

$$q_{d_1,d_2}(k) = \sum_{b=1}^{B} \frac{\tilde{\boldsymbol{\theta}}_{d_1,k}^{(b)} \tilde{\boldsymbol{\theta}}_{d_2,k}^{(b)} \hat{\boldsymbol{\eta}}_{k}^{(i)}}{\sum_{k'=1}^{K} \tilde{\boldsymbol{\theta}}_{d_1,k'}^{(b)} \tilde{\boldsymbol{\theta}}_{d_2,k'}^{(b)} \hat{\boldsymbol{\eta}}_{k'}^{(i)}}, \tag{3.19}$$

$$\hat{\boldsymbol{\eta}}_{k}^{(i+1)} = \frac{\sum_{d_1,d_2=1}^{D} \boldsymbol{A}_{d_1,d_2} \cdot q_{d_1,d_2}(k)}{\sum_{b=1}^{B} \left(\sum_{d=1}^{D} \tilde{\boldsymbol{\theta}}_{d,k}^{(b)} \right)^2}. \tag{3.20}$$

We fit our proposed model to the statistics paper citation network data set using the above estimation algorithm. The details can be found in Section 3.5. We monitor the objective function in Equation (3.15) as the algorithm progresses, and plot it in Figure 3.4. In the plot, we dropped the first iteration, since the increase in the objective function is very large during the first iteration, which squeezes the rest of the points in the plot. We can see from the plot that the objective function is generally increasing, with minor fluctuations due to the stochastic nature of the algorithm. The objective function stabilizes after about 25 iterations. In practice, when we fit the model, we set a maximum number of iterations and a tolerance value, and we terminate the estimation algorithm if it reaches the maximum number of iterations or the relative change in the objective function becomes less than the tolerance value.

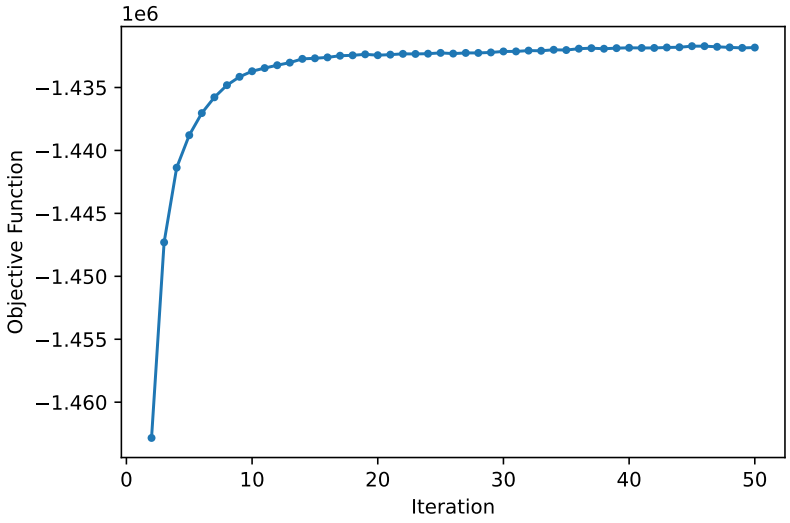


Figure 3.4: Plot of the objective function against the number of iterations.

3.4 Model Evaluation

3.4.1 Held-out Likelihood

To assess the generalization performance of the topic models, we evaluate the likelihood of a held-out test data set. We randomly select a subset of the documents, and for each selected document, we hold out a fraction of its words. We then fit models to the remaining training data, and evaluate the per-word likelihood of the held-out test set. The definition of per-word held-out likelihood is defined as

$$\frac{1}{|D^{test}|} \sum_{d \in D^{test}} \frac{\log P(\boldsymbol{w}_d^{test})}{N_d^{test}}. \tag{3.21}$$

We note that in many topic model literature, a related quantity called perplexity is used. The perplexity is inversely related to the per-word likelihood, and therefore captures the same aspect of the model fitting. In topic modeling literature, perplexity and held-out likelihood are the main metrics for assessing how well the model fits the

data. However, we also note that neither of these two metrics is directly attached to the human interpretability of the discovered topics. For example, Chang et al. (2009) showed that topic models that do better on held-out likelihood might be less semantically interpretable to humans.

3.4.2 Metrics for Semantic Interpretability

As an alternative evaluation metric to the held-out likelihood, Chang et al. (2009) proposed two human evaluation tasks, namely, word intrusion and topic intrusion. In these evaluation tasks, a human annotator is presented with a list of words within a topic as well as an intruder word that does not belong to the topic, or a list of topics within a document as well as an intruder topic that does not belong to the document. Then, the annotator is asked to identify the intruder word (or topic), and the probability of identifying the correct intruder word (or topic) is used as a quality metric for topic modeling. A more straightforward human evaluation task is topic rating, in which an annotator rates the quality of the presented topic on a scale of, say, 1 to 3.

These tasks require the participation of humans and are therefore expensive to deploy on a large scale. Newman et al. (2010) tested a group of automated coherence metrics for topic models, and compared them to real human evaluation scores. They found that the PMI-based term co-occurrence within Wikipedia achieves a high correlation with human evaluation scores. The automated coherence metric gives a single score to each topic. For a topic model that returns several topics, we can calculate the average coherence score of all its topics, which is then used as a quality metric for the topic model. The automated coherence metric enables fast and large-scale evaluation of topic models, and is widely used as a proxy for human evaluations.

However, a recent study also shows that, for recent neural topic models, there could be a substantial gap between human evaluation and the coherence metrics: coherence metrics can declare a winner, whereas the corresponding human judgment does not (Hoyle et al., 2021). In addition, as we will demonstrate later, coherence metrics are also heavily influenced by word frequencies, and generic words often lead to higher coherence scores regardless of the actual interpretability of the topics.

In this section, we introduce two automated coherence metrics. For topic k, we select m representative words, which we denote as $\boldsymbol{v}(k) = (v_1, v_2, ..., v_m)$. Then, the coherence metrics we use are the following:

Co-occurrence Coherence:

The first coherence metric is based on co-occurrences of the m representative words (Mimno et al., 2011). Specifically, let $D(v)$ be the number of documents containing word v, and $D(v, v')$ be the number of documents containing both v and v'. Then, the co-occurrence coherence is defined as

$$\text{coherence}(k) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \log \frac{D(v_i, v_j)}{\sqrt{D(v_i)D(v_j)}}. \tag{3.22}$$

Intuitively, if a topic is meaningful to humans, its representative words will reinforce each other semantically, and we can expect that its representative words co-occur more frequently in the same documents, whereas if a topic is random and not meaningful, the co-occurrence rate will be low.

Cosine Coherence:

Similar to co-occurrence coherence, we can also use cosine similarity to measure co-occurrence. For the document-word matrix $\boldsymbol{N}$, let $\boldsymbol{N}_{*,i}$ be its i-th column. This

column corresponds to word i. Then, the cosine coherence is defined as

$$\text{cosine}(k) = \frac{2}{m(m-1)} \sum_{\substack{i,j \in \boldsymbol{v}(k), \\ i \neq j}} \sum_{d=1}^{D} \frac{\boldsymbol{N}_{d,i}}{\|\boldsymbol{N}_{*,i}\|_2} * \frac{\boldsymbol{N}_{d,j}}{\|\boldsymbol{N}_{*,j}\|_2}. \tag{3.23}$$

The above two metrics require that we have selected m representative words for each topic. In many applications, the most straightforward selection method is just picking m words with the highest probabilities in each topic. Besides, we can also try to filter out the generic words by subtracting the background word distribution from the topic-word distribution, which essentially "centers" the topic-word distribution. Therefore, generic words that have high probabilities in every topic are less likely to be selected as representative words.

In our experiment, for topic k, we define the following transformation:

$$\boldsymbol{\phi}'_{k,w} = \arcsin \sqrt{\boldsymbol{\phi}_{k,w}} - \arcsin \sqrt{\boldsymbol{\phi}_{\cdot,w}}, \tag{3.24}$$

where $\boldsymbol{\phi}_{\cdot,w} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\phi}_{k,w}$ is the average probability of word w over all K topics. After the transformation, we select m words that have the highest $\boldsymbol{\phi}'_{k,w}$ for each topic as its representative words.

3.4.3 Metrics for Stability

Another aspect from which topic models can be evaluated is stability. We use the procedure proposed in Mantyla et al. (2018), namely, making replicated runs, clustering the resulting topics from all the runs, and measuring the dissimilarity of topics within each cluster. The idea of clustering the topics from replicated runs also appeared in Chuang et al. (2015), but the latter did not propose a quantitative measure for stability. Let S be the number of replicated runs. Let $\boldsymbol{\phi}^{(s)}$ be the topic-word

matrix from the s-th model run. We stack the resulting topic-word matrices $\boldsymbol{\phi}^{(s)}$, and let $\boldsymbol{\Phi} = \left(\boldsymbol{\phi}^{(1)^t}, \boldsymbol{\phi}^{(2)^t}, ..., \boldsymbol{\phi}^{(S)^t}\right)^t$. Then we apply the K-means clustering to the rows of $\boldsymbol{\Phi}$, with K equal to the number of topics in each individual run. Then, the stability measure is calculated as the sum of the squared Euclidean distance between each row and its cluster center.

Stability measure:

$$\text{stability} = \frac{1}{S} \sum_i \left\| \boldsymbol{\Phi}_{i,*} - \boldsymbol{C}_{km(i)} \right\|_2^2, \tag{3.25}$$

where $\boldsymbol{C}_{km(i)}$ denotes the cluster center for the i-th row. In Chuang et al. (2015), a clustering algorithm is also applied to the rows of $\boldsymbol{\Phi}$, but with the additional constraint that each resulting cluster cannot contain multiple topics from the same run. Therefore, in addition to the regular K-means clustering, we also applied the constrained K-means clustering from Wagstaff et al. (2001), which ensures that different topics from the same model run will not appear in the same cluster.

3.5 Application to Statistics Papers with Citation Network Data Set

In this section, we apply our topic model to the statistics papers with citation network data set collected by Ji and Jin (2016). We use our proposed topic model to model the abstract of these papers while taking into account the document-level features and the citation network available in this data set. After removing papers with no abstract, there are 3214 papers in the data set. Standard text preprocessing is applied

to the abstracts, including stopwords removal[1], punctuation removal, stemming, etc. We also use tf-idf to filter the words: For each word, we calculate its tf-idf weights, and accumulate the weights over all documents. We only keep the top 1000 words with the highest cumulative tf-idf weights. Then, we remove words that only appear in one document, and 990 distinct words remain for our topic modeling task.

We use the publication year of each document as the assortative feature $\boldsymbol{X}$, and assume that topic distribution priors are B-spline functions with 5 degrees of freedom of the publication year. We use the publication journal as the generative feature $\boldsymbol{Y}$. This feature has four categories: Annals of Statistics (AOS), Biometrika (Biomet), Journal of American Statistical Association (JASA), and Journal of Royal Statistical Society Series B (RSS). In the preprocessed data set, there are 954, 750, 1103, and 407 papers in each of these four journals, respectively. For our proposed model, we consider two versions: one with both the document-level features and the citation network, and one with only document-level features but no network.

We compare our proposed model to Structural Topic Model (STM) and Latent Dirichlet Allocation (LDA). For STM, the same set of document-level features, publication year and journal, are used, and both are used as prevalence features. In addition, no network information is included. For LDA, only text data are used, and neither document-level features nor the network is included. We set the number of topics to $K = 5, 10, 20$, and calculate the evaluation metrics introduced in the previous section. The results are shown in Tables 3.1 to 3.3. Note that for topic stability, lower values indicate better performance, while for other metrics, higher values indicate better performance. The results are based on 50 replicated runs: We present the mean values over the 50 replicated runs for these metrics, and two times

[1]We use the SMART information retrieval system. The list can be found in the `stopwords` R package

standard deviations of the means are shown in the parentheses; for stability metrics, we draw subsamples of size 20 from the 50 replicated runs, calculate the stability metrics using the subsamples, and average them over 300 independent draws. The standard deviations of the stability metrics are negligibly small and are not shown.

As we can see in the table, while STM achieves the best coherence and LDA achieves the best stability, our proposed model achieves the best or close-to-the-best performances in held-out likelihood, coherence, and stability. LDA produces stable results. However, its discovered topics are less semantically coherent, which limits its usefulness. This observation implies that the document-level features indeed carry topical information, and are therefore helpful for discovering meaningful topics. STM is able to find semantically coherent topics, but is relatively unstable. In addition, since STM does not consider the causal directions between topic distributions and the document-level features, users must be cautious when interpreting its model parameters. For our proposed model, the inclusion of the network into the model mainly improves the stability metrics.

Next, we focus on our proposed model and interpret the model fitting result. For this objective, we fix $K = 10$. First, in Table 3.4, we listed the top-20 most representative words for each topic. The representativeness of each word in each topic is defined as in Equation (3.24). We also named each topic based on these 20 words. We also note that because of the instability of topic models, each time we run our model, we may get a slightly different set of topics and their top-20 words. From our replicated runs, we found that "High-Dimensional Stat/Variable Selection", "Clinical Trials", "Bayesian", "Hypothesis Testing", and "Density Function" are the highly consistent topics across multiple runs. They almost always show up in each individual run. On the other hand, there are several interpretable topics that show up most of the time, such as "Experimental Design" and "Spatial/Temporal". The

Table 3.1: Topic Model Evaluation ($K = 5$)

Methods	Held-out Likelihood	Coherence		Stability	
		Co-occurrence	Cosine	Unconstrained	Constrained
Ours	**−6.002(±0.005)**	−383.94(±5.63)	0.1394(±0.0014)	0.0052	0.0067
Ours (no network)	−6.005(±0.005)	−373.20(±4.63)	0.1424(±0.0013)	0.0066	0.0082
Ours (no features)	−6.004(±0.005)	−378.70(±6.19)	0.1398(±0.0018)	**0.0051**	**0.0061**
Ours (permuted)	−6.022(±0.005)	−372.38(±5.01)	0.1431(±0.0016)	0.0062	0.0073
STM	−6.099(±0.005)	**−347.45(±2.72)**	**0.1457(±0.0015)**	0.0171	0.0202
LDA	−6.043(±0.006)	−460.02(±10.89)	0.1199(±0.0013)	0.0056	0.0069

Table 3.2: Topic Model Evaluation ($K = 10$)

Methods	Held-out Likelihood	Coherence		Stability	
		Co-occurrence	Cosine	Unconstrained	Constrained
Ours	**−5.922(±0.006)**	−435.25(±7.19)	**0.1314(±0.0009)**	0.023	0.030
Ours (no network)	−5.928(±0.006)	−431.80(±5.58)	0.1311(±0.0008)	0.029	0.036
Ours (no features)	−5.924(±0.006)	−445.38(±6.35)	0.1296(±0.0009)	0.024	0.031
Ours (permuted)	−5.940(±0.006)	−444.09(±7.31)	0.1299(±0.0009)	0.033	0.040
STM	−6.029(±0.006)	**−408.41(±5.18)**	0.1309(±0.0010)	0.074	0.088
LDA	−6.049(±0.007)	−548.20(±12.40)	0.1199(±0.0013)	**0.020**	**0.027**

Table 3.3: Topic Model Evaluation ($K = 20$)

Methods	Held-out Likelihood	Coherence		Stability	
		Co-occurrence	Cosine	Unconstrained	Constrained
Ours	**−5.839(±0.006)**	−504.22(±6.47)	**0.1217(±0.0006)**	0.091	0.151
Ours (no network)	−5.848(±0.006)	−501.65(±5.24)	0.1216(±0.0006)	0.106	0.168
Ours (no features)	−5.843(±0.006)	−499.52(±5.90)	0.1213(±0.0007)	0.086	0.141
Ours (permuted)	−5.854(±0.007)	−520.58(±6.96)	0.1198(±0.0007)	0.126	0.185
STM	−5.967(±0.005)	**−494.81(±5.65)**	0.1118(±0.0008)	0.337	0.422
LDA	−6.244(±0.012)	−668.60(±9.91)	0.1082(±0.0007)	**0.076**	**0.107**

Table 3.4: The most representative words for each topic. These words are sorted in descending order based on Equation (3.24). The numbers in the parentheses are the word probabilities in the corresponding topic.

Id	Name	Most representative words
1	Clinical Trials	effect(0.035), treatment(0.024), outcom(0.014), studi(0.026), random(0.019), trial(0.01), miss(0.01), diseas(0.009), respons(0.013), cancer(0.008)
2	High-dimensional Stat	select(0.042), variabl(0.022), dimens(0.013), dimension(0.015), regress(0.023), lasso(0.009), high(0.014), penalti(0.009), classif(0.008), penal(0.008)
3	Bayesian	prior(0.027), bayesian(0.027), distribut(0.039), model(0.061), carlo(0.017), mont(0.017), posterior(0.016), mixtur(0.016), markov(0.015), algorithm(0.021)
4	Mixed/unclear	sampl(0.058), estim(0.073), error(0.029), varianc(0.019), bootstrap(0.014), interv(0.014), confid(0.012), robust(0.014), popul(0.012), small(0.011)
5	Mixed/unclear	estim(0.088), likelihood(0.041), model(0.065), semiparametr(0.015), paramet(0.027), regress(0.025), parametr(0.014), effici(0.016), propos(0.03), maximum(0.013)
6	Spatial/Temporal	time(0.026), spatial(0.015), model(0.055), process(0.019), data(0.032), forecast(0.006), articl(0.011), seri(0.009), dynam(0.006), onlin(0.005)
7	Hypothesis Testing	test(0.111), procedur(0.039), statist(0.037), power(0.018), null(0.017), hypothesi(0.016), control(0.018), fals(0.012), hypothes(0.011), rank(0.012)
8	Experimental Design	design(0.049), optim(0.026), class(0.027), span(0.015), graphic(0.012), graph(0.008), space(0.012), inlin(0.008), formula(0.007), shape(0.007)
9	Functional Data	function(0.057), cluster(0.026), compon(0.028), data(0.046), smooth(0.019), correl(0.018), curv(0.014), gene(0.013), analysi(0.021), spline(0.011)
10	Density Function	densiti(0.021), function(0.036), converg(0.017), rate(0.019), estim(0.052), bound(0.012), gaussian(0.011), process(0.017), kernel(0.009), minimax(0.007)

topics "Functional Data", "Time Series", and "Survival Analysis" occasionally show up.

Then, we look at the effect of the publication year on the topic distributions. Specifically, let $\boldsymbol{x}$ contain the B-spline bases at certain publication years, and then we plot $\boldsymbol{x}\hat{\boldsymbol{\beta}}$ against the corresponding publication year in Figure 3.5. In our model assumption, the topic distribution depends on the assortative features. In this plot, for a given year, the heights of the lines indicate the prior distributions for the topics, and can be interpreted as the general popularity of the topics. For example, Topic 2, which seems to be about high dimensional statistics and variable selection, became more prevalent as the years went by. In Figure 3.6, we plot the estimated coefficients $\hat{\boldsymbol{\alpha}}$ for the generative features. In this data example, papers focusing on Topic 10 (Density Function) are more likely to appear in AOS, while papers focusing on Topic 6 (Spatial/Temporal) are more likely to appear in JASA. Interestingly, Topic 1 (Clinical Trials) seems to have a negative effect on the likelihood of being published on AOS. The parameter $\hat{\boldsymbol{\eta}}$ is the weight of the inner-product in Equation (3.8), and it describes to which extent each topic affects the formation of citations. We plot its values in Figure 3.7, from which we can see that Topic 2 (High-dimensional Stat), Topic 7 (Hypothesis Testing), and Topic 9 (Functional Data) has the highest weight, while Topic 4 (Mixed/unclear) and Topic 6 (Spatial/Temporal) have the lowest weights. This suggests that if a pair of papers both focus on Topics 2, 7, or 9, it is more likely for them to form a citation compared with pairs of papers focusing on other topics.

Next, we look at the topic distributions of the papers with high degrees in the citation network. We select the papers with ≥ 35 degrees, which gives us 12 papers. In Figure 3.8, we plot their topic distributions. We note that at the time of writing this thesis, these papers all have received hundreds or even thousands of citations as indicated by Google Scholar. Therefore, the topic distributions of these papers

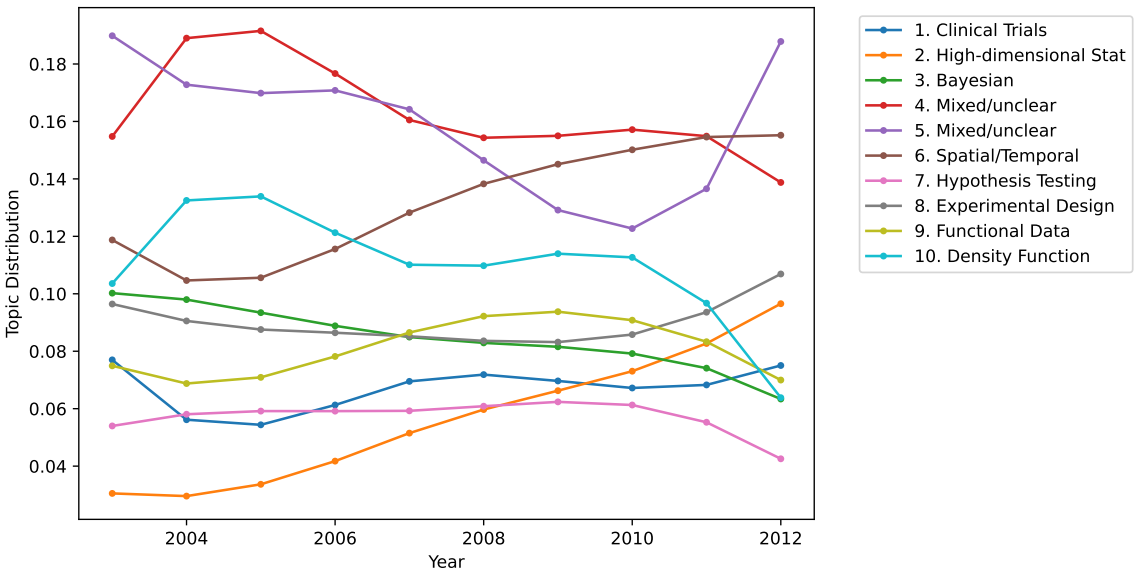


Figure 3.5: Effects of the assortative features (publication year) on the Topic distributions.

might be interesting to the general audience. In Figure 3.8, we can see that 9 out of 12 papers are all focused on Topic 2 (High-dimensional Stat). There are also 2 papers focusing on hypothesis testing, and 1 paper focusing on functional data. This observation suggests that high-dimensional statistics has generated some of the most-cited research papers in statistics during the time between 2003 and 2012. We note that the titles of these papers are also indicative of the topics of the papers. We also did the same analysis with STM: we plotted the topic distributions of the same 9 papers with $\boldsymbol{\theta}$ estimated by STM. Although STM does not return the exact same set of topics, the topics with the highest probabilities are also about high-dimensional statistics/variable selection and hypothesis testing.

What if we include a false network?

We also considered the problem of including a false network, in which case there is no true association between the network structure and the documents. To simulate this

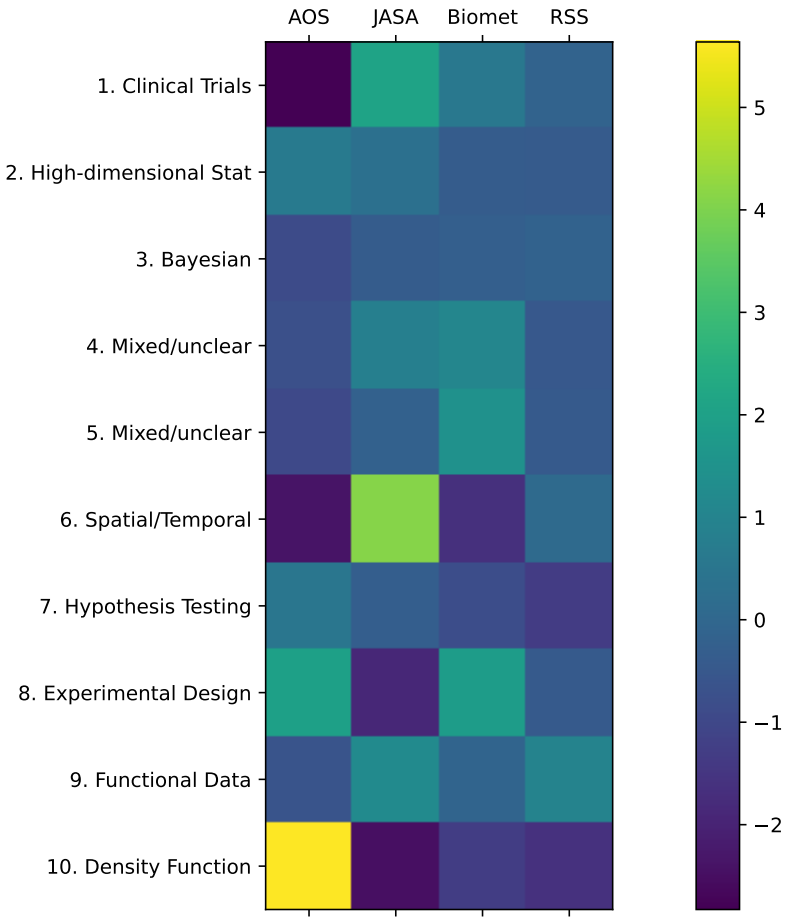


Figure 3.6: Generative feature: Topic distributions vs Publication Journal.

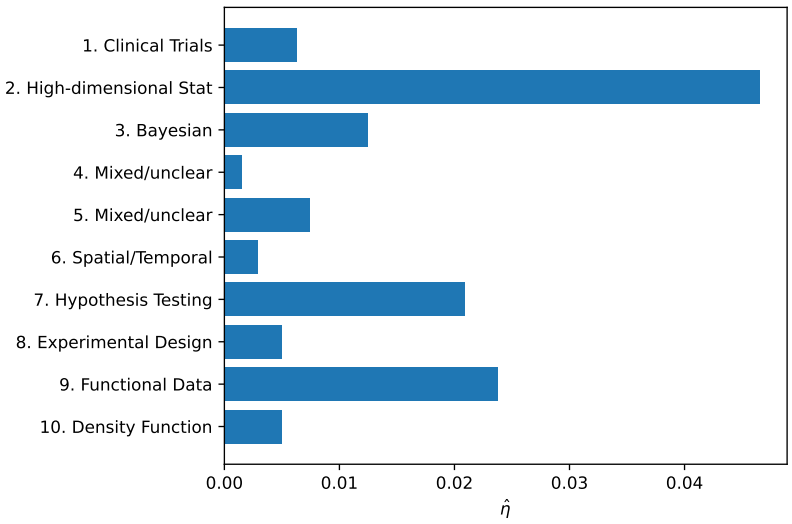


Figure 3.7: Estimated topic assortative weight $\hat{\boldsymbol{\eta}}$.

scenario with our current data set, we randomly permute the order of the network nodes and assign the permuted network nodes to the document, thus breaking the association between the network structure and the documents. Then, we fit the topic model with the permuted network, and evaluate the same metrics as in the previous section. The results are also shown in Tables 3.1 to 3.3. The topic models with the permuted networks consistently lead to a lower held-out likelihood than models with the true network or without a network. For $K = 10$ and 20, using the permuted networks also result in worse coherence and stability. In practice, to test whether there is a true association between the network and the documents, we can fit two models with and without the network, and see if the one with the network has a worse held-out likelihood.

When there is no true association between the network structure and the documents, the estimated $\hat{\boldsymbol{\eta}}$ is no longer meaningful. We tried to find the null distribution of $\hat{\boldsymbol{\eta}}$ in this case, and it turns out the null distribution still depends on the network structure: When we use a simulated non-informative Erdös-Renyi network, all entries

Figure 3.8: Topic distributions of the papers with the highest degrees in the citation network.

in $\hat{\boldsymbol{\eta}}$ have similar values; When we use the permuted networks, where there is degree heterogeneity, one of the entries of $\hat{\boldsymbol{\eta}}$ will have large values, while other entries are close to zero.

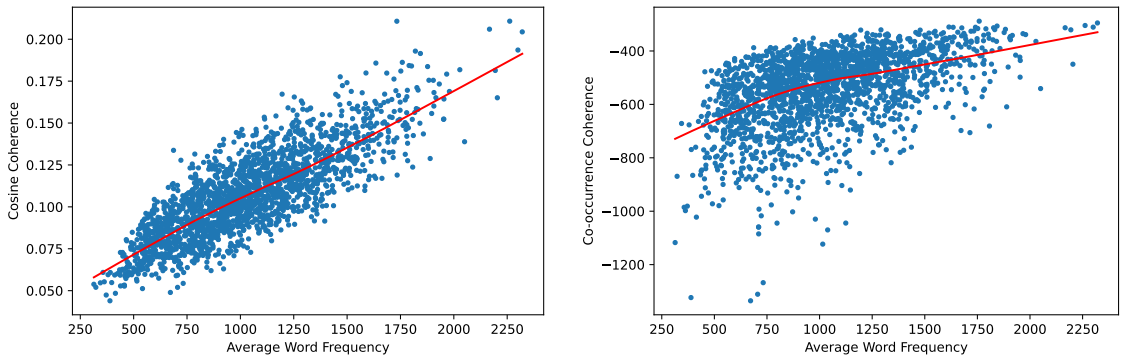
Limitation of Coherence

Coherence is one of the most widely used automated evaluation metrics for topic models. However, coherence can be heavily influenced by word frequencies. For example, in the statistics papers with citation network data set, if we select the top-20 words with the highest word frequency and treat them as a topic, the cosine coherence will be 0.287 and the co-occurrence coherence will be -179.66, both of which are much higher than the coherence scores in Tables 3.1 to 3.3, despite not being an actual topic.

More generally, to show the dependence of coherence on word frequency, we randomly sample 20 words without replacement, with the sampling probability of each word proportional to its overall frequency. Then, for each sample of 20 words, we calculate its coherence, and we plot the coherence against the average word frequency of the 20 words. The results are shown in Figure 3.9. As can be seen in the figures, coherence scores generally increase as the average word frequency increases.

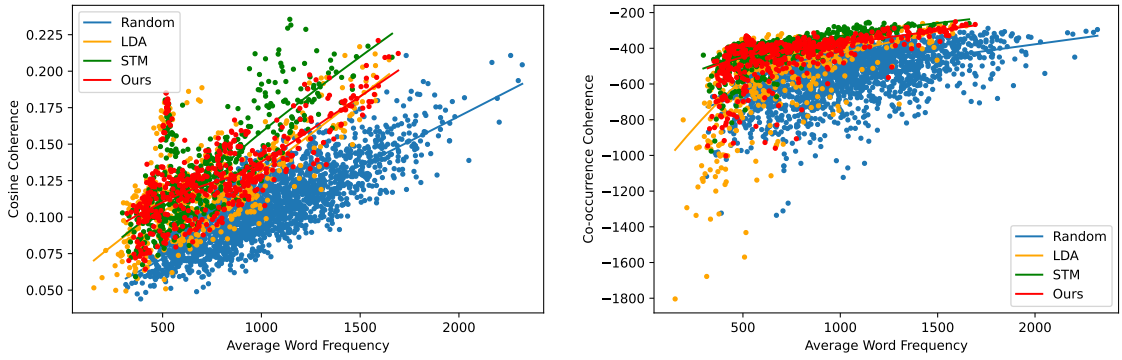
Next, we plot the coherence against average word frequency for the actual topics found by our model, STM, and LDA. We also compare them to the coherence of randomly sampled words. The results are shown in Figure 3.10. For these actual topics, the coherence scores also seem to increase as the average word frequency increases. In addition, for most of the actual topics, their coherence scores are above that of the randomly sampled words at any fixed word frequency level, which indicates that these models do find signals that are not purely random.

In practice, coherence metrics should be used with caution for model selection.



(a) (b)

Figure 3.9: Coherence scores of 20 randomly sampled words vs. Average overall word frequency.



(a) (b)

Figure 3.10: Coherence scores of actual topics and random words vs. Average overall word frequency.

If a topic model only returns topics with high-frequency generic words, this model might receive a deceivingly high coherence score.

3.6 Conclusion and Discussion

In this chapter, we developed a topic model that jointly models the text, assortative features, generative features, and network data. We borrowed the idea of assortative features and generative features from network community detection research and adapted it for topic modeling. This adaption allows a more natural interpretation of the model-fitting result. Meanwhile, we compared our model to both STM and LDA in terms of several automated evaluation metrics, and showed that our model is able to simultaneously achieve high held-out likelihood, coherence, and stability. By comparing our models with and without the citation network, we found that the main advantage of including the network is the improved stability of the discovered topics.

Our proposed model is highly modular and flexible: Even if some of the components are missing, our model can still be applied. For example, we have seen in the model evaluation that our model can be applied without the network data or the document-level features. Another interesting situation is when there is no text data. In this case, our model becomes a network community detection model similar to the model in Zare et al. (2019), and the only distinction is that we treat the community membership $\boldsymbol{\theta}$ as a hidden unobserved variable, whereas in their model, $\boldsymbol{\theta}$ is a model parameter. Investigating whether this distinction will have an impact on performance in the context of community detection can be an interesting future direction.

The current main limitation of our proposed model is its relatively slow model fitting process. While STM and LDA require a few minutes to fit to the current data

set, our proposed model currently requires about 40 minutes on a personal laptop. Since our model currently relies on a generic estimation procedure, a potential future work is developing a more efficient estimation algorithm to speed up the model fitting process.

Another potentially interesting future direction is the model evaluation. We have seen that the coherence metrics are heavily influenced by the overall word frequency. Devising a more principled model evaluation metric that is immune to the influence of word frequency can be beneficial to the topic modeling community. A potential modification to the coherence metrics is treating the word frequency as a confounding variable, and controlling for it when comparing coherence scores across different models.

The current implementation of our proposed model and the data example in this chapter can be found in `https://github.com/RuizhongMiao/Topic-Model-with-Metadata`.

Chapter 4

Appendix

4.1 Proofs and Additional Simulation Results

4.1.1 Proofs of the Main Theorems

Proofs under the ER-type model

Let $\boldsymbol{U}$, $\boldsymbol{\Lambda}$, $\hat{\boldsymbol{U}}$, $\hat{\boldsymbol{\Lambda}}$ be defined as in (2.23) and (2.24). We introduce the following additional notations to be used:

- $p^* = \max_{1\le i,j\le n} \boldsymbol{P}_{i,j}$.
- $\Delta = |\lambda_r| - |\lambda_{r+1}|$.
- $\kappa = \min\{|\lambda_1/\lambda_r|, 2r\}$.
- $R = (\gamma+1)\log n + r$, where $\gamma > 0$.
- $g = \sqrt{d_{\max}} + \frac{R}{\alpha \log R}$, where $\alpha \in (0,1)$.
- $B(r) = 10\min\{r, 1+\log_2(|\lambda_1/\lambda_r|)\}$.

- $d_{\min} = \min_{1\le i\le n} \sum_{j=1}^{n} \boldsymbol{P}_{i,j}$.
- $d_{\max} = \max_{1\le i\le n} \sum_{j=1}^{n} \boldsymbol{P}_{i,j}$.

As preparation, the following lemmas will be used in our proofs.

Lemma 1 (Lei (2019)). *If $\Delta \succeq \kappa g$ and $|\lambda_r| \succeq \frac{np^*}{\sqrt{n}\|\boldsymbol{U}\|_{2,\infty}}$, we have*

$$\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} \preceq \sqrt{n}\left(\frac{\kappa g}{\Delta}\|\boldsymbol{U}\|_{2,\infty} + \frac{\sqrt{Rp^*}}{|\lambda_r|}\right)\|\boldsymbol{U}\|_{2,\infty}|\lambda_1| + \sqrt{n}\|\boldsymbol{U}\|_{2,\infty}^2\|\boldsymbol{E}\|_2 + \sqrt{n}\left(\frac{\kappa g}{\Delta}\|\boldsymbol{U}\|_{2,\infty} + \frac{\sqrt{Rp^*}}{|\lambda_r|}\right)^2(|\lambda_1| + \|\boldsymbol{E}\|_2), \quad (4.1)$$

with probability $1-(B(r)+1)n^{-\gamma}$.

Proof. This lemma can be proved by combining the Corollary 3.6 and the result in Section 7.4 from Lei (2019). □

Lemma 2 (Theorem 5.2 of Lei and Rinaldo (2015)). *For $c_0 > 0$ and $\gamma > 0$ there exists a constant $C = C(\gamma, c_0)$ such that*

$$\|\boldsymbol{E}\|_2 \le C\max\{\sqrt{np^*}, \sqrt{c_0\log n}\} \quad (4.2)$$

with probability at least $1-n^{-\gamma}$.

Combining Lemma 1 and Lemma 2 would leads to a concentration bound of low-rank approximation with respect to the $\|\cdot\|_{2,\infty}$.

Lemma 3. *Suppose $np^* \succeq \log n$, $\Delta \succeq \kappa g$, and $|\lambda_r| \succeq \frac{np^*}{\sqrt{n}\|\boldsymbol{U}\|_{2,\infty}}$. Then, with probability*

at least $1-(B(r)+2)n^{-\gamma}$*, we have*

$$\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t-\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty}\preceq\sqrt{n}\bigg(\frac{\kappa g}{\Delta}\left\|\boldsymbol{U}\right\|_{2,\infty}^2|\lambda_1|+\frac{\sqrt{Rp^*}}{|\lambda_r|}\left\|\boldsymbol{U}\right\|_{2,\infty}|\lambda_1| \\ +\left\|\boldsymbol{U}\right\|_{2,\infty}^2\sqrt{np^*}+\left(\frac{\kappa g}{\Delta}\right)^2\left\|\boldsymbol{U}\right\|_{2,\infty}^2(|\lambda_1|+\sqrt{np^*})+\frac{Rp^*}{\lambda_r^2}(|\lambda_1|+\sqrt{np^*})\bigg). \quad (4.3)$$

Furthermore, if Assumption 2 holds, (4.3) *becomes*

$$\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t-\hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty}\preceq\frac{\mu_0^2r\kappa g}{\sqrt{n}\Delta}|\lambda_1|+\mu_0|\lambda_1|\frac{\sqrt{rRp^*}}{|\lambda_r|}+\mu_0^2r\sqrt{p^*} \\ +\left(\frac{\kappa g}{\Delta}\right)^2\mu_0^2\frac{r}{\sqrt{n}}(|\lambda_1|+\sqrt{np^*})+\frac{Rp^*}{\lambda_r^2}(\sqrt{n}|\lambda_1|+n\sqrt{p^*}). \quad (4.4)$$

Proof. Plugging (4.2) into (4.1), and applying union bound, we get (4.3). Plugging $\left\|\boldsymbol{U}\right\|_{2,\infty}\leq\mu_0\sqrt{\frac{r}{n}}$ into (4.3), we get (4.4).

□

We now introduce the following theorem that includes Theorem 1 as a special case.

Theorem 5. *Assume the network* $\boldsymbol{A}$ *is generated from the ER-type model in Model 1 under Assumption 2. Suppose* $\Delta\succeq\kappa g$*,* $|\lambda_r|\succeq\frac{np^*}{\sqrt{n}\|\boldsymbol{U}\|_{2,\infty}}$*, and* $np^*\succeq\log n$*. Furthermore, if we have*

$$h(n)\geq C\bigg[\frac{\mu_0^2r}{\sqrt{n}}|\lambda_1|\left(\frac{\kappa g}{\Delta}+\frac{R}{|\lambda_r|}\right)+\mu_0|\lambda_1|\frac{\sqrt{rp^*R}}{|\lambda_r|}+\mu_0^2r\sqrt{p^*} \\ +\left(\frac{\kappa g}{\Delta}+\frac{R}{|\lambda_r|}\right)^2\frac{\mu_0^2r}{\sqrt{n}}(|\lambda_1|+\sqrt{np^*})+\frac{p^*R\sqrt{n}}{\lambda_r^2}(|\lambda_1|+\sqrt{np^*})\bigg]+2|\lambda_{r+1}|+p^*, \quad (4.5)$$

then, for sufficiently large n*, Algorithm 1 exactly identifies the core and periphery set with probability* $1-(B(r)+2)n^{-\gamma}$*.*

Proof of Theorem 5 and Theorem 1. To achieve an exact separation between the core and periphery, we need

$$\min_{i\in\mathcal{C}} \left\|\hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}\right\|_2 > \max_{i\in\mathcal{P}} \left\|\hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}\right\|_2. \tag{4.6}$$

By triangular inequality, we have the following:

$$\text{For } i\in\mathcal{C}\text{: } \left\|\hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}\right\|_2 \geq \|\boldsymbol{P}_{i,*}\boldsymbol{H}\|_2 - \left\|\boldsymbol{P}_{i,*}\boldsymbol{H} - \hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}\right\|_2 \geq h(n) - \left\|\boldsymbol{P}\boldsymbol{H} - \hat{\boldsymbol{P}}\boldsymbol{H}\right\|_{2,\infty}.$$

$$\text{For } i\in\mathcal{P}\text{: } \left\|\hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}\right\|_2 \leq \|\boldsymbol{P}_{i,*}\boldsymbol{H}\|_2 + \left\|\boldsymbol{P}_{i,*}\boldsymbol{H} - \hat{\boldsymbol{P}}_{i,*}\boldsymbol{H}\right\|_2 \leq p^* + \left\|\boldsymbol{P}\boldsymbol{H} - \hat{\boldsymbol{P}}\boldsymbol{H}\right\|_{2,\infty}.$$

Therefore, to satisfy (4.6), it is thus sufficient to ensure that

$$\left\|\boldsymbol{P}\boldsymbol{H} - \hat{\boldsymbol{P}}\boldsymbol{H}\right\|_{2,\infty} \leq \frac{1}{2}(h(n) - p^*). \tag{4.7}$$

By the basic properties of $\|\cdot\|_{2,\infty}$, we have

$$\begin{aligned}
&\left\|\boldsymbol{P}\boldsymbol{H} - \hat{\boldsymbol{P}}\boldsymbol{H}\right\|_{2,\infty} \leq \left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_{2,\infty} \|\boldsymbol{H}\|_2 = \left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_{2,\infty} \\
&= \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t + \boldsymbol{U}_\perp\boldsymbol{\Lambda}_\perp\boldsymbol{U}_\perp^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} \leq \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} + \left\|\boldsymbol{U}_\perp\boldsymbol{\Lambda}_\perp\boldsymbol{U}_\perp^t\right\|_{2,\infty} \\
&\leq \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} + \left\|\boldsymbol{U}_\perp\boldsymbol{\Lambda}_\perp\boldsymbol{U}_\perp^t\right\|_2 = \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} + |\lambda_{r+1}|. \quad (4.8)
\end{aligned}$$

By (4.8) and (4.3) of Lemma 3, (4.7) holds with probability at least $1-(B(r)+2)n^{-\gamma}$ as long as

$$h(n) \geq C\left[\frac{\mu_0^2 r\kappa g}{\sqrt{n}\Delta}|\lambda_1| + \mu_0|\lambda_1|\frac{\sqrt{rRp^*}}{|\lambda_r|} + \mu_0^2 r\sqrt{p^*} \right.$$
$$\left. + \left(\frac{\kappa g}{\Delta}\right)^2 \mu_0^2 \frac{r}{\sqrt{n}}(|\lambda_1| + \sqrt{np^*}) + \frac{Rp^*}{\lambda_r^2}(\sqrt{n}|\lambda_1| + n\sqrt{p^*})\right] + 2|\lambda_{r+1}| + p^*.$$

as assumed by Theorem 5.

Now we proceed to prove Theorem 1, under the additional conditions of $p^* \succeq \max\left\{\frac{\mu_0^2 r\log n}{n}, \frac{\mu_0^2 r^2}{n}\right\}$, boundedness of $|\lambda_1/\lambda_r|$ and Assumption 1. Combined with the fact that

$$g \preceq \sqrt{np^*} + \log n + r,$$

these conditions lead to

$$|\lambda_r| \succ |\lambda_{r+1}|$$

and

$$\Delta \simeq |\lambda_r| \succeq \frac{np^*}{\mu_0\sqrt{r}} \succeq g.$$

Therefore the conditions of Equation (4.3) hold. Inserting the result of Lemma 4 into the third step of (4.8) leads to

$$\left\|\boldsymbol{PH} - \hat{\boldsymbol{P}}\boldsymbol{H}\right\|_{2,\infty} \preceq \mu_0\sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}$$

Therefore, (4.7) is satisfied if

$$h(n) \succeq \mu_0\sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}.$$

as assumed in Theorem 1. □

Lemma 4. *Under the same conditions in Lemma 3, suppose Assumption 1 and As-*

sumption 2 hold. If $p^* \succeq \max\left\{\frac{\mu_0^2 r \log n}{n}, \frac{\mu_0^2 r^2}{n}\right\}$*, and* $|\lambda_1/\lambda_r|$ *is bounded, we have*

$$\left\|\hat{\boldsymbol{P}} - \boldsymbol{P}\right\|_{2,\infty} \preceq \mu_0 \sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}, \tag{4.9}$$

with probability at least $1 - (B(r) + 2)n^{-\gamma}$.

Proof of Lemma 4. Since $p^* \succeq \frac{\mu_0^2 r \log n}{n} \succ \frac{r \log n}{n^2}$, Assumption 1 indicates that $|\lambda_r| \succ |\lambda_{r+1}|$. Together with the boundedness assumption of $|\lambda_1/\lambda_r|$, we know that $|\lambda_1/\Delta|$ is also bounded. In addition, κ also becomes bounded. (4.4) becomes

$$\begin{aligned}\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} \preceq{}& \frac{\mu_0^2 r g}{\sqrt{n}} + \mu_0\sqrt{rRp^*} + \mu_0^2 r\sqrt{p^*} \\ &+ \left(\frac{g}{\Delta}\right)^2 \frac{\mu_0^2 r}{\sqrt{n}}(|\lambda_1| + \sqrt{np^*}) + \frac{Rp^*}{\lambda_r^2}(\sqrt{n}|\lambda_1| + n\sqrt{p^*}).\end{aligned} \tag{4.10}$$

Note that $\|\boldsymbol{U}\|_{2,\infty} \geq \sqrt{\frac{r}{n}}$, so we have $\mu_0 \geq 1$. Therefore, $|\lambda_1| \geq |\lambda_r| \geq \frac{np^*}{\mu_0\sqrt{r}}$. When $p^* \succeq \frac{\mu_0^2 r}{n}$, we have $|\lambda_1| \succeq \sqrt{np^*}$, and (4.10) becomes

$$\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} \preceq \frac{\mu_0^2 r g}{\sqrt{n}} + \mu_0\sqrt{rRp^*} + \mu_0^2 r\sqrt{p^*} + \frac{g^2}{\Delta}\frac{\mu_0^2 r}{\sqrt{n}} + \frac{Rp^*}{|\lambda_r|}\sqrt{n}. \tag{4.11}$$

Furthermore, due to the assumption $r \leq \sqrt{np^*}$, we have $g \preceq \Delta$ and $\frac{g^2}{\Delta}\frac{\mu_0^2 r}{\sqrt{n}} \preceq \mu_0^2 r\sqrt{p^*}$. So (4.11) becomes

$$\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} \preceq \frac{\mu_0^2 r g}{\sqrt{n}} + \mu_0\sqrt{rRp^*} + \mu_0^2 r\sqrt{p^*} + \frac{Rp^*}{|\lambda_r|}\sqrt{n}. \tag{4.12}$$

Plugging in $|\lambda_r| \succeq \frac{np^*}{\sqrt{r}}$, we get

$$\begin{aligned}\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} &\preceq \frac{\mu_0^2 rg}{\sqrt{n}} + \mu_0\sqrt{rRp^*} + \mu_0^2 r\sqrt{p^*} + \frac{\sqrt{r}R}{\sqrt{n}} \\ &\preceq \frac{\mu_0^2 r(\sqrt{np^*} + \log n + r)}{\sqrt{n}} + \mu_0\sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}. \quad (4.13)\end{aligned}$$

Taking into account the condition $p^* \succeq \max\left\{\frac{\mu_0^2 r\log n}{n}, \frac{\mu_0^2 r^2}{n}\right\}$, (4.13) becomes

$$\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} \preceq \mu_0\sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*}. \quad (4.14)$$

Finally, we have

$$\begin{aligned}\left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_{2,\infty} &\leq \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} + \left\|\boldsymbol{U}_\perp\boldsymbol{\Lambda}_\perp\boldsymbol{U}_\perp^t\right\|_{2,\infty} \\ &\leq \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} + \left\|\boldsymbol{U}_\perp\boldsymbol{\Lambda}_\perp\boldsymbol{U}_\perp^t\right\|_2 \\ &\leq \left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^t - \hat{\boldsymbol{U}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{U}}^t\right\|_{2,\infty} + |\lambda_{r+1}| \\ &\preceq \mu_0\sqrt{r(\log n + r)p^*} + \mu_0^2 r\sqrt{p^*},\end{aligned}$$

with probability at least $1 - (B(r) + 2)n^{-\gamma}$. □

Corollary 1 can be proved by specifying a data-driven estimate of the separation order between the scores of the core and periphery components.

Proof of Corollary 1. For any $i \in \mathcal{P}$, we have $\|\boldsymbol{P}_{i,*}\boldsymbol{H}\| < p^*$. Under the event of

Lemma 4, by the boundedness of μ_0 and r, we have

$$\begin{aligned} S_i &= \left\| \hat{\boldsymbol{P}}_{i,*} \boldsymbol{H} \right\|_2 \\ &\leq \left\| \boldsymbol{P}_{i,*} \boldsymbol{H} \right\|_2 + \left\| \hat{\boldsymbol{P}}_{i,*} \boldsymbol{H} - \boldsymbol{P}_{i,*} \boldsymbol{H} \right\|_2 \\ &< p^* + \left\| \hat{\boldsymbol{P}}_{i,*} - \boldsymbol{P}_{i,*} \right\|_2 \\ &\preceq \sqrt{p^* \log n}; \end{aligned}$$

Similarly for any $i \in \mathcal{C}$, since $\|\boldsymbol{P}_{i,*}\boldsymbol{H}\| \geq h(n)$, we have

$$\begin{aligned} S_i &= \left\| \hat{\boldsymbol{P}}_{i,*} \boldsymbol{H} \right\|_2 \\ &\geq h(n) - \left\| \hat{\boldsymbol{P}}_{i,*} \boldsymbol{H} - \boldsymbol{P}_{i,*} \boldsymbol{H} \right\|_2 \\ &\succeq h(n) - \sqrt{p^* \log n}. \end{aligned}$$

Recall that $\hat{p} = \frac{2}{n^2-n} \sum_{i<j} \boldsymbol{A}_{i,j}$ and $\min_{1\leq i,j\leq n} \boldsymbol{P}_{i,j} \simeq \max_{1\leq i,j\leq n} \boldsymbol{P}_{i,j} = p^*$. By Hoeffding's inequality, we know that $\hat{p} \simeq p^*$ with probability greater than $1 - 2n^{-\gamma}$. Therefore, if $h(n) \succ \sqrt{p^{*(1-\epsilon)} \log n}$, for some small constant ϵ, $S_i \succ \sqrt{\hat{p}^{1-\epsilon} \log n}$ for $i \in \mathcal{C}$, and $S_i \preceq \sqrt{\hat{p} \log n}$ for $i \in \mathcal{P}$, for sufficiently large n with probability greater than $1 - (B(r) + 4)n^{-\gamma}$.

□

Finally, the weak consistency can be proved by concentration results with respect to Frobenius norm.

Proof of Theorem 2. First, we want to bound $\left\|\boldsymbol{PH}-\hat{\boldsymbol{P}}\boldsymbol{H}\right\|_F^2$.

$$\begin{aligned}
\left\|\boldsymbol{PH}-\hat{\boldsymbol{P}}\boldsymbol{H}\right\|_F^2 &\preceq \operatorname{rank}(\boldsymbol{PH}-\hat{\boldsymbol{P}}\boldsymbol{H})\cdot\left\|\boldsymbol{PH}-\hat{\boldsymbol{P}}\boldsymbol{H}\right\|_2^2 \\
&\preceq \max\{\operatorname{rank}(\boldsymbol{PH}),\operatorname{rank}(\hat{\boldsymbol{P}}\boldsymbol{H})\}\cdot\left\|\boldsymbol{P}-\hat{\boldsymbol{P}}\right\|_2^2 \\
&\leq \max\{\operatorname{rank}(\boldsymbol{P}),\operatorname{rank}(\hat{\boldsymbol{P}})\}\cdot\left\|\boldsymbol{P}-\hat{\boldsymbol{P}}\right\|_2^2 \\
&= \max\{\operatorname{rank}(\boldsymbol{P}),r\}\cdot\left\|\boldsymbol{P}-\hat{\boldsymbol{P}}\right\|_2^2 \\
&= \max\{\operatorname{rank}(\boldsymbol{P}),r\}\cdot\left\|\boldsymbol{P}-\boldsymbol{A}+\boldsymbol{A}-\hat{\boldsymbol{P}}\right\|_2^2 \\
&\preceq \max\{\operatorname{rank}(\boldsymbol{P}),r\}\cdot\left(\|\boldsymbol{P}-\boldsymbol{A}\|_2+\left\|\boldsymbol{A}-\hat{\boldsymbol{P}}\right\|_2\right)^2 \\
&\preceq \max\{\operatorname{rank}(\boldsymbol{P}),r\}\cdot\left(\|\boldsymbol{P}-\boldsymbol{A}\|_2+|\hat{\lambda}_{r+1}|\right)^2 \\
&\preceq \max\{\operatorname{rank}(\boldsymbol{P}),r\}\cdot\left(\|\boldsymbol{P}-\boldsymbol{A}\|_2+\|\boldsymbol{P}-\boldsymbol{A}\|_2+|\lambda_{r+1}|\right)^2 \\
&\preceq \max\{\operatorname{rank}(\boldsymbol{P}),r\}\cdot\left(\|\boldsymbol{P}-\boldsymbol{A}\|_2+|\lambda_{r+1}|\right)^2.
\end{aligned}$$

Each misclassification necessarily involves a squared deviation of order at least $(h(n)-p^*)^2$. Given the total squared deviation bounded by the above inequality, we can show that the number of misclassified nodes is at most

$$M \preceq \frac{\left\|\boldsymbol{PH}-\hat{\boldsymbol{P}}\boldsymbol{H}\right\|_F^2}{(h(n)-p^*)^2} = C\cdot\max\{r,\operatorname{rank}(\boldsymbol{P})\}\cdot\frac{\left(\|\boldsymbol{P}-\boldsymbol{A}\|_2+|\lambda_{r+1}|\right)^2}{(h(n)-p^*)^2},$$

where C is some constant. Applying Lemma 2, we can get

$$M \preceq \max\{r,\operatorname{rank}(\boldsymbol{P})\}\cdot\frac{\left(\max\{\sqrt{np^*},\sqrt{\log n}\}+|\lambda_{r+1}|\right)^2}{(h(n)-p^*)^2},$$

with probability at least $1-n^{-\gamma}$. □

Proofs under the configuration-type model

We first introduce the ancillary lemmas:

Lemma 5 (Qin and Rohe (2013))**.** *Suppose* $d_{\min} > 3(\gamma+1)\log n$*. Then,*

$$\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2 < \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}$$

with probability greater than $1 - \frac{2}{n^{\gamma}}$*.*

Proof. This lemma is indirectly proved in the proof of Theorem 4.1 in Qin and Rohe (2013). By setting $\tau = 0$, $\epsilon = 4n^{-\gamma}$, and $a = \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}$ in their proof, for each i, we can have

$$P\left(\left|\frac{\hat{\boldsymbol{D}}_{i,i}}{\boldsymbol{D}_{i,i}} - 1\right| \geq \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}\right) \leq 2n^{-\gamma-1}.$$

Then,

$$\begin{aligned}
P\left(\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2 \geq \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}\right) &= P\left(\max_i \left|\frac{\hat{\boldsymbol{D}}_{i,i}}{\boldsymbol{D}_{i,i}} - 1\right| \geq \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}\right) \\
&= P\left(\cup_i \left\{\left|\frac{\hat{\boldsymbol{D}}_{i,i}}{\boldsymbol{D}_{i,i}} - 1\right| \geq \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}\right\}\right) \\
&\leq \sum_{i=1}^{n} P\left(\left|\frac{\hat{\boldsymbol{D}}_{i,i}}{\boldsymbol{D}_{i,i}} - 1\right| \geq \sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}\right) \\
&\leq 2n^{-\gamma}.
\end{aligned}$$

□

Lemma 6. *Under Model 2, we have*

$$\max_{i\in\mathcal{P}} \left\|\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}\boldsymbol{H}\right\|_2 \leq \frac{d_{\max}}{(n-1)d_{\min}}.$$

Proof. We assume the diagonal entries of $\boldsymbol{P}$ are 0. By definition, for $i \in \mathcal{P}$ and $i \neq j$,

$$[\boldsymbol{P}\boldsymbol{D}^{-1}]_{i,j} = \frac{d_i}{\sum_{k\neq i} d_k}.$$

So,

$$[\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H}]_{i,j} = \frac{d_i}{\sum_{k\neq i} d_k} - \frac{n-1}{n}\frac{d_i}{\sum_{k\neq i} d_k} = \frac{d_i}{n\sum_{k\neq i} d_k}$$

for $i \neq j$, and

$$[\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H}]_{i,i} = -\frac{n-1}{n}\frac{d_i}{\sum_{k\neq i} d_k}.$$

Therefore, we have

$$\left\|\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}\boldsymbol{H}\right\|_2 = \sqrt{(n-1)\left(\frac{1}{n}\right)^2 + \left(\frac{n-1}{n}\right)^2}\frac{d_i}{\sum_{k\neq i} d_k} = \sqrt{\frac{n-1}{n}}\frac{d_i}{\sum_{k\neq i} d_k} < \frac{d_{\max}}{(n-1)d_{\min}}.$$

□

We give a more general theorem that includes Theorem 3 as a special case.

Theorem 6. *Assume the network $\boldsymbol{A}$ is generated from the configuration-type model in Model 2 under Assumption 2. Suppose $\Delta \succeq \kappa g$, $|\lambda_r| \succeq \frac{np^*}{\sqrt{n}\|\boldsymbol{U}\|_{2,\infty}}$, $d_{\min} \succ \log n$. If we have*

$$\begin{aligned} h'(n) \succ \frac{\mu_0^2 r|\lambda_1|}{d_{\min}\sqrt{n}}\left(\frac{\kappa g}{\Delta} + \frac{R}{|\lambda_r|}\right) + \frac{\mu_0|\lambda_1|\sqrt{rp^*R}}{d_{\min}|\lambda_r|} + \frac{\mu_0^2 r\sqrt{p^*}}{d_{\min}} \\ + \left(\frac{\kappa g}{\Delta} + \frac{R}{|\lambda_r|}\right)^2 \frac{\mu_0^2 r}{d_{\min}\sqrt{n}}(|\lambda_1| + \sqrt{np^*}) + \frac{p^*R\sqrt{n}}{\lambda_r^2 d_{\min}}(|\lambda_1| + \sqrt{np^*}) + \frac{|\lambda_{r+1}|}{d_{\min}} \\ + \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}} + \frac{d_{\max}}{nd_{\min}}, \quad (4.15) \end{aligned}$$

then, Algorithm 2 exactly identifies the core and periphery set with probability greater than $1 - (B(r)+4)n^{-\gamma}$, for some positive constant γ.

Proof of Theorem 6 and Theorem 3. First, we have $\|\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}\boldsymbol{H}\|_2 \geq h'(n)$ for $i \in \mathcal{C}$. Also, by Lemma 6, we have that $\|\boldsymbol{P}_{i,*}\boldsymbol{D}^{-1}\boldsymbol{H}\|_2 \leq \frac{d_{\max}}{(n-1)d_{\min}}$ for $i \in \mathcal{P}$. To achieve strong consistency, we need to have

$$h'(n) > 2\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_{2,\infty} + \frac{d_{\max}}{(n-1)d_{\min}}. \tag{4.16}$$

In the following, we give a bound for $\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_{2,\infty}$. Notice that

$$\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_{2,\infty} \leq \left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty}\|\boldsymbol{H}\|_2 \leq \left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty}. \tag{4.17}$$

Meanwhile, we have

$$\begin{aligned}
\left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty} &= \left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\boldsymbol{D}^{-1} + \hat{\boldsymbol{P}}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty} \\
&= \left\|(\boldsymbol{P} - \hat{\boldsymbol{P}})\boldsymbol{D}^{-1} + \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}(\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I})\right\|_{2,\infty} \\
&= \left\|(\boldsymbol{P} - \hat{\boldsymbol{P}})\boldsymbol{D}^{-1} + (\hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1} - \boldsymbol{P}\boldsymbol{D}^{-1} + \boldsymbol{P}\boldsymbol{D}^{-1})(\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I})\right\|_{2,\infty} \\
\leq \left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_{2,\infty}&\left\|\boldsymbol{D}^{-1}\right\|_2 + \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2 + \left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty}\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2.
\end{aligned}$$

Moving the term $\left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty}\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2$ from the right to the left, we get

$$\left(1 - \left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2\right)\left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty} \leq \left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_{2,\infty}\left\|\boldsymbol{D}^{-1}\right\|_2 + \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2. \tag{4.18}$$

By Lemma 5, if $d_{\min} \succ \log n$, $\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1} - \boldsymbol{I}\right\|_2$ is vanishing for sufficiently large n

with high probability. Therefore, we have

$$\begin{aligned}\left\|\boldsymbol{P}\boldsymbol{D}^{-1}-\hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_{2,\infty} &\preceq \left\|\boldsymbol{P}-\hat{\boldsymbol{P}}\right\|_{2,\infty}\left\|\boldsymbol{D}^{-1}\right\|_2+\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\left\|\hat{\boldsymbol{D}}\boldsymbol{D}^{-1}-\boldsymbol{I}\right\|_2 \\ &\preceq \left\|\boldsymbol{P}-\hat{\boldsymbol{P}}\right\|_{2,\infty}\frac{1}{d_{\min}}+\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}} \quad (4.19)\end{aligned}$$

with probability greater than $1-\frac{2}{n^\gamma}$.

Under Assumption 2, applying Lemma 3 and (4.17), we can see that (4.16) are satisfied with probability greater than $1-(B(r)+4)n^{-\gamma}$, if

$$\begin{aligned}h'(n) \succ\ & \frac{\mu_0^2 r\kappa g|\lambda_1|}{\Delta d_{\min}\sqrt{n}}+\frac{\mu_0|\lambda_1|\sqrt{rRp^*}}{|\lambda_r|d_{\min}}+\frac{\mu_0^2 r\sqrt{p^*}}{d_{\min}} \\ &+\frac{\mu_0^2 r\kappa^2 g^2}{\Delta^2 d_{\min}\sqrt{n}}(|\lambda_1|+\sqrt{np^*})+\frac{Rp^*}{\lambda_r^2 d_{\min}}(\sqrt{n}|\lambda_1|+n\sqrt{p^*})+\frac{|\lambda_{r+1}|}{d_{\min}} \\ &+\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\sqrt{\frac{3(\gamma+1)\log n}{d_{\min}}}+\frac{d_{\max}}{nd_{\min}}\end{aligned}$$

as stated in the theorem.

Furthermore, to see how this leads to Theorem 3, suppose Assumption 1 holds, and assume $p^* \succeq \max\left\{\frac{\mu_0^2 r\log n}{n}, \frac{\mu_0^2 r^2}{n}\right\}$, and $|\lambda_1/\lambda_r|$ is bounded. Applying Lemma 4 to (4.15) gives

$$h'(n) \succ \frac{1}{d_{\min}}\left(\mu_0\sqrt{r(\log n+r)p^*}+\mu_0^2 r\sqrt{p^*}\right)+\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_{2,\infty}\sqrt{\frac{\log n}{d_{\min}}}$$

as stated in Theorem 3. □

Proof of Corollary 2. For any $i \in \mathcal{P}$, by Lemma 6 and Lemma 4, and the boundedness

of μ_0 and r, (4.17) and (4.19) lead to

$$\begin{aligned} S_i &= \left\| \hat{\boldsymbol{P}}_{i,*} \hat{\boldsymbol{D}}^{-1} \boldsymbol{H} \right\|_2 \\ &\leq \left\| \boldsymbol{P}_{i,*} \boldsymbol{D}^{-1} \boldsymbol{H} \right\|_2 + \left\| \hat{\boldsymbol{P}}_{i,*} \hat{\boldsymbol{D}}^{-1} \boldsymbol{H} - \boldsymbol{P}_{i,*} \boldsymbol{D}^{-1} \boldsymbol{H} \right\|_2 \\ &\leq \frac{d_{\max}}{(n-1) d_{\min}} + \left\| \hat{\boldsymbol{P}}_{i,*} \hat{\boldsymbol{D}}^{-1} \boldsymbol{H} - \boldsymbol{P}_{i,*} \boldsymbol{D}^{-1} \boldsymbol{H} \right\|_2 \\ &\preceq \frac{\sqrt{p^* \log n}}{d_{\min}} + \left\| \boldsymbol{P} \boldsymbol{D}^{-1} \right\|_{2,\infty} \sqrt{\frac{\log n}{d_{\min}}}; \end{aligned}$$

Similarly for any $i \in \mathcal{C}$, using the fact that $\|\boldsymbol{P}_{i,*} \boldsymbol{D}^{-1} \boldsymbol{H}\| \geq h'(n)$ and Lemma 4,

$$\begin{aligned} S_i &= \left\| \hat{\boldsymbol{P}}_{i,*} \hat{\boldsymbol{D}}^{-1} \boldsymbol{H} \right\|_2 \\ &\geq h'(n) - \left\| \hat{\boldsymbol{P}}_{i,*} \hat{\boldsymbol{D}}^{-1} \boldsymbol{H} - \boldsymbol{P}_{i,*} \boldsymbol{D}^{-1} \boldsymbol{H} \right\|_2 \\ &\succeq h'(n) - \frac{\sqrt{p^* \log n}}{d_{\min}} - \left\| \boldsymbol{P} \boldsymbol{D}^{-1} \right\|_{2,\infty} \sqrt{\frac{\log n}{d_{\min}}}, \end{aligned}$$

for sufficiently large n with probability $1 - (B(r) + 4) n^{-\gamma}$.

When $\min_{1 \leq i,j \leq n} \boldsymbol{P}_{i,j} \simeq \max_{1 \leq i,j \leq n} \boldsymbol{P}_{i,j} = p^*$, we have

$$\begin{aligned} \frac{\sqrt{p^* \log n}}{d_{\min}} + \left\| \boldsymbol{P} \boldsymbol{D}^{-1} \right\|_{2,\infty} \sqrt{\frac{\log n}{d_{\min}}} &\simeq \frac{\sqrt{p^* \log n}}{n p^*} + \left\| \boldsymbol{P} \boldsymbol{D}^{-1} \right\|_{2,\infty} \sqrt{\frac{\log n}{n p^*}} \\ &\preceq \frac{\sqrt{\log n}}{n \sqrt{p^*}} + \|\boldsymbol{P}\|_{2,\infty} \left\| \boldsymbol{D}^{-1} \right\|_2 \sqrt{\frac{\log n}{n p^*}} \\ &\simeq \frac{\sqrt{\log n}}{n \sqrt{p^*}} + \frac{\sqrt{n} p^*}{n p^*} \sqrt{\frac{\log n}{n p^*}} \\ &\simeq \frac{\sqrt{\log n}}{n \sqrt{p^*}}. \end{aligned}$$

Furthermore, in the proof of Corollary 1, we have shown that $\hat{p} \simeq p^*$ with probability greater than $1 - 2n^{-\gamma}$. In this case, if $h'(n) \succ \frac{\sqrt{\log n}}{n \sqrt{p^{*1+\epsilon}}}$, we have $S_i \succ \frac{\sqrt{\log n}}{n \sqrt{\hat{p}^{1+\epsilon}}}$

for $i \in \mathcal{C}$, and $S_i \preceq \frac{\sqrt{\log n}}{n\sqrt{\bar{p}}}$ for $i \in \mathcal{P}$.

□

Proof of Theorem 4. The key idea remains the same as in the proof of Theorem 2. We want to bound $\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_F^2$.

$$\begin{aligned}\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_F^2 &\preceq \max\{\operatorname{rank}(\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H}), \operatorname{rank}(\hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H})\} \cdot \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_2^2 \\ &\leq \max\{\operatorname{rank}(\boldsymbol{P}), r\} \cdot \left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_2^2\end{aligned}$$

Using an argument similar to (4.18), and (4.19), with probability greater than $1 - \frac{2}{n^\gamma}$ we have

$$\left\|\boldsymbol{P}\boldsymbol{D}^{-1} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\right\|_2 \preceq \left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_2 \frac{1}{d_{\min}} + \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_2 \sqrt{\frac{\log n}{d_{\min}}}.$$

Meanwhile, by Lemma 2, we also have

$$\begin{aligned}\left\|\boldsymbol{P} - \hat{\boldsymbol{P}}\right\|_2 &= \left\|\boldsymbol{P} - \boldsymbol{A} + \boldsymbol{A} - \hat{\boldsymbol{P}}\right\|_2 \\ &\leq \|\boldsymbol{P} - \boldsymbol{A}\|_2 + \left\|\boldsymbol{A} - \hat{\boldsymbol{P}}\right\|_2 \\ &\leq \|\boldsymbol{P} - \boldsymbol{A}\|_2 + |\hat{\lambda}_{r+1}| \\ &\leq 2\|\boldsymbol{P} - \boldsymbol{A}\|_2 + |\lambda_{r+1}| \\ &\preceq \sqrt{np^*} + |\lambda_{r+1}|,\end{aligned}$$

with probability at least $1 - \frac{1}{n^\gamma}$. Therefore, combining the above equations, we get

$$\left\|\boldsymbol{P}\boldsymbol{D}^{-1}\boldsymbol{H} - \hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1}\boldsymbol{H}\right\|_F^2 \preceq \max\{\operatorname{rank}(\boldsymbol{P}), r\} \cdot \left[\frac{np^* + \lambda_{r+1}^2}{d_{\min}^2} + \left\|\boldsymbol{P}\boldsymbol{D}^{-1}\right\|_2^2 \frac{\log n}{d_{\min}}\right],$$

with probability at least $1 - \frac{3}{n^\gamma}$, and the number of misclassified nodes satisfies

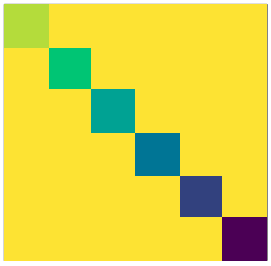
$$M' \preceq \max\{\text{rank}(\boldsymbol{P}), r\} \cdot \frac{\left[np^* + \lambda_{r+1}^2 + \|\boldsymbol{P}\boldsymbol{D}^{-1}\|_2^2 \cdot d_{\min} \cdot \log n\right]}{d_{\min}^2 \left[h'(n) - \frac{d_{\max}}{(n-1)d_{\min}}\right]^2}.$$

□

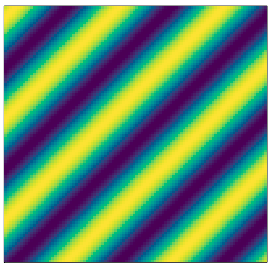
4.1.2 Additional Simulation Results

In this section, we include the additional simulation results, where the core size and the periphery size are different. We can see that our method achieves the best performance across different settings, which is consistent with the balanced cases.

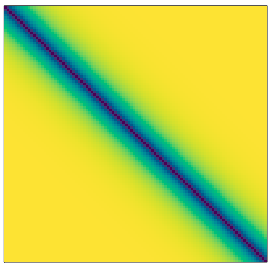
4.2 Additional Topic Modeling Results



(a) Graphon 1

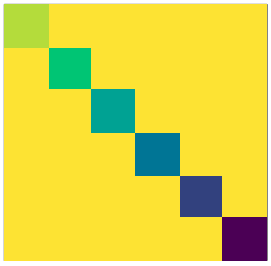
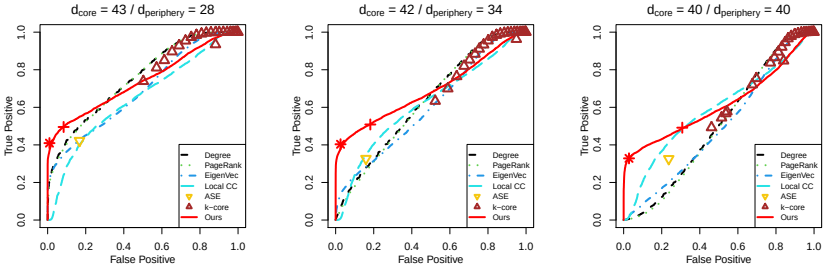


(b) Graphon 2

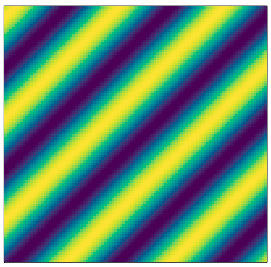
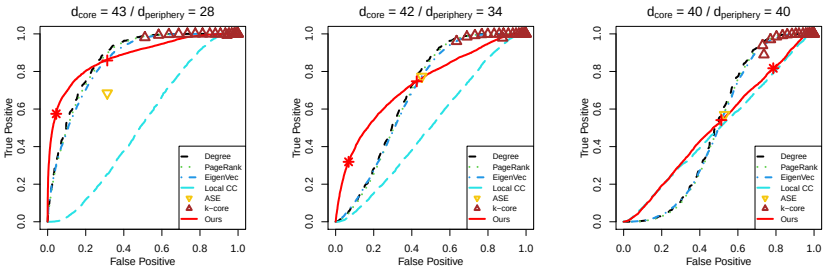


(c) Graphon 3

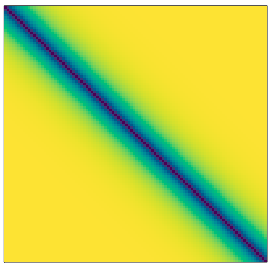
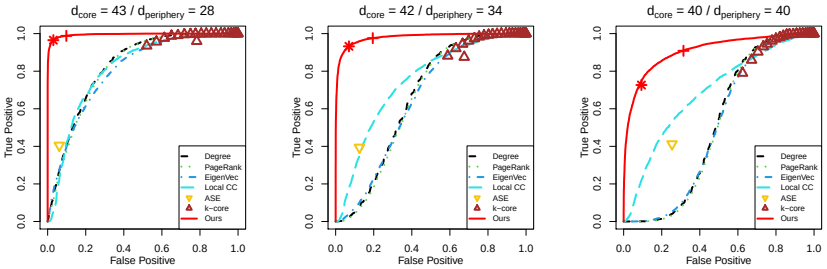
Figure 4.1: Erdös-Renyi periphery. $N_{\mathcal{C}} = 700$, $N_{\mathcal{P}} = 1300$.

(a) Graphon 1

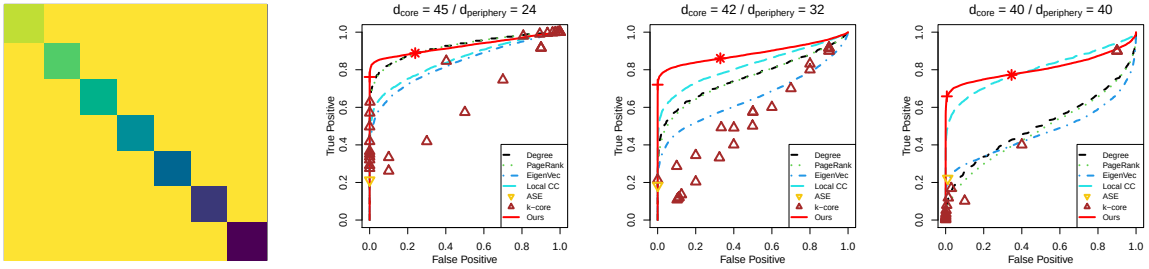



(b) Graphon 2

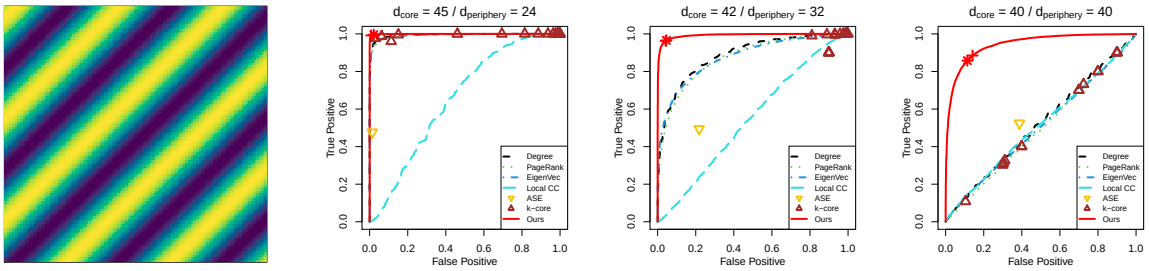



(c) Graphon 3

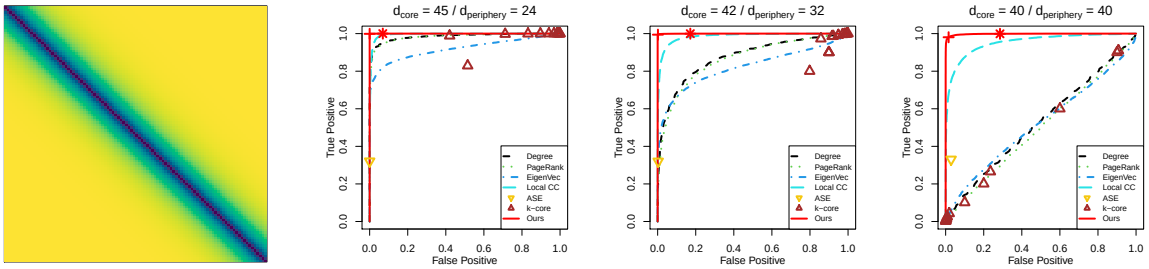
Figure 4.2: Configuration periphery. $N_{\mathcal{C}} = 700$, $N_{\mathcal{P}} = 1300$.



(a) Graphon 1

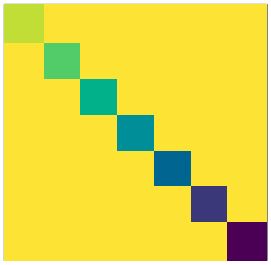


(b) Graphon 2

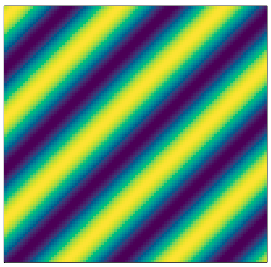


(c) Graphon 3

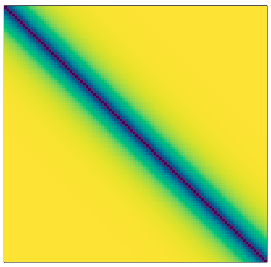
Figure 4.3: Erdös-Renyi periphery. $N_{\mathcal{C}} = 1300$, $N_{\mathcal{P}} = 700$.



(a) Graphon 1



(b) Graphon 2



(c) Graphon 3

Figure 4.4: Configuration periphery. $N_{\mathcal{C}} = 1300$, $N_{\mathcal{P}} = 700$.

4.2.1 $K = 5$

Table 4.1: The most representative words for each topic ($K = 5$). These words are sorted in descending order based on Equation (3.24). The numbers in the parentheses are the word probabilities in the corresponding topic.

Id	Most representative words
1	model(0.058), process(0.023), time(0.021), distribut(0.028), prior(0.012), posterior(0.008), bayesian(0.01), seri(0.008), mont(0.007), carlo(0.007)
2	estim(0.076), regress(0.029), select(0.018), covari(0.017), linear(0.014), propos(0.023), predictor(0.008), coeffici(0.008), method(0.027), effici(0.01)
3	effect(0.02), data(0.031), treatment(0.011), cluster(0.008), outcom(0.006), gene(0.005), studi(0.017), analysi(0.011), subject(0.005), random(0.011)
4	design(0.019), optim(0.016), adapt(0.009), converg(0.009), bound(0.008), densiti(0.009), function(0.021), space(0.008), problem(0.015), nois(0.005)
5	test(0.059), procedur(0.024), statist(0.026), bootstrap(0.01), power(0.01), confid(0.01), null(0.009), hypothesi(0.008), control(0.01), interv(0.009)

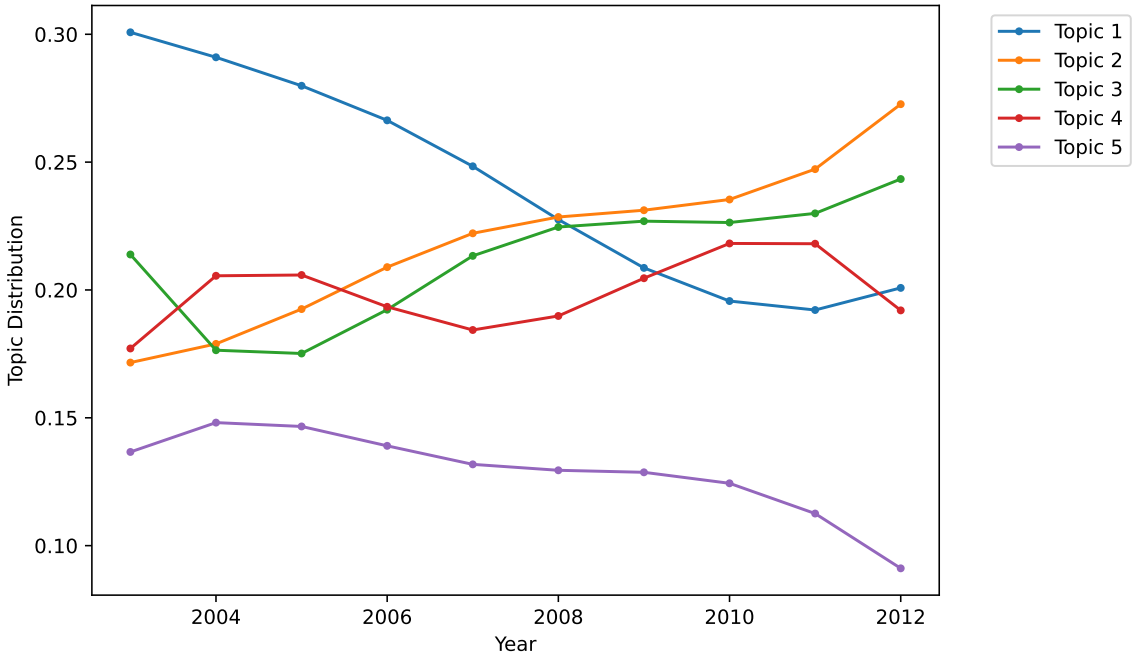


Figure 4.5: Effects of the assortative features (publication year) on the Topic distributions.

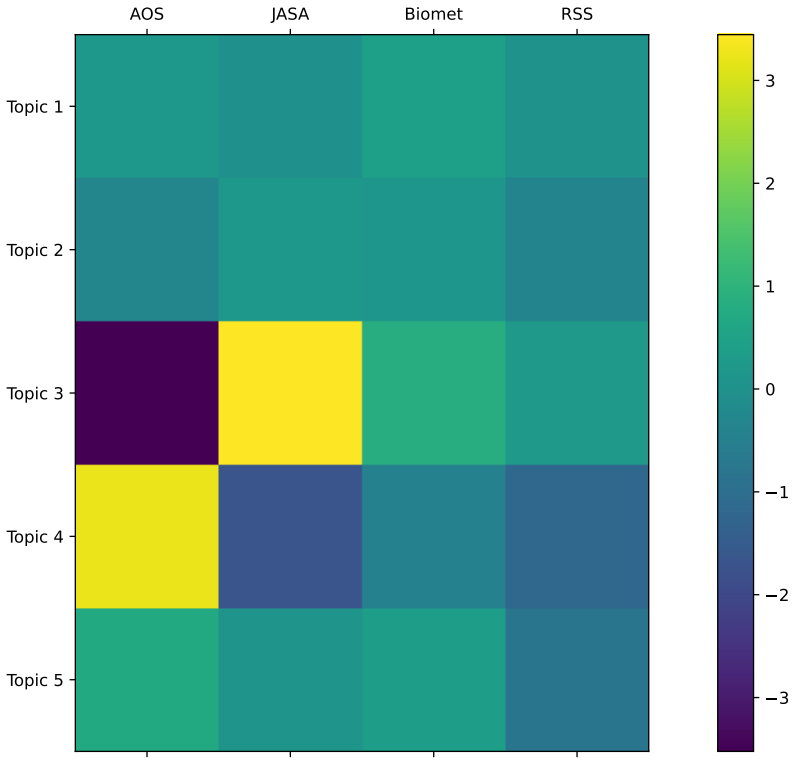


Figure 4.6: Generative feature: Topic distributions vs Publication Journal.

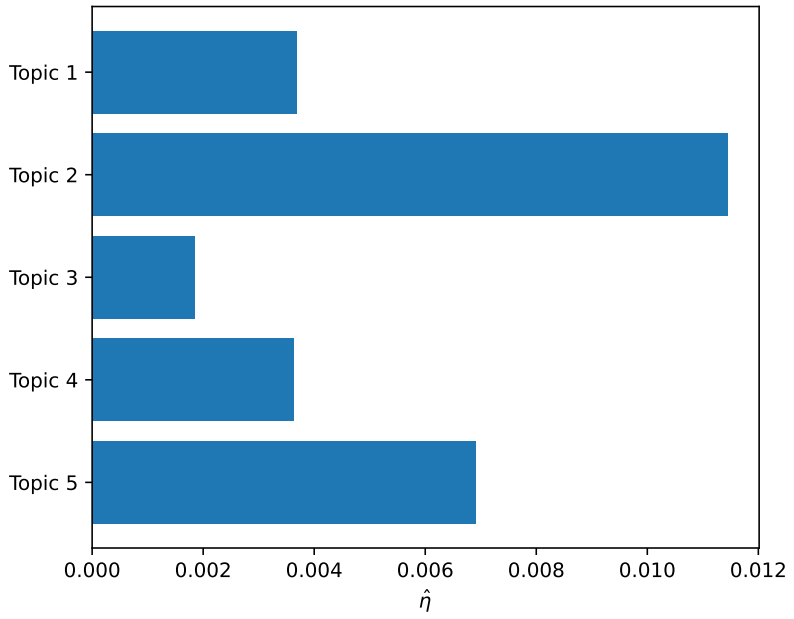


Figure 4.7: Estimated topic assortative weight $\hat{\boldsymbol{\eta}}$.

4.2.2 $K = 20$

Table 4.2: The most representative words for each topic ($K = 20$). These words are sorted in descending order based on Equation (3.24). The numbers in the parentheses are the word probabilities in the corresponding topic.

Id	Most representative words
1	prior(0.053), model(0.11), bayesian(0.049), mixtur(0.032), posterior(0.032), distribut(0.035), bay(0.014), dirichlet(0.011), paramet(0.025), frequentist(0.008)
2	test(0.177), procedur(0.055), power(0.029), null(0.028), hypothesi(0.026), control(0.027), fals(0.02), hypothes(0.018), statist(0.033), discoveri(0.014)
3	effect(0.075), treatment(0.048), random(0.035), outcom(0.024), trial(0.017), causal(0.014), patient(0.014), exposur(0.012), adjust(0.013), assign(0.011)
4	design(0.116), optim(0.056), project(0.014), construct(0.018), factor(0.015), balanc(0.01), orthogon(0.011), experi(0.013), minimum(0.01), run(0.009)
5	covari(0.053), estim(0.078), miss(0.021), data(0.049), quantil(0.016), effici(0.024), weight(0.018), imput(0.012), robust(0.016), correl(0.013)
6	estim(0.148), error(0.052), varianc(0.038), bootstrap(0.027), confid(0.025), interv(0.024), bias(0.017), measur(0.02), small(0.013), squar(0.014)
7	predict(0.039), model(0.083), forecast(0.014), survey(0.011), health(0.009), inform(0.016), system(0.009), year(0.007), calibr(0.007), uncertainti(0.007)
8	sampl(0.093), size(0.037), problem(0.039), larg(0.023), classif(0.013), classifi(0.01), theori(0.013), number(0.017), theoret(0.012), small(0.011)
9	spatial(0.035), point(0.025), field(0.016), data(0.042), process(0.027), extrem(0.011), region(0.012), space(0.014), intens(0.009), structur(0.014)
10	class(0.115), span(0.048), graphic(0.039), residu(0.031), amp(0.029), inlin(0.025), formula(0.025), alt(0.017), fit(0.021), model(0.063)
11	function(0.072), dimens(0.032), predictor(0.033), curv(0.021), dimension(0.026), compon(0.025), reduct(0.017), princip(0.015), regress(0.026), analysi(0.022)
12	distribut(0.096), condit(0.057), multivari(0.024), famili(0.018), independ(0.02), rank(0.016), variabl(0.026), tail(0.011), general(0.024), case(0.02)
13	select(0.071), variabl(0.04), lasso(0.019), spars(0.017), penalti(0.015), penal(0.014), regular(0.012), oracl(0.009), dimension(0.015), sparsiti(0.008)
14	cluster(0.036), data(0.052), gene(0.02), express(0.015), articl(0.019), onlin(0.011), materi(0.011), correl(0.015), genet(0.009), studi(0.027)
15	time(0.043), surviv(0.022), censor(0.022), hazard(0.02), event(0.021), failur(0.013), studi(0.031), data(0.037), proport(0.011), cancer(0.01)
16	time(0.087), process(0.069), seri(0.047), frequenc(0.016), spectral(0.015), autoregress(0.015), dynam(0.015), stationari(0.014), stochast(0.015), volatil(0.012)
17	likelihood(0.125), paramet(0.093), asymptot(0.068), maximum(0.045), normal(0.043), estim(0.095), ratio(0.022), empir(0.021), consist(0.022), nuisanc(0.009)
18	algorithm(0.079), mont(0.039), carlo(0.039), markov(0.034), chain(0.024), comput(0.034), approxim(0.027), method(0.042), state(0.012), filter(0.009)
19	model(0.098), linear(0.043), regress(0.047), smooth(0.028), nonparametr(0.028), parametr(0.021), local(0.019), spline(0.014), coeffici(0.017), addit(0.018)
20	densiti(0.035), rate(0.037), function(0.053), bound(0.025), converg(0.028), minimax(0.013), risk(0.015), adapt(0.016), estim(0.058), download(0.008)

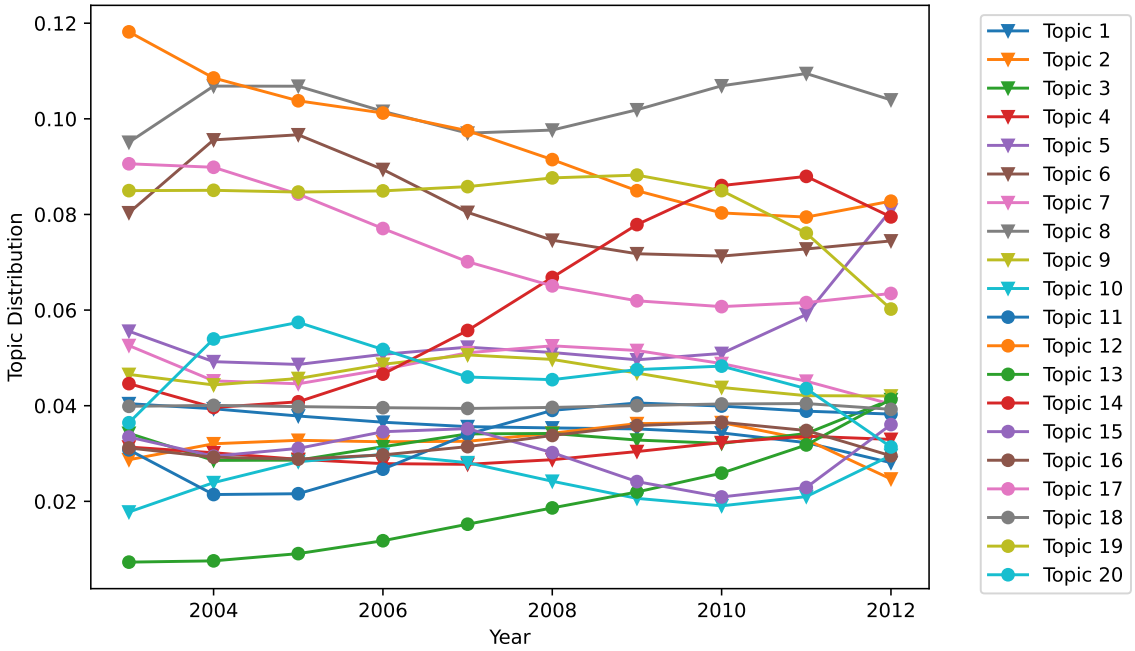


Figure 4.8: Effects of the assortative features (publication year) on the Topic distributions.

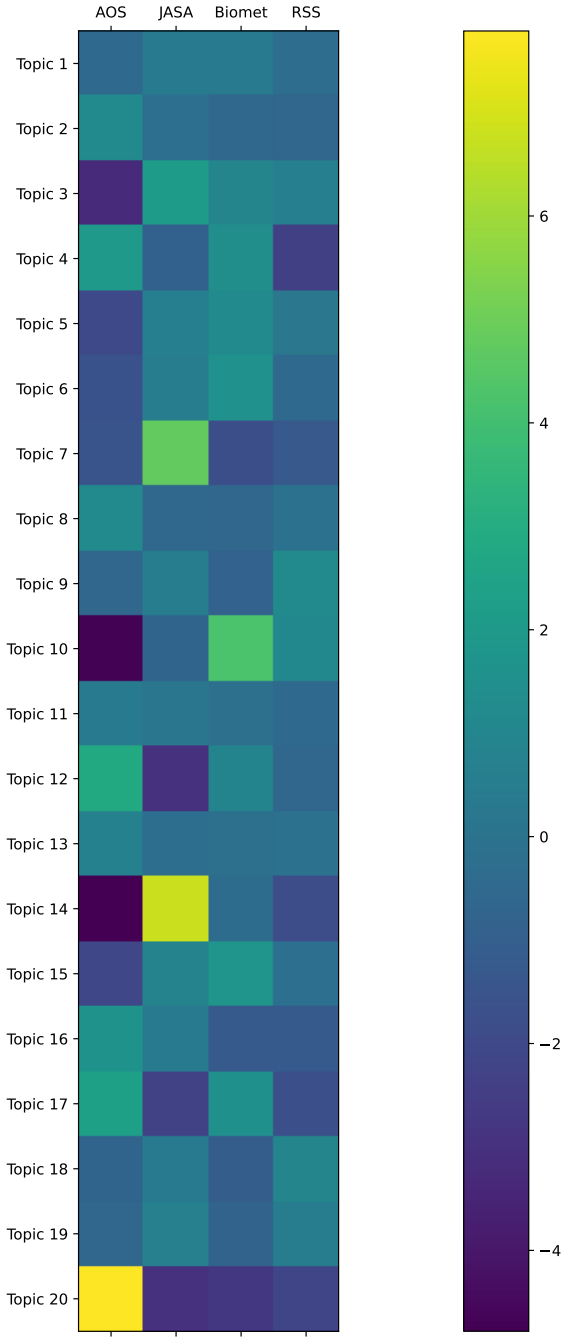


Figure 4.9: Generative feature: Topic distributions vs Publication Journal.

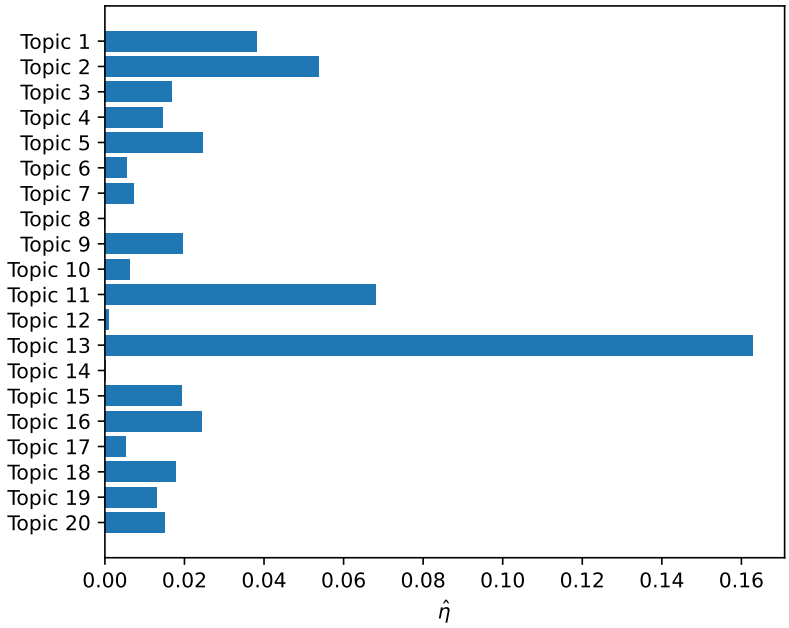


Figure 4.10: Estimated topic assortative weight $\hat{\boldsymbol{\eta}}$.

Bibliography

Abbe, E., Fan, J., Wang, K., Zhong, Y., et al. (2020). Entrywise eigenvector analysis of random matrices with low expected rank. *Annals of Statistics*, 48(3):1452–1474.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2015). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Alba, R. D. and Moore, G. (1978). Elite social circles. *Sociological Methods & Research*, 7(2):167–188.

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47.

Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598.

Alon, N., Krivelevich, M., and Sudakov, B. (1998). Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466.

Alvarez-Hamelin, J. I., Dall'Asta, L., Barrat, A., and Vespignani, A. (2006). Large scale networks fingerprinting and visualization using the k-core decomposition. In *Advances in neural information processing systems*, pages 41–50.

Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T.,

Levin, K., Lyzinski, V., and Qin, Y. (2017). Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, 18(1):8393–8484.

Baker, D. R. (1992). A structural analysis of social work journal network: 1985-1986. *Journal of Social Service Research*, 15(3-4):153–168.

Barucca, P., Tantari, D., and Lillo, F. (2016). Centrality metrics and localization in core-periphery networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(2):023401.

Baxter, G., Dorogovtsev, S., Lee, K.-E., Mendes, J., and Goltsev, A. (2015). Critical dynamics of the k-core pruning process. *Physical Review X*, 5(3):031017.

Bhawalkar, K., Kleinberg, J., Lewi, K., Roughgarden, T., and Sharma, A. (2015). Preventing unraveling in social networks: the anchored k-core problem. *SIAM Journal on Discrete Mathematics*, 29(3):1452–1475.

Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd international Conference on Machine Learning. ICML*, volume 6, pages 113–120.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Bollobás, B. (1980). A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316.

Bonacich, P. (1987). Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182.

Borgatti, S. P. and Everett, M. G. (2000). Models of core/periphery structures. *Social networks*, 21(4):375–395.

Boutsidis, C., Kambadur, P., and Gittens, A. (2015). Spectral clustering via the power method-provably. In *International Conference on Machine Learning*, pages 40–48.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web. *Computer networks*, 33(1-6):309–320.

Butucea, C., Ingster, Y. I., and Suslina, I. A. (2015). Sharp variable selection of a sparse submatrix in a high-dimensional noisy matrix. *ESAIM: Probability and Statistics*, 19:115–134.

Cai, T. T., Liang, T., Rakhlin, A., et al. (2017). Computational and statistical boundaries for submatrix localization in a large noisy matrix. *The Annals of Statistics*, 45(4):1403–1430.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717.

Cape, J., Tang, M., and Priebe, C. E. (2019). The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *The Annals of Statistics*, 47(5):2405–2439.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214.

Chen, Y. (2015). Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923.

Chuang, J., Roberts, M. E., Stewart, B. M., Weiss, R., Tingley, D., Grimmer, J., and Heer, J. (2015). Topiccheck: Interactive alignment for assessing topic model stability. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–184.

Chung, F. and Lu, L. (2002). The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882.

Comrey, A. L. (1962). The minimum residual method of factor analysis. *Psychological Reports*, 11(1):15–18.

Cucuringu, M., Rombach, P., Lee, S. H., and Porter, M. A. (2016). Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics*, 27(6):846–887.

Da Silva, M. R., Ma, H., and Zeng, A.-P. (2008). Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. *Proceedings of the IEEE*, 96(8):1411–1420.

Dekel, Y., Gurel-Gurevich, O., and Peres, Y. (2014). Finding hidden cliques in linear time with high probability. *Combinatorics, Probability and Computing*, 23(1):29–49.

Della Rossa, F., Dercole, F., and Piccardi, C. (2013). Profiling core-periphery network structure by random walkers. *Scientific reports*, 3:1467.

Deshpande, Y. and Montanari, A. (2015). Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems. In *Conference on Learning Theory*, pages 523–562.

Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. F. (2006). K-core organization of complex networks. *Physical review letters*, 96(4):040601.

Elliott, A., Chiu, A., Bazzi, M., Reinert, G., and Cucuringu, M. (2020). Core–periphery structure in directed networks. *Proceedings of the Royal Society A*, 476(2241):20190783.

Erdös, P. (1959). On random graphs. *Publicationes mathematicae*, 6:290–297.

Fan, J., Wang, W., and Zhong, Y. (2018). An ℓ_∞ eigenvector perturbation bound and its application to robust covariance estimation. *Journal of Machine Learning Research*, 18(207):1–42.

Fei, Y. and Chen, Y. (2018). Exponential error rates of sdp for block models: Beyond grothendieck's inequality. *IEEE Transactions on Information Theory*, 65(1):551–571.

Feldman, V., Grigorescu, E., Reyzin, L., Vempala, S. S., and Xiao, Y. (2017). Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2):1–37.

Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345.

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

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Gao, C. and Lafferty, J. (2017). Testing for global network structure using small subgraph statistics. *arXiv preprint arXiv:1710.00862*.

Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. (2017). Achieving optimal misclassification proportion in stochastic block models. *The Journal of Machine Learning Research*, 18(1):1980–2024.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Gu, S., Xia, C. H., Ciric, R., Moore, T. M., Gur, R. C., Gur, R. E., Satterthwaite, T. D., and Bassett, D. S. (2020). Unifying the notions of modularity and core–periphery structure in functional brain networks during youth. *Cerebral Cortex*, 30(3):1087–1102.

Guimera, R. and Amaral, L. A. N. (2005). Functional cartography of complex metabolic networks. *nature*, 433(7028):895.

Hajek, B., Wu, Y., and Xu, J. (2017). Information limits for recovering a hidden community. *IEEE Transactions on Information Theory*, 63(8):4729–4745.

Hazan, E. and Krauthgamer, R. (2011). How hard is it to approximate the best nash equilibrium? *SIAM Journal on Computing*, 40(1):79–91.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Holme, P. (2005). Core-periphery organization of complex networks. *Physical Review E*, 72(4):046111.

Hoover, D. N. (1979). Relations on probability spaces and arrays of random variables. *Preprint, Institute for Advanced Study, Princeton, NJ*, 2.

Hoyle, A., Goel, P., Hian-Cheong, A., Peskov, D., Boyd-Graber, J., and Resnik, P. (2021). Is automated topic model evaluation broken? the incoherence of coherence. *Advances in Neural Information Processing Systems*, 34.

Ji, P. and Jin, J. (2016). Coauthorship and citation networks for statisticians. *The Annals of Applied Statistics*, 10(4):1779–1812.

Jia, J. and Benson, A. R. (2019). Random spatial network models for core-periphery structure. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 366–374.

Jin, J. (2015). Fast community detection by score. *The Annals of Statistics*, 43(1):57–89.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632.

Kojaku, S. and Masuda, N. (2017). Finding multiple core-periphery pairs in networks. *Physical Review E*, 96(5):052313.

Kojaku, S. and Masuda, N. (2018). Core-periphery structure requires something else in the network. *New Journal of Physics*, 20(4):043012.

Kojaku, S., Xu, M., Xia, H., and Masuda, N. (2019). Multiscale core-periphery structure in a global liner shipping network. *Scientific reports*, 9(1):1–15.

Kučera, L. (1995). Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212.

Le, C. M. and Li, T. (2020). Linear regression and its inference on noisy network-linked data. *arXiv preprint arXiv:2007.00803*.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Lee, S. H., Cucuringu, M., and Porter, M. A. (2014). Density-based and transport-based core-periphery structures in networks. *Physical Review E*, 89(3):032810.

Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237.

Lei, L. (2019). Unified $\ell_{2\to\infty}$ eigenspace perturbation theory for symmetric random matrices. *arXiv preprint arXiv:1909.04798*.

Lei, L., Li, X., and Lou, X. (2020). Consistency of spectral clustering on hierarchical stochastic block models. *arXiv preprint arXiv:2004.14531*.

Li, T., Lei, L., Bhattacharyya, S., Van den Berge, K., Sarkar, P., Bickel, P. J., and Levina, E. (2020a). Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, pages 1–39.

Li, T., Levina, E., and Zhu, J. (2020b). Community models for partially observed networks from surveys. *arXiv preprint arXiv:2008.03652*.

Li, T., Levina, E., and Zhu, J. (2020c). Network cross-validation by edge sampling. *Biometrika*, 107(2):257–276.

Li, T., Levina, E., Zhu, J., et al. (2019). Prediction models for network-linked data. *The Annals of Applied Statistics*, 13(1):132–164.

Lim, K. W. and Buntine, W. (2015). Bibliographic analysis with the citation network topic model. In *Asian conference on machine learning*, pages 142–158. PMLR.

Lim, K. W., Chen, C., and Buntine, W. (2016). Twitter-network topic model: A full bayesian treatment for social network and text modeling. *arXiv preprint arXiv:1609.06791*.

Liu, Y., Niculescu-Mizil, A., and Gryc, W. (2009). Topic-link lda: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*, pages 665–672.

Ma, Z., Ma, Z., and Yuan, H. (2020). Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4):1–67.

Mantyla, M. V., Claes, M., and Farooq, U. (2018). Measuring lda topic stability from clusters of replicated runs. In *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*, pages 1–4.

Milgram, S. (1967). The small world problem. *Psychology today*, 2(1):60–67.

Mimno, D., Wallach, H., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 262–272.

Mossel, E., Neeman, J., and Sly, A. (2012). Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499*.

Mossel, E., Neeman, J., and Sly, A. (2018). A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708.

Mukherjee, S. S., Sarkar, P., Wang, Y. R., and Yan, B. (2018). Mean field for the stochastic blockmodel: optimization landscape and convergence issues. In *Advances in Neural Information Processing Systems*, pages 10694–10704.

Mullins, N. C., Hargens, L. L., Hecht, P. K., and Kick, E. L. (1977). The group structure of cocitation clusters: A comparative study. *American sociological review*, pages 552–562.

Naik, C., Caron, F., and Rousseau, J. (2021). Sparse networks with core-periphery structure. *Electronic Journal of Statistics*, 15(1):1814–1868.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108.

Newman, M. (2018). The configuration model. In *Networks*. Oxford University Press.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Newman, M. E. (2016a). Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315.

Newman, M. E. (2016b). Mathematics of networks. *The new Palgrave dictionary of economics*, pages 1–8.

Newman, M. E. and Peixoto, T. P. (2015). Generalized communities in networks. *Physical review letters*, 115(8):088701.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856.

Nielsen, S. F. (2000). The stochastic em algorithm: estimation and asymptotic results. *Bernoulli*, pages 457–489.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Priebe, C. E., Park, Y., Vogelstein, J. T., Conroy, J. M., Lyzinski, V., Tang, M., Athreya, A., Cape, J., and Bridgeford, E. (2019). On a two-truths phenomenon in spectral graph clustering. *Proceedings of the National Academy of Sciences*, 116(13):5995–6000.

Qin, T. and Rohe, K. (2013). Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, pages 3120–3128.

Rajaraman, A. and Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Roberts, M. E., Stewart, B. M., and Airoldi, E. M. (2016). A model of text for experimentation in the social sciences. *Journal of the American Statistical Association*, 111(515):988–1003.

Rohe, K., Chatterjee, S., and Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915.

Rombach, P., Porter, M. A., Fowler, J. H., and Mucha, P. J. (2017). Core-periphery structure in networks (revisited). *SIAM Review*, 59(3):619–646.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. (2012). The author-topic model for authors and documents. *arXiv preprint arXiv:1207.4169*.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581–603.

Seidman, S. B. (1983). Network structure and minimum degree. *Social networks*, 5(3):269–287.

Shahnaz, F., Berry, M. W., Pauca, V. P., and Plemmons, R. J. (2006). Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386.

Strogatz, S. H. (2001). Exploring complex networks. *nature*, 410(6825):268.

Sussman, D. L., Tang, M., Fishkind, D. E., and Priebe, C. E. (2012). A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128.

Ugander, J., Backstrom, L., and Kleinberg, J. (2013). Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1307–1318.

Verma, T., Russmann, F., Araújo, N. A., Nagler, J., and Herrmann, H. J. (2016). Emergence of core–peripheries in networks. *Nature communications*, 7:10441.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584.

Wang, C. and Blei, D. M. (2013). Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 14(4).

Wang, C., Blei, D. M., and Heckerman, D. (2008). Continuous time dynamic topic models. In *UAI'08 Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*.

Wang, S., Rohe, K., et al. (2016). Discussion of "coauthorship and citation networks for statisticians". *The Annals of Applied Statistics*, 10(4):1820–1826.

Wang, Z., Liang, Y., and Ji, P. (2020). Spectral algorithms for community detection in directed networks. *Journal of Machine Learning Research*, 21:1–45.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442.

Zare, H., Hajiabadi, M., and Jalili, M. (2019). Detection of community structures in networks with nodal features based on generative probabilistic approach. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, X., Martin, T., and Newman, M. E. (2015). Identification of core-periphery structure in networks. *Physical Review E*, 91(3):032803.

Zhang, Y., Levina, E., and Zhu, J. (2017). Estimating network edge probabilities by neighbourhood smoothing. *Biometrika*, 104(4):771–783.

Zhao, Y., Levina, E., Zhu, J., et al. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292.

Zhu, Y., Yan, X., Getoor, L., and Moore, C. (2013). Scalable text and link analysis with mixed-topic link models. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 473–481.