Title
On Some Inference Problems for Networks

Author
Mukherjee, Soumendu Sundar

Publication Date
2018

Peer reviewed|Thesis/dissertation

On Some Inference Problems for Networks

by

Soumendu Sundar Mukherjee

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Peter J. Bickel, Chair
Professor Adityanand Guntuboyina
Professor Alan Hammond
Professor Luca Trevisan

Spring 2018

On Some Inference Problems for Networks

Abstract

On Some Inference Problems for Networks

by

Soumendu Sundar Mukherjee

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Peter J. Bickel, Chair

Networks are abstract representations of relationships between a set of entities. As such they can be used to represent data in a variety of complex interactive systems such as people and their social connections, researchers and their collaborations, proteins and their interactions, and so on. Vast amounts of such interaction data are being collected routinely in a range of disciplines and thus call for the attention of the statistician. Due to their large size (number of observations scales as the square of the number of nodes), traditional statistical methods are usually not scalable and one needs to come up with more computationally feasible inference techniques.

A concrete example of the issue is the problem of community detection in networks. Traditional likelihood based methods are computationally intractable, so researchers have come up with various computation-friendly alternatives. Although these methods work well on small to moderately large networks, most of them cannot handle truly large networks in a reasonable amount of time.

In this dissertation, we first advance divide and conquer strategies for community detection. We propose two algorithms which perform clustering on a number of small subgraphs and finally patch the results into a single clustering. The main advantage of these algorithms is that they bring down significantly the computational cost of traditional algorithms, including spectral clustering, semidefinite programs, modularity based methods, likelihood based methods, etc., without losing on accuracy and even improving accuracy at times. These algorithms are also, by nature, parallelizable. Thus, exploiting the facts that most traditional algorithms are accurate and the corresponding optimization problems are much simpler in small problems, our divide and conquer methods provide an omnibus recipe for scaling traditional algorithms up to large networks. We prove consistency of these algorithms under various subgraph selection procedures and perform extensive simulations and real data analysis to understand the advantages of the divide and conquer approach in various settings.

We then extend these divide and conquer methods to the more realistic situation of mixed memberships. Models that can be tackled are the mixed membership blockmodel, topic models, etc.

Next we focus on the problem of network comparison. We tackle two aspects of this problem: clustering and changepoint detection.

While being able to cluster within a network, in the sense of community detection, is important, there are emerging needs to be able to *cluster multiple networks*. This is largely motivated by the routine collection of network data that are generated from potentially different populations. These networks may or may not have node correspondence. For example, brain networks of a group of patients have node correspondence, whereas collaboration networks of researchers in different disciplines such as Computer Science, Mathematics or Statistics will have little node correspondence. When node correspondence is present, we cluster networks by summarizing a network by its graphon estimate, whereas when node correspondence is not present, we propose a novel solution for clustering such networks by associating a computationally feasible feature vector to each network based on traces of powers of the adjacency matrix. We illustrate our methods using both simulated and real data sets, and theoretical justifications are provided in terms of consistency.

In the changepoint problem, one observes a series of networks indexed by time and wishes the check if there is some significant change in the structure of these networks at some point of time. Potential applications are in, for instance, brain imaging, where one has brain scans of individuals collected over time and is looking for abnormalities, ecological networks observed over time, where one wonders if there is a structural change. We consider a CUSUM (short for cumulative sum) statistic for this problem, and prove its consistency. We find that in this high dimensional setting, the estimation error rate is better than the classical rate for fixed dimensional changepoint problems. As applications, we detect changepoints in the MIT reality mining data and the US senate roll call data.

To my parents, Chandana Mukherjee and Basudev Mukherjee.

To all my teachers.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I am very fortunate to have been advised by Professor Peter Bickel. The FRG seminar that Peter ran in 2014-2015, when I was a first year student, kindled my interest in networks. He welcomed me right away when I approached him for a project. Over the next three years, I have learnt so much from his deep insights into virtually every part of Statistics.

I am also very grateful to Professor Alan Hammond for his guidance. I have learnt a great deal about Probability from his classes and numerous conversations with him. Thanks to Alan and Peter for their careful reading of this dissertation and their many helpful comments and suggestions.

In addition to Alan and Peter, my heartfelt thanks goes to Professor Purnamrita Sarkar, who I have collaborated with in a number of projects. She kindly invited me to spend the summer of 2016 at UT Austin which was a great learning experience.

I have also had the privilege to work with Sören Künzel, Lizhen Lin, Satyaki Mukherjee, Rajarshi Mukherjee, Bodhisattva Sen, Bowei Yang and Rachel Wang in various projects for which I am grateful to them. I would like to thank my co-authors for permitting me to use material from some of our joint publications in this dissertation.

Berkeley is a dream place do one's PhD in. I have learnt a great deal about Statistics, Mathematics and Computer Science from classes, group meetings and seminars here. I am especially thankful to Professors Peter Bartlett, Adityanand Guntuboyina, Jim Pitman, Nikhil Srivastava and Rajarshi Mukherjee for numerous illuminating conversations. I am grateful to Professors Adityanand Guntuboyina, Alan Hammond and Luca Trevisan for agreeing to be on my qualifying examination committee, as well as my dissertation committee. I also thank Professor Bin Yu for chairing my qualifying examination committee. Many thanks are due to La Shana Porlaris and Mary Melinn for their kind and prompt help in numerous official matters.

Settling into life in a foreign place would have been so much difficult were it not for the warm friendship of Zsolt Bartha, Nick Bhattacharya, Vipul Gupta, Milind Hegde, Koulik Khamaru, Sören Künzel, Satyaki Mukherjee, Sujayam Saha, Sourav Sarkar and Jason Wu. I shall fondly remember the numerous badminton, squash and table tennis matches with Koulik, Satyaki, Vipul and Zsolt, and the conversations we had afterwards. Special thanks to Sujayam for being an awesome housemate, and to Satyaki who has, in his own way, taught me to be more skeptical about the non-mathematical sides of life. I would also like to thank my batchmates and seniors from ISI, especially Riddhipratim Basu, Sumanta Basu, Sharmodeep Bhattacharyya, Nirupam Chakrabarty, Moumita Chakraborty, Sutanoy Dasgupta, Paromita Dubey, Sourav Ghosh, Aritra Guha, Antony Joseph, Nilanjana Laha, Pragya Sur and Rajarshi Mukherjee for all the enjoyable times we spent together.

Ananya, my fiancé, kept me sane in a difficult period of my life. She would resent me if I thank her here, but it would have been so much harder if not for her love and support during the last eight years.

Finally, I thank my parents, Chandana Mukherjee and Basudev Mukherjee, for their unconditional love and support, and all their sacrifices throughout my life. It was a dream of my late father that I pursue higher education, and he did so many things to make that happen. This dissertation is dedicated to them and to all my teachers who have influenced and shaped me in many ways in the course of life.

Chapter 1

Introduction

1.1 A brief overview of some network models

Networks are abstract representations of relationships between a set of entities. Formally, a graph or network $G = (V, E)$ consists of a set of vertices (also called nodes, actors, entities) and a set E of edges (also called links, connections) between them. As a concrete example, V could be the set of all students in a class and an edge in E could then tell whether a student considers another student as a friend. So, G, in this case, is a network of friendships among students in a class. Other examples could be social networks, food-web (predator-prey) networks, co-authorship networks and so on.

Network data are being collected at an ever-growing volume in modern sciences. Due to their complex structure representing interactions among a number of entities, these datasets pose new challenges for a statistician. As the data dimension scales as the square of the number of entities, traditional statistical methods of inference are not scalable beyond small networks. So one has to come up with novel methods of inference which are scalable. A major focus of this dissertation thus will be computation — we want to develop methods that can be applied to large graphs in a reasonable amount of computing time.

Statisticians deal with data by postulating reasonable probabilistic models and then learning parameters of these models. Over the years people have come up with various probabilistic models for networks, depending on the focus of study. The simplest such model is the *Erdös-Rényi (ER) model* [29], where the edges are independent and identically distributed as a Bernoulli(p) variable. Although mathematically already quite interesting, this model is essentially a null model of randomness and fails to capture phenomena seen in real world networks, such as variable (especially heavy tailed) degree distribution, community structure, etc. There are many alternative models which address some of the issues. For instance, in the *configuration model*, graphs are drawn at random from a specified degree distribution; in the *preferential attachment model*, one can see the "rich-get-richer" phenomenon, seen, for example, in citation networks. For more details on these models and networks in general, see [57]. Anyway, we will not deal with these models in this dissertation. General

network models which are appropriate for the problems considered in this dissertation are latent space models (loosely called "graphons" sometimes), inhomogeneous random graphs, or kernel random graphs [18].

Among the problems considered in this dissertation, a recurring one will be the problem of community estimation or network clustering. In various real networks one observes a group of nodes tightly connected between themselves and sparsely connected to the rest of the nodes. Such densely connected groups are called *communities* and the goal is two extract communities from observed networks. The simplest statistical model for networks with communities is the stochastic blockmodel (SBM) [37]. The key idea in an SBM is to enforce stochastic equivalence, i.e. two nodes in the same latent community have identical probabilities of connection to all nodes in the network.

For an SBM generating a network with n nodes and K communities, one has a hidden/unknown community/cluster membership matrix $Z \in \{0,1\}^{n\times K}$, where $Z_{ik} = 1$ if node i is in community $k \in [K]$. Given these memberships, the link formation probabilities are given as

$$\mathbb{P}(A_{ij} = 1 \mid Z_{ik} = 1, Z_{jk'} = 1) = B_{kk'},$$

where B is a $K \times K$ symmetric parameter matrix of probabilities. The elements of B may decay to zero as n grows to infinity, to model sparse networks. Note that for $K = 1$, this reduces to the Erdös-Rényi model.

There are many extensions of SBM. The degree corrected blockmodel (DCBM) [40] allows one to model varied degrees in the same community, whereas a standard SBM does not. Mixed membership blockmodels (MMBM) [4] allow a node to belong to multiple communities, whereas in an SBM, a node can belong to exactly one cluster.

We will now do a brief literature review of some of the methods used for community detection and the computational challenges that arise, before we give an overview of the work in this dissertation.

Typically, the goal is to estimate the latent memberships consistently. A method outputting an estimate $\hat{Z}$ is called strongly consistent if $\mathbb{P}(\hat{Z} = Z\Pi) \to 1$ for some $K \times K$ permutation matrix Π, as $n \to \infty$. A weaker notion of consistency is when the fraction of misclustered nodes goes to zero as n goes to infinity. Typically, most of the consistency results are derived where average degree of the network grows faster than the logarithm of n. This is often called the semi-dense regime. When average degree is bounded, we are in the sparse regime. In the sparse regime, one cannot hope to reconstruct the community assignments consistently. Then best one can do is to construct an estimate which has a better-than-random correlation with the truth. Even this is not possible below the famous phase-transition threshold [54, 53] where no algorithm is better than a random guess.

On the algorithmic side, there are a handful of methods. These include likelihood based methods [8], modularity based methods [75, 61, 13], spectral methods [71], semidefinite programming (SDP) based approaches [21], etc. Among these spectral methods are scalable since the main bottleneck is computing top K eigenvectors of a large and often sparse matrix. While the theoretical guarantees of spectral methods are typically proven in the semi-dense

regime [51, 71, 43], a regularized version of it has been shown to perform better than a random predictor for sparse networks [42]. Profile likelihood methods [13] involve greedy search over all possible membership matrices, which makes them computationally expensive. Semidefinite programs are robust to outliers [21] and are shown to be strongly consistent in the dense regime [7] and yield a small but non-vanishing error in the sparse regime [34]. However, semidefinite programs are slow and, typically, only scale to thousands of nodes, not millions of nodes.

Methods like spectral clustering on geodesic distances [11] are provably consistent in the semi-dense case, and can give a small error in sparse cases. However, it requires computing all pairs of shortest paths between all nodes, which can pose serious problems for both computation and storage for very large graphs.

Monte Carlo methods [75, 62], which are popular tools in Bayesian frameworks, are typically not scalable. More scalable alternatives such as variational methods [32] do not have provable guarantees for consistency, and often suffer from bad local optima.

So far we have discussed community detection methods which only look at the network connections and not node attributes which are also often available and may possess useful information on the community structure (see, e.g., [60]). There are extensions of the methods mentioned earlier which accommodate node attributes, e.g., modularity based [85], spectral [14], SDP based [81], etc. These methods come with theoretical guarantees and have good performance in moderately sized networks. While existing Bayesian methods [50, 60, 80] are more amenable to incorporating covariates in the inference procedure, they are often computationally expensive and lack rigorous theoretical guarantees.

The same situations prevail for the generalizations DCBM and MMBM. Most of the above mentioned methods have direct counterparts for DCBM. Some notable algorithms for MMBM are variational inference [3], tensor decomposition methods [9], eigenvector-based methods [48] among others.

While the above mentioned array of algorithms are diverse and each has its unique aspects, in order to scale them to very large datasets, one has to apply different computational tools tailored to different algorithmic settings. While stochastic variational updates may be suitable to scale Bayesian methods, pseudo likelihood methods are better optimized using row sums of edges inside different node blocks. Thus it is an important problem to devise general techniques that aid computation.

1.2 Overview of the work in this dissertation

Chapters 2 and 3 of this dissertation are devoted to scalable community membership estimation. Chapters 4 and 5 are devoted to detecting patterns in a number of given networks. Below we give short summaries of the work presented in each chapter.

Scalable community detection

In Chapter 2, we advance divide and conquer strategies for solving the community detection problem. We propose two algorithms which perform clustering on a number of small subgraphs and finally patch the results into a single clustering. The main advantage of these algorithms is that they bring down significantly the computational cost of traditional algorithms, including spectral clustering, semidefinite programs, modularity based methods, likelihood based methods, etc., without losing on accuracy and even improving accuracy at times. These algorithms are also, by nature, parallelizable. Thus, exploiting the facts that most traditional algorithms are accurate and the corresponding optimization problems are much simpler in small problems, our divide and conquer methods provide an omnibus recipe for scaling traditional algorithms up to large networks. We prove consistency of these algorithms under various subgraph selection procedures and perform extensive simulations and real data analysis to understand the advantages of the divide and conquer approach in various settings. This work is available at the arXiv [55].

Scalable estimation in mixed membership models

In Chapter 3, we extend the methods of Chapter 2 to handle the more realistic scenario of mixed memberships. This in particular includes the mixed membership blockmodel (MMBM), and topic models (TM) among others. We derive conditions under which alignment of estimated mixed memberships are possible for different subsets of nodes.

Clustering network-valued objects

In Chapter 4, we consider clustering network valued objects. While being able to cluster within a network is important, there are emerging needs to be able to cluster multiple networks. This is largely motivated by the routine collection of network data that are generated from potentially different populations. These networks may or may not have node correspondence. For example, brain networks of a group of patients have node correspondence, whereas collaboration networks of researchers in different disciplines such as Computer Science, Mathematics or Statistics will have little node correspondence. When node correspondence is present, we cluster networks by summarizing a network by its graphon estimate, whereas when node correspondence is not present, we propose a novel solution for clustering such networks by associating a computationally feasible feature vector to each network based on trace of powers of the adjacency matrix. We illustrate our methods using both simulated and real datasets, and theoretical justifications are provided in terms of consistency. This work has appeared in NIPS (2017) [56].

Changepoint detection

In Chapter 5, we consider a temporal analog of the problem considered in Chapter 3, namely changepoint detection, given a network valued time series. Changepoint detection is a classical problem in statistics, going all the way back to the early days of statistical quality control [65, 66, 31]. There is a huge literature on the univariate changepoint problem, where one tries to detect changes in a scalar stochastic process. An excellent treatment can be found in the book [20].

The network version of the problem is also of practical interest. Potential applications are in, for instance, brain imaging, where one has brain scans of individuals collected over time and is looking for abnormalities, ecological networks observed over time, where one wonders if there is a structural change, and so on. There is a significant literature on the related problem of anomaly detection in graphs. See, e.g., [70] for a survey.

Although some empirical work has been done [68, 67], not much theory can be found, and most theoretical results focus on particular structures or specialized models. The classical CUSUM (short for cumulative sum) statistic [66] for univariate changepoint problems can be used in the network problem as well, and provides a unified way of constructing statistics. It is also amenable to theoretical analysis because of the averaging structure present. We consider the offline case where the entire series is observed beforehand and analyze the CUSUM statistic in this problem, proving the consistency of the resulting estimator, with asymptotics involving the number of networks in the series, the size of the networks, and their sparsity. In particular, we show that the classical rate of estimation for fixed dimensional changepoint problems can be surpassed in the high-dimensional setting of networks. We analyze the MIT reality mining data and the US senate roll call data as applications.

Chapter 2

Divide and conquer for community detection

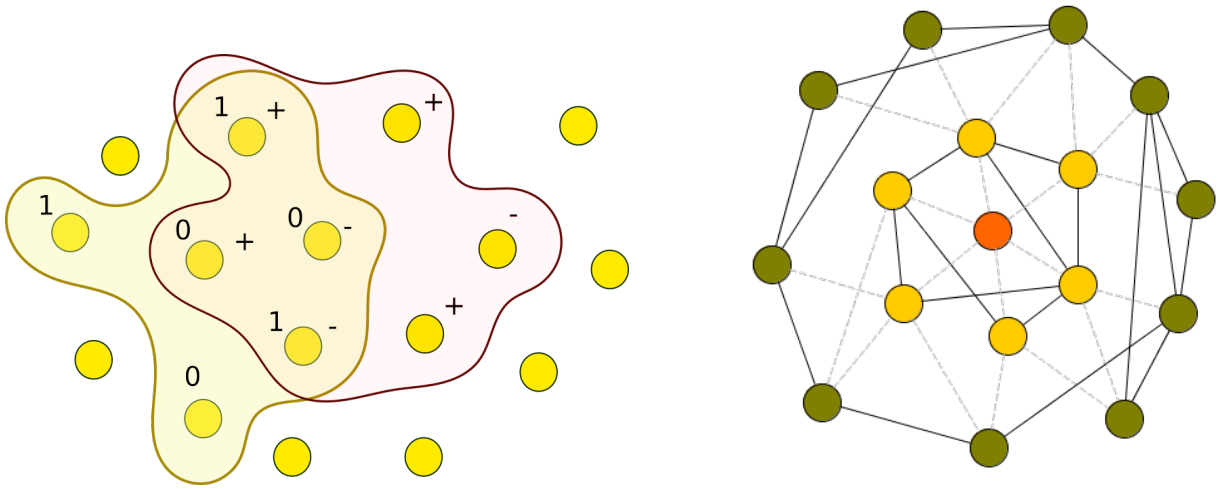
2.1 Introduction

As mentioned in Chapter 1, most existing algorithms for community detection can be used only on small to moderately large networks. In this chapter, we propose a divide and conquer approach to community detection. The idea is to apply a community detection method on small subgraphs of a large graph, and somehow stitch the results together. If we could achieve this, we would be able to scale up any community detection method (which may involve covariates as well as the network structure) that is computationally feasible on small graphs, but is difficult to execute on very large networks. This would be especially useful for computationally expensive community detection methods (such as SDPs, modularity based methods, Bayesian methods, etc.). Another possible advantage concerns variational likelihood methods (such as mean field) with a large number (depending on n) of local parameters, which typically have an optimization landscape riddled with local minima. For smaller graphs there are less parameters to fit, and the optimization problem often becomes easier.

Clearly, the principal difficulty in doing this is matching the possibly conflicting label assignments from different subgraphs (see Figure 2.1(a) for an example). This immediately rules out a simple-minded averaging of estimates $\hat{Z}_S$ of cluster membership matrices Z_S, for various subgraphs S, as a viable stitching method.

In this regard, we propose two different stitching algorithms. The first is called Piecewise Averaged Community Estimation (PACE); in which we focus on estimating the clustering matrix $C = ZZ^\top$, which is labeling invariant, since the (i, j)-th element of this matrix being one simply means that nodes i and j belong to the same cluster, whereas the value zero means i and j belongs to two different clusters. Thus we first compute estimates of $Z_S Z_S^\top$ for various subgraphs S and then average over these matrices to obtain an estimate $\hat{C}$ of C. Finally we apply some computationally cheap clustering algorithm like approximate

Figure 2.1: (a) Conflicting label assignments on two subgraphs, on one labels are denoted by 0/1, on the other by $+/-$, (b) onion neighborhood: the yellow vertices and the solid edges between them constitute the ego network of the root vertex colored red. The green and yellow vertices together with the solid edges between them constitute the 2-hop onion neighborhood of the root (see the paragraph preceding Corollary 2.3.2 for details).



(a) (b)

K-means, DGCluster[†], spectral clustering, etc. on $\hat{C}$ to recover an estimate of Z.

We also propose another algorithm called Global Alignment of Local Estimates (GALE), where we first take a sequence of subgraphs, such that any two consecutive subgraphs on this sequence have a large intersection, and then traverse through this sequence, aligning the clustering based on a subgraph with an averaged clustering of the union of all its predecessor subgraphs in the sequence, which have already been aligned. The alignment is done via an algorithm called Match which identifies the right permutation to align two clusterings on two subgraphs by computing the confusion matrix of these two clusterings restricted to the intersection of the two subgraphs. Whereas a naive approach would entail searching through all K permutations, Match finds the right permutation in $K \log K$ time. Once this alignment step is complete, we get an averaged clustering of the union of all subgraphs (which covers all the vertices). By design GALE works with estimates of cluster membership matrices Z_S directly to output an estimate of Z, and thus, unlike PACE, avoids the extra overhead of recovering such an estimate from $\hat{C}$.

The rest of the chapter is organized as follows. In Section 2.2 we describe our algorithms. In Section 2.3 we state our main results and some applications. Section 2.4 contains simulations and real data experiments. In Section 2.5 we provide proofs of our main results, while relegating some of the details to Appendix A. Finally, in Section 2.6 we conclude with a summarizing discussion.

[†]This is a greedy algorithm detailed in Section A.3.

2.2 Two divide and conquer algorithms

As we discussed in the introduction, the main issue with divide and conquer algorithms for clustering is that one has to somehow match up various potentially conflicting label assignments. We propose two algorithms to accomplish this task. Both algorithms first compute the clustering on small patches of a network; these patches can be the induced subgraph of a random subsample of nodes, or neighborhoods. However, the stitching procedures are different.

Algorithm 1 PACE: Piecewise Averaged Community Estimation

1: **Subgraph selection:** Fix a positive integer threshold $m_\star$ for minimum required subgraph size. Fix another positive integer T, that will be the number of subgraphs we will sample. Given A, choose T subsets $S_1, \ldots, S_T$ of the nodes by some procedure, e.g., select $m \geq m_\star$ nodes at random, or pick h-hop neighborhoods of vertices in G, or ego-neighborhood of vertices. By A_{S_ℓ} denote the adjacency matrix of the network induced by S_ℓ.

2: **Clustering on subgraphs:** Perform any of the standard clustering algorithms like profile likelihood (PL), mean field likelihood (MFL), spectral clustering (SC), semidefinite programming (SDP), etc. on each of these T subgraphs which have size at least $m_\star$ to obtain estimated clustering matrices $\hat{C}^{(S_\ell)} = \hat{C}^{(\ell)}$. For the rest of the subgraphs, set $\hat{C}^{(\ell)} \equiv 0$. Extend $\hat{C}^{(\ell)}$ to an $n \times n$ matrix by setting $\hat{C}^{(\ell)}_{ij} = 0$ if at least one of i, j was not selected in S_ℓ. Denote the resulting matrix again by $\hat{C}^{(\ell)}$.

3: **Patching up:** Let $y^{(\ell)}_{ij}$ denote the indicator of the event that both i, j were selected in S_ℓ. Set $N_{ij} = \sum_{\ell=1}^{T} y^{(\ell)}_{ij}$. Define the combined estimator $\hat{C} = \hat{C}_\tau$ by

$$\hat{C}_{ij} = \hat{C}_{\tau,ij} = \frac{\mathbf{1}_{\{N_{ij} \geq \tau\}} \sum_{\ell=1}^{T} y^{(\ell)}_{ij} \hat{C}^{(\ell)}_{ij}}{N_{ij}} = \frac{\mathbf{1}_{\{N_{ij} \geq \tau\}} \sum_{\ell=1}^{T} \hat{C}^{(\ell)}_{ij}}{N_{ij}}. \tag{2.1}$$

Here $1 \leq \tau \leq T$ is an integer tuning parameter. We will call $\hat{C}_\tau$ as Piecewise Averaged Community Estimator (also abbreviated as PACE).

PACE: an averaging algorithm

Suppose A is the adjacency matrix of a network with true cluster membership of its nodes being given by the $n \times K$ matrix Z where there are K clusters. Set $C = ZZ^\top$ to be the clustering matrix whose (i, j)-th entry is the indicator of whether nodes i, j belong to the same cluster. Given A we will perform a local clustering algorithm to obtain an estimate of C, from which an estimate $\hat{Z}$ of the cluster memberships may be reconstructed.

The τ parameter in PACE reduces variance in estimation quality as it discards information from less credible sources — if a pair of nodes has appeared in only a few subgraphs, we do

not trust what the patching has to say about them. Setting $\tau = \theta\binom{n}{2}^{-1}\sum_{i<j} N_{ij}$, for some $0 < \theta < 1$, or some quantile (say 0.4-th) of the N_{ij}'s seems to work well in practice (this is also somewhat justified by our theory).

A slight variant of Algorithm 1 is where we allow subgraph and/or node-pair specific weights $w_{\ell,i,j}$ in the computation of the final estimate, i.e.

$$\hat{C}_{ij} = \hat{C}_{\tau,ij} = \frac{\mathbf{1}_{\{N_{ij}\geq\tau\}}\sum_{\ell=1}^{T} w_{\ell,i,j} y_{ij}^{(\ell)} \hat{C}_{ij}^{(\ell)}}{N_{ij}} = \frac{\mathbf{1}_{\{N_{ij}\geq\tau\}}\sum_{\ell=1}^{T} w_{\ell,i,j} \hat{C}_{ij}^{(\ell)}}{N_{ij}}, \tag{2.2}$$

where N_{ij} now equals $\sum_{\ell=1}^{T} w_{\ell,i,j} y_{ij}^{(\ell)}$. We may call this estimator w-PACE standing for weighted-PACE. If the weights are all equal, w-PACE becomes equivalent to ordinary PACE. There are natural nontrivial choices, including

(i) $w_{\ell,i,j} = |S_\ell|$, which will place more weight to estimates based on large subgraphs,

(ii) $w_{\ell,i,j} = \deg_{S_\ell}(i) + \deg_{S_\ell}(j)$, where $\deg_S(u)$ denotes the degree of node u in subgraph S (this will put more weight on pairs which have high degree in S_ℓ).

The first prescription above is intimately related to the following sampling scheme for ordinary PACE: pick subgraphs with probability proportional to their sizes. For instance, in Section 2.4 we analyze the political blog data of [1] where neighborhood subgraphs are chosen by selecting their roots with probability proportional to degree.

In real world applications, it might make more sense to choose these weights based on domain knowledge (for instance, it may be that certain subnetworks are known to be important). Another (minor) advantage of having weights is that when $T = 1$ and $|S_1| = n$, we have $N_{ij} = w_{1,i,j}$ and so if $w_{1,i,j} \geq \tau$, then

$$\hat{C}_{ij} = \hat{C}_{ij}^{(1)},$$

i.e. w-PACE becomes the estimator based on the full graph. This is, for example, true with $w_{\ell,i,j} = |S_\ell|$, because τ is typically much smaller than n. However, ordinary PACE lacks this property unless $\tau = 1$, in fact, with $\tau > 1$, the estimate returned by PACE is identically 0. Anyway, in what follows, we will stick with ordinary PACE because of its simplicity.

Before we discuss how to reconstruct an estimate $\hat{Z}$ of Z from $\hat{C}$, let us note that we may obtain a binary matrix $\hat{C}_\eta$ by thresholding $\hat{C}$ at some level η (for example, $\eta = 1/2$):

$$\hat{C}_\eta := [\hat{C} > \eta].$$

This thresholding does not change consistency properties (see Lemma A.4.1). Looking at a plot of this matrix gives a good visual representation of the community structure. In what follows, we work with unthresholded $\hat{C}$.

Reconstruction of $\hat{Z}$

How do we actually reconstruct $\hat{Z}$ from $\hat{C}$? The key is to note that members of the same community have identical rows in C and that, thanks to PACE, we have gotten hold of a consistent estimate of C. Thus we may use any clustering algorithm on the rows of $\hat{C}$ to recover the community memberships. Another option would be to run spectral clustering on the matrix $\hat{C}$ itself. However, as the rows of $\hat{C}$ are n-vectors, most clustering algorithms will typically have a running time of $O(n^3)$[‡] at best. Indeed, the main computational bottleneck of any distance based clustering algorithm in a high dimensional situation like the present one is computing $d_{ij} = \|\hat{C}_{i\star} - \hat{C}_{j\star}\|$ which takes $O(n)$ bit operations. However, since we have gotten a good estimate of C, we can project the rows of $\hat{C}$ onto lower dimensions, without distorting the distances too much. The famous Johnson-Lindenstrauss Lemma for random projections says that by projecting onto $\Omega(\log n/\epsilon^2)$ dimensions, one can keep, with probability at least $1 - O(1/n)$, the distances between projected vectors within a factor of $(1 \pm \epsilon)$ of the true distances. Choosing ϵ as inverse polylog(n) we need to project onto polylog(n) dimensions and this would then readily bring the computational cost of any distance based algorithm down from $O(n^3)$ to $O(n^2\text{polylog}(n))$.

Following the discussion of the above paragraph, we first do a random projection of the rows of $\hat{C}$ onto $s\,(=\text{polylog}(n))$ dimensions and then apply a (distance based) clustering algorithm.

Algorithm 2 Recovering $\hat{Z}$ from $\hat{C}$: random projection followed by distance based clustering

1: Select a dimension s for random projection. Let $\mathsf{Cluster}(\cdot, K)$ be a clustering algorithm that operates on the rows of its first argument and outputs K clusters.
2: $\hat{C}_{proj} \leftarrow \hat{C}R/\sqrt{s}$, where R is a standard Gaussian matrix of dimensions $n \times s$.
3: $\hat{Z} \leftarrow \mathsf{Cluster}(\hat{C}_{proj}, K)$.

As for $\mathsf{Cluster}(\cdot, K)$, we may use approximate K-means or any other distance based clustering algorithm, e.g., $\mathsf{DGCluster}(\hat{C}, \hat{C}_{proj}, K)$, a greedy algorithm presented in Section A.3 as Algorithm 10.

GALE: a sequential algorithm

First we introduce a simple algorithm for computing the best permutation to align labels of one clustering (Z_1) to another (Z_2) of the same set of nodes (with fixed ordering) in a set S. The idea is to first compute the confusion matrix between two clusterings. Note that if the two labelings each have low error with respect to some unknown true labeling, then

[‡]In addition to the standard 'O', 'Ω', 'Θ', 'o' notations, and their probabilistic counterparts, we will use the following less standard alternatives: (i) $A \asymp B$ to mean $A = \Theta(B)$, and (ii) $A \gg B$ to mean $B = o(A)$. We will also sometimes use a 'tilde' over the standard notations to hide polylogarithmic factors.

the confusion matrix will be close to a diagonal matrix up to permutations. The following algorithm below essentially finds a best permutation to align one clustering to another.

Algorithm 3 Match: An algorithm for aligning two clusterings of the same set S of nodes. Input $Z_1, Z_2 \in \{0,1\}^{|S|\times K}$.

1: Compute $K \times K$ confusion matrix $M = Z_1^\top Z_2$. Set $\Pi \leftarrow 0_{K\times K}$.
2: **while** there are no rows/columns left **do**
   (a) Find i, j, such that $M_{ij} = \max_{i',j'} M_{i'j'}$ (a tie can be broken arbitrarily).
   (b) Set $\Pi_{ij} = 1$.
   (c) Replace the i-th row and j-th columns in M with -1.
3: Return the permutation matrix Π.

Remark 2.2.1. *One can also compute the optimal permutation by searching through all $K!$ permutations of the labels and picking the one which gives smallest mismatch between the two; but* Match *brings the dependence on K down from exponential to quadratic. Note that if one of the clusterings are poor, then* Match *may not retrieve the optimal permutation. However, our goal is to cluster many subgraphs using an algorithm which has good accuracy with high probability (but may be computationally intensive, e.g., profile likelihood or semidefinite programming) and then use the intersections between the subgraphs to align one to another. As we shall show later, as long as there are enough members from each community in S, the simple algorithm sketched above suffices to find an optimal permutation.*

Now we present our sequential algorithm which aligns labelings across different subgraphs. The idea is to first fix the indexing of the nodes; cluster the subgraphs (possibly with a parallel implementation) using some algorithm, and then align the clusterings along a sequence of subgraphs. To make things precise, we make the following definition.

Definition 2.2.1. *Let $\mathcal{S}_{m,T} = (V, E)$ (vertex set V and edge set E) denote a "super-graph of subgraphs" where each subgraph is a node, and two nodes are connected if the corresponding subgraphs have a substantial overlap. For random m-subgraphs, this threshold is $m_1 := \left\lceil \frac{m^2}{2n} \right\rceil$.*

Define a traversal through a spanning tree $\mathcal{T}$ of $\mathcal{S}_{m,T}$ as a sequence $x_1, \ldots, x_J$, $x_j \in [T]$, $T \le J \le 2T$, covering all the vertices, such that along the sequence x_i and x_{i+1} are adjacent, for $1 \le i \le J-1$ (i.e. a traversal is just a walk on $\mathcal{T}$ of length at most $2T-1$ passing through each vertex at least once).

After we construct a traversal, we travel through this traversal such that at any step, we align the current subgraph's labels using the Match algorithm (Algorithm 3) on its intersection with the union of the previously aligned subgraphs. At the end, all subgraph labellings are aligned to the labeling of the starting subgraph. Now we can simply take an average or a majority vote between these.

Algorithm 4 GALE: Global Alignment of Local Estimates. Input: adjacency matrix A; parameters $m, T, \tau_i, i \in [T]$; a base algorithm $\mathcal{A}$ (e.g., PL, MFL, SC, SDP, etc.)

1: **Subgraph selection:** Given A, choose T subsets $S_1, \ldots, S_T$ of the nodes by some procedure, e.g., select them at random, or select T random nodes and then pick their h-hop neighborhoods. By A_{S_ℓ} denote the adjacency matrix of the network induced by S_ℓ.

2: **Clustering on subgraphs:** Perform algorithm $\mathcal{A}$ on each of these T subgraphs to obtain estimated cluster membership matrices $\hat{Z}_\ell = \hat{Z}_{S_\ell}$. Extend $\hat{Z}_\ell$ to a $n \times K$ matrix by setting $(\hat{Z}_\ell)_{jk} = 0$ for all $k \in [K]$ if $j \notin S_\ell$.

3: **Traversal of subgraphs:** Construct a traversal $S_{x_1}, \ldots, S_{x_J}$ through the T subgraphs.

4: **Initial estimate of Z:** $\hat{Z} \leftarrow \hat{Z}_{x_1}$. Also set $\hat{Z}^{(x_1)} := \hat{Z}_{x_1}$.

5: **Sequential label aligning:** For subgraph S_{x_i} on the traversal ($i = 1, \ldots, J$), if x_i has not been visited yet,

(a) Compute the overlap between the current subgraph with all subgraphs previously visited, i.e. let $S = S_{x_i} \cap \cup_{\ell=1}^{i-1} S_{x_\ell}$.

(b) Compute the best permutation to match the clustering $\hat{Z}_{x_i}, \hat{Z}$ on this set S, i.e. compute $\hat{\Pi}_i = \mathsf{Match}(\hat{Z}_{x_i}\big|_S, \hat{Z}\big|_S)$.

(c) Permute the labels of the nodes of S_{x_i} to get an aligned cluster membership matrix $\hat{Z}^{(x_i)} \leftarrow \hat{Z}_{x_i}\hat{\Pi}$.

(d) Update $\hat{Z}_{jk} \leftarrow \frac{\sum_{\ell \in \{x_1,\ldots,x_i\}} \hat{Z}_{jk}^{(x_\ell)} \mathbf{1}_{\{j \in S_{x_\ell}\}}}{\sum_{\ell \in \{x_1,\ldots,x_i\}} \mathbf{1}_{\{j \in S_{x_\ell}\}}} \mathbf{1}_{\{\sum_{\ell \in \{x_1,\ldots,x_i\}} \mathbf{1}_{\{j \in S_{x_\ell}\}} > \tau_i\}}$, for some threshold τ_i.

(e) Mark S_{x_i} as visited.

Implementation details

Constructing a traversal of the subgraphs can be done using a depth first search of the supergraph $\mathcal{S}_{m,T}$ of subgraphs. For our implementation, we start with a large enough subgraph (the parent), pick another subgraph that has a large overlap with it (the child), align it and note that this subgraph has been visited. Now recursively find another unvisited child of the current subgraph, and so on. It is possible that a particular path did not cover all vertices, and hence it is ideal to estimate clusterings with multiple traversals with different starting subgraphs and then align all these clusterings, and take an average. This is what we do for real networks. We also note that at any step, if we find a poorly clustered subgraph, then this can give a bad permutation which may deteriorate the quality of aligning the subsequent subgraphs on the path. In order to avoid this we use a self validation routine. Let S be intersection of current subgraph with union of the previously visited subgraphs. After aligning the current subgraph's clustering, we compute the classification accuracy of the current labeling of S with the previous labeling of S. If this accuracy is large enough, we

use this subgraph, and if not we move to the next subgraph on the path. For implementation, we use a threshold of 0.55.

Computational time and storage

The main computational bottleneck in GALE is in building a traversal through the random subgraphs. Let $\eta_{m,T}$ be the time for computing the clusterings for T subgraphs in parallel. A naive implementation would require computing intersections between all $\binom{T}{2}$ pairs of m-subsets. As we will show in our theoretical analysis, we take $m = \omega(\sqrt{n/\pi_{\min}})$, where $\pi_{\min} := \min_k \pi_k$ (here $\pi_k := n_k/n$, where n_k is the size of the k-th cluster) and $T = \tilde{\Omega}(n/m)$. Taking $\pi_{\min} = \Theta(1)$, the computation of intersections takes $O(T^2 m) = \tilde{O}(n^{3/2})$ time. Further, a naive comparison for computing subsets similar or close to a given one would require $T \log T$ time for each subset leading to $T^2 \log T = \tilde{O}(n)$ computation. However, for building a traversal one only needs access to subsets with large overlap with a given subset, which is a classic example of nearest neighbor search in Computer Science.

One such method is the widely used and theoretically analyzed technique of Locality Sensitive Hashing (LSH). A hash function maps any data object of an arbitrary size to another object of a fixed size. In our case, we map the characteristic vector of a subset to a number. The idea of LSH is to compute hash functions for two subsets A and B such that the two functions are the same with high probability if A and B are "close". In fact, the amount of overlap normalized by m is simply the cosine similarity between the characteristic vectors $\chi(A)$ and $\chi(B)$ of the two subsets, for which efficient hashing schemes $h : \{0,1\}^n \to \{0,1\}$ using random projections exist [24], with

$$\mathbb{P}(h(A) = h(B)) = 1 - \arccos(\chi(A)^T \chi(B)/m)/\pi.$$

For LSH schemes, one needs to build $L := T^\rho$ hash tables, for some $\rho < 1$, that governs the approximation quality. In each hash table, a "bucket" corresponding to an index stores all subsets which have been hashed to this index. For any query point, one evaluates $O(L)$ hash functions and examines $O(L)$ subsets hashed to those buckets in the respective hash tables. Now for these subsets, the distance is computed exactly. The preprocessing time is $O(T^{1+\rho}) = \tilde{O}(n^{\frac{1+\rho}{2}})$, with storage being $O(T^{1+\rho} + Tm) = \tilde{O}(n)$, and total query time being $O(T^{1+\rho} m) = \tilde{O}(n^{1+\rho/2})$. This brings down the running time added to the algorithm specific $\eta_{m,T}$ from sub-quadratic time $\tilde{O}(n^{3/2})$ to nearly linear time, i.e. $\tilde{O}(n^{1+\rho/2})$.

Thus, for other nearly linear time clustering algorithms, GALE may not lead to computational savings. However, for algorithms like profile likelihood or SDP which are well known to be computationally intensive, GALE can lead to a significant computational saving without requiring a lot of storage.

Remarks on sampling schemes

With PACE we have mainly used random m-subgraphs, h-hop neighborhoods, and onion neighborhoods, but many other subgraph sampling schemes are possible. For instance,

choosing roots of hop neighborhoods with probability proportional to degree, or sampling roots from high degree nodes (we have done this in our analysis of the political blog data, in Section 2.4). As discussed earlier, this weighted sampling scheme is related to w-PACE. A natural question regarding h-hop neighborhoods is how many hops to use. While we do not have a theory for this yet, because of "small world phenomenon" we expect not to need many hops; typically, in moderately sparse networks, 2-3 hops should be enough. However, an adaptive procedure (e.g., cross-validation type) for choosing h would be welcome. Also, since neighborhood size increases exponentially with hop size, an alternative to choosing full hop-neighborhoods is to choose a smaller hop-neighborhood and then add some (but not all) randomly chosen neighbors of the already chosen vertices. Other possibilities include sampling a certain proportion of edges at random, and then consider the subgraph induced by the participating nodes [45].

We have analyzed GALE under the random sampling scheme. For any other scheme, one will have to understand the behavior of the intersection of two samples or neighborhoods. For example, if one takes h-hop neighborhoods, for sparse graphs, each neighborhood predominantly has nodes from mainly one cluster. Hence GALE often suffers with this scheme. We show this empirically in Section 2.4, where GALE's accuracy is much improved under a random m-subgraph sampling scheme.

2.3 Main results

In this section we will state and discuss our main results on PACE and GALE, along with a few applications.

Results on PACE

Let σ and σ' be two clusterings (of n objects into K clusters), usually their discrepancy is measured by

$$\delta_c(\sigma, \sigma') = \inf_{\xi \in S_K} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{\sigma(i) \neq \xi(\sigma'(i))\}} = \frac{1}{n} \inf_{\xi \in S_K} \|\xi(\sigma) - \sigma'\|_0,$$

where S_K is the permutation group on $[K]$. If Z, Z' are the corresponding $n \times K$ binary matrices, then a related measure of discrepancy between the two clusterings is $\delta(Z, Z') = \inf_{Q \text{ perm.}} \frac{1}{n}\|ZQ - Z'\|_0$*. It is easy to see that $\delta(Z, Z') = 2\delta_c(\sigma, \sigma')$. (To elaborate, let Q_ξ be the permutation matrix corresponding to the permutation ξ, i.e. $Q_{ij} = \mathbf{1}_{\{\xi(i)=j\}}$. Then $\xi(\sigma(i) \neq \sigma'(i)$, if and only if $(ZQ_\xi)_{i\star} \neq Z'_{i\star}$, i.e. $\|(ZQ_\xi)_{i\star} - Z'_{i\star}\|_0 = 2$.) For our purposes, however, a more useful measure of discrepancy would be the normalized Frobenius squared distance between the corresponding clustering matrices $C = ZZ^\top$ and $C' = Z'Z'^\top$, i.e.

$$\tilde{\delta}(C, C') = \frac{1}{n^2}\|C - C'\|_F^2.$$

*In this chapter and the next one, $\|M\|_p$ will denote the ℓ_p norm of the vectorized version of the matrix M, whereas $\|M\|_F$ and $\|M\|$ will denote the Frobenius and ℓ_2 operator norms respectively.

Now we compare these two notions of discrepancies.

Proposition 2.3.1. *We have* $\tilde{\delta}(C, C') \le 4\delta(Z, Z') = 8\delta_c(\sigma, \sigma')$.

Incidentally, if the cluster sizes are equal, then one can show that

$$\tilde{\delta}(C, C') \le \frac{4}{K}\delta(Z, Z') = \frac{8}{K}\delta_c(\sigma, \sigma').$$

Although we do not have a lower bound on $\tilde{\delta}(C, C')$ in terms of $\delta(Z, Z')$, Lemma A.1 of [77] gives us (with $X = Z$, $Y = Z'$) that there exists an orthogonal matrix $\mathcal{O}$ such that

$$\|Z\mathcal{O} - Z'\|_F \le \frac{\|C - C'\|(\sqrt{K\|C\|} + \sqrt{K\|C'\|})}{\lambda_{\min}(C)} \le \frac{2\sqrt{Kn}\|C - C'\|}{n_{\min}(Z)},$$

where we used the fact that $\|C\| = \lambda_{\max}(C) = n_{\max}(Z) \le n$. The caveat here is that the matrix $\mathcal{O}$ need not be a permutation matrix.

To prove consistency of PACE we have to assume that the clustering algorithm $\mathcal{A}$ we use has some consistency properties. For example, it will suffice to assume that for a randomly chosen subgraph S (under our subgraph selection procedure), $\mathbb{E}\delta(\hat{Z}_S, Z_S)$[†] is small. The following is our main result on the expected misclustering rate of PACE.

Theorem 2.3.1 (Expected misclustering rate of PACE)**.** *Let S be a randomly chosen subgraph according to our sampling scheme. Let $\pi_{\max} = \max_k \pi_k$. We have*

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \le \frac{T}{\tau n^2} \times \mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2 + \pi_{\max} \times \max_{i,j} \mathbb{P}(N_{ij} < \tau), \tag{2.3}$$

where

$$\mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2 \le n^2\mathbb{P}(|S| < m_\star) + 4\mathbb{E}|S|^2\delta(\hat{Z}_S, Z_S)\mathbf{1}_{(|S| \ge m_\star)}.$$

The first term in (2.3) essentially measures the performance of the clustering algorithm we use on a randomly chosen subgraph. The second term measures how well we have covered the full graph by the chosen subgraphs, and only depends on the subgraph selection procedure. The effect of the algorithm we use is felt through the first term only.

We can now specialize Theorem 2.3.1 to various subgraph selection schemes. First, we consider randomly chosen m-subgraphs, which is an easy corollary.

Corollary 2.3.1 (Subgraphs are induced by $m \ge m_\star$ randomly chosen nodes)**.** *Let $p = \frac{m(m-1)}{n(n-1)}$, $0 < \theta < 1$ and $\tau = \theta T p$. We have*

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \le \frac{m}{\theta(m-1)} \times \mathbb{E}\tilde{\delta}(\hat{C}^{(S)}, C^{(S)}) + \pi_{\max} \times e^{-(1-\theta)^2 Tp/2}. \tag{2.4}$$

[†]The expectation is taken over both the randomness in the graph and the randomness of the sampling mechanism.

Notice that the constant $\frac{m}{\theta(m-1)}$ in (2.4) can be made as close to 1 as one desires, which means that the above bound is essentially optimal.

Full h-hop neighborhood subgraphs are much harder to analyze and will not be pursued here. However, ego networks, which are 1-hop neighborhoods minus the root node (see Figure 2.1(b)), are easy to deal with. One can also extend our analysis to h-hop onion neighborhoods which are recursively defined as follows: $\mathcal{O}_1(v) = S_1(v)$ is just the ego network of vertex v; in general, the h-th shell $S_h(v) := \uplus_{u \in S_{h-1}(v)}[\mathcal{O}_1(u) \setminus \mathcal{O}_{h-1}(v) \cup \{v\}]$, and $\mathcal{O}_h(v) = \mathcal{O}_{h-1}(v) \uplus S_h(v)$, where the operation $\uplus$ denotes superposition of networks. Here, for ease of exposition, we choose to work with ego networks (1-hop onion neighborhoods).

Corollary 2.3.2 (Ego networks under a stochastic blockmodel)**.** *Let* $B_\# = \max B_{ab}, B_\star = \min B_{ab}$ *and* $\tau = \frac{TB_\star^2}{4}$ *and* $m_\star \leq (n-1)B_\star/2$. *Let* $\theta > 1$. *Then*[†]

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \leq \frac{16(1+\theta)^2 B_\#^2}{B_\star^2} \times \mathbb{E}\delta(\hat{Z}_S, Z_S)\mathbf{1}_{(|S| \geq m_\star)} + \Delta, \tag{2.5}$$

where

$$\Delta \leq \frac{4}{B_\star^2} \times \exp\left(-\frac{nB_\star^2}{16B_\#}\right) + \frac{4}{n^2 B_\star^2} \times \exp\left(-\frac{\theta^2 nB_\#}{6}\right) + \pi_{\max} \times \left[2\exp\left(-\frac{nB_\star^4}{16B_\#^2}\right) + \exp\left(-\frac{TB_\star^2}{16}\right)\right].$$

Results on GALE

We denote the unnormalized miscustering error between estimated labels $\hat{Z}$ and the true labels Z, ($\hat{Z}, Z \in \{0,1\}^{n \times K}$) of the same set of nodes as $\mathcal{M}(Z, \hat{Z}) := n\delta(Z, Z') = \min_{Q \text{ perm.}} \|\hat{Z} - ZQ\|_0$. Note that since $\hat{Z}, Z$ are binary, the $\|\hat{Z} - ZQ\|_0 = \|\hat{Z} - ZQ\|_1 = \|\hat{Z} - ZQ\|_F^2$. As noted earlier, the number of misclustered nodes will be half of this.

The main idea behind GALE is simple. Every approximately accurate clustering of a set of nodes is only accurate up to a permutation, which can never be recovered without the true labels. However, we can align a labeling to a permuted version of the truth, where the permutation is estimated from another labeling of the same set of vertices. This is done by calculating the confusion matrix between two clusterings. We call two clusterings aligned if cluster i from one clustering has a large overlap with cluster i from the other clustering. If the labels are "aligned" and the clusterings agree, this confusion matrix will be a matrix with large diagonal entries. This idea is used in the Match algorithm, where we estimate the permutation matrix to align one clustering to another.

Now we present our main result. We prove consistency of a slightly modified and weaker version of Algorithm 4. In Algorithm 4, at every step of a traversal, we apply the Match

[†]Actually, we can allow $\tau = \theta' TB_\star^2$, for any $0 < \theta' < 1$.

algorithm on the intersection of the current subgraph and the union of all subgraphs previously aligned to estimate the permutation of the yet unaligned current subgraph. However, in the theorem presented below we use the intersection between the unaligned current subgraph with the last aligned subgraph. Empirically, it is better to use the scheme presented in Algorithm 4 since it increases the size of the intersection which requires weaker conditions on the clustering accuracy of any individual subgraph.

We now formally define our estimator $\hat{Z}^{\mathsf{GALE}}$. Let $y_i^{(\ell)} = \mathbf{1}_{\{i \in S_\ell\}}$. Let $\hat{Z}^{(\ell)}$ denote the aligned clustering of subgraph S_ℓ and let $N_i = \sum_{\ell=1}^{T} y_i^{(\ell)}$. Define

$$\hat{Z}_{ik}^{\mathsf{GALE}} := \frac{\sum_{\ell=1}^{T} y_i^{(\ell)} \hat{Z}_{ik}^{(\ell)}}{N_i} \mathbf{1}_{\{N_i \geq \tau\}}. \tag{2.6}$$

The entries of $\hat{Z}^{\mathsf{GALE}}$ will be fractions, but as we show in Lemma A.4.2, rounding it to a binary matrix will not change consistency properties.

Note that GALE depends on the spanning tree we use and, in particular, the traversal of that spanning tree. Let $\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}$ be the outcome of GALE on the traversal $(x_1, \ldots, x_J)$ of the spanning tree $\mathcal{T}$ of $\mathcal{S}_{m,T}$.

Theorem 2.3.2 (Misclustering rate of GALE). *Let $0 < \theta < 1$ and $r, r' > 0$. Let $m = \Omega_{r,r',\theta}\left(\sqrt{\frac{n \log n}{\pi_{\min}}}\right)$, $T = \Omega_{r,r',\theta}(n \log n/m)$, and $\tau = \frac{\theta T m}{n}$. Consider an algorithm $\mathcal{A}$ which labels any random m-subgraph with error $\leq m_1 \pi_{\min}/12$ with probability at least $1 - \delta$. Then we have, with probability at least $1 - T\delta - O(1/n^{r'})$, that*

$$\max_{\mathcal{T}} \max_{(x_1,\ldots,x_J)} \delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) \leq \frac{1}{\theta T} \sum_{\ell=1}^{T} \delta(\hat{Z}_\ell, Z) + O\left(\frac{1}{n^r}\right). \tag{2.7}$$

Again, the constant θ can be taken as close to 1 as one desires. Thus the above bound is also essentially optimal.

We will now use existing consistency results on several clustering algorithms $\mathcal{A}$, in conjunction with the above bounds to see what conditions (i.e. conditions on the model parameters, and m, T, etc.) are required for PACE and GALE to be consistent. We will use stochastic blockmodel as the generative model and for simplicity will assume that the link probability matrix has the following simple form

$$B = \alpha_n[\lambda I + (1 - \lambda)\mathbf{1}\mathbf{1}^\top] \text{ with } 0 < \lambda < 1. \tag{2.8}$$

Corollary 2.3.3 (PACE and GALE with spectral clustering and SDP). *Consider the blockmodel (2.8) and also assume balancedness, i.e. $\pi_{\min}, \pi_{\max} \asymp 1/K$. Suppose we use PACE or GALE with T random m-subgraphs and adjacency spectral clustering (ASP) or SDP as algorithm $\mathcal{A}$. Then, in order for PACE to give a consistent clustering, it suffices to take*

$$m \gg \frac{K^2}{\lambda^2 \alpha_n}, \quad T \gg \frac{n^2}{m^2}, \tag{2.9}$$

and for GALE, it suffices to take

$$m \gg \frac{K^2}{\lambda^2 \alpha_n}, \quad m = \Omega(\sqrt{Kn \log n}), \quad T = \Omega\left(\frac{n \log n}{m}\right). \tag{2.10}$$

If ego-subgraphs and ASP are used with PACE, then for consistency in model (2.8), with balanced block sizes and fixed K, λ, it is sufficient to have

$$\min\{n\alpha_n^2, T\alpha_n^2\} \to \infty. \tag{2.11}$$

Thus we see that for large K or small α_n (i.e. high sparsity) or a small separation between the blocks (i.e. λ), we would need a large m, which is natural to expect. For fixed K, λ, the average degree $d_n \asymp n\alpha_n$. Rephrasing in terms of d_n, we need $m \gg nd_n^{-1}$. So, if our implementation of $\mathcal{A}$ has complexity $O(n^3)$, then the complexity of PACE and GALE will be:

$$O\left(\frac{Tm^3}{N_c}\right) = \begin{cases} \tilde{O}(\frac{n^2}{m^2 N_c} m^3) = \tilde{O}(\frac{n^3}{d_n N_c}) & \text{for PACE,} \\ O(\frac{n \log n}{m N_c} m^3) = \tilde{O}(\max(\frac{n^2 (\log n)^2}{N_c}, \frac{n^3 \log n}{d_n^2 N_c})) & \text{for GALE,} \end{cases}$$

where $\tilde{O}$ hides slowly growing factors that may come from $\gg$, and N_c is the number of parallel processor cores we use. We see that we may get a significant boost in terms of computation, if the average degree $d_n = \Omega(\log n)$. Even in the bounded degree case, the benefit from parallelization may be significant from a practical point of view.

On the other hand, the condition (2.11) translates to $d_n \gg \sqrt{n}$, and $T \gg \frac{n^2}{d_n^2}$. That with ego neighborhoods we can not go down to $d_n = \Theta(\log n)$ is not surprising, since these ego networks are rather sparse in this case. One would want to use larger neighborhoods.

Although not clear from our analysis, in numerical simulations, we have found that PACE with neighborhood subgraphs works quite well in sparse settings. For example, in sparse and unbalanced graphs PACE with (regularized) spectral clustering vastly outperforms ordinary (regularized) spectral clustering (see Table 2.4). It seems that the reason why PACE works well in sparse settings lies in the weights N_{ij}. With h-hop neighborhoods as the chosen subgraphs, if $P_{uv} = \mathbf{1}_{\{\rho_g(u,v) \le h\}}$, where ρ_g is the geodesic distance on G, then $N_{ij} = (P^2)_{ij}$. It is known that spectral clustering on the matrix of geodesic distances works well in sparse settings [12]. PACE seems to inherit that property through N; proving this is part of our ongoing work.

Quantitative versions of Corollary 2.3.3 appear in Appendix A. Needless to say, there is nothing special about using spectral clustering or SDP. Similar results can be derived readily for other clustering algorithms.

2.4 Simulations and real data analysis[†]

In Table 2.1 we present a qualitative comparison of PACE and GALE with four representative global community detection methods profile likelihood (PL), mean field likelihood (MFL),

[†]Code used in this chapter is publicly available at https://github.com/soumendu041/divide-and-conquer-community-detection.

spectral clustering (SC) and semidefinite programming (SDP).

Table 2.1: Qualitative comparison of various methods; TC = theoretical complexity, RWS = real world scalability in terms of the size of the network, P = parallelizability.

	PL	MFL	SC	SDP	PACE	GALE
TC	NP hard	$O(n^{\theta_1})^{\ddagger}$	$O(n^3)$	$O(n^{\theta_2})^{\ddagger}$	$O(n^{2+\epsilon})^{\S}$	$O(n^{2+\epsilon})^{\S}$
RWS	$10^2 - 10^3$	$10^2 - 10^3$	10^6	$10^2 - 10^3$	$\gg 10^6$	$\gg 10^6$
P	✗	✗	✗	✗	✓	✓

‡ θ_j's (≥ 3) depend on details of implementation and numerical accuracy sought.
§ with $O(n^3)$ algorithms.

Simulations: comparison against traditional algorithms

For simulations we will use the planted partition model with $B = (p-q)I + qJ = \rho_n a((1-r)I + rJ)$, where I is the K dimensional identity matrix and J is the $K \times K$ matrix of all ones. Here ρ_n denotes the density, r denotes the ratio between the within between block linkage probabilities p and q. For cluster proportions $\pi_i, i = 1, \ldots, K$, the average degree d is given by $d_n = (n-1)\rho_n a((1-r)\sum_i \pi_i^2 + r)$.

In order to understand and emphasize the role of PACE and GALE in reducing computational time while maintaining good clustering accuracy, we use different settings of sparsity for different methods. For recovering $\hat{Z}$ from $\hat{C}$ in PACE, we have used random projection plus K-means (abbreviated as RPKmeans below), and spectral clustering (SC). We will denote these two variants of PACE by PACE-1 and PACE-2 respectively. We also want to point out that, for sparse unbalanced networks GALE may return more than K clusters, typically when a very small fraction of nodes has not been visited. However, it is possible that the unvisited nodes have important information about the network structure. For example, all subgraphs may be chosen from the larger clusters, thereby leaving the smallest cluster unvisited. We take care of this by computing the smallest error between the $(K+1)!$ permutations of GALE's clustering to the ground truth. This essentially treats the smallest cluster returned by GALE as misclustered. In real and simulated networks we have almost never seen GALE return a large number of unvisited nodes. All experiments except those of Section 2.4 were carried out on a Dell PowerEdge R620 server with Intel Xeon E-26XX v2 processors, 48 cores, 384GB RAM. The experiments of Section 2.4 were carried out on a server with AMD Opteron Processor 8384, 32 cores, 62 GB RAM.

SDP with ADMM

Interior point methods for SDPs are not very fast in practice. We have solved SDPs using an ADMM based implementation of [81]. From Table 2.2 we see that PACE and GALE significantly reduces the running time of SDP without losing accuracy too much. In fact,

if we use spectral clustering to estimate $\hat{Z}$ from $\hat{C}$ in the last step of PACE, we get zero misclustering error (ME).

Mean field likelihood (MFL)

Our implementation of global MFL did not converge to an acceptable solution even after *five and half hours*, while both PACE and GALE return better solutions (Table 2.3) in about *two minutes*. Interestingly, the misclustering error of PACE-2 is only 0.14, which is quite low, which begs the question if this improvement is due to spectral clustering step of PACE-2 only. In the next simulation, we show certain parameter settings where PACE and GALE lead to significant improvements in terms of accuracy and running time even when spectral clustering is used as the base algorithm.

Table 2.2: PACE and GALE with SDP. $n = 5000$, $d_n = 128$, $m = 500$, 4 equal sized clusters, $T = 100$, 20 workers.

Algorithm	ME(%)	Time
SDP	0	1588s
SDP + PACE-1	9.1	281s
SDP + PACE-2	0	288s
SDP + GALE	1.2	281s

Table 2.3: PACE and GALE with MFL. $n = 5000$, $d_n = 13$, 2-hop neighborhood, 2 equal sized clusters, $T = 100$, 20 workers.

Algorithm	ME(%)	Time
MFL	50	20439s
MFL + PACE-1	36.5	125s
MFL + PACE-2	1.4	131s
MFL + GALE	19.2	126s

Regularized spectral clustering (RSC)

In sparse unbalanced settings (Table 2.4), RSC with PACE and GALE performs significantly better than global RSC. Remarkably, with PACE-2 we can hit about 5% error or below. As mentioned before, this is probably due to the connection of the N_{ij}'s with geodesics. In Section 2.4 we will see that PACE and GALE also stabilize spectral clustering in terms of clustering degree 1 vertices.

Profile likelihood (PL) with tabu search

PL involves combinatorial optimization and is hard to scale, even if we ignore the problem of local minima. In Table 2.5 we show that the local versions of PL (optimized using tabu search [87]) significantly improve the running time without sacrificing accuracy too much. We also apply PL on 5000 node graphs with 20 workers. Although PACE and GALE finished in about 22 minutes, the global method did not finish in 3 days. Hence we use 1000 node networks instead.

Remark 2.4.1. *So far we have seen that, for recovering $\hat{Z}$ from $\hat{C}$ in PACE, spectral clustering outperforms the random projection based algorithms (e.g., RPKmeans). For smaller*

Table 2.4: PACE and GALE with RSC. Simulation settings: $n = 5000$, average degree $= 7$, cluster proportions $\boldsymbol{\pi} = (0.2, 0.8)$, $T = 100$, parallel implementation in Matlab with 20 workers.

	Random 1500-subgraph		3-hop neighborhood		5-hop onion	
Algorithm	ME(%)	Time	ME(%)	Time	ME(%)	Time
RSC	39.6	87s				
RSC+PACE-1	34.7	20s	34.2	14s	18	53s
RSC+PACE-2	11.1	26s	3.4	21s	5.1	59s
RSC + GALE	17.9	23s	33.6	13s	29.7	52s

Table 2.5: PACE and GALE with PL. $n = 1000$, average degree $= 18.47$, Cluster proportion $\boldsymbol{\pi} = (0.4, 0.6)$, $T = 50$, parallel implementation in Matlab with 12 workers. Roots of the 2-hop neighborhoods were sampled uniformly at random from nodes with degree above the 0.1th lower quantile ($= 12$) of the degree distribution (average neighborhood size was 310). Ordinary PACE with such a scheme may be thought of as w-PACE (see Section 2.2).

	Random 310-subgraph		2-hop neighborhood	
Algorithm	ME(%)	Time	ME(%)	Time
PL	0	70m		
PL + PACE-1	3.9	30m	3.5	38m
PL + GALE	1.2	30m	29.5	38m

networks, this is not an issue (e.g., spectral clustering on the dense 5000×5000 *matrix* $\hat{C}$ *in the context of Table 2.3 took only about 7-8 seconds). However, for networks of much larger scale (say, with several million nodes), that last step would be costly if spectral clustering is used.*

Real data analysis with political blog data

This is a directed network (see Figure 2.2) of hyperlinks between 1490 blogs (2004) that are either liberal or conservative [1]; we have ground truth labels available for comparison, 758 are liberal, 732 are conservative. We convert it into an undirected network by connecting blogs i and j if there is at least one directed edge between them.

Since the resulting network has many isolated nodes and edges, we focus on the largest connected component. To correct for its heterogeneous degree distribution (so a degree-corrected model would be more appropriate), we use normalized spectral clustering [69] with PACE.

Figure 2.2: Network of political blogs, red nodes are conservative, blues are liberal; picture courtesy: [1].

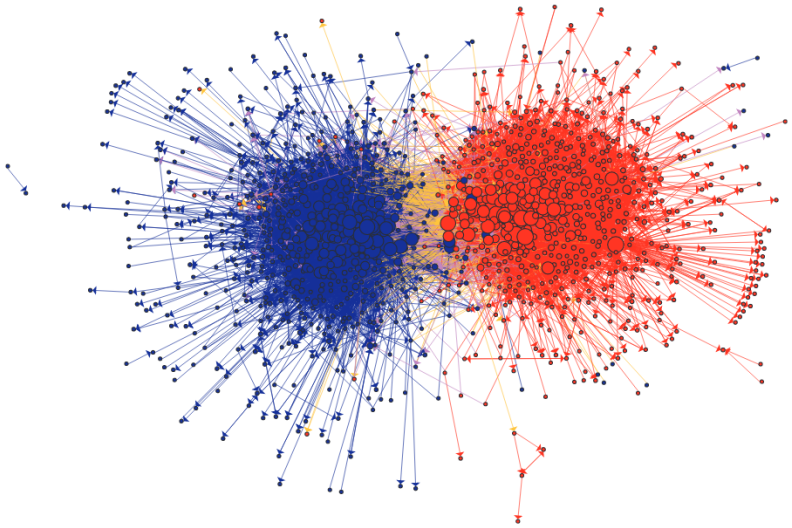


Table 2.6: Misclustering rate in the political blog data. There are 1222 nodes in the graph w/ leaves, and 1087 w/o leaves. In (a), PACE is used with $T = 10$, and h-hop neighborhoods with $h = 2$ (the roots are chosen at random from high degree nodes). In (b), PACE and GALE are used with $T = 50$, and random m-subgraphs with $m = 300$.

	w/ leaves	w/o leaves
RSC	18.74%	11.87%
SC	48.12%	3.13%
RSC + PACE-2	6.79%	4.23%
SC + PACE-2	6.55%	3.86%

(a)

	w/ leaves	w/o leaves
RSC + PACE-2	13.34%	12.8%
RSC + GALE	11.62%	10.0%
SC + PACE-2	7.86%	7.28%
SC + GALE	5.81%	6.7%

(b)

Table 2.6 shows that PACE and GALE add stability (possibly in eigenvector computation) to spectral clustering. Indeed, PACE and GALE can cluster "leaf" vertices (i.e. vertices of degree 1) with significantly higher accuracy.

2.5 Proofs

Results on PACE

We will first prove Theorem 2.3.1. The proof will be broken down into two propositions. First we decompose

$$\hat{C}_{ij} - C_{ij} = \underbrace{\hat{C}_{ij} - C_{ij}\mathbf{1}_{\{N_{ij}\geq\tau\}}}_{=:(E_1)_{ij}} + \underbrace{C_{ij}\mathbf{1}_{\{N_{ij}\geq\tau\}} - C_{ij}}_{=:(E_2)_{ij}} \tag{2.12}$$

Now observe that $(E_1)_{ij} = \mathbf{1}_{\{N_{ij}\geq\tau\}} \sum_{\ell=1}^{T} y_{ij}^{(\ell)}(\hat{C}_{ij}^{(\ell)} - C_{ij})/N_{ij}$, and $(E_2)_{ij} = -\mathbf{1}_{\{N_{ij}<\tau\}}C_{ij}$, so that $(E_1)_{ij}(E_2)_{ij} = 0$. Therefore

$$\frac{1}{n^2}\|\hat{C} - C\|_F^2 = \frac{1}{n^2}(\|E_1\|_F^2 + \|E_2\|_F^2). \tag{2.13}$$

We will estimate $\|E_1\|_F$ and $\|E_2\|_F$ separately.

Proposition 2.5.1. *We have*

$$\mathbb{E}\|E_1\|_F^2 \leq \frac{T}{\tau}\mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2, \tag{2.14}$$

where S is a randomly chosen subgraph under our subgraph selection scheme.

Proof. Let $W_{ij} := \frac{\mathbf{1}_{\{N_{ij}\geq\tau\}}}{N_{ij}}$. Then $(E_1)_{ij} = W_{ij}\sum_{\ell=1}^{T} y_{ij}^{(\ell)}(\hat{C}_{ij}^{(\ell)} - C_{ij})$. Therefore $\|E_1\|_F^2 = \sum_{i,j} W_{ij}^2\Big(\sum_{\ell=1}^{T} y_{ij}^{(\ell)}(\hat{C}_{ij}^{(\ell)} - C_{ij})\Big)^2$. So, by an application of Cauchy-Schwartz, we can upper bound this by

$$\sum_{i,j} W_{ij}^2\left(\sum_{\ell=1}^{T} y_{ij}^{(\ell)}\right)\left(\sum_{\ell=1}^{T} y_{ij}^{(\ell)}(\hat{C}_{ij}^{(\ell)} - C_{ij})^2\right) = \sum_{\ell=1}^{T}\sum_{i,j} W_{ij}y_{ij}^{(\ell)}(\hat{C}_{ij}^{(\ell)} - C_{ij})^2.$$

Noting that $W_{ij} \leq \frac{1}{\tau}$, and $\sum_{i,j} y_{ij}^{(\ell)}(\hat{C}_{ij}^{(\ell)} - C_{ij})^2 = \|\hat{C}^{(\ell)} - C^{(\ell)}\|_F^2$, we finally obtain $\|E_1\|_F^2 \leq \frac{1}{\tau} \times \sum_{\ell=1}^{T}\|\hat{C}^{(\ell)} - C^{(\ell)}\|_F^2$. Since the subgraphs are chosen independently using the same sampling scheme, the $\|\hat{C}^{(\ell)} - C^{(\ell)}\|_F$ are identically distributed. Now taking expectation yields the desired result. □

Proposition 2.5.2. *Let $n_{\max}$ be the largest block size. Then we have*

$$\mathbb{E}\|E_2\|^2 \leq n_{\max} n \max_{i,j}\mathbb{P}(N_{ij} < \tau). \tag{2.15}$$

Proof. Since $-(E_2)_{ij} = C_{ij}\mathbf{1}_{\{N_{ij}<\tau\}}$, we have $\|E_2\|_F^2 = \sum_{i,j} C_{ij}^2\mathbf{1}_{\{N_{ij}<\tau\}}$, and by taking expectations we get

$$\mathbb{E}\|E_2\|_F^2 = \sum_{i,j} C_{ij}^2\mathbb{P}(N_{ij} < \tau) \leq \max_{i,j}\mathbb{P}(N_{ij} < \tau)\sum_{i,j} C_{ij}^2.$$

The result now follows since $\sum_{i,j} C_{ij}^2 = \sum_{a=1}^{K} n_a^2 \leq n_{\max} n$. □

Proof of Theorem 2.3.1. Combining Propositions 2.5.1 and 2.5.2, we get (2.3). Finally,

$$\mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2 \mathbf{1}_{(|S|<m_\star)} \leq n^2 \mathbb{P}(|S| < m_\star),$$

and

$$\mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2 \mathbf{1}_{(|S|\geq m_\star)} \leq 4\mathbb{E}|S|^2 \delta(\hat{Z}_S, Z_S)\mathbf{1}_{(|S|\geq m_\star)},$$

yielding the last claim. □

Proof of Corollary 2.3.1. For this sampling scheme $|S| = m \geq m_\star$ and with $p = \frac{m(m-1)}{n(n-1)}$, $N_{ij} \sim \mathsf{Binomial}(T, p)$ so that we have, using the Chernoff bound[‡] for binomial lower tail, that

$$P(N_{ij} < \theta T p) \leq e^{-(1-\theta)^2 T p/2}.$$

Plugging in these parameter values and estimates in (2.3) gives (2.4). □

Proof of Corollary 2.3.2. The most crucial thing to observe here is that if one removes the root node and its adjacent edges from a 1-hop neighborhood, then the remaining "ego network" has again a blockmodel structure. Indeed, let S be a random ego neighborhood of size $\geq s$ with root R, i.e. $V(S) = \{j : A_{Rj} = 1\}$. Then conditional on $V(S)$ being R's neighbors, and the latent cluster memberships, edges in $E(S)$ are independently generated, i.e. for $j, k, \ell, m \in V(S)$, and $s, t \in \{0, 1\}$, we have

$$\mathbb{P}(A_{jk} = s, A_{\ell m} = t|S, Z) = \mathbb{P}(A_{jk} = s|Z)\mathbb{P}(A_{\ell m} = t|Z).$$

This is because the "spoke" edges A_{Rj} are independent of $A_{j,k}, j, k \in V(S)$. Therefore, conditional on S, this ego-subgraph is one instantiation of a blockmodel with the same parameters on $|S|$ vertices.

Now for ego networks, $y_{ij}^{(\ell)} \sim \mathsf{Bernoulli}(n_{ij}/n)$, where n_{ij} is the total number of ego-subgraphs containing both i and j. Notice that

$$n_{ij} = \sum_{\ell \neq i,j} \mathbf{1}_{\{A_{i\ell}=1, A_{j\ell}=1\}},$$

that is, n_{ij} is the sum of $(n-2)$ independent Bernoulli random variables

$$\mathbf{1}_{\{A_{i\ell}=1, A_{j\ell}=1\}} \sim \mathsf{Bernoulli}(B_{\sigma(i)\sigma(\ell)} B_{\sigma(j)\sigma(\ell)}).$$

So

$$(n-2)B_\#^2 \geq \mathbb{E}n_{ij} = \sum_{\ell \neq i,j} B_{\sigma(i)\sigma(\ell)} B_{\sigma(j)\sigma(\ell)} \geq (n-2)B_\star^2,$$

[‡]We use a slightly loose but convenient form of the Chernoff bounds: (i) $\mathbb{P}(X \leq (1-\delta)\mu) \leq \exp(-\delta^2\mu/2)$, and (ii) $\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp(-\delta^2\mu/3)$, where $X = X_1 + \cdots + X_n$ and X_i's are independent binary random variables, with $\mathbb{E}X = \mu$, and $0 < \delta < 1$.

and we have, by the Chernoff bound, that

$$\begin{aligned}\mathbb{P}(n_{ij} \le (n-2)B_\star^2 - \Delta) &\le \exp\left(-\frac{(\mathbb{E}n_{ij} - (n-2)B_\star^2 + \Delta)^2}{2\mathbb{E}n_{ij}}\right) \\ &\le \exp\left(-\frac{\Delta^2}{2(n-2)B_\#^2}\right). \end{aligned} \tag{2.16}$$

In order to apply Theorem 2.3.1 we need (i) an estimate of $|S|$, and (ii) an estimate of $\mathbb{P}(N_{ij} < \tau)$.

(i) Estimate of $|S|$. Note that $|S| = \sum_k A_{kR}$. Since $A_{kR}, 1 \le k \le n$ are independent, and

$$(n-1)B_\star \le \mathbb{E}(|S| \,|\, R) = \sum_{j \ne R} B_{\sigma(k)\sigma(R)} \le (n-1)B_\#,$$

we have, by Chernoff's inequality, that

$$\begin{aligned}\mathbb{P}(|S| < m_\star \,|\, R) &\le \exp\left(-\frac{(\mathbb{E}(|S| \,|\, R) - m_\star)^2}{2\mathbb{E}(|S| \,|\, R)}\right) \\ &\le \exp\left(-\frac{((n-1)B_\star - m_\star)^2}{2(n-1)B_\#}\right),\end{aligned}$$

where $m_\star \le (n-1)B_\star$. Therefore the same upper bound holds for $\mathbb{P}(|S| < m_\star)$. In particular, for $m_\star \le (n-1)B_\star/2$ we have

$$\mathbb{P}(|S| < m_\star) \le \exp\left(-\frac{(n-1)B_\star^2}{8B_\#}\right) \le \exp\left(-\frac{nB_\star^2}{16B_\#}\right). \tag{2.17}$$

Similarly, using Chernoff's inequality for binomial upper tail, we can show that, for $\theta > 0$,

$$\mathbb{P}(|S| > (1+\theta)nB_\#) \le \exp\left(-\frac{\theta^2 nB_\#}{6}\right). \tag{2.18}$$

(ii) Estimate of $\mathbb{P}(N_{ij} < \tau)$. Recall that $N_{ij} \,|\, n_{ij} \sim \mathsf{Binomial}(T, n_{ij}/n)$. Then

$$\mathbb{P}(N_{ij} < \tau) = \underbrace{\mathbb{P}\left(N_{ij} < \tau, n_{ij} < \frac{2n\tau}{T}\right)}_{=:P_1} + \underbrace{\mathbb{P}\left(N_{ij} < \tau, n_{ij} \ge \frac{2n\tau}{T}\right)}_{=:P_2}.$$

Clearly

$$P_1 \le \mathbb{P}\left(n_{ij} < \frac{2n\tau}{T}\right) \le \mathbb{P}\left(n_{ij} < \frac{nB_\star^2}{2}\right).$$

Now, given n_{ij} such that $n_{ij} \ge \frac{2n\tau}{T}$, we can invoke Chernoff's inequality to get

$$\mathbb{P}(N_{ij} < \tau \,|\, n_{ij}) \le \exp\left(-\frac{(\mathbb{E}(N_{ij} \,|\, n_{ij}) - \tau)^2}{2\mathbb{E}(N_{ij} \,|\, n_{ij})}\right) \le \exp\left(-\frac{Tn_{ij}}{8n}\right).$$

Therefore

$$\begin{aligned} P_2 &\le \mathbb{E}\exp\left(-\frac{Tn_{ij}}{8n}\right)\mathbf{1}_{\{n_{ij}\ge\frac{2n\tau}{T}\}} \\ &\le \mathbb{E}\exp\left(-\frac{Tn_{ij}}{8n}\right) \\ &= \mathbb{E}\exp\left(-\frac{Tn_{ij}}{8n}\right)\mathbf{1}_{\{n_{ij}<\frac{nB_\star^2}{2}\}} + \mathbb{E}\exp\left(-\frac{Tn_{ij}}{8n}\right)\mathbf{1}_{(n_{ij}\ge\frac{nB_\star^2}{2})} \\ &\le \mathbb{P}\left(n_{ij}<\frac{nB_\star^2}{2}\right) + \exp\left(-\frac{TB_\star^2}{16}\right). \end{aligned}$$

Thus

$$\mathbb{P}(N_{ij}<\tau) \le 2\mathbb{P}\left(n_{ij}<\frac{nB_\star^2}{2}\right) + \exp\left(-\frac{TB_\star^2}{16}\right).$$

But by (2.16)

$$\begin{aligned} \mathbb{P}\left(n_{ij}<\frac{nB_\star^2}{2}\right) \le \exp\left(-\frac{((n-2)B_\star^2-\frac{nB_\star^2}{2})^2}{2(n-2)B_\#^2}\right) &\le \exp\left(-\frac{(n-2)}{8}\frac{B_\star^4}{B_\#^2}\right) \\ &\le \exp\left(-\frac{nB_\star^4}{16B_\#^2}\right). \end{aligned}$$

Thus

$$\mathbb{P}(N_{ij}<\tau) \le 2\exp\left(-\frac{nB_\star^4}{16B_\#^2}\right) + \exp\left(-\frac{TB_\star^2}{16}\right).$$

Now we are ready to use (2.3). Using our estimates on $|S|$ we get that

$$\begin{aligned} \mathbb{E}|S|^2\delta(\hat{Z}_S,Z_S)\mathbf{1}_{(|S|\ge m_\star)} \le (1+\theta)^2n^2B_\#^2\mathbb{E}\delta(\hat{Z}_S,Z_S)\mathbf{1}_{(|S|\ge m_\star)} \\ + \exp\left(-\frac{\theta^2nB_\#}{6}\right). \end{aligned}$$

Next we plug into (2.3) all the estimates we derived in this subsection to get the desired bound (2.5) As mentioned before, we can allow $\tau = \theta' TB_\star^2$, for any $0<\theta'<1$. As the resulting bound involves complicated constants depending on θ' and does not add anything extra as to the nature of the bounds, we have chosen to work with a particular θ' $(=1/4)$ to ease our exposition. With a general θ', the constant multiplier in the first term in (2.5) will be $4\frac{(1+\theta)^2}{\theta'}$, instead of $16(1+\theta)^2$. □

Results on GALE

For any clustering $\hat{Z}_i$ on subgraph S_i, let $\Pi_i \in \arg\min_{Q \text{ perm.}} \|\hat{Z}_i - Z_iQ\|_1$, where Z_i is used a shorthand for $Z_{S_i} = Z\big|_{S_i}$, the true cluster membership matrix for the members of S_i. Define the matrix F_i by the requirement

$$\hat{Z}_i = Z_i\Pi_i + F_i, \quad F_i \in \{0,\pm1\}^{n\times K}. \tag{2.19}$$

In other words, $\|F_i\|_1 = \mathcal{M}(\hat{Z}_i, Z_i)$.

We now analyze Algorithm 3, i.e. Match. Recall that, if two clusterings on some set S agree, then the confusion matrix will be diagonal up to permutations, with the entries in the diagonal corresponding to the cluster sizes in either of the clusterings. In the following lemma, we consider a noisy version of this, where the two clusterings are not in perfect agreement. This lemma essentially establishes that if supplied with two clusterings whose confusion matrix is a diagonal matrix up to permutations plus noise, then Match will recover the correct aligning permutation, if the noise is not too large.

Lemma 2.5.1. *Let $d \in \mathbb{R}_+^K$. Also let $M = \Pi_2^\top \mathrm{diag}(d)\Pi_1 + \Gamma$, where $\Gamma \in \mathbb{R}^{K\times K}$, $\|\Gamma\|_\infty \le \min_i d_i/3$. Then Match returns $\Pi = \Pi_2^\top \Pi_1$, when applied on the confusion matrix M.*

Proof of Lemma 2.5.1. Let $D := \mathrm{diag}(d)$. Let ξ_i be the permutation encoded in Π_i, for $i \in [2]$, i.e. $(\Pi_i)_{uv} = \mathbf{1}_{\{v=\xi_i(u)\}}$. It is easy to see that $M_{uv} = D_{\xi_1^{-2}(u),\xi_2^{-1}(v)} + \Gamma_{uv}$. We have for all u, v such that $(\Pi_2^\top \Pi_1))_{uv} = 0$, $M_{uv} \le \|\Gamma\|_\infty$, whereas for all u, v such that $(\Pi_2^\top \Pi_1)_{uv} = 1$, $M_{uv} \ge \min_a D_{aa} - \|\Gamma\|_\infty$. Hence

$$\min_{u,v\,:\,(\Pi_2^\top \Pi_1)_{uv}=1} M_{uv} \ge \min_a D_{aa} - \|\Gamma\|_\infty \ge 2\|\Gamma\|_\infty \ge 2 \max_{u,v\,:\,(\Pi_2^\top \Pi_1)_{uv}=0} M_{uv}.$$

Hence the top K (recall that K is the number of rows in M) elements are the diagonal elements in D. Thus the elements of Π learned by the Match algorithm will be $\Pi_{uv} = \mathbf{1}_{\{\xi_2^{-1}(u)=\xi_1^{-1}(v)\}}$. This is equivalent to $\Pi = \Pi_2^\top \Pi_1$. □

Now we will establish that post-alignment misclustering errors on subgraphs equal the original misclustering errors. For this we first need a lemma on what happens when we align two subgraphs based on their intersection.

Lemma 2.5.2. *Consider two random m-subgraphs S_1 and S_2. Suppose our clustering algorithm $\mathcal{A}$ outputs clusterings $\hat{Z}_i, i = 1, 2$. Let $S = S_1 \cap S_2$ be of size at least $m_1 = \left\lceil \frac{m^2}{2n} \right\rceil$. Assume that $\mathcal{M}(\hat{Z}_i, Z_i) \le m_1\pi_{\min}/12$ and that $\min_{k\in[K]} n_k^{(S)} \ge m_1\pi_{\min}$. Then $\|\hat{Z}_2\Pi - Z\Pi_1\| = \mathcal{M}(\hat{Z}_2, Z)$, where Π is the output of Match$(\hat{Z}_2\big|_S, \hat{Z}_1\big|_S)$.*

Proof of Lemma 2.5.2. To ease notation, let us write $\tilde{Z}_i := \hat{Z}_i\big|_S$, $\tilde{F}_i := F_i\big|_S, i = 1, 2$, and $\tilde{Z} := Z\big|_S$. Then by restricting (2.19) to S, we get

$$\tilde{Z}_i = \tilde{Z}\Pi_i + \tilde{F}_i, \quad \tilde{F}_i \in \{0, \pm 1\}^{n\times K}.$$

Now

$$\tilde{Z}_2^\top \tilde{Z}_1 = \Pi_2^\top \tilde{Z}^\top \tilde{Z}\Pi_1 + \underbrace{\Pi_2^\top \tilde{Z}^\top \tilde{F}_1 + \tilde{F}_2^\top \tilde{Z}\Pi_1 + \tilde{F}_2^\top \tilde{F}_1}_{=:\Gamma}.$$

Note that (a) multiplication by a permutation matrix does not change the $\|\cdot\|_1$ norm, and (b) for any matrix A, we have $\|A\|_1 = \|A^\top\|_1$. Therefore

$$\begin{aligned}\|\Pi_2^\top \tilde{Z}^\top \tilde{F}_1\|_1 = \|\tilde{Z}^\top \tilde{F}_1\|_1 &= \sum_{a,b} |\sum_i (\tilde{Z}^\top)_{ai}(\tilde{F}_1)_{ib}| \\ &\le \sum_{a,b}\sum_i \tilde{Z}_{ia}|(\tilde{F}_1)_{ib}| = \sum_{i,b} |(\tilde{F}_1)_{ib}| \times \underbrace{\sum_a \tilde{Z}_{ia}}_{=1} = \|\tilde{F}_1\|_1.\end{aligned}$$

Similarly,

$$\|\tilde{F}_2^\top \tilde{Z}\Pi_1\|_1 = \|\Pi_1^\top \tilde{Z}^\top \tilde{F}_2\|_1 \le \|\tilde{F}_2\|_1.$$

Finally,

$$\|\tilde{F}_2^\top \tilde{F}_1\|_1 \le \sum_{a,b}\sum_i |(\tilde{F}_2)_{ia}(\tilde{F}_1)_{ib}| = \sum_{i,b} |(\tilde{F}_1)_{ib}| \times \underbrace{\sum_a |(\tilde{F}_2)_{ia})|}_{=2} = 2\|\tilde{F}_1\|_1,$$

where we have used the fact that each row of $\tilde{F}_i$ has exactly one 1 and one -1. By (b), $\|\tilde{F}_2^\top \tilde{F}_1\|_1 = \|\tilde{F}_1^\top \tilde{F}_2\|_1 \le 2\|\tilde{F}_2\|_1$, and therefore

$$\|\tilde{F}_2^\top \tilde{F}_1\|_1 \le \|\tilde{F}_1\|_1 + \|\tilde{F}_2\|_1.$$

Note that, by our assumptions on the individual misclustering errors,

$$\|\tilde{F}_i\|_1 \le \|F_i\|_1 = \mathcal{M}(\hat{Z}_i, Z_i) \le m_1\pi_{\min}/12.$$

Therefore

$$\|\Gamma\|_\infty \le 2(\|\tilde{F}_1\|_1 + \|\tilde{F}_2\|_1) \le m_1\pi_{\min}/3.$$

Since by our assumption, $(\tilde{Z}^\top \tilde{Z})_{kk} = n_k^{(S)} \ge m_1\pi_{\min}$, we can apply Lemma 2.5.1 to see that the output $\mathsf{Match}(\tilde{Z}_2, \tilde{Z}_1)$ is $\Pi = \Pi_2^\top \Pi_1$. Hence

$$\|\hat{Z}_2\Pi - Z\Pi_1\|_1 = \|\hat{Z}_2\Pi_2^\top\Pi_1 - Z\Pi_1\|_1 = \|\hat{Z}_2\Pi_2^\top - Z\|_1 = \|\hat{Z}_2 - Z\Pi_2\|_1 = \|F_2\|_1.$$

□

Proposition 2.5.3. *Let the T subsets, $(S_i)_{i=1}^T$ be associated with estimated clusterings $\hat{Z}_i$ with misclustering error $\mathcal{M}(\hat{Z}_i, Z_i) \le m_1\pi_{\min}/12$. Consider a traversal $x_1 \sim x_2 \sim \cdots \sim x_J$ of some spanning tree $\mathcal{T}$ of $\mathcal{S}_{m,T}$ as in Algorithm 4, satisfying*

$$\min_{2\le i\le J}\min_{k\in[K]} n_k^{(S_{i,i-1})} \ge m_1\pi_{\min},$$

where $n_k^{(S)}$ is the number of nodes from cluster k in a subgraph S, and $S_{i,i-1} := S_{x_i} \cap S_{x_{i-1}}$. Apply GALE on this walk. Let $\hat{Z}^{(x_1)} := \hat{Z}_{x_1}, \hat{\Pi}_{x_1} = I$ and for $2 \le i \le J$, define recursively

$$\hat{\Pi}_{x_i} = \begin{cases} I & \text{if } x_i = x_j \text{ for some } 1 \le j < i, \\ \mathsf{Match}(\hat{Z}_{x_i}\big|_{S_{i,i-1}}, \hat{Z}^{(x_{i-1})}\big|_{S_{i,i-1}}) & \text{otherwise,} \end{cases}$$

and set $\hat{Z}^{(x_i)} = \hat{Z}_{x_i}\hat{\Pi}_{x_i}$. *Then, for any* $1 \le i \le J$, *we have that*

$$\mathcal{M}(\hat{Z}^{(x_i)}, Z_{x_i}) = \mathcal{M}(\hat{Z}_{x_i}, Z_{x_i}).$$

Proof. We claim that, for all $1 \le i \le J$, we have

$$\Pi_{x_1} \in \arg\min_Q \|\hat{Z}^{(x_i)} - Z_{x_i}Q\|, \text{ and } \mathcal{M}(\hat{Z}^{(x_i)}, Z_{x_i}) = \mathcal{M}(\hat{Z}_{x_i}, Z_{x_i}). \tag{2.20}$$

We use strong induction to prove this claim. (2.20) is true by definition, for $i = 1$. Now assume that it is true for all $i \le \ell$. Then, for all $i \le \ell$, we have the representation $\hat{Z}^{(x_i)} = Z_{x_i}\Pi_{x_1} + \bar{F}_{x_i}$, for some matrix $\bar{F}_{x_i} \in \{0, \pm 1\}^{m\times K}$. If $x_{\ell+1}$ has been visited before, i.e. $x_{\ell+1} = x_j$ for some $j \le \ell$, then (2.20) holds by our induction hypothesis. Otherwise, we can apply Lemma 2.5.2 on the two clusterings $\hat{Z}^{(x_\ell)}$ on S_{x_ℓ} and $\hat{Z}_{x_{\ell+1}}$ on $S_{x_{\ell+1}}$, to conclude that

$$\|\hat{Z}^{(x_{\ell+1})} - Z_{x_{\ell+1}}\Pi_{x_1}\|_1 = \|\hat{Z}_{x_{\ell+1}}\hat{\Pi}_{x_{\ell+1}} - Z_{x_{\ell+1}}\Pi_{x_1}\|_1 = \mathcal{M}(\hat{Z}_{x_{\ell+1}}, Z_{x_{\ell+1}}).$$

But $\hat{Z}^{(x_{\ell+1})} = \hat{Z}_{x_{\ell+1}}\hat{\Pi}_{x_{\ell+1}}$, and $\hat{\Pi}_{x_{\ell+1}} = \Pi_{x_{\ell+1}}^\top\Pi_{x_1}$. So, for any permutation matrix Q we have

$$\begin{aligned}\|\hat{Z}^{(x_{\ell+1})} - Z_{x_{\ell+1}}Q\|_1 &= \|\hat{Z}_{x_{\ell+1}}\hat{\Pi}_{x_{\ell+1}} - Z_{x_{\ell+1}}Q\|_1 = \|\hat{Z}_{x\ell+1} - Z_{x_{\ell+1}}Q\hat{\Pi}_{x_{\ell+1}}^\top\|_1 \\ &\ge \mathcal{M}(\hat{Z}_{x_{\ell+1}}, Z_{x_{\ell+1}}) = \|\hat{Z}^{(x_{\ell+1})} - Z_{x_{\ell+1}}\Pi_1\|_1.\end{aligned}$$

So, indeed, $\Pi_{x_1} \in \arg\min_Q \|\hat{Z}^{(x_{\ell+1})} - Z_{x_\ell}Q\|$ and hence $\mathcal{M}(\hat{Z}^{(x_{\ell+1})}, Z_{x_{\ell+1}})$ and $\mathcal{M}(\hat{Z}_{x_{\ell+1}}, Z_{x_{\ell+1}})$ are equal. The induction is now complete. □

Now we establish an upper bound on the error of the estimator $\hat{Z}^{\mathsf{GALE}}$ in terms of the errors of the aligned clusterings $\hat{Z}^{(\ell)}$.

Proposition 2.5.4. *Let* $0 < \theta < 1$, *and set* $\tau = \frac{\theta T m}{n}$. *Then, for any cluster membership matrix* $\bar{Z} \in \{0,1\}^{n\times K}$,

$$\frac{\|\hat{Z}^{\mathsf{GALE}} - \bar{Z}\|_F^2}{n} \le \frac{1}{\theta T}\sum_{\ell=1}^{T}\frac{\|\hat{Z}^{(\ell)} - \bar{Z}_\ell\|_F^2}{m} + \sum_i \mathbf{1}_{\{N_i<\tau\}}. \tag{2.21}$$

Proof. The proof is very similar to that of Theorem 2.3.1. We decompose

$$\bar{Z}_{ik} = \bar{Z}_{ik}\mathbf{1}_{\{N_i\ge\tau\}} + \bar{Z}_{ik}\mathbf{1}_{\{N_i<\tau\}}.$$

Using this we have

$$\hat{Z}_{ik}^{\mathsf{GALE}} - \bar{Z}_{ik} = \frac{\sum_{\ell=1}^{T} y_i^{(\ell)}(\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik})}{N_i}\mathbf{1}_{\{N_i\ge\tau\}} - \bar{Z}_{ik}\mathbf{1}_{\{N_i<\tau\}}.$$

Therefore

$$\|\hat{Z}^{\mathsf{GALE}} - \bar{Z}\|^2 = \sum_{i,k} \left(\frac{\sum_{\ell=1}^T y_i^{(\ell)} (\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik})}{N_i} \right)^2 \mathbf{1}_{\{N_i \geq \tau\}} + \sum_{i,k} \bar{Z}_{ik}^2 \mathbf{1}_{\{N_i < \tau\}}$$
$$\leq \underbrace{\sum_{i,k} \left(\frac{\sum_{\ell=1}^T y_i^{(\ell)} (\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik})}{N_i} \right)^2 \mathbf{1}_{\{N_i \geq \tau\}}}_{=:E} + \sum_i \mathbf{1}_{\{N_i < \tau\}}.$$

Let $W_i := \frac{\mathbf{1}_{(N_i \geq \tau)}}{N_i}$. Noting that $E = \sum_{i,k} W_i^2 (\sum_{\ell=1}^T y_i^{(\ell)} (\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik}))^2$, an application of Cauchy-Schwartz inequality gives

$$E \leq \sum_{i,k} W_i^2 \left(\sum_{\ell=1}^T y_i^{(\ell)} \right) \sum_{\ell=1}^T y_i^{(\ell)} (\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik})^2 = \sum_{\ell=1}^T \sum_{i,k} W_i y_i^{(\ell)} (\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik})^2. \tag{2.22}$$

Using $W_i \leq \frac{1}{\tau} = \frac{n}{\theta T m}$, and $\sum_{i,k} y_i^{(\ell)} (\hat{Z}_{ik}^{(\ell)} - \bar{Z}_{ik})^2 = \|\hat{Z}^{(\ell)} - \bar{Z}_\ell\|_F^2$, we get the required bound on E. □

We now mention some auxiliary results, whose proofs are deferred to Appendix A. These will help us control the probability of a "bad" event, defined in the proof of Theorem 2.3.2.

Lemma 2.5.3. *Let $\tau = \frac{\theta T m}{n}$, where $0 < \theta < 1$. Let $r > 0$. Then, with probability at least $1 - \frac{1}{n^r}$, we have*

$$\sum_i \mathbf{1}_{\{N_i < \tau\}} \leq n e^{-(1-\theta)^2 T m / 2n} \left(1 + \sqrt{\frac{3r \log n}{n e^{-(1-\theta)^2 T m / 2n}}} \right).$$

Recall that we view the T random m-subsets of $[n]$ as nodes of a super-graph $\mathcal{S}_{m,T}$ and put an edge between nodes $a (\equiv S_a)$ and $b (\equiv S_b)$ in $\mathcal{S}_{m,T}$ (we use the shorthand $a \sim b$ to denote an edge between a and b), if $Y_{ab} := |S_a \cap S_b| \geq m_1 = \left\lceil \frac{m^2}{2n} \right\rceil$. The next lemma shows that $\mathcal{S}_{m,T}$ is in fact an Erdös-Rényi random graph.

Lemma 2.5.4. *The super-graph $\mathcal{S}_{m,T}$ is an Erdös-Rényi random graph with*

$$P(a \sim b) = P\left(Y_{ab} \geq \left\lceil \frac{m^2}{2n} \right\rceil \right) \geq 1 - \exp\left(-\frac{m^2}{16n} \right). \tag{2.23}$$

Because of our assumptions on m, it follows that $S_{m,T}$ is well above the connectivity threshold for Erdös-Rényi random graphs.

Lemma 2.5.5. *The super-graph $\mathcal{S}_{m,T}$ is connected with probability at least $1 - \exp(-O(n))$.*

The next lemma states that intersection of two random m-subgraphs contains enough representatives from each cluster, with high probability.

Lemma 2.5.6. *Consider two random m-subgraphs S_a and S_b. Let $r > 0$ and $\frac{m^2\pi_k}{n\log n} \geq 20r$. Then*

$$P\left(n_k^{(S_a\cap S_b)} < \frac{m^2\pi_k}{n}\left(1 - O\left(\sqrt{\frac{2rn\log n}{m^2\pi_k}}\right)\right)\right) \leq \frac{2}{n^r}.$$

We are now ready to prove our main result on GALE.

Proof of Theorem 2.3.2. First we construct a good set in the sample space. Consider the following "bad" events

$$\mathcal{B}_1 := \{(S_1,\ldots,S_T) \mid \mathcal{S}_{m,T} \text{ is connected}\},$$
$$\mathcal{B}_2 := \left\{(S_1,\ldots,S_T) \mid \min_{(i,j)}\min_k n_k^{(S_i\cap S_j)} < m_1\pi_{\min}\right\},$$

$$\mathcal{B}_3 := \left\{(S_1,\ldots,S_T) \mid \sum_i \mathbf{1}_{\{N_i<\tau\}} > e^{-\frac{(1-\theta)^2Tm}{2n}}\left(1 + \sqrt{\frac{3r'\log n}{ne^{-\frac{(1-\theta)^2Tm}{2n}}}}\right)\right\},$$
$$\mathcal{B}_4 := \{(A,S_1,\ldots,S_T) \mid \max_{1\leq i\leq T}\mathcal{M}(\hat{Z}_i,Z_i) > m_1\pi_{\min}/12\}.$$

Let $\mathcal{B} := \cup_{i=1}^4\mathcal{B}_i$. Let $r'' > 0$. By Lemma 2.5.5, we have $\mathbb{P}(\mathcal{B}_1) \leq \exp(-O(n))$. Lemma 2.5.6, and a union bound gives $\mathbb{P}(\mathcal{B}_2) \leq \binom{T}{2} \times K \times \frac{2}{n^{r''}} \leq \frac{KT^2}{n^{r''}}$. By Lemma 2.5.3, $\mathbb{P}(\mathcal{B}_3) \leq \frac{1}{n^{r'}}$. Finally, by our hypothesis on individual misclustering errors, we have, by a union bound, that $\mathbb{P}(\mathcal{B}_4) \leq T\delta$. Therefore

$$\mathbb{P}(\mathcal{B}) \leq \sum_{i=1}^4 \mathbb{P}(\mathcal{B}_i) \leq \exp(-O(n)) + \frac{KT^2}{n^{r''}} + \frac{1}{n^{r'}} + T\delta \leq T\delta + O\left(\frac{1}{n^{r'}}\right),$$

by choosing r'' suitably large.

Now on the good set $\mathcal{B}^c$, for any spanning tree $\mathcal{T}$ of $\mathcal{S}_{m,T}$ and for any traversal $(x_1,\ldots,x_J)$ of $\mathcal{T}$, the hypotheses of Propositions 2.5.3 and 2.5.4 are satisfied. Now note that

$$\delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) \leq \frac{\|\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)} - Z\Pi_{x_1}\|_1}{n} = \frac{\|\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)} - Z\Pi_{x_1}\|_F^2}{n}.$$

By Proposition 2.5.3, we also have, for any $1 \leq \ell \leq T$, that

$$\frac{\|\hat{Z}^{(\ell)} - Z_\ell\Pi_{x_1}\|_F^2}{m} = \frac{\|\hat{Z}^{(\ell)} - Z_\ell\Pi_{x_1}\|_1}{m} = \delta(\hat{Z}^{(\ell)}, Z) = \delta(\hat{Z}_\ell, Z).$$

Thus, by of Proposition 2.5.4, taking $\bar{Z} = Z\Pi_{x_1}$ and noting then that $\bar{Z}_\ell = Z_\ell\Pi_{x_1}$, we have

$$\delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) \leq \frac{1}{\theta T}\sum_{\ell=1}^T \delta(\hat{Z}_\ell, Z) + \sum_i \mathbf{1}_{\{N_i<\tau\}}.$$

Since the bound on the RHS does not depend on the particular spanning tree used, or a particular traversal thereof, on the good event $\mathcal{B}^c$, we have:

$$\max_{\mathcal{T}} \max_{(x_1,\ldots,x_J)} \delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) \leq \frac{1}{\theta T} \sum_{\ell=1}^{T} \delta(\hat{Z}_\ell, Z) + \sum_i \mathbf{1}_{\{N_i<\tau\}}.$$

Now, for our choice of τ, and $T \geq \frac{2r'''n\log n}{(1-\theta)^2 m}$, we have $e^{-(1-\theta)^2 Tm/2n} \leq \frac{1}{n^{r'''}}$. Therefore, on the event $\mathcal{B}^c$,

$$\sum_i \mathbf{1}_{\{N_i<\tau\}} = O\left(\sqrt{\frac{\log n}{n^{1+r'''}}}\right) = O\left(\frac{1}{n^r}\right),$$

by choosing r''' suitably large. Thus we conclude that with probability at least $1 - T\delta - O\left(\frac{1}{n^r}\right)$, we have

$$\max_{\mathcal{T}} \max_{(x_1,\ldots,x_J)} \delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) \leq \frac{1}{\theta T} \sum_{\ell=1}^{T} \delta(\hat{Z}_\ell, Z) + O\left(\frac{1}{n^r}\right). \qquad \square$$

2.6 Discussion

To summarize, we have proposed two divide and conquer type algorithms for community detection, PACE and GALE, which can lead to significant computational advantages without sacrificing accuracy. The main idea behind these methods is to compute the clustering for each individual subgraph and then "stitch" them together to produce a global clustering of the entire network. The main challenge of such a stitching procedure comes from the fundamental problem of unidentifiability of label assignments. That is, if two subgraphs overlap, the clustering assignment of a pair of nodes in the overlap may be inconsistent between the two subgraphs.

PACE addresses this problem by estimating the clustering matrix for each subgraph and then estimating the global clustering matrix by averaging over the subgraphs. GALE takes a different approach by using overlaps between two subgraphs to calculate the best alignment between the cluster memberships of nodes in the subgraphs. We prove that, in addition to being computationally much more efficient than base methods which typically run in $\Omega(n^2)$ time, these methods have accuracy at least as good as the base algorithm's typical accuracy on the type of subgraphs used, with high probability. Experimentally, we show something more interesting — we identify parameter regimes where a local implementation of a base algorithm based on PACE or GALE in fact outperforms the corresponding global algorithm. One example of this is the mean field algorithm, which typically suffers from bad local optima for large networks. Empirically, we have seen that on a smaller subgraph, with a reasonable number of restarts, it finds local optima that are often highly correlated with the ground truth. PACE and GALE take advantage of this phenomenon to improve on accuracy/running time significantly. Another example is regularized spectral clustering on sparse unbalanced networks.

Chapter 3

Divide and conquer for mixed memberships

3.1 Introduction

Latent space models are very popular tools in modern Statistics in modeling rich structures in real world data. These models typically involve a large number of latent variables, and fitting them is no easy task, especially when the dataset is large. Serious computational challenges arise from non-convexity of the relevant objective functions resulting from the multitude of parameters to learn. In this chapter, we shall discuss divide and conquer approaches to several important latent space models which aim to alleviate some of the computational issues.

Mixed membership blockmodels (MMBM) [3] are natural extensions of stochastic blockmodels, allowing for a node to be in multiple communities with varying degrees of strength. The model is specified as follows: there are node specific membership probability distributions $\theta_i \overset{i.i.d.}{\sim} \mathsf{Dirichlet}(\alpha)$, $\alpha = (\alpha_1, \ldots, \alpha_K)$, and a block probability matrix B. Two nodes i and j are connected with probability $\theta_i^\top B \theta_j$ independently of other pairs*. A principal goal is to estimate the membership probabilities θ_i. There are identifiability issues with this model. For a discussion and necessary/sufficient conditions for identifiability, see [48]. There are several algorithms in the literature for estimating θ_i's, such as variational Bayes [3], tensor decomposition methods [9], eigenvector-based methods [48] among others.

Topic models (TM) are mixed membership models used for text analysis. The most simple generative topic model is latent Dirichlet allocation (LDA) [17], where one observes a collection of D (independent) documents $W_{1:D,1:N}$ (a document is a collection of N words) — $W_{d,n}$ representing the n-th word in document d. These words are from a vocabulary of V

*This is a simplified version of the MMBM of [3], where for each ordered pair of nodes (i, j), there are two latent random community assignment vectors $z_{i\to j} \sim \mathrm{Mult}(\theta_i)$, $z_{j\to i} \sim \mathrm{Mult}(\theta_j)$, and a directed edge is created with probability $z_{i\to j}^\top B z_{j\to i}$. Our methods address the problem of estimating θ_i's and so are applicable to this model as well.

words. A topic is a distribution over the vocabulary, modeled as $\mathsf{Dirichlet}(\alpha)$. Suppose there are K topics, $\beta_{1:K}$ with $\beta_k \overset{i.i.d.}{\sim} \mathsf{Dirichlet}(\alpha)$. A document d is a mixture of topics, modeled as $\theta_d \sim \mathsf{Dirichlet}(\eta)$. Thus we have a set of D independent latent mixing distributions $\theta_{1:D}$. For each word, there is a topic assignment parameter $z_{d,n} \sim \text{Mult}(\theta_d)$, and the word is drawn as $W_{d,n} \sim \text{Mult}(\beta_{z_{d,n}})$. The topic assignment parameters are independent of each other, and given these, the words are also drawn independently. Here the topic distributions $\beta_{1:K}$, the document representations $\theta_{1:D}$ and the word-topic-assignments $z_{d,n}$ are all latent parameters, and the goal is to learn their posterior distribution given the document corpus. This is typically done by using the mean field variational approximation to the likelihood. There are extensions of the simple LDA, such as correlated topic models (CTM) [15], dynamic topic models (DTM) [16] among others.

A related but simpler problem is that of *biclustering* [36], where there are two types of clusters in an observed network: row clusters and column clusters. An example would be a predator-prey network where an individual species can be both a predator and a prey, and there are two different types of clusterings based on these roles (of course, this can be generalized to m-ary clustering, where there are m roles/dimensions). Thus there are row cluster assignments $Y \in \{0,1\}^{n \times K_r}$ and column cluster assignments $Z \in \{0,1\}^{n \times K_c}$, and the network is drawn as an inhomogeneous Erdös-Rényi random graph with link probability matrix $P = YBZ^\top$, where B_{ab} now denotes the probability of link formation between a member of the row cluster a and a member of the column cluster b. Again, one can have mixed membership versions of this model.

In Chapter 2, we proposed divide and conquer wrapper algorithms for the related problem of community detection (where a node is assumed to belong to a single community). In this chapter, we will generalize those algorithms for mixed membership structures. Thus they can be used for MMBM, TM or mixed membership versions of m-ary clustering (topic models are closer in spirit to such models).

3.2 Two divide and conquer algorithms

In all of the models described in the introduction, the following feature is present: there is a set of K clusters/communities, and nodes have hidden community assignment distributions θ_i. These θ_i could be degenerate (stochastic blockmodels, m-ary clustering) or mixed (MMBM, topic models, m-ary mixed memberships, etc.). The goal is to recover these θ_i's.

As in Chapter 2, the main issue with divide and conquer algorithms is that one has to somehow match up various potentially conflicting label assignments, arising from subsets of nodes. For the problem of clustering (degenerate θ_i's) we tackled this issue in Chapter 2 with the two algorithms PACE and GALE.

We can actually use the these two algorithms with minor modifications for mixed θ_i's. In this chapter, we will describe the modified algorithms and prove some results towards their consistency. Several new issues also arise with mixed θ_i which will also be addressed.

PACE: an averaging algorithm

Suppose the true cluster membership of the nodes of the data in concern is given by the $n \times K$ mixed-membership matrix Θ where there are K clusters. Set $C = \Theta\Theta^\top$ to be the "clustering" matrix whose (i, j)-th entry is a measure of the correlation of the mixed cluster membership structures of i and j. Given the data, we will perform a local clustering algorithm to obtain an estimate of C, from which an estimate $\hat{\Theta}$ of the cluster memberships may be reconstructed.

Algorithm 5 PACE: Piecewise Averaged Community Estimation

1: **Subset selection:** Fix a positive integer threshold $m_\star$ for minimum required subset size. Fix another positive integer T, that will be the number of subsets we will sample. Given the data, choose T subsets $S_1, \ldots, S_T$ of the nodes by some procedure, e.g., select $m \geq m_\star$ nodes at random, or pick h-hop neighborhoods (or onion neighborhoods) of vertices in the underlying graph (in case of MMBM or biclustering).

2: **Clustering on subsets:** Apply any existing clustering algorithm $\mathcal{A}$ of choice for the problem in question (such as the eigenvector-based method of [48] for MMBM, variational Bayes or tensor-decomposition for MMBM or TM, spectral clustering for biclustering, etc.) on each of these T subsets which have size at least $m_\star$ to obtain estimated clustering matrices $\hat{C}^{(S_l)} = \hat{C}^{(l)}$. For the rest of the subsets, set $\hat{C}^{(l)} \equiv 0$. Extend $\hat{C}^{(l)}$ to an $n \times n$ matrix by setting $\hat{C}^{(l)}_{ij} = 0$ if at least one of i, j was not selected in S_l. Denote the resulting matrix again by $\hat{C}^{(l)}$.

3: **Patching up:** Let $y^{(l)}_{ij}$ denote the indicator of the event that both i, j were selected in S_l. Set $N_{ij} = \sum_{l=1}^T y^{(l)}_{ij}$. Define the combined estimator $\hat{C} = \hat{C}_\tau$ by

$$\hat{C}_{ij} = \hat{C}_{\tau,ij} = \frac{\mathbf{1}_{\{N_{ij} \geq \tau\}} \sum_{l=1}^T y^{(l)}_{ij} \hat{C}^{(l)}_{ij}}{N_{ij}} = \frac{\mathbf{1}_{\{N_{ij} \geq \tau\}} \sum_{l=1}^T \hat{C}^{(l)}_{ij}}{N_{ij}}. \tag{3.1}$$

Here $1 \leq \tau \leq T$ is an integer tuning parameter. We will call $\hat{C}_\tau$ as Piecewise Averaged Community Estimator (also abbreviated as PACE).

Discussions on the PACE algorithm for community detection from Chapter 2 also apply here. In particular, one may consider a weighted version of PACE.

Reconstruction of $\hat{\Theta}$: How do we actually reconstruct $\hat{\Theta}$ from $\hat{C}$? One can apply any cheap mixed membership estimation algorithm such as the sequential projection algorithm SPACL of [48] on $\hat{C}$ to construct an estimate. Note that reconstructing $\hat{\Theta}$ from $\hat{C}$ is same as (approximate) symmetric non-negative matrix factorization (SNMF) under the constraint of the solution having 1 as an eigenvalue with eigenvector $\mathbf{1}$. SPACL is one such algorithm. Developing a more computationally efficient algorithm would be an interesting topic of future research.

GALE: a sequential algorithm

First we extend Match to compute the best permutation to align labels of one clustering (Θ_1) to another (Θ_2) of the same set of nodes in a set S.

Algorithm 6 Match: A Matching algorithm for matching the two labelings

1: Compute $k \times k$ confusion matrix $M = \Theta_1^\top \Theta_2$.
2: While there are no rows/columns left,
   1: Find i, j, such that $M_{ij} = \max_{i',j'} M_{i',j'}$.
   2: Set $v(i) = j$
   3: Replace the i^{th} row and j^{th} columns in M with -1.
   4: Repeat.

Now we can extend GALE.

Algorithm 7 GALE: Global Alignment of Local Estimates. Input: problem specific data such as adjacency matrix A for network models, collection of documents $W_{1:D,1:N}$ for topic models; parameters $T, \tau_i, i \in [T]$; a base algorithm $\mathcal{A}$ (e.g., tensor decomposition, variational Bayes, eigenvector-based methods, etc.)

1: **Subset selection:** Given A, choose T subsets $S_1, \ldots, S_T$ of the nodes by some procedure, e.g., select them at random, or select T random nodes and then pick their h-hop neighborhoods (in case of MMBM or biclustering).
2: **Clustering on subsets:** Perform algorithm $\mathcal{A}$ on each of these T subsets to obtain estimated cluster membership matrices $\hat{\Theta}_\ell = \hat{\Theta}_{S_\ell}$. Extend $\hat{\Theta}_\ell$ to an $n \times K$ matrix by setting $(\hat{\Theta}_\ell)_{jk} = 0$ for all $k \in [K]$ if $j \notin S_\ell$.
3: **Traversal of subsets:** Construct a traversal $S_{x_1}, \ldots, S_{x_J}$ through the T subsets.
4: **Initial estimate of Θ:** $\hat{\Theta} \leftarrow \hat{\Theta}_{x_1}$. Also set $\hat{\Theta}^{(x_1)} := \hat{\Theta}_{x_1}$.
5: **Sequential label aligning:** For subset S_{x_i} on the traversal ($i = 1, \ldots, J$), if x_i has not been visited yet,
   (a) Compute the overlap between the current subset with all subsets previously visited, i.e. let $S = S_{x_i} \cap \cup_{\ell=1}^{i-1} S_{x_\ell}$.
   (b) Compute the best permutation to match the clustering $\hat{\Theta}_{x_i}, \hat{\Theta}$ on this set S, i.e. compute $\hat{\Pi}_i = \mathsf{Match}(\hat{\Theta}_{x_i}\big|_S, \hat{\Theta}\big|_S)$.
   (c) Permute the labels of the nodes of S_{x_i} to get an aligned cluster membership matrix $\hat{\Theta}^{(x_i)} \leftarrow \hat{\Theta}_{x_i}\hat{\Pi}$.
   (d) Update $\hat{\Theta}_{jk} \leftarrow \frac{\sum_{\ell \in \{x_1,\ldots,x_i\}} \hat{\Theta}^{(x_\ell)}_{jk} \mathbf{1}_{\{j \in S_{x_\ell}\}}}{\sum_{\ell \in \{x_1,\ldots,x_i\}} \mathbf{1}_{\{j \in S_{x_\ell}\}}} \mathbf{1}_{\{\sum_{\ell \in \{x_1,\ldots,x_i\}} \mathbf{1}_{\{j \in S_{x_\ell}\}} > \tau_i\}}$, for some threshold τ_i.
   (e) Mark S_{x_i} as visited.

Again the discussion in Chapter 2 about construction of paths of subsets also applies here.

3.3 Main results

In this section we will state and discuss our main results on PACE and GALE for mixed memberships. The results will be similar in flavor to those of Chapter 2.

Results on PACE

To prove consistency of PACE we have to assume that the clustering algorithm $\mathcal{A}$ we use has some consistency properties. For example, it will suffice to assume that for a randomly chosen subset S (under our subset selection procedure), $\mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2$[†] is small. The following is our main result on the expected misclustering rate of PACE.

Theorem 3.3.1 (Expected error of PACE)**.** *Let S be a randomly chosen subset of nodes according to our sampling scheme. Then we have*

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \leq \frac{T}{\tau n^2} \times \mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2 + \max_{i,j} \mathbb{P}(N_{ij} < \tau). \tag{3.2}$$

The proof of Theorem 3.3.1 is the same as the proof of Theorem 2.3.1 of Chapter 2 modulo minor changes. We omit the details.

For randomly chosen m-subsets, we have an easy corollary.

Corollary 3.3.1 (Subsets formed by $m \geq m_\star$ randomly chosen nodes)**.** *Let $p = \frac{m(m-1)}{n(n-1)}$, $0 < \theta < 1$ and $\tau = \theta T p$. We have*

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \leq \frac{1}{\theta m(m-1)} \times \mathbb{E}\|\hat{C}^{(S)} - C^{(S)}\|_F^2 + e^{-(1-\theta)^2 Tp/2}. \tag{3.3}$$

Results on GALE

We first prove a deterministic result concerning the correctness of Match. This is similar to Lemma 2.5.1. There, however, the matrix D below was a diagonal matrix.

Lemma 3.3.1 (Correctness of Match)**.** *Let $M = \Pi_2^\top D \Pi_1 + \Gamma$, where $\Pi_i, i = 1, 2$ are permutation matrices, and D, Γ satisfy*

(1) $\max_{a \neq b} D_{ab} \leq \gamma_1 \min_a D_{aa}$,

(2) $\|\Gamma\|_\infty \leq \gamma_2 \min_a D_{aa}$,

[†]The expectation here is taken with respect to the randomness of the sampling procedure and the randomness of the network, given the latent variables Θ.

where $0 < \gamma_1 + 2\gamma_2 < 1$. *Then* Match *applied to* M *returns* $\Pi_2^\top \Pi_1$.

Proof. Let π_i be the permutations corresponding to Π_i. We note that if a, b are such that $\pi_2^{-1}(a) \neq \pi_1^{-1}(b)$, i.e. $(\Pi_2^\top \Pi_1)_{ab} = 0$, then

$$M_{ab} = D_{\pi_2^{-1}(a),\pi_1^{-1}(b)} + \Gamma_{ab} \leq (\gamma_1 + \gamma_2) \min_a D_{aa}.$$

On the other hand, if a, b are such that $\pi_2^{-1}(a) = \pi_1^{-1}(b)$, i.e. $(\Pi_2^\top \Pi_1)_{ab} = 1$, then

$$M_{ab} = D_{\pi_2^{-1}(a),\pi_2^{-1}(a)} + \Gamma_{ab} \geq (1 - \gamma_2) \min_a D_{aa}.$$

Combining we get

$$\min_{a,b:(\Pi_2^\top \Pi_1)_{ab}=1} M_{ab} \geq \left(\frac{1-\gamma_2}{\gamma_1+\gamma_2}\right) \max_{a,b:(\Pi_2^\top \Pi_1)_{ab}=1} M_{ab}.$$

So, if $\frac{1-\gamma_2}{\gamma_1+\gamma_2} > 1$, i.e. if $\gamma_1 + 2\gamma_2 < 1$, then the K largest entries of M appear at positions that map via Π_i to the diagonal entries of $\Pi_2^\top \Pi_1$. Hence the result. □

Suppose we have two conflicting estimates $\hat{\Theta}_i = \Theta\Pi_i + E_i$ of Θ (based on the same set S of nodes). Then their mismatch matrix

$$M = \hat{\Theta}_2^\top \hat{\Theta}_1 = \Pi_2^\top \Theta^T \Theta \Pi_1 + \Gamma,$$

where $\Gamma = \Pi_2^\top \Theta^\top E_1 + E_2^\top \Theta \Pi_1 + E_2^\top E_1$. To align these two estimates, we can apply Match on M. Condition (1) of Lemma 3.3.1 becomes

$$\max_{a \neq b} (\Theta^\top \Theta)_{ab} \leq \gamma_1 \min_a (\Theta^T \Theta)_{aa}. \tag{3.4}$$

In all the models described before, the rows of Θ are sampled i.i.d. from Dirichlet(α). Under a reasonable assumption on α, one can prove that Condition (1) is satisfied with high probability. (We emphasize that such a Dirichlet assumption is not necessary, one only needs (3.4).)

Lemma 3.3.2 (Dirichlet prior)**.** *Suppose that the rows* θ_i *of* Θ *are i.i.d.* Dirichlet(α)*, and for* $\gamma_3 \in (0, 1)$*, we have*

$$(1 + \gamma_3)\alpha_{\max}^2 \leq \gamma_1(\alpha_{\min} + \alpha_{\min}^2). \tag{3.5}$$

Then, with probability at least $1 - \exp(-\Omega_{\alpha,\gamma_1,\gamma_3}(n))$*, one has*

$$\max_{a \neq b} (\Theta^\top \Theta)_{ab} \leq \gamma_1 \min_a (\Theta^T \Theta)_{aa}.$$

In particular, if $\alpha = \eta \mathbf{1}$*, then* (3.5) *translates to* $\eta \leq \frac{\gamma_1}{1-\gamma_1+\gamma_3}$.

Proof. It can be shown by using matrix Bernstein type inequalities that $\Theta^\top\Theta$ concentrates around its expectation $\mathbb{E}(\Theta^\top\Theta) = \frac{n(\mathrm{diag}(\alpha)+\alpha\alpha^\top)}{1+\alpha^\top\mathbf{1}}$ (see, e.g., Lemma A.2 of [48]):

$$\mathbb{P}(\|\Theta^\top\Theta - \mathbb{E}(\Theta^\top\Theta)\| \geq t) \leq K \exp\left(-\frac{t^2}{8n + 4t/3}\right). \tag{3.6}$$

Let us write $\Theta^\top\Theta = \mathbb{E}(\Theta^\top\Theta) + E$, where $\|E\|$ is small with high probability. Then we need

$$\max_{a\neq b} \alpha_a\alpha_b + (1+\alpha^\top\mathbf{1})E_{ab}/n \leq \gamma_1 \min_a(\alpha_a + \alpha_a^2 + (1+\alpha^\top\mathbf{1})E_{ab}/n). \tag{3.7}$$

The LHS of (3.7) $\leq \alpha_{\max}^2 + \frac{1+\alpha^\top\mathbf{1}}{n}\|E\|$, whereas the RHS $\geq \gamma_1(\alpha_{\min} + \alpha_{\min}^2 + \frac{1+\alpha^\top\mathbf{1}}{n}\|E\|)$. Thus a simple sufficient condition is

$$\alpha_{\max}^2 + \frac{1+\alpha^\top\mathbf{1}}{n}\|E\| \leq \gamma_1(\alpha_{\min} + \alpha_{\min}^2 + \frac{1+\alpha^\top\mathbf{1}}{n}\|E\|),$$

which is satisfied if

$$\alpha_{\max}^2 + (1-\gamma_1)\frac{1+\alpha^\top\mathbf{1}}{n}\|E\| \leq \gamma_1(\alpha_{\min} + \alpha_{\min}^2).$$

Now, by the concentration result (3.6), for $\gamma_3 \in (0,1)$,

$$\|E\| \leq \frac{n\gamma_3\alpha_{\max}^2}{(1-\gamma_1)(1+\alpha^\top\mathbf{1})},$$

with probability $\geq 1 - \exp(-\Omega_{\alpha,\gamma_1,\gamma_3}(n))$. Therefore the condition

$$(1+\gamma_3)\alpha_{\max}^2 \leq \gamma_1(\alpha_{\min} + \alpha_{\min}^2) \tag{3.8}$$

suffices. □

In general (e.g., in CTM), one often uses distributions on the simplex other than the Dirichlet, one example being logistic-normal. One can derive sufficient conditions for such distributions as well, by deriving expressions for $\mathbb{E}(\Theta^\top\Theta)$.

As for Condition (2) of Lemma 3.3.1, we can prove, as in the proof of Lemma 2.5.2, that

$$\|\Gamma\|_\infty \leq \|\Gamma\|_1 \leq 2(\|E_1\|_1 + \|E_2\|_1).$$

Thus we will need bounds on the error of the algorithm in use for subsets to control $\|\Gamma\|_\infty$. Under assumptions on the parameters of the model in hand, we can derive analogues of Theorem 2.3.2. Further development of such results is left for future work.

Remark 3.3.1. *For identifiability of MMBM, a necessary condition is to have one* pure *(i.e. a node which only belongs to one cluster) node from each cluster. Therefore it seems that, in order for the algorithm $\mathcal{A}$ we use to produce meaningful and comparable estimates on each subset, we would need one pure node per cluster in each subset, which is an annoying restriction on the subset sampling scheme. Whether this issue can be resolved is another intersting question for future investigation.*

Relation to assignment problem: Note that Match is essentially a special case of the standard Hungarian algorithm [41] for the assignment problem, for cost matrices M which are diagonal up to a permutation of rows/columns. As our results show, Match can work even in cases where the diagonal entries of the cost function are sufficiently larger than the off-diagonal entries. We can substitute Match with any algorithm for the assignment problem, such as the Hungarian algorithm. It would be an interesting topic for future work to verify if the conditions on D can be relaxed by using such an algorithm.

3.4 Discussion

In this chapter, we have extended PACE and GALE to handle mixed memberships. Thus they can be used with topic models, mixed membership blockmodels, models for m-ary mixed memberships, etc. We have discussed which parts of the theoretical arguments from Chapter 2 need to be extended, and worked out a correctness criterion for the alignment algorithm Match. Derivation of problem specific results are left for future work. Another interesting direction would be investigate if divide and conquer has any statistical benefits.

Chapter 4

Clustering network-valued objects

4.1 Introduction

The majority of the works in the community detection literature including those mentioned in the Chapter 1 focus on finding communities among the nodes in *a single network*. While this is still a very important problem with many open questions, there is an emerging need to be able to detect *clusters among multiple network-valued objects*, where a network itself is a fundamental unit of data. This is largely motivated by the routine collection of populations or subpopulations of network-valued data objects. Technological advancement and the explosion of complex data in many domains has made this a somewhat common practice.

There has been some notable work on graph kernels in the Computer Science literature [79, 73]. In these works the goal is to efficiently compute different types of kernel based similarity measures (or their approximations) between networks. In contrast, we ask the following statistical questions. Can we cluster networks *consistently* from a mixture of graphons, when 1) there is node correspondence and 2) when there isn't? The first situation arises, for example, when one has a network evolving over time, or multiple instances of a network between well-defined objects. If one thinks of them as random samples from a mixture of graphons, then can we cluster them? We propose a simple and general algorithm to address this question, which operates by first obtaining a graphon estimate of each of the networks, constructing a distance matrix between those graphon estimates, and then performing spectral clustering on the distance matrix. We call this algorithm Network Clustering based on Graphon Estimates (NCGE).

The second situation arises when one is interested in global properties of a network. This setting is closer to that of graph kernels. Say we have co-authorship networks from Computer Science and High Energy Physics. Are these different types of networks? There has been a lot of empirical and algorithmic work on featurizing networks or computing kernels between networks. But most of these features require expensive computation. We propose a simple feature based on traces of powers of the adjacency matrix for this purpose which is very cheap to compute as it involves only matrix multiplication. We cluster the networks based

on these features and call this method Network Clustering based on Log Moments (NCLM).

We provide some theoretical guarantees for our algorithms in terms of consistency, in addition to extensive simulations and real data examples. The simulation results show that NCGE clearly outperform the naive yet popular method of clustering (vectorized) adjacency matrices in various settings. We also show that, in absence of node correspondence, NCLM is consistently better and faster than methods which featurize networks with different global statistics and graphlet kernels. We also apply NCLM to separate out a mixed bag of real world networks, like co-authorship networks form different domains and ego networks.

4.2 Related work

Our focus is on 1) clustering networks which have node correspondence based on estimating the underlying graphon and 2) clustering networks without node correspondence based on global properties of the networks. In this section we first cite two methods of obtaining graphon estimates, which we will use in our first algorithm. Second, we cite existing work that summarizes a network using different statistics and compares those to obtain a measure of similarity.

A prominent estimator of graphons is the so-called Universal Singular Value Thresholding (USVT) estimator proposed by [25]. The main idea behind USVT is to essentially estimate the low rank structure of the population matrix by thresholding the singular values of the observed matrix at an universal cutoff, and then use retained singular values and the corresponding singular vectors to construct an estimate of the population matrix.

Another recent work [86] proposes a novel, statistically consistent and computationally efficient approach for estimating the link probability matrix by neighborhood smoothing.

Typically, for large networks, USVT is a lot more scalable than the neighborhood-smoothing approach. There are several other methods for graphon estimation, e.g., by fitting a stochastic blockmodel [63]. These methods can also be used in our algorithm.

In [26], a graph-based method for changepoint detection is proposed, where an independent sequence of observations are considered. These are generated i.i.d. under the null hypothesis, whereas under the alternative, after a changepoint, the underlying distribution changes. The goal is to find this changepoint. The observations can be high-dimensional vectors or even networks, with the latter bearing some resemblance with our first framework. This can be viewed as clustering the observations into "past" and "future". We remark here that our graphon-estimation based clustering algorithm suggests an alternative method for changepoint detection in networks, namely by looking at the second eigenvector of the distance matrix between estimated graphons. This will be described more elaborately in Chapter 5. Another related work is due to [30] which aims to extend the classical large sample theory to model network-valued objects.

For comparing global properties of networks, there have been many interesting works that featurize networks, see, for instance, [6]. In the Computer Science literature, graph

kernels have gained much attention [79, 73]. In these works the goal is to efficiently compute different types of kernel based similarity measures (exact or approximate) between networks.

4.3 A framework for clustering networks

Let G be a binary random network or graph with n nodes. Denote by A its adjacency matrix, which is an n by n symmetric matrix with binary entries. That is, $A_{ij} = A_{ji} \in \{0, 1\}, 1 \leq i < j \leq n$, where $A_{ij} = 1$ if there is an observed edge between nodes i and j, and $A_{ij} = 0$ otherwise. All the diagonal elements of A are structured to be zero (i.e. $A_{ii} = 0$). We assume the following random Bernoulli model with

$$A_{ij} \mid P_{ij} \sim \text{Bernoulli}(P_{ij}),\ i < j, \tag{4.1}$$

where $P_{ij} = P(A_{ij} = 1)$ is the probability of link formation between nodes i and j. We denote the link probability matrix as $P = ((P_{ij}))$. The edge probabilities are often modeled using the so-called *graphons*. A graphon f is a nonnegative bounded, measurable symmetric function $f : [0, 1]^2 \to [0, 1]$. Given such an f, one can use the model

$$P_{ij} = f(\xi_i, \xi_j), \tag{4.2}$$

where ξ_i, ξ_j are *i.i.d. uniform random variables* on $(0, 1)$. In fact, any (infinite) exchangeable network arises in this way (by Aldous-Hoover representation [5, 38]).

Our current work focuses on the problem of *clustering networks*. Unlike in a traditional setup, where one observes a single network (with potentially growing number of nodes) and the goal often is to cluster the nodes, here we observe multiple networks and are interested in clustering these networks viewed as fundamental data units.

Node correspondence present

A simple and natural model for this is what we call the *graphon mixture model* for obvious reasons: there are only K (fixed) underlying graphons $f_1, \ldots, f_K$ giving rise to link probability matrices $\Pi_1, \ldots, \Pi_K$ and we observe T networks sampled i.i.d. from the mixture model

$$\mathbb{P}_{mix}(A) = \sum_{i=1}^{K} q_i \mathbb{P}_{\Pi_i}(A), \tag{4.3}$$

where the q_i's are the mixing proportions and $\mathbb{P}_P(A) = \prod_{u<v} P_{uv}^{A_{uv}}(1 - P_{uv})^{1-A_{uv}}$ is the probability of observing the adjacency matrix A when the link probability matrix is given by P. Consider n nodes, and T independent networks $A_i, i \in [T]$, which define edges between these n nodes. We propose the following simple and general algorithm (Algorithm 8) for clustering them.

Algorithm 8 Network Clustering based on Graphon Estimates (NCGE)

1: **Graphon estimation:** Given $A_1, \ldots, A_T$, estimate their corresponding link probability matrices $P_1, \ldots, P_T$ using any one of the "blackbox" algorithms such as USVT ([25]), the neighborhood smoothing approach by [86], etc. Call these estimates $\hat{P}_1, \ldots, \hat{P}_T$.
2: **Forming a distance matrix:** Compute the T by T distance matrix $\hat{D}$ with $\hat{D}_{ij} = \|\hat{P}_i - \hat{P}_j\|_F$, where $\|\cdot\|_F$ is the Frobenius norm.
3: **Clustering:** Apply the spectral clustering algorithm to the distance matrix $\hat{D}$.

We will from now on denote the above algorithm with the different graphon estimation ("blackbox") approaches as follows: the algorithm with USVT as blackbox will be denoted by CL-USVT and the one with the neighborhood smoothing method as blackbox will be denoted by CL-NBS. We will compare these two algorithms with the CL-NAIVE method which does not estimate the underlying graphon, but clusters vectorized adjacency matrices directly (in the spirit of [26]).

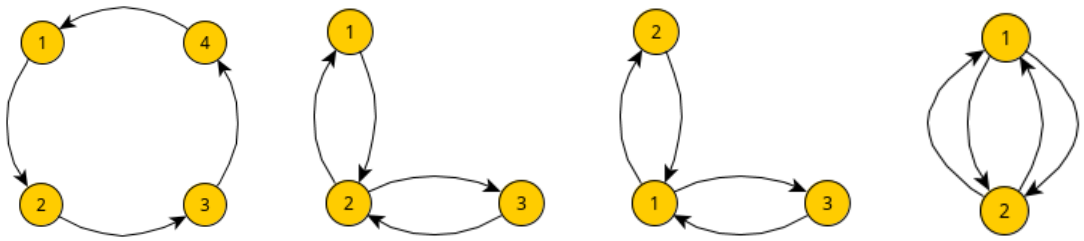
Node correspondence absent

We use certain graph statistics to construct a feature vector. The basic statistics we choose are the trace of powers of the adjacency matrix, suitably normalized and we call them *graph moments*:

$$m_k(A) = \text{trace}(A/n)^k. \tag{4.4}$$

These statistics are related to various path/subgraph counts. For example, $m_2(A)$ is the normalized count of the total number of edges, $m_3(A)$ is the normalized triangle count of A. Higher order moments are actually counts of closed walks (or directed circuits). Figure 4.1 shows the circuits corresponding to $k = 4$.

Figure 4.1: Circuits related to $m_4(A)$.



(A) (B) (C) (D)

The reason we use graph moments instead of subgraph counts is that the latter are quite difficult to compute and present day algorithms work only for subgraphs up to size 5. On the contrary, graph moments are easy to compute as they only involve matrix multiplication.

While it may seem that this is essentially the same as comparing the eigenspectrum, it is not clear how many eigenvalues one should use. Even if one could estimate the number of large eigenvalues using an USVT type estimator, the length is different for different networks. The trace takes into account the relative magnitudes of the eigenvalues naturally. In fact, we tried (see Section 4.5) using the top few eigenvalues as the sole features, but the results were not as satisfactory as using m_k.

We now present our second algorithm (Algorithm 9). In step 2 below we take d to be the standard Euclidean metric.

Algorithm 9 Network Clustering based on Log Moments (NCLM)

1: **Moment calculation:** For each network $A_i, i \in [T]$ and a positive integer J, compute the feature vector $g_J(A_i) := (\log m_1(A_i), \log m_2(A_i), \ldots, \log m_J(A_i))$.
2: **Forming a distance matrix:** For some metric d, set $\hat{D}_{ij} = d(g_J(A_i), g_J(A_j))$.
3: **Clustering:** Apply the spectral clustering algorithm to the distance matrix $\hat{D}$.

Note: The rationale behind taking a logarithm of the graph moments is that if we have two graphs with the same degree density but different sizes, then the degree density will not play any role in the the distance (which is necessary because the degree density will subdue any other differences otherwise). The parameter J counts, in some sense, the effective number of eigenvalues we are using.

4.4 Main Results

We will now mention our main results and discuss some of the consequences. All the proofs and further details are relegated to Section 4.6.

Results on NCGE

We can think of $\hat{D}_{ij}$ as estimating $D_{ij} = \|P_i - P_j\|_F$.

Theorem 4.4.1. *Suppose $D = ((D_{ij}))$ has rank K. Let V (resp. $\hat{V}$) be the $T \times K$ matrix whose columns correspond to the leading K eigenvectors (corresponding to the K largest-in-magnitude eigenvalues) of D (resp. $\hat{D}$). Let $\gamma = \gamma(K, n, T)$ be the K-th smallest eigenvalue value of D in magnitude. Then there exists an orthogonal matrix $\hat{O}$ such that*

$$\|\hat{V}\hat{O} - V\|_F^2 \leq \frac{64T}{\gamma^2} \sum_i \|\hat{P}_i - P_i\|_F^2.$$

Corollary 4.4.1. *Assume for some absolute constants $\alpha, \beta > 0$ the following holds for each $i \in [T]$:*

$$\frac{\|\hat{P}_i - P_i\|_F^2}{n^2} \leq C_i n^{-\alpha} (\log n)^{\beta}, \tag{4.5}$$

either in expectation or with high probability ($\geq 1 - \epsilon_{i,n}$). Then in expectation or with high probability ($\geq 1 - \sum_i \epsilon_{i,n}$) we have that

$$\|\hat{V}\hat{O} - V\|_F^2 \leq \frac{64 C_T T^2 n^{2-\alpha} (\log n)^\beta}{\gamma^2}, \tag{4.6}$$

where $C_T = \max_{i \leq i \leq T} C_i$.

If there are K (fixed, not growing with T) underlying graphons, then the constant C_T does not depend on T. Table 4.1 reports values of α, β for various graphon estimation procedures (under assumptions on the underlying graphons, that are described in Section 4.6).

Table 4.1: Values of α, β for various graphon estimation procedures.

Procedure	USVT	NBS	Minimax rate
α	1/3	1/2	1
β	0	1/2	1

While it is hard to obtain an explicit lower bound on γ in general, let us consider a simple equal weight mixture of two graphons to illustrate the relationship between γ and separation between graphons. Let the distance between the population graphons be dn. Then we have $D = Z \begin{pmatrix} 0 & dn \\ dn & 0 \end{pmatrix} Z^T$, where the i-th row of the binary matrix Z has a single one at position l if network A_i is sampled from Π_l. The nonzero eigenvalues of this matrix are $Tnd/2$ and $-Tnd/2$. Thus, in this case, $\gamma = Tnd/2$. As a result (4.6) becomes

$$\|\hat{V}\hat{O} - V\|_F^2 \leq \frac{256 C_T n^{-\alpha} (\log n)^\beta}{d^2}. \tag{4.7}$$

Let us look at a more specific case of blockmodels with the same number ($= m$) of clusters of equal sizes ($= n/m$) to gain some insight into d. Let C be a $n \times m$ binary matrix of memberships such that $C_{ib} = 1$ if node i within a blockmodel comes from cluster b. Consider two blockmodels $\Pi_1 = CB_1C^T$ with $B_1 = (p-q)I_m + qE_m$ and $\Pi_2 = CB_2C^T$ with $B_2 = (p'-q')I_m + q'E_m$, where I_m is the identity matrix of order m (here the only difference between the models come from link formation probabilities within/between blocks, the blocks remaining the same). In this case

$$d^2 = \frac{\|\Pi_1 - \Pi_2\|_F^2}{n^2} = \frac{1}{m}(p-p')^2 + \left(1 - \frac{1}{m}\right)(q-q')^2.$$

The bound (4.6) can be turned into a bound on the proportion of "misclustered" networks, defined appropriately. There are several ways to define misclustered nodes in the context of community detection in stochastic blockmodels that are easy to analyze with spectral clustering (see, e.g., [71, 43]). These definitions work in our context too. For example, if we

use Definition 4 of [71] and denote by $\mathcal{M}$ the set of misclustered networks, then from the proof of their Theorem 1, we have

$$|\mathcal{M}| \leq 8m_T\|\hat{V}\hat{O} - V\|_F^2,$$

where $m_T = \max_{j=1,\ldots,K}(Z^TZ)_{jj}$ is the maximum number of networks coming from any of the graphons.

Results on NCLM

We first establish concentration of trace(A^k). The proof uses Talagrand's concentration inequality, which requires additional results on Lipschitz continuity and convexity. This is obtained via decomposing $A \mapsto$ trace(A^k) into a linear combination of convex-Lipschitz functions.

Theorem 4.4.2 (Concentration of moments)**.** *Let A be the adjacency matrix of a random graph with link-probability matrix P. Then for any k. Let $\psi_k(A) := \frac{n}{k\sqrt{2}}m_k(A)$. Then*

$$\mathbb{P}(|\psi_k(A) - \mathbb{E}\psi_k(A)| > t) \leq 4\exp(-(t - 4\sqrt{2})^2/16).$$

As a consequence, $g_J(A)$ concentrates around $\bar{g}_J(A) := (\log \mathbb{E}m_2(A), \ldots, \log \mathbb{E}m_J(A))$.

Theorem 4.4.3 (Concentration of $g_J(A)$)**.** *Let $\mathbb{E}A = \rho S$, where $\rho \in (0,1)$, $\min_{i,j} S_{ij} = \Omega(1)$, and $\sum_{i,j} S_{ij} = n^2$. Then $\|\bar{g}_J(A)\| = \Theta(J^{3/2}\log(1/\rho))$, and for any $0 < \delta < 1$ satisfying $\delta J \log(1/\rho) = \Omega(1)$, we have*

$$\mathbb{P}(\|g_J(A) - \bar{g}_J(A)\| \geq \delta J^{3/2}\log(1/\rho)) \leq JC_1e^{-C_2n^2\rho^{2J}}.$$

We expect that $\bar{g}_J$ will be a good population level summary for many models. In general, it is hard to show an explicit separation result for $\bar{g}_J$. However, in simple models, we can do explicit computations to show separation. For example, in a two parameter blockmodel $B = (p-q)I_m + qE_m$, with equal block sizes, we have $\mathbb{E}m_2(A) = (p/m + (m-1)q/m)(1+o(1))$, $\mathbb{E}m_3(A) = (p^3/m^2 + (m-1)pq^2/m^2 + (m-1)(m-2)q^3/6m^2)(1+o(1))$ and so on. Thus we see that if $m = 2$, then $\bar{g}_2$ should be able to distinguish between such blockmodels (i.e. different p, q).

Note: After this work was submitted, we came to know of a concurrent work [49] that provides a topological/combinatorial perspective on the expected graph moments $\mathbb{E}m_k(A)$. Theorem 1 in [49] shows that under some mild assumptions on the model (satisfied, for example, by generalized random graphs with bounded kernels as long as the average degree grows to infinity), $\mathbb{E}$trace$(A^k) = \mathbb{E}$(# of closed k-walks) will be asymptotic to $\mathbb{E}$(# of closed k-walks that trace out a k-cycle) plus $\mathbf{1}_{\{k \text{ even}\}}\mathbb{E}$(# of closed k-walks that trace out a $(k/2+1)$-tree). For even k, if the degree grows fast enough, k-cycles tend to dominate, whereas for sparser graphs trees tend to dominate. From this and our concentration results, we can

expect NCLM to be able to tell apart graphs which are different in terms the counts of these simpler closed k-walks. Incidentally, the authors of [49] also show that the expected count of closed non-backtracking walks of length k is dominated by walks tracing out k-cycles. Thus, if one uses counts of closed non-backtracking k-walks (i.e. moments of the non-backtracking matrix) instead of just closed k-walks as features, one would expect similar performance on denser networks, but in sparser settings it may lead to improvements because of the absence of the non-informative trees in lower order even moments.

4.5 Simulation study and data analysis

In this section, we describe the results of our experiments with simulated and real data to evaluate the performance of NCGE and NCLM. We measure performance in terms of clustering error which is the minimum normalized hamming distance between the estimated label vector and all $K!$ permutations of the true label assignment. Clustering accuracy is one minus clustering error.

Node correspondence present: We provide two simulated data experiments* for clustering networks with node correspondence. In each experiment twenty 150-node networks were generated from a mixture of two graphons, 13 networks from the first and the other 7 from the second. We also used a scalar multiplier with the graphons to ensure that the networks are not too dense. The average degree for all these experiments were around 20-25. We report the average error bars from a few random runs.

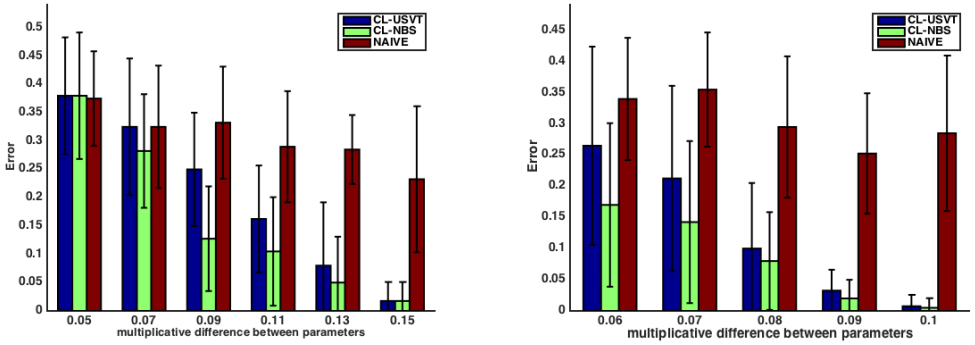
First we generate a mixture of graphons from two blockmodels, with probability matrices $(p_i - q_i)I_m + q_i E_m$ with $i \in \{1, 2\}$. We use $p_2 = p_1(1 + \epsilon)$ and $q_2 = q_1(1 + \epsilon)$ and measure clustering accuracy as the multiplicative error ϵ is increased from 0.05 to 0.15. We compare CL-USVT, CL-NBS and CL-NAIVE and the results are summarized in Figure 4.2(A). We have observed two things. First, CL-USVT and CL-NBS start distinguishing the graphons better as ϵ increases (as the theory suggests). Second, the naive approach does not do a good job even when ϵ increases.

In the second simulation, we generate the networks from two smooth graphons Π_1 and Π_2, where $\Pi_2 = \Pi_1(1 + \epsilon)$ (here Π_1 corresponds to the graphon 3 appearing in Table 1 of [86]). As is seen from Figure 4.2(B), here also CL-USVT and CL-NBS outperform the naive algorithm by a huge margin. Also, CL-NBS is consistently better than CL-USVT, which shows that the accuracy of the graphon estimation procedure is important (for example, USVT is known to perform worse as the network becomes sparser).

Node correspondence absent: We show the efficacy of our approach via two sets of experiments. We compare our log-moment based method NCLM with three other methods. The first is Graphlet Kernels [73] with 3, 4 and 5 graphlets, denoted by GK3, GK4 and

*Code used in this chapter is publicly available at https://github.com/soumendu041/clustering-network-valued-data.

Figure 4.2: Behavior of CL-USVT, CL-NBS and CL-NAIVE when ϵ increases, and the underlying network is generated from (A) a blockmodel, and (B) a smooth graphon.



(A) (B)

GK5 respectively. In the second method, we use six different network-based statistics to summarize each graph; these statistics are the algebraic connectivity, the local and global clustering coefficients [59], the distance distribution [47] for 3 hops, the Pearson correlation coefficient [58] and the rich-club metric [88]. We also compare against graphs summarized by the top J eigenvalues of A/n (TopEig).

The algebraic connectivity is the second smallest eigenvalue of the Laplacian. However, to make this metric free of the size of a graph, we use the second smallest eigenvalue of the *normalized* Laplacian of the largest connected component of a graph. The need for using the largest connected component is that most real graphs without any preprocessing have isolated nodes, or small components. The global clustering coefficient measures the ratio of the number of triangles to the number of connected triplets. In contrast, the local clustering coefficient computes the average of the ratios of the number of triangles connected to a node and the number of triplets centered at that node. The distance distribution for h hops essentially calculates the fraction of all pairs of nodes that are within shortest path or geodesic distance of h hops. Essentially this metric calculates how far a pair of nodes are in a graph on average. The Pearson correlation coefficient of a graph measures the assortativity by computing the correlation coefficient between the degrees of the endpoints of the edges in the graph. Finally, the rich-club metric calculates the edge density of the subgraph induced by nodes with degree above a given threshold. For this metric we chose to use the 0.8-th quantile of the degree sequence of a graph.

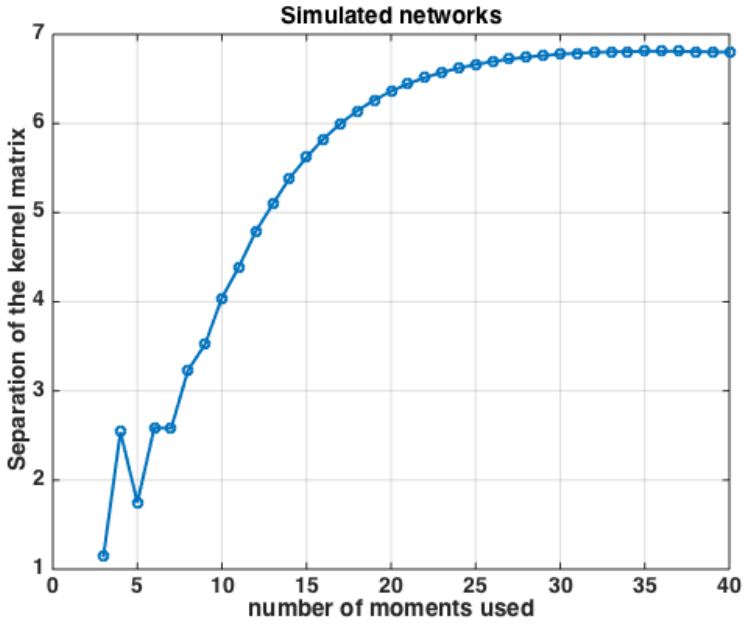
For each distance matrix $\hat{D}$ we compute with NCLM, GraphStats and TopEig, we calculate a similarity matrix $\mathcal{K} = \exp(-t\hat{D})$ where t is chosen as the value, within a range, which maximizes the relative eigengap $(\lambda_K(\mathcal{K}) - \lambda_{K+1}(\mathcal{K}))/\lambda_{K+1}(\mathcal{K})$. It would be interesting to have a data dependent range for t.

For each matrix $\mathcal{K}$ we calculate the top few eigenvectors, say N many, and do K-means on them to get the final clustering. We use $N = K$; however, for GK3, GK4, and GK5, we had to use a smaller N which boosted their clustering accuracy.

First we construct four sets of parameters for the two parameter blockmodel (also known as the planted partition model): $\Theta_1 = \{p = 0.1, q = 0.05, K = 2, \rho = 0.6\}$, $\Theta_2 = \{p = 0.1, q = 0.05, K = 2, \rho = 1\}$, $\Theta_3 = \{p = 0.1, q = 0.05, K = 8, \rho = 0.6\}$, and $\Theta_4 = \{p = 0.2, q = 0.1, K = 8, \rho = 0.6\}$. Note that the first two settings differ only in the density parameter ρ. The second two settings differ in the within and across cluster probabilities. The first two and second two differ in K. For each parameter setting we generate two sets of 20 graphs, one with $n = 500$ and the other with $n = 1000$.

For choosing J, we calculate the moments for a large J; compute a kernel similarity matrix for each choice of J and report the one with largest relative eigengap between the K^{th} and $(K+1)^{th}$ eigenvalue. See Figure 4.3.

Figure 4.3: Tuning for J in the simulated networks: we plot the separation ($= (\lambda_{\mathcal{K}} - \lambda_{\mathcal{K}+1})/\lambda_{\mathcal{K}+1}$) found in the kernel matrix $\mathcal{K}$ against the value of J used in NCLM in our simulation settings.



We see that the eigengap increases and levels off after a point. However, as J increases, the computation time increases, so there is a trade-off. We report the accuracy of $J = 5$, whereas $J = 8$ also returns the same in 48 seconds.

Table 4.2: Clustering error of six different methods on simulated networks.

	NCLM ($J = 5$)	GK3	GK4	GK5	GraphStats ($J = 6$)	TopEig ($J = 5$)
Error	**0**	0.5	0.36	0.26	0.37	0.18
Time (s)	25	14	16	38	94	8

We see that NCLM performs the best. For GK3, GK4 and GK5, if one uses the top two eigenvectors, and clusters those into 4 groups (since there are four parameter settings), the errors are respectively 0.08, 0.025 and 0.03. This means that, for clustering, one needs to estimate the effective rank of the graphlet kernels as well. TopEig performs better than GraphStats, which has trouble separating out Θ_2 and Θ_4.

Note: Intuitively one would expect that, if there is node correspondence between the graphs, clustering based on graphon estimates would work better, because it aims to estimate the underlying probabilistic model for comparison. However, in our experiments we found that a properly tuned NCLM matched the performance of NCGE. This is probably because a properly tuned NCLM captures the global features that distinguish two graphons. We leave it for future work to compare their performance theoretically.

Real networks: We cluster about fifty real world networks. We use 11 co-authorship networks between 15,000 researchers from the High Energy Physics corpus of the arXiv, 11 co-authorship networks with 21,000 nodes from Citeseer (which had Machine Learning in their abstracts), 17 co-authorship networks (each with about 3000 nodes) from the NIPS conference and finally 10 Facebook ego networks[†]. The average degrees vary between 0.2 to 0.4 for co-authorship networks and are around 10 for the ego networks. Each co-authorship network is dynamic, i.e. a node corresponds to an author in that corpus and this node index is preserved in the different networks over time. The ego networks are different in that sense, since each network is the subgraph of Facebook induced by the neighbors of a given central or "ego" node. The sizes of these networks vary between 350 to 4000.

Table 4.3: Clustering error of six different methods on a collection of real world networks consisting of co-authorship networks from Citeseer, High Energy Physics (HEP-Th) corpus of arXiv, NIPS and ego networks from Facebook.

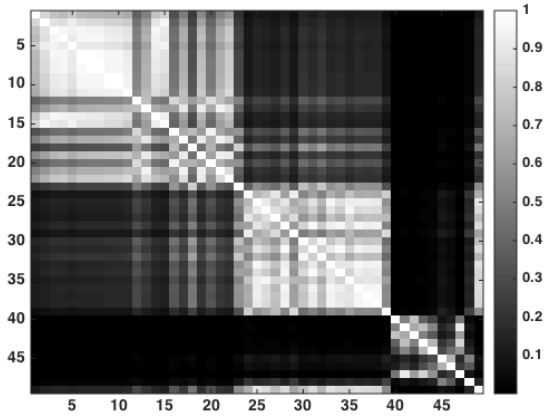
	NCLM ($J = 8$)	GK3	GK4	GK5	GraphStats ($J = 8$)	TopEig ($J = 30$)
Error	0.1	0.6	0.6	0.6	0.16	0.32
Time (s)	2.7	45	50	60	765	14

Table 4.3 summarizes the performance of different algorithms and their running time to compute distance between the graphs. We use the different sources of networks as ground truth labels, i.e. HEP-Th will be one cluster, etc. We explore different choices of J, and see that the best performance is from NCLM, with $J = 8$, followed closely by GraphStats. TopEig (J in this case is where the eigenspectra of the larger networks have a knee) and the graph kernels do not perform very well. GraphStats take 765 seconds to complete, whereas NCLM finishes in 2.7 seconds. This is because the networks are large but extremely sparse, and so calculation of matrix powers is comparatively cheap.

In Figure 4.4, we plot the kernel similarity matrix obtained using NCLM on the real networks (higher the value, more similar the points are). The first 11 networks are from

[†]https://snap.stanford.edu/data/egonets-Facebook.html

Figure 4.4: Kernel matrix for NCLM on 49 real networks.



HEP-Th, whereas the next 11 are from Citeseer. The next 16 are from NIPS and the remaining ones are the ego networks from Facebook. First note that {HEP-Th, Citeseer}, NIPS and Facebook are well separated. However, HEP-Th and Citeseer are hard to separate out. This is also verified by the bad performance of TopEig in separating out the first two (shown in Section 4.5). However, in Figure 4.4, we can see that the Citeseer networks are different from HEP-Th in the sense that they are not as strongly connected inside as HEP-Th.

4.6 Proofs and related discussions

Proofs of results on NCGE

Proposition 4.6.1. *We have*

$$\|\hat{D} - D\|_F^2 \leq 4T \sum_i \|\hat{P}_i - P_i\|_F^2.$$

Proof. The proof is straightforward. By triangle inequality we have

$$|\hat{D}_{ij} - D_{ij}| = \left| \|\hat{P}_i - \hat{P}_j\|_F - \|P_i - P_j\|_F \right| \leq \|\hat{P}_i - P_i\|_F + \|\hat{P}_j - P_j\|_F.$$

Therefore

$$\begin{aligned} \|\hat{D} - D\|_F^2 = \sum_{i,j} |\hat{D}_{ij} - D_{ij}|^2 &\leq \sum_{i,j} (\|\hat{P}_i - P_i\|_F + \|\hat{P}_j - P_j\|_F)^2 \\ &\leq 2 \sum_{i,j} (\|\hat{P}_i - P_i\|_F^2 + \|\hat{P}_j - P_j\|_F^2) = 4T \sum_i \|\hat{P}_i - P_i\|_F^2. \end{aligned}$$

□

Proposition 4.6.2 (Davis-Kahan)**.** *Suppose D has rank K. Let V (resp. $\hat{V}$) be the $T \times K$ matrix whose columns correspond to the leading K eigenvectors (corresponding to the K largest-in-magnitude eigenvalues) of D (resp. $\hat{D}$). Let $\gamma = \gamma(K, n, T)$ be the K-th smallest eigenvalue value of D in magnitude. Then there exists an orthogonal matrix $\hat{O}$ such that*

$$\|\hat{V}\hat{O} - V\|_F \leq \frac{4\|\hat{D} - D\|_F}{\gamma}.$$

Proof. This follows from a slight variant of Davis-Kahan theorem that appears in [83]. Since D is a Euclidean distance matrix of rank K, its eigenvalues must be of the form

$$\lambda_1 \geq \cdots \geq \lambda_u > 0 = \cdots = 0 > \lambda_v \geq \cdots \geq \lambda_n,$$

with $u + n - v + 1 = K$. Applying Theorem 2 of [83] with $r = 1$, $s = u$ we get that if V_+ denotes matrix whose columns are the eigenvectors of D corresponding to $\lambda_1, \ldots, \lambda_u$ and $\hat{V}_+$ denotes the corresponding matrix for $\hat{D}$, then there exists an orthogonal matrix $\hat{O}_+$ such that

$$\|\hat{V}_+\hat{O}_+ - V_+\|_F \leq \frac{2\sqrt{2}\|\hat{D} - D\|_F}{\lambda_u}.$$

Similarly, considering the eigenvalues $\lambda_v, \ldots, \lambda_n$, and applying Theorem 2 of [83] with $r = v$ and $s = n$ we get that

$$\|\hat{V}_-\hat{O}_- - V_-\|_F \leq \frac{2\sqrt{2}\|\hat{D} - D\|_F}{-\lambda_v},$$

where V_-, $\hat{V}_-$ and $\hat{O}_-$ are the relevant matrices. Set $V = [V_+ : V_-]$, $\hat{V} = [\hat{V}_+ : \hat{V}_-]$ and $\hat{O} = \begin{pmatrix} \hat{O}_+ & 0 \\ 0 & \hat{O}_- \end{pmatrix}$. Then note that the columns of V are eigenvectors of D corresponding to its K largest-in-magnitude eigenvalues, that O is orthogonal and also that $\gamma = \min\{\lambda_u, -\lambda_v\}$. Thus

$$\begin{aligned} \|\hat{V}\hat{O} - V\|_F^2 &= \|\hat{V}_+\hat{O}_+ - V_+\|_F^2 + \|\hat{V}_-\hat{O}_- - V_-\|_F^2 \\ &\leq \frac{8\|\hat{D} - D\|_F^2}{\lambda_u^2} + \frac{8\|\hat{D} - D\|_F^2}{\lambda_v^2} \\ &\leq \frac{16\|\hat{D} - D\|_F^2}{\gamma^2}, \end{aligned}$$

which is the desired bound. □

Proof of Theorem 4.4.1. Follows immediately from Propositions 4.6.1-4.6.2. □

Proposition 4.6.3. *Suppose there are K underlying graphons, as in the graphon mixture model (4.3), and assume that in our sample there is at least one representative from each of these. Then D has the form $Z\mathcal{D}Z^T$ where the ith row of the binary matrix Z has a single*

one at position l if network A_i is sampled from Π_l, and $\mathcal{D}$ is the $K \times K$ matrix of distances between the Π_l. As a consequence D is of rank K. Then there exists a $T \times K$ matrix V whose columns are eigenvectors of D corresponding to the K nonzero eigenvalues, such that

$$V_{i\star} = V_{j\star} \Leftrightarrow Z_{i\star} = Z_{j\star},$$

so that knowing V, one can recover the clusters perfectly.

Proof. The proof is standard. Note that Z^TZ is positive definite. Consider the matrix $(Z^TZ)^{1/2}B(Z^TZ)^{1/2}$ and let $U\Delta U^T$ be its spectral decomposition. Then the matrix $V = Z(Z^TZ)^{-1/2}U$ has the required properties. □

USVT: Theorem 2.7 of [25] tells us that if the underlying graphons are Lipschitz then $\mathbb{E}\frac{\|\hat{P}_i - P_i\|_F^2}{n^2} \leq C_i n^{-1/3}$, where the constant C_i depends only on the Lipschitz constant of the underlying graphon f_i. So, the condition (4.5) of Corollary 4.4.1 is satisfied with $\alpha = 1/3$, $\beta = 0$.

Neighborhood smoothing: The authors of [86] work with a class $\mathcal{F}_{\delta,L}$ of piecewise Lipschitz graphons, see Definition 2.1 of their paper. The proof of Theorem 2.2 of [86] reveals that if $f_i \in \mathcal{F}_{\delta_i,L_i}$, then there exist a global constant C and constants $C_i \equiv C_i(L_i)$, such that for all $n \geq N_i \equiv N_i(\delta_i)$, with probability at least $1-n^{-C}$, we have $\frac{\|\hat{P}_i - P_i\|_F^2}{n^2} \leq C_i\sqrt{\frac{\log n}{n}}$. Thus the condition (4.5) of Corollary 4.4.1 is satisfied with $\alpha = \beta = 1/2$, for all $n \geq N_T := \max_{1\leq i\leq T} N_i$.

Remark 4.6.1. *In the case of the graphon mixture model (4.3), the constants C_T and N_T will be free of T as there are only K (fixed) underlying graphons. Also, if each $f_i \in \mathcal{F}_{\delta,L_i}$, then N_T will not depend on T and if the Lipschitz constants are the same for each graphon, then C_T will not depend on T.*

Remark 4.6.2. *There are combinatorial algorithms that can achieve the minimax rate, $n^{-1}\log n$, of graphon estimation [28]. Albeit impractical, these algorithms can be used to achieve the optimal bound of $4C_Tn^{-1}\log n$ in Proposition 4.6.1.*

Remark 4.6.3. *We do not expect* CL-NAIVE *to perform well simply because A is not a good estimate of P in Frobenius norm in general. Indeed,*

$$\frac{1}{n^2}\mathbb{E}\|A-P\|_F^2 = \frac{1}{n^2}\sum_{i,j}\mathbb{E}(A_{ij}-P_{ij})^2 = \frac{1}{n^2}\sum_{i\neq j}P_{ij}(1-P_{ij}) + \frac{1}{n^2}\sum_i P_{ii}^2 \leq \frac{1}{4}(1+o(1)),$$

with equality, for example, when each $P_{ij} = \frac{1}{2} + o(1)$.

Proofs of results on NCLM

Proposition 4.6.4 (Lipschitz continuity)**.** *Suppose A and $A+U$ are matrices with entries in $[-1,1]$, then*

1. $|trace((A+U)^k_+) - trace(A^k_+)| \leq kn^{k-1}\|U\|_F$,
2. $|trace((A+U)^k_-) - trace(A^k_-)| \leq kn^{k-1}\|U\|_F$,

i.e. the map $\phi_{+,k} : S^{n\times n}_{[-1,1]} \to [0,\infty)$ *defined on the space* $S^{n\times n}_{[-1,1]}$ *of symmetric matrices with entries in* $[-1,1]$ *by* $\phi_{+,k}(A) = trace(A^k_+)$ *is Lipschitz with constant* kn^{k-1}. *Note that this implies that the map* $\tilde{\phi}_{+,k} : [0,1]^{n(n-1)/2} \to [0,\infty)$ *defined by*

$$\tilde{\phi}_{+,k}((a_{ij})_{1\leq i<j\leq n}) = trace(A^k_+),$$

where $A_{ij} = A_{ji} = a_{ij}$, *for* $1 \leq i < j \leq n$ *and* $A_{ii} = 0$, *is Lipschitz with constant* $\sqrt{2}kn^{k-1}$. *The same statements hold for analogously defined* $\phi_{-,k}$ *and* $\tilde{\phi}_{-,k}$.

Proof. Let $\lambda_1 \geq \cdots \geq \lambda_n$ be the ordered eigenvalues of $A+U$, whereas $\nu_1 \geq \cdots \geq \nu_n$ be the ordered eigenvalues of A. Let $\sigma_1 \geq \cdots \geq \sigma_n$ be the ordered eigenvalues of U. First note that since these matrices have entries in $[-1,1]$, their Frobenius norm is at most n. Thus all their eigenvalues are in $[-n,n]$.

We now compute the derivative of $\text{trace}A^k$ with respect to a particular variable A_{ij}. We claim that

$$\frac{d}{dA_{ij}}\text{trace}A^k = 2kA^{k-1}_{ij}.$$

To do this we shall work with the linear map interpretation of derivative (in this case multiplication by a number). First consider the map $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined as $f(A) = A^k$. Then consider the map $g : \mathbb{R}^{n\times n} \to \mathbb{R}$ defined by $g(A) = \text{trace}(A)$. Finally consider the map $h : \mathbb{R} \to \mathbb{R}^{n\times n}$ defined as $h(x) = A + (x - A_{ij})e_ie_j^T + (x - A_{ij})e_je_i^T$. Now we view the function $\text{trace}A^k$ for A symmetric as a function of A_{ij} as $g \circ f \circ h(A_{ij})$. Therefore, by the chain rule

$$\frac{d}{dA_{ij}}\text{trace}A^k(u) = D_{f\circ h(A_{ij})}g \circ D_{h(A_{ij})}f \circ D_{A_{ij}}h(u).$$

Now g is a linear function. Therefore $D_Ag = g$. On the other hand, it is easy to see that $D_{A_{ij}}h(u) = ue_ie_j^T + ue_je_i^T$. Finally f can be viewed as the composition of two maps $\alpha(A_1,\ldots,A_k) = A_1A_2\cdots A_k$ and $\beta(A) = (A,\ldots,A)$. Notice that $D_{(A_1,\ldots,A_k)}\alpha(H_1,\ldots,H_k) = H_1A_2\cdots A_k + A_1H_2\cdots A_k + \ldots + A_1A_2\cdots H_k)$, and $D_A\beta = \beta$. Thus

$$D_Af(H) = D_{\beta(A)}\alpha \circ D_A\beta(H) = D_{\beta(A)}\alpha(H,\ldots,H) = HA^{k-1} + \cdots + HA^{k-1} = kHA^{k-1}.$$

Therefore

$$\frac{d}{dA_{ij}}\text{trace}A^k(u) = g \circ D_{h(A_{ij})}f(ue_ie_j^T + ue_je_i^T).$$

Noting that $h(A_{ij}) = A$ we get

$$\frac{d}{dA_{ij}}\text{trace}A^k(u) = g(k(ue_ie_j^T + ue_je_i^T)A^{k-1}) = 2kA^{k-1}_{ij}u.$$

Therefore the map $\tilde{\phi}_k : \mathbb{R}^{n(n-1)/2} \to \mathbb{R}$ defined by $\tilde{\phi}_k((A_{ij})_{i<j}) \equiv \tilde{\phi}_k(A) = \text{trace}(A^k)$ has gradient $2k(A_{ij}^{k-1})_{i<j}$.

Therefore

$$|\tilde{\phi}_k(A) - \tilde{\phi}_k(B)| \leq \|\nabla\tilde{\phi}_k\|_2 \|(A_{ij}) - (B_{ij})\|_2.$$

But $\|\nabla\tilde{\phi}_k\|_2 = \sqrt{2}k\|A^{k-1}\|_F \leq \sqrt{2}kn^{k-1}$ (by repeated application of the inequality $\|XY\|_F \leq \|X\|_F\|Y\|_{op}$ and noting that $\|A\|_F \leq n$), and $\|(A_{ij}) - (B_{ij})\|_2 = \|A - B\|_F/\sqrt{2}$. Therefore

$$|\tilde{\phi}_k(A) - \tilde{\phi}_k(B)| \leq kn^{k-1}\|A - B\|_F.$$

Also as before

$$\begin{aligned} |\text{trace}(A+U)_+^k - \text{trace}A_+^k| &= |\tilde{\phi}_k((A+U)_+) - \tilde{\phi}_k(A_+)| \\ &\leq kn^{k-1}\|(A+U)_+ - A_+\|_F \\ &\leq kn^{k-1}\|U\|_F, \end{aligned}$$

where in the last step we have used the fact that $A \mapsto A_+$ is projection onto the PSD cone and hence non-expansive (i.e. 1-Lipschitz). Part 2. may now be obtained easily by noting that $A_- = (-A)_+$. □

Proposition 4.6.5 (Convexity)**.** *The functions $\phi_{\pm,k}$ and $\tilde{\phi}_{\pm,k}$ are convex on their respective domains.*

Proof. We recall the standard result that if a continuous map $t \mapsto f(t)$ is convex, so is $A \mapsto \text{trace}f(A)$ on the space of Hermitian matrices, and it is strictly convex if f is strictly convex (See, for example, Theorem 2.10 of [23]). To use this we note that $x \mapsto x_+^k$ is continuous and convex, and so is $x \mapsto x_-^k$. This establishes convexity of $\phi_{\pm,k}$. Convexity of $\tilde{\phi}_{\pm,k}$ is an immediate consequence. □

Proof of Theorem 4.4.2. The idea is to use Talagrand's concentration inequality for convex-Lipschitz functions (cf. [19], Theorem 7.12). First note that for k even, we have

$$\psi_k(A) = \psi_{+,k}(A) + \psi_{-,k}(A)$$

and for k odd

$$\psi_k(A) = \psi_{+,k}(A) - \psi_{-,k}(A),$$

where

$$\psi_{\pm,k}(A) = \frac{1}{\sqrt{2}kn^{k-1}}\tilde{\phi}_{\pm,k}(A).$$

Viewed as a map from $[0,1]^{n(n-1)/2}$ to $[0,\infty)$, both $\psi_{\pm,k}$ are convex, 1-Lipschitz. Therefore, by Talagrand's inequality,

$$\mathbb{P}(|\psi_{\pm,k}(A) - \mathbb{M}\psi_{\pm,k}(A)| > t) \leq 2\exp(-t^2/4),$$

where $\mathbb{M}\psi_{\pm,k}(A)$ is a median of $\psi_{\pm,k}(A)$. By Exercise 2.2 of [19], we have

$$|\mathbb{M}\psi_{\pm,k}(A) - \mathbb{E}\psi_{\pm,k}(A)| \leq 2\sqrt{2},$$

which implies that

$$\mathbb{P}(|\psi_{\pm,k}(A) - \mathbb{E}\psi_{\pm,k}(A)| > t) \leq 2\exp(-(t-2\sqrt{2})^2/4).$$

Therefore

$$\begin{aligned}\mathbb{P}(|\psi_k(A) - \mathbb{E}\psi_k(A)| > t) &\leq \mathbb{P}(|\psi_{+,k}(A) - \mathbb{E}\psi_{+,k}(A)| > t/2) + \mathbb{P}(|\psi_{-,k}(A) - \mathbb{E}\psi_{-,k}(A)| > t/2) \\ &\leq 4\exp(-(t-4\sqrt{2})^2/16),\end{aligned}$$

as desired. □

Proposition 4.6.6 (Order of expectation)**.** *Let* $\mathbb{E}A = P = \rho S$*, where* $\rho \in (0,1)$*,* $\min_{i,j} S_{ij} = \Omega(1)$*, and* $\sum_{i,j} S_{ij} = n^2$*. Then,*

$$\rho^k \preceq \mathbb{E}m_k(A) \preceq \rho^{k-1}.$$

Proof. Note that

$$\operatorname{trace}(A^k) = \sum_{i_1,i_2,\ldots,i_k} A_{i_1 i_2} A_{i_2 i_3} \cdots A_{i_k i_1}.$$

Since A_{ij}'s are Bernoulli random variables, Letting $P_\star := \min_{ij} P_{ij}$ and $P_\# := \max P_{ij}$, we see that

$$P_\star^\ell \leq \mathbb{E}A_{i_1 i_2} A_{i_2 i_3} \cdots A_{i_k i_1} \leq P_\#^\ell,$$

where $1 \leq \ell \leq k$ is the number of distinct sets among $\{i_1, i_2\}, \{i_2, i_3\}, \ldots, \{i_k, i_1\}$. We call ℓ the weight of the sequence $i_1, \ldots, i_k$. We can easily see that the total number of sequences is bounded by n^k, and the number of sequences with weight ℓ, call it $N(\ell; k, n)$, is bounded above by $n^{\ell+1}$. In fact,

$$N(k; k, n) = n(n-1)(n-2)^{k-3}(n-3) \asymp n^k.$$

We thus have

$$\sum_{\ell=1}^{k} N(\ell; k, n) P_\star^\ell \leq \mathbb{E}\operatorname{trace}(A^k) \leq \sum_{\ell=1}^{k} N(\ell; k, n) P_\#^\ell.$$

This gives us trivial upper and lower bounds ($C_1, C_2 > 0$ are absolute constants whose values are adjusted as necessary)

$$\begin{aligned}C_1 n^k \rho^k \leq C_1 n^k P_\star^k \leq N(k; k, n) \leq \mathbb{E}\operatorname{trace}(A^k) &\leq \sum_{\ell=1}^{k-1} n^{\ell+1} P_\#^\ell + n^k P_\#^k \\ &= n\frac{(nP_\#)^k - (nP_\#)}{nP_\# - 1} + n^k P_\#^k \\ &\leq C_2 n^k {P_\#}^{k-1} \leq C_2 n^k \rho^{k-1}.\end{aligned}$$

This completes the proof. □

In the following, we will again use $C_1, C_2 > 0$ as absolute constants whose values may change from line to line.

Proof of Theorem 4.4.3. First of all, by Proposition 4.6.6 we have

$$|\log \mathbb{E}m_k(A)| = \Theta(k \log(1/\rho)),$$

from which we conclude that $\|\bar{g}_J(A)\| = \Theta(J^{3/2} \log(1/\rho))$.

Writing $\mu_k = \mathbb{E}m_k(A)$, and using Theorem 4.4.2, we get

$$\begin{aligned}
\mathbb{P}(|\log m_k(A) - \log \mu_k| > t) &= \mathbb{P}(\frac{m_k}{\mu_k} - 1 > e^t - 1) + \mathbb{P}(\frac{m_k}{\mu_k} - 1 < -(1 - e^{-t})) \\
&\leq \mathbb{P}(|m_k - \mu_k| > (e^t - 1)\mu_k) + \mathbb{P}(|m_k - \mu_k| > (1 - e^{-t})\mu_k) \\
&\leq C_1 e^{-C_2 \frac{n^2 \mu_k^2 (e^t - 1)^2}{k^2}} + C_1 e^{-C_2 \frac{n^2 \mu_k^2 (1 - e^{-t})^2}{k^2}} \\
&\leq C_1 e^{-C_2 \frac{n^2 \rho^{2k} (e^t - 1)^2}{k^2}} + C_1 e^{-C_2 \frac{n^2 \rho^{2k} (1 - e^{-t})^2}{k^2}},
\end{aligned}$$

where in the last line we have used Proposition 4.6.6. Using this along with an union bound, we get

$$\begin{aligned}
\mathbb{P}(\|g_J(A) - \bar{g}_J(A)\| \geq t) &\leq \sum_{k=2}^{J} \mathbb{P}(|\log m_k(A) - \log \mathbb{E}m_k(A)| > \frac{t}{\sqrt{J}}) \\
&\leq \sum_{k=2}^{J} C_1 e^{-C_2 \frac{n^2 \rho^{2k} (e^{t/\sqrt{J}} - 1)^2}{k^2}} + C_1 e^{-C_2 \frac{n^2 \rho^{2k} (1 - e^{-t/\sqrt{J}})^2}{k^2}}.
\end{aligned}$$

Choosing $t = \delta J^{3/2} \log(1/\rho)$, where $\delta J \log(1/\rho) = \Omega(1)$, we see that $e^{t/\sqrt{J}} - 1 = \rho^{-\delta J} - 1 = \Omega(\rho^{-\delta J})$ and $1 - e^{-t/\sqrt{J}} = 1 - \rho^{\delta J} = \Omega(1)$. Also note that $\rho^{2k}/k^2 \geq \rho^{2J}/4$. Therefore we have

$$\begin{aligned}
\mathbb{P}(\|g_J(A) - \bar{g}_J(A)\| \geq \delta J^{3/2} \log(1/\rho)) &\leq J(C_1 e^{-C_2 n^2 \rho^{2J} \rho^{-2\delta J}} + C_1 e^{-C_2 n^2 \rho^{2J}}) \\
&\leq J C_1 e^{-C_2 n^2 \rho^{2J}},
\end{aligned}$$

which completes the proof. □

4.7 Discussion

We consider the problem of clustering network-valued data for two settings, both of which are prevalent in practice. In the first setting, different network objects have node correspondence. This includes clustering brain networks obtained from FMRI data where each node corresponds to a specific region in the brain, or co-authorship networks between a set of authors where the connections vary from one year to another. In the second setting, node correspondence is not present, e.g., when one wishes to compare different types of networks:

co-authorship networks, Facebook ego networks, etc. One may be interested in seeing if co-authorship networks are more "similar" to each other than ego or friendship networks.

We present two algorithms for these two settings based on a simple general theme: summarize a network into a possibly high dimensional feature vector and then cluster these feature vectors. In the first setting, we propose NCGE, where each network is represented using its graphon-estimate. We can use a variety of graphon estimation algorithms for this purpose. We show that if the graphon estimation is consistent, then NCGE can cluster networks generated from a finite mixture of graphons in a consistent way, if those graphons are sufficiently different. In the second setting, we propose to represent a network using an easy-to-compute summary statistic, namely the vector of the log-traces of the first few powers of a suitably normalized version of the adjacency matrix. We call this method NCLM and show that the summary statistic concentrates around its expectation, and argue that this expectation should be able to separate networks generated from different models. Using simulated and real data experiments we show that NCGE is vastly superior to the naive but often-used method of comparing adjacency matrices directly, and NCLM outperforms most computationally expensive alternatives for differentiating networks without node correspondence. In conclusion, we believe that these methods will provide practitioners with a powerful and computationally tractable tool for comparing network-structured data in a range of disciplines.

Chapter 5

Changepoint detection

5.1 Introduction

In this chapter, we tackle the problem of changepoint detection in temporal network data, that is one observes a series of networks indexed by time and wishes the check if there is a timepoint (so-called changepoint) when there is a significant change in the structure of these networks. Potential applications are in, for instance, brain imaging, where one has brain scans of individuals collected over time and is looking for abnormalities, ecological networks observed over time, where one wonders if there is a structural change, and so on.

Note: We stress here that we observe the whole time series ahead of our analysis, this is thus an *offline* or *a posteriori* changepoint problem. We will not discuss the online version of the problem here, which is also quite interesting.

5.2 Related work

Changepoint detection is a classical problem in statistics going all the way back to the early days of statistical quality control [65, 66, 31]. There is a huge literature on the univariate changepoint problem. An excellent treatment can be found in the book [20].

The multivariate versions of the problem are significantly more complex. Some notable works are [84, 74, 76, 39] in the parametric setting, and [35, 46, 26] in the non-parametric setting.

There has been some recent works on the problem of network changepoints. For example, [68] postulate a hierarchical random graph model and use a Bayesian procedure to detect changepoints. [67] use local graph statistics for changepoint and/or anomaly detection in dynamic networks. For a survey of techniques used in the related problem of anomaly detection in graphs, see [70]. Most two sample graph tests can be used for the changepoint problem viewed as a multiple testing problem. For example, an eigenvalue based test for the ER vs SBM problem is worked out in [22]. Although much empirical work has been done,

not much theory can be found, and most theoretical results focus on particular structures or specialized models. An exception is [72], where the authors model networks as a Markov random field and estimate the changepoint using a penalized pseudo-likelihood and prove its consistency at a near classical (i.e. fixed dimensional) rate under a restricted strong convexity type assumption on the log-pseudo-likelihood. Although their results are in a high-dimensional setting, and allow more complicated node interaction than random graphs with independent edges, the role of network sparsity in their setup is not clear.

The classical CUSUM statistic [66] for univariate changepoint problems can be used in the network problem as well, and provides a unified way of constructing estimates of changepoints. It is also amenable to theoretical analysis because of the averaging structure present. In this chapter, we will investigate its theoretical properties in a quite general setup.

5.3 Setup and methodology

Single changepoint

Suppose we observe T networks $A_1, \ldots, A_T$ over time on the same set of nodes $1, \ldots, n$, where the edges in A_i are independent and $\mathbb{E}A_i = P_i$[†]. A natural problem is whether we can tell if structural properties of these networks change over time. Let $\xi_i = \xi(P_i)$ denote a (population level) property of interest of the i-th network, e.g., the expected number of copies of a subgraph of certain type, the average connectivity, or the full distribution itself. We wish to test if the ξ_i's change over time. The simplest such problem would be to test against a single changepoint alternative

$$H_0 : \xi_i = \xi, 1 \le i \le T,$$

versus

$$H_1 : \exists\ 1 \le \tau \le T-1 \text{ such that } \xi_i = \begin{cases} \xi_1 & 1 \le i \le \tau \\ \xi_2 & \tau + 1 \le i \le T. \end{cases}$$

We now discuss some natural statistics for this problem. The excellent monograph [20] contains a thorough overview of the first two in the univariate changepoint setting. We will not deal with the testing problem here, and focus on estimating τ under the alternative.

CUSUM statistic

A natural statistic to consider under our single changepoint alternative is based on the cumulative averages of estimates of the property of interest. Such CUSUM statistics are

[†]As for the mutual dependence of the A_i's, we will assume their independence to prove concentration bounds, this can possibly be relaxed to include, e.g., Markovian or martingale type dependence (results other than the concentration bounds do not require any assumption on the mutual dependence of these networks).

very widely used in changepoint problems. Let $\hat{\xi}_i = \hat{\xi}_i(A_i)$ be an estimate of the property ξ_i based on the observed network A_i. Define, for $0 \le \zeta \le 1$ and $1 \le t \le T-1$,

$$G_t = G_t^\zeta := \left(\frac{t(T-t)}{T^2}\right)^\zeta \left(\frac{1}{t}\sum_{i=1}^{t} \hat{\xi}_i - \frac{1}{T-t}\sum_{i=t+1}^{T} \hat{\xi}_i\right).$$

If there is a changepoint at $1 \le a \le \tau \le b \le T-1$, one expects that $Z_{a,b} := \max_{a \le t \le b} \theta_n \|G_t\|$ would be large and the maximizer would be close to τ. Here $\|\cdot\|$ is some appropriate norm and θ_n is a scalar parameter whose purpose is to normalize the statistic according to size of the ξ_i's. In this work, we will take $\zeta = 0$ for simplicity of exposition, the general case only differs in terms of constants and may be handled similarly. We mention in passing that if the $\hat{\xi}_i$ were i.i.d. $\mathcal{N}(\theta, 1)$ where $\theta = \theta_0 \mathbf{1}_{1 \le i \le \tau} + \theta_1 \mathbf{1}_{\tau+1 \le i \le T}$, then $\arg\max_{1 \le t \le T-1} (G_t^{1/2})^2$ would be the maximum likelihood estimator (MLE) of τ. In many univariate settings, such CUSUM statistics are minimax optimal (see, e.g., [20]).

Likelihood ratio (LR) statistic

Another widely used statistic in the univariate changepoint literature is the likelihood ratio (LR) statistic. When we test for a changepoint at t, this amounts to calculating

$$L_t = \frac{\sup_{\mathbf{P} \in \mathcal{P}_{H_1}} L_{\mathbf{P}}(A_i, 1 \le i \le T)}{\sup_{\mathbf{P} \in \mathcal{P}_{H_0}} L_{\mathbf{P}}(A_i, 1 \le i \le T)}, \tag{5.1}$$

where $\mathbf{P} = (P_1, \ldots, P_T)$, and $\mathcal{P}_{H_i}$ denotes the parameter space for H_0 and H_1. For testing against an unknown changepoint, one then would look at $\sup_{a \le t \le b} L_t$. For our graph problem, however, it is quite difficult to even compute the LR statistic, as the alternative involves a multitude of parameters P_{ij}, Q_{ij}, whose maximum likelihood estimates are all linked together by τ. Thus optimization of the profile likelihood (i.e. likelihood after maximizing over P, Q) over τ (which would then produce the maximum likelihood estimator of τ) is not an easy task and hence will not be considered here.

A method inspired by the clustering problem of Chapter 4

Under the single changepoint hypothesis, there are two clusters of values: pre-change and post-change. So we can cluster the $\hat{\xi}_i$ using some off-the-shelf clustering algorithm and then look at the clusters to pinpoint the location of the change. Spectral clustering is a natural choice here, because the second eigenvector of the kernel matrix will reveal if there is a changepoint or not. More elaborately, let K be a kernel matrix computed from the $\hat{\xi}_i$, e.g., $K_{ij} = \exp(-c\theta_n \|\hat{\xi}_i - \hat{\xi}_j\|^2)$. Then in absence of any changepoint the second eigenvector of the corresponding Laplacian matrix is expected to be flat, while under the single changepoint model, it is expected to be a step function with a jump at the value of the changepoint. If v is the second eigenvector, then one can use univariate changepoint detection methods on

v to locate the changepoint. A naive estimator of τ would be

$$\hat{\tau} = \arg \max_{1 \le i \le T-1} |v_i - v_{i+1}|.$$

Theoretical investigation of this estimator is, however, not easy, as it would require quite accurate estimates for the second eigenvector of a kernel random matrix. We leave this as a direction for future work.

Differences from usual univariate changepoint detection

If one is interested in testing changes in an univariate statistic $\xi(P)$ (such as density of triangles), one can certainly use univariate changepoint detection methods. However, there is a significant difference between such a graph statistic and, say, a normal mean. This should be intuitively clear, because a graph has about n^2 independent observations, and often possesses low rank structures (e.g., communities), and thus a single realization of a graph alone can produce very good estimates of the underlying parameters, which can hardly be said about a univariate normal mean. This phenomenon can be helpful in mitigating edge effects, and can significantly reduce requirements on the minimum separation of the pre and post change distributions necessary for univariate changepoint detection. To give a very simplistic example, if we know that our networks are all Erdös-Rényi, then the corresponding parameter p can be estimated consistently from a single observation of the network, and so we can detect changes even in a series of two networks. Thus it seems worthwhile to derive asymptotics involving both the network sizes (and sparsity) and the length of the time series.

Multiple changepoints

If we believe that there are multiple changepoints, a local version of a single changepoint detection method may be put to use. Fixing a width parameter γ, we may scan intervals of length γ for changepoints, i.e. we may maximize

$$G_{\gamma,t,t+s} := \frac{1}{s} \sum_{i=t+1}^{t+s} \hat{\xi}_i - \frac{1}{\gamma - s} \sum_{i=t+s+1}^{t+\gamma} \hat{\xi}_i,$$

over $s \in [t, t+\gamma]$, for a range of values of t. A number of issues arise, such us how to choose γ in a data dependent way, how to choose the range of t and so on. Again we leave these for future work.

Nodes are changing or node correspondence not available

If nodes are changing, then we cannot use the graph based method as described above. Also, we must make it clear what do we mean by $A_i \sim \mathcal{M}_0$ in this case. Perhaps in this case, we would be more interested in finding out whether some characteristic (such as normalized subgraph counts) changes over time. Another statistic of interest is the vector of a certain

number of normalized traces of powers of the adjacency matrix, which we used for NCLM in Chapter 4.

5.4 Main results

Consider the property $\xi(P) = P$, i.e. the distribution itself. The CUSUM estimate of τ is

$$\hat{\tau} = \arg\max_{a \le t \le b} \theta_n \|G_t\|,$$

where $\|\cdot\|$ is some appropriate matrix norm (such as the Frobenius norm or the operator norm; if we use the Frobenius norm, we will denote the resulting estimate as $\hat{\tau}_F$, and as $\hat{\tau}_{op}$, for the operator norm), $\theta_n \asymp \|P_i\|^{-1} \asymp (n\rho_n)^{-1}$, and $1 \le a \le b \le T-1$ are some known lower and upper bounds on the changepoint. As for $\hat{P}_i$, we may use A_i itself, or compute some estimate from A_i (for example, neighborhood smoothing [86], blockmodel approximation [2], USVT [25], etc.). Here ρ_n is a measure of the sparsity of the network in that $n\rho_n$ scales like the average degree. We will assume that all the networks are in the same scale of sparsity, because otherwise comparisons can be made just on the basis of the average degree. Moreover, we will assume that the expected degrees of vertices are uniform, i.e. $\min_{k,l}(P_i)_{kl} \asymp \max_{k,l}(P_i)_{kl} \asymp \rho_n$. Our goal will be to prove consistency of $\hat{\tau}$ and obtain asymptotics in n, ρ_n, and T. In practice, ρ_n is not known, but we can estimate $n\rho_n$ by the average of the average degrees of all networks.

In the following, we assume that $a = \alpha_0 T$, $b = \alpha_1 T$, for known $0 < \alpha_0 < \alpha_1 < 1$, and $\tau = \lceil \alpha T \rceil$, for some $\alpha \in [\alpha_0, \alpha_1]$. Let $\gamma = \gamma(\alpha_0, \alpha_1) := \min\{\alpha_0, 1 - \alpha_1\}$. This is a measure of the edge effect. Let P denote the expected adjacency matrix before time τ, and let Q denote the expected adjacency matrix post time τ. Define the population version of G_t as

$$\bar{G}_t := \begin{cases} \frac{T-\tau}{T-t}(P-Q) & \text{if } t \le \tau, \\ \frac{\tau}{t}(P-Q) & \text{if } t > \tau. \end{cases},$$

Let $\phi(t) = (n\rho_n)^{-1}\|G_t\|$ and let $\bar{\phi}(t) = (n\rho_n)^{-1}\|\bar{G}_t\|$. Clearly, if $P \neq Q$, then $\bar{\phi}(t)$ attains its unique maximum at $t = \tau$. Thus $\bar{\phi}(\tau)$ is a measure of the signal present in the problem: the smaller its value, the harder it is to detect the changepoint.

We will now present our main results. The proofs will be deferred to Section 5.6. There are two regimes of interest:

1. $n\rho_n$ is not large enough so that a single graph alone is not sufficient to estimate the underlying network structure. In this case, we can exploit the averaging present in the CUSUM statistic. Note also that, in very sparse cases, it is usually not possible to estimate the whole matrix P consistently, but one may be able to use some other graph statistic, such as subgraph counts, for identifying changepoints. For example, in Section 5.5, we will see that for the MIT reality mining data, the degree density itself is a good measure for changepoint detection.

2. $n\rho_n$ is large enough so that each graph alone is sufficient to estimate the underlying network structure.

Theorem 5.4.1 (Consistency of $\hat{\tau}$, Regime 1)**.** *Assume that for any fixed $\beta \in (\alpha_0, \alpha_1) \cup (1-\alpha_1, 1-\alpha_0)$, we have*

$$\mathbb{P}(\frac{1}{n\rho_n}\|\frac{1}{\beta T}\sum_{i=1}^{\beta T}(\hat{P}_i - P_i)\| > \epsilon) \leq \delta_{\alpha_0,\alpha_1}(n, T, \epsilon), \tag{5.2}$$

for some function $\delta_{\alpha_0,\alpha_1}(n, T, \epsilon)$, where the P_i's are all equal. Then

$$\mathbb{P}(|\hat{\tau} - \tau| \geq \eta T) \leq 6T\delta_{\alpha_0,\alpha_1}(n, T, \frac{\gamma\eta\bar{\phi}(\tau)}{8}). \tag{5.3}$$

Remark 5.4.1. *The extra factor of T in the error probability in (5.3) comes from a union bound and above can be avoided if, in place of (5.2), we could prove a uniform deviation result like*

$$\mathbb{P}(\frac{1}{n\rho_n}\sup_{\beta\in(\alpha_0,\alpha_1)\cup(1-\alpha_1,1-\alpha_0)}\|\frac{1}{\beta T}\sum_{i=1}^{\beta T}(\hat{P}_i - P_i)\| > \epsilon) \leq \delta_{\alpha_0,\alpha_1}(n, T, \epsilon), \tag{5.4}$$

but this is, naturally, harder to obtain.

Remark 5.4.2. *Notice the presence of the factor $\gamma = \min\{\alpha_0, 1-\alpha_1\}$. If $n\rho_n$ is small, we can not hope to prove a uniform concentration result including t very near the edges of the series, because then one of the two averages in G_t will be a poor estimate of the underlying structure. This edge effect is a characteristic of all fixed dimensional changepoint problems.*

We can prove concentration results like (5.2), with $\hat{P} = A$, to derive explicit rates.

Theorem 5.4.2 (Operator norm)**.** *Assume that the A_i's are independent. Let $\hat{\tau}_{op}$ be CUSUM estimate of τ with $\hat{P} = A$ and $\|\cdot\| = \|\cdot\|_{op}$. Then*

$$\frac{|\hat{\tau}_{op} - \tau|}{T} = O\left(\frac{1}{\gamma\bar{\phi}(\tau)\sqrt{n\rho_n T}}\right),$$

with probability at least $1 - O(nT\exp(-C\sqrt{n\rho_n/T}))$, which goes to 1, as long as $\frac{n\rho_n}{T} = \Omega((\log(nT))^2)$.

Theorem 5.4.3 (Frobenius norm)**.** *Assume that the A_i's are independent. Let $\hat{\tau}_F$ be CUSUM estimate of τ with $\hat{P} = A$ and $\|\cdot\| = \|\cdot\|_F$. Then*

$$\frac{|\hat{\tau}_F - \tau|}{T} = O\left(\frac{1}{\gamma\bar{\phi}(\tau)\sqrt{T\rho_n}}\right),$$

with probability at least $1 - O(T\exp(-Cn^2\rho_n^2))$, which goes to 1, as long as $n\rho_n = \Omega(\sqrt{\log T})$.

These results may be compared with the classical fixed dimensional changepoint problem (see, e.g., [10]), where the minimax rate is $T^{-1}|\hat{\tau} - \tau| = O_P(1/T)$, achieved by the maximum likelihood estimator. Note that, by Theorem 5.4.2, with high probability,

$$\frac{|\hat{\tau}_{op} - \tau|}{T} \ll \frac{1}{T},$$

as long as $n\rho_n = \Omega(T \log(nT))$. Therefore, in this high dimensional case, our estimation error rate is better than the classical rate of $\frac{1}{T}$. We also see that using the operator norm is better, although it requires a larger value of $n\rho_n$ compared to T. In the next section, we will see from our simulations that the operator norm is much more tolerant to the edge effect than the Frobenius norm.

If we are unable to prove a deviation result like (5.2), but $n\rho_n$ is large enough so that we can estimate P from just one instance of A, in that case we can still derive some weaker consistency results.

Theorem 5.4.4 (Consistency of $\hat{\tau}$, Regime 2)**.** *Assume that the smoothing procedure produces $\hat{P}_i$ such that*

$$\mathbb{P}(\frac{1}{n\rho_n}\|\hat{P}_i - P_i\| \geq \epsilon) \leq \delta(n, \epsilon), \tag{5.5}$$

for some function $\delta(n, \epsilon)$. Then we have

$$\mathbb{P}(|\hat{\tau} - \tau| \geq \eta T) \leq 2T\delta(n, \frac{\eta}{4}\bar{\phi}(\tau)). \tag{5.6}$$

Remark 5.4.3. *Again, the extra factor of T in the error probability in (5.6) comes from a crude union bound for $\max_{1\leq i\leq T} \frac{1}{n\rho_n}\|\hat{P}_i - P_i\|$ (and hence requires no assumptions on the mutual dependence of the A_i's) and in some cases one may be able to use a more refined concentration result (by possibly exploiting the mutual dependence structure of the A_i's) in order to avoid it.*

Remark 5.4.4. *Note the absence of γ in (5.6). Indeed, if we can get very good estimates of P just from a single instance of A, then the edge effect disappears, we can even detect a changepoint in a time series of length two (which can be thought of as testing difference of two graphs).*

For dense graphs, we can use graphon estimation techniques to estimate P.

Example 5.4.1. *As mentioned in Section 4.6, the neighborhood smoothing (NBS) estimate of P due to [86] satisfies (for P coming from piecewise L-Lipschitz graphons, see Definition 2.1 of [86]), with probability at least $1 - n^{-C}$,*

$$\frac{\|\hat{P} - P\|_F^2}{n^2} \leq C_L\sqrt{\frac{\log n}{n}},$$

for some global constant $C > 0$ and a constant C_L that depends on the Lipschitz constant L.

Example 5.4.2. *As mentioned in Section 4.6, the USVT estimate of P due to [25] satisfies (for P coming from L-Lipschitz graphons)*

$$\mathbb{E}\frac{\|\hat{P}-P\|_F^2}{n^2} \leq C_L n^{-1/3},$$

for some constant $C_L > 0$ depending on the Lipschitz constant L. Therefore, for any $\mu \in (0, 1/3)$, with probability at least $1 - n^{-\mu}$, we have

$$\frac{\|\hat{P}-P\|_F^2}{n^2} \leq C_L n^{-(1/3-\mu)}.$$

We can use these results in conjunction with Theorem 5.4.4 to get explicit concentration bounds for $\hat{\tau}$. For instance, if we use the NBS estimate, then, with probability at least $1 - O(Tn^{-C})$, we have

$$\frac{|\hat{\tau}_F - \tau|}{T} = O\left(\frac{1}{\bar{\phi}(\tau)}\sqrt{\frac{\log n}{n}}\right).$$

Therefore, as long as $n^C \gg T$, we have, with high probability, that

$$\frac{|\hat{\tau}_F - \tau|}{T} \ll \frac{1}{\bar{\phi}(\tau)}\sqrt{\frac{\log n}{T^{\frac{1}{2C}}}}.$$

5.5 Simulations and real data analysis

Simulations

We provide two sets of simulations, one in a dense setting, the other sparse. In both cases, we generate $T = 100$ networks of size $n = 500$ each from the ER model with $p = \frac{\kappa \log n}{n}$. We also generate the 100 networks from a simple balanced two parameter stochastic blockmodel with block probability matrix $B = (1 - r)pI + rpJ$, where $r = 0.1$. Finally, we consider a sequence of 100 networks, where the first 60 are from the ER model and the last 40 are from the SBM.

In the dense setting, we set $\kappa = 10$ (average degree of the blockmodel graphs becomes ≈ 34). See Figures 5.1-5.2. We use the naive estimate A as well as the USVT estimate in this simulation. We see that the USVT version is much less prone to edge effects. Also, if we use A, the operator norm is preferable. This is reasonable because A is not a good estimate of P in the Frobenius norm in general (see Remark 4.6.3), so that near the edges we do not get good estimates from the averages. On the other hand, A concentrates around P in the operator norm as long as the network has average degree growing like $\log n$.

In the sparse setting, we set $\kappa = 1$ (average degree of the blockmodel graphs becomes ≈ 3.4). See Figure 5.3. We use the naive estimate A as USVT does not work in such extremely sparse cases. We see a good separation only when we use the operator norm.

Figure 5.1: Dense setting: simulation with a single changepoint that marks a transition from an ER model to an SBM. In the left column, we have used A as an estimate of P, whereas in the right column, USVT is used to get an estimate of P. We have used the Frobenius norm here.

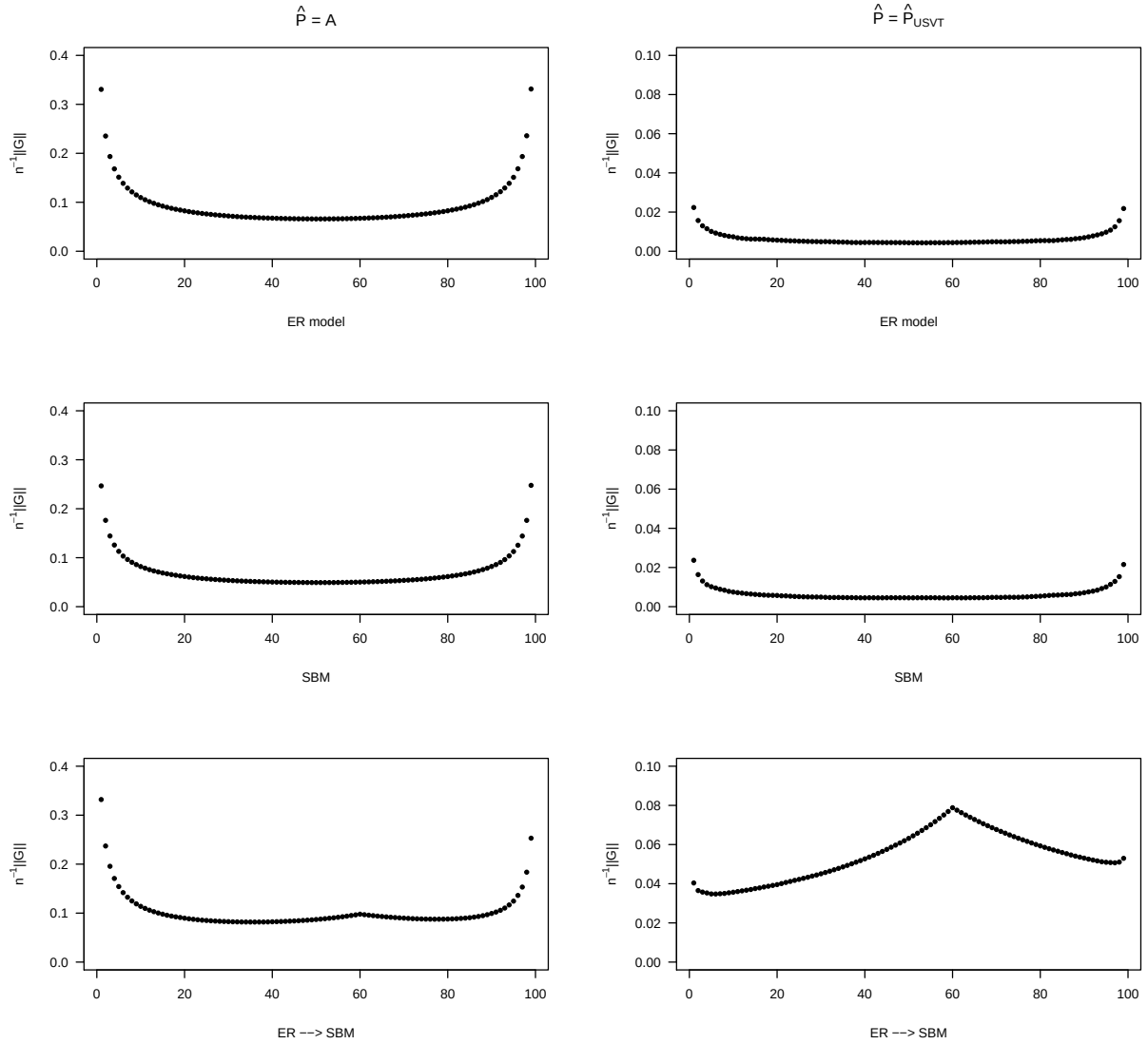


Figure 5.2: Dense setting: simulation with a single changepoint that marks a transition from an ER model to an SBM. In the left column, we have used A as an estimate of P, whereas in the right column, USVT is used to get an estimate of P. We have used the operator norm here.

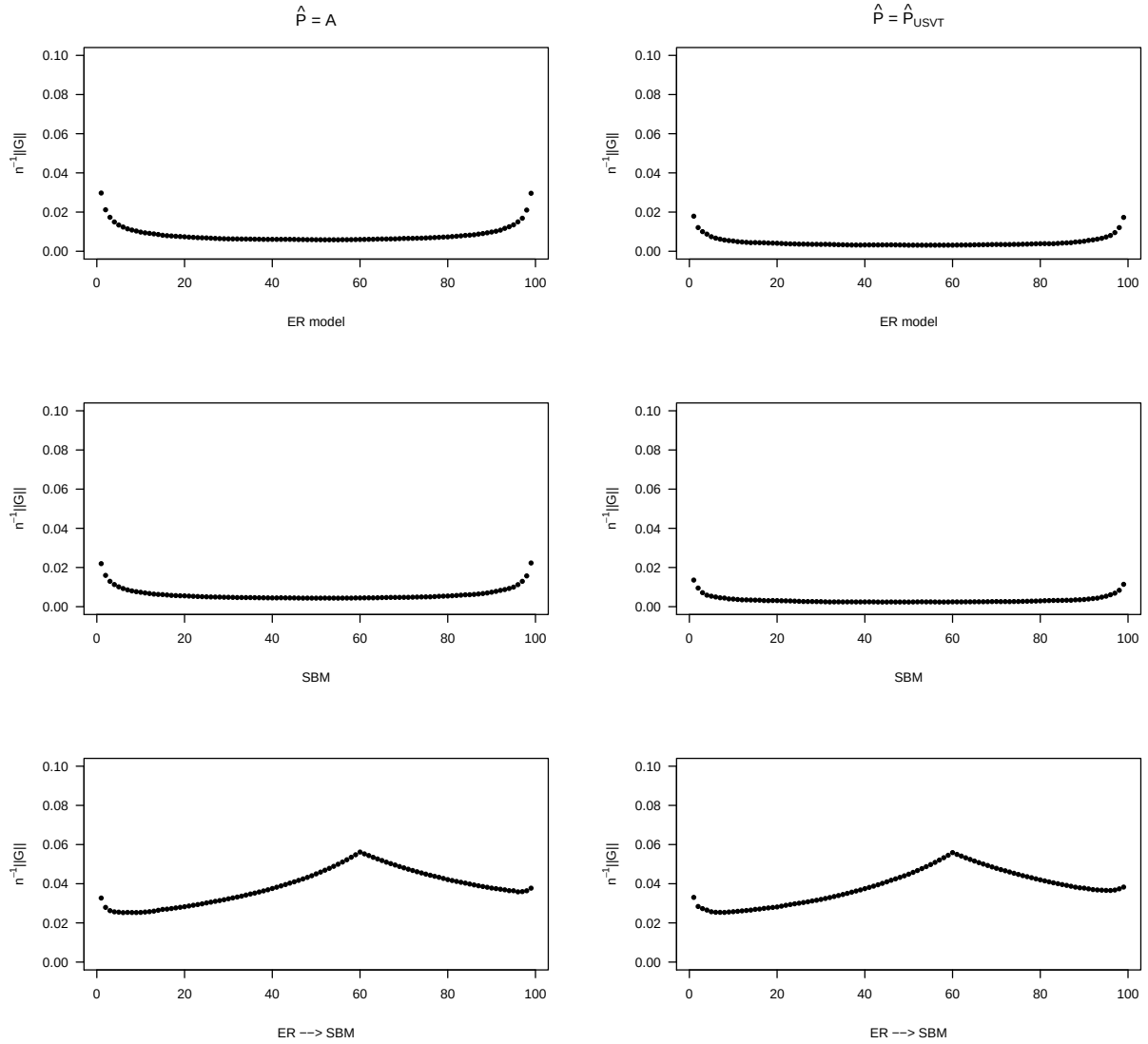
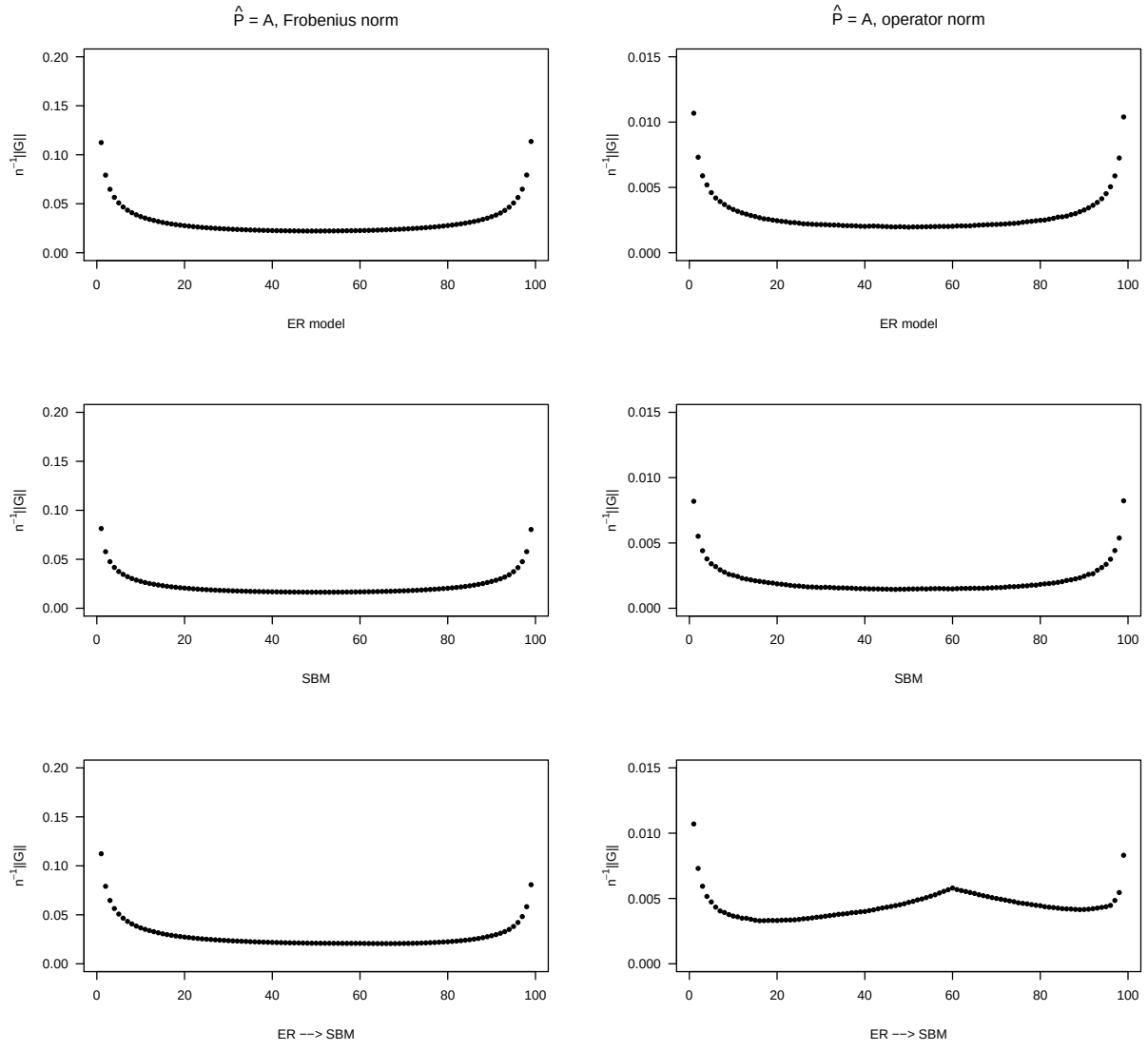


Figure 5.3: Sparse setting: simulation with a single changepoint that marks a transition from an ER model to an SBM.

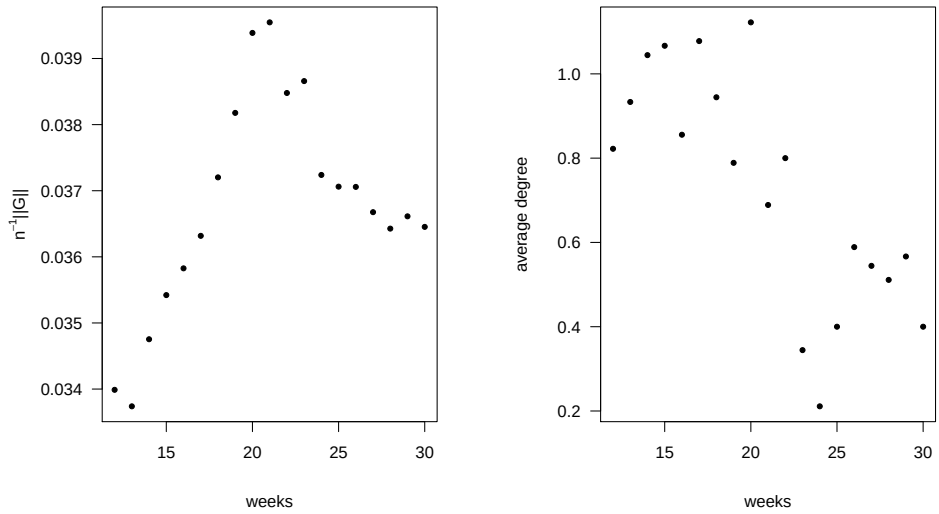


Real Data

MIT reality mining data

We use the MIT reality mining data [27] to construct a series of networks involving 90 individuals (staff and students at the university). The data consists of call logs between these individuals from 20th July 2004 to 14th June 2005. We construct 48 weekly networks, where in the i-th network, an edge is put between two vertices if they had been in at least one call during week i. These networks are extremely sparse, with the maximum average degree being barely over 1. The first three, and the last six weeks had especially low average degree (< 0.12) compared to the rest, so we discard these. On the remaining time series, our (Frobenius norm based) CUSUM estimate is week number 21, which roughly corresponds to the beginning of the winter break. [26] also found a changepoint at around this time. We use a conservative choice of $a = 12, b = 30$ to eliminate edge effects. See Figure 5.4. Due to the extreme sparsity of these networks, the average degree itself is a good measure to detect changes, and standard univariate changepoint algorithms run on the sequence of average degrees detect changepoints at week numbers $9, 22$, and 42.

Figure 5.4: MIT reality mining caller-callee networks, the peak at $t = 21$ in the left panel, corresponds roughly to the beginning of the winter break. We have used A as an estimate of P, and the Frobenius norm version of the CUSUM estimate. The average degrees of these networks are plotted in the right panel, we see a sharp change at $t = 22$.

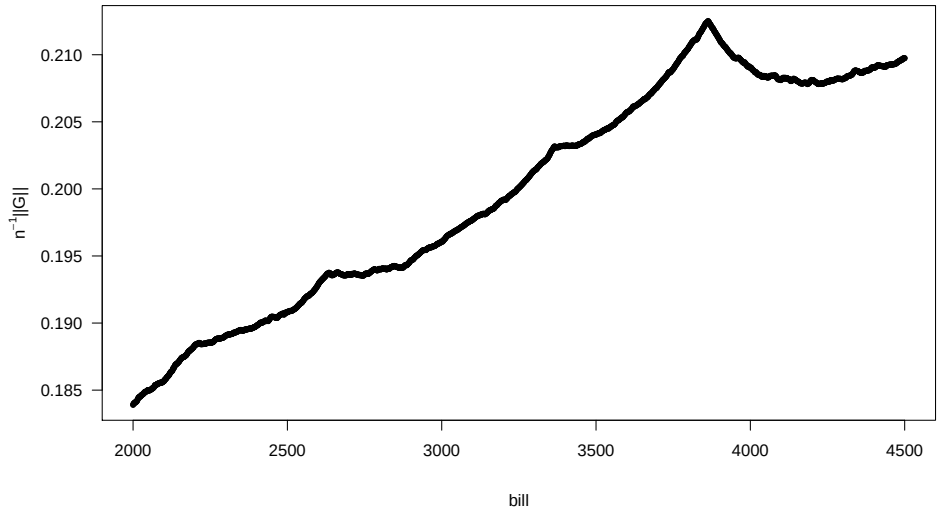


US senate roll call data

We use the US senate roll call data (1979 - 2012) [44] for another illustration of our changepoint detection method. Each state has 2 senate seats, so the networks have size 100, the votes are "yay/nay", and sometimes missing/present, such cases are imputed by taking the majority stand in the respective party, or if that is not possible, taking the winning majority stand. We discard roll calls wherein the majority ≥ 75, because such unanimous roll calls do not reveal much structure. This results in $T = 7863$ roll calls. We convert these to networks by putting an edge between seats that take the same stand on the proposed bill. Our (Frobenius norm based) CUSUM estimate is Dec 1, 1994. See Figure 5.5. There are possibly other changepoints in this data. We use the conservative lower and upper bounds of $a = 2000$, $b = 4500$, which correspond respectively to 18th March, 1986 and 23rd July, 1996.

The detected changepoint is close to the November 1994 election when the Republican Party took over the House of Representatives for the first time after 1956. The Markov random field method of [72] found a changepoint on January 17, 1995, which is also close to the November 1994 elections. To quote the authors of [72], *"As discussed in the political science literature, the 1994 election marked the end of the "Conservative Coalition", a bipartisan coalition of conservative oriented Republicans and Democrats on President RooseveltâĂŹs "New Deal" policies, which had often managed to control Congressional outcomes since the "New Deal" era."* Other analyses, such as [52], also indicate a significant change happening after the November 1994 election.

Figure 5.5: US senate roll call data, the peak at $t = 3863$ corresponds to a roll call on Dec 1, 1994. We have used A as an estimate of P, and the Frobenius norm version of the CUSUM estimate. Using USVT and/or operator norm results in the same estimate.



5.6 Proofs

As $\hat{\tau}$ is obtained as a maximizer of a criterion, the consistency proofs will bear similarities with the consistency proof for M-estimators. To prove consistency of $\hat{\tau}$, it suffices to show uniform concentration of ϕ around $\bar{\phi}$. We begin with a preliminary result.

Lemma 5.6.1. *For any $\eta \in (0,1)$, we have*

$$\mathbb{P}(|\hat{\tau}-\tau| \geq \eta T) \leq 2\mathbb{P}(\max_{t\in[\alpha_0 T,\alpha_1 T]} |\phi(t)-\bar{\phi}(t)| \geq \frac{\eta}{4}\bar{\phi}(\tau)).$$

Proof. We have

$$\begin{aligned}
\mathbb{P}(|\hat{\tau}-\tau| \geq \eta T) &= \mathbb{P}(\hat{\tau} \geq \tau+\eta T) + \mathbb{P}(\hat{\tau} \leq \tau-\eta T) \\
&\leq \mathbb{P}(\bar{\phi}(\hat{\tau}) \leq \bar{\phi}(\tau+\eta T)) + \mathbb{P}(\bar{\phi}(\hat{\tau}) \leq \bar{\phi}(\tau-\eta T)) \\
&= \mathbb{P}(\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) \geq \bar{\phi}(\tau)-\bar{\phi}(\tau-\eta T)) \\
&\qquad + \mathbb{P}(\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) \geq \bar{\phi}(\tau)-\bar{\phi}(\tau-\eta T)) \\
&= \mathbb{P}(\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) \geq \frac{\eta T}{\tau+\eta T}\bar{\phi}(\tau)) + \mathbb{P}(\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) \geq \frac{\eta T}{T-\tau+\eta T}\bar{\phi}(\tau)) \\
&\leq 2\mathbb{P}(\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) \geq \frac{\eta}{2}\bar{\phi}(\tau)).
\end{aligned}$$

Now observe that

$$\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) = \bar{\phi}(\tau)-\phi(\tau)+\phi(\hat{\tau})-\bar{\phi}(\hat{\tau}) + \underbrace{\phi(\tau)-\phi(\hat{\tau})}_{\leq 0} \leq 2\max_{t\in[\alpha_0 T,\alpha_1 T]} |\phi(t)-\bar{\phi}(t)|.$$

Therefore

$$\mathbb{P}(|\hat{\tau}-\tau| \geq \eta T) \leq 2\mathbb{P}(\bar{\phi}(\tau)-\bar{\phi}(\hat{\tau}) \geq \frac{\eta}{2}\bar{\phi}(\tau)) \leq 2\mathbb{P}(\max_{t\in[\alpha_0 T,\alpha_1 T]} |\phi(t)-\bar{\phi}(t)| \geq \frac{\eta}{4}\bar{\phi}(\tau)),$$

and we are done. □

To prove uniform concentration, we may use the averaging structure present in the CUSUM estimate.

Lemma 5.6.2 (Uniform concentration of $\phi(t)$ around $\bar{\phi}(t)$)**.** *Assume (5.2). Then, with probability at least $1-3T\delta_{\alpha_0,\alpha_1}(n,T,\epsilon)$, we have*

$$\max_{t\in[\alpha_0 T,\alpha_1 T]} |\phi(t)-\bar{\phi}(t)| \leq \frac{2\epsilon}{\gamma},$$

where $\gamma = \gamma(\alpha_0,\alpha_1) := \min\{\alpha_0, 1-\alpha_1\}$.

Proof. Note first that $n\rho_n|\phi(t) - \bar{\phi}(t)| \leq \|G_t - \bar{G}_t\|$. Now we have, for any $t \in [\alpha_0 T, \alpha_1 T]$, with $t < \tau$,

$$\begin{aligned}
\|G_t - \bar{G}_t\| &= \left\| \frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) - \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i) \right\| \\
&= \left\| \frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) - \frac{1}{T-t}\left[\sum_{i=1}^{\tau}(\hat{P}_i - P_i) + \sum_{i=\tau+1}^{T}(\hat{P}_i - P_i) - \sum_{i=1}^{t}(\hat{P}_i - P_i)\right] \right\| \\
&\leq \left\| \frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) \right\| + \frac{\tau}{T-t}\left\| \frac{1}{\tau}\sum_{i=1}^{\tau}(\hat{P}_i - P_i) \right\| \\
&\qquad + \frac{T-\tau}{T-t}\left\| \frac{1}{T-\tau}\sum_{i=\tau+1}^{T}(\hat{P}_i - P_i) \right\| + \frac{t}{T-t}\left\| \frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) \right\| \\
&= \frac{T}{T-t}\left\| \frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) \right\| + \frac{\tau}{T-t}\left\| \frac{1}{\tau}\sum_{i=1}^{\tau}(\hat{P}_i - P_i) \right\| \\
&\qquad + \frac{T-\tau}{T-t}\left\| \frac{1}{T-\tau}\sum_{i=\tau+1}^{T}(\hat{P}_i - P_i) \right\|.
\end{aligned}$$

Similarly, for any $t \in [\alpha_0 T, \alpha_1 T]$, with $t \geq \tau$, we have

$$\begin{aligned}
\|G_t - \bar{G}_t\| &= \|\frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) - \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i)\| \\
&= \left\| \frac{1}{t}\left[\sum_{i=1}^{\tau}(\hat{P}_i - P_i) + \sum_{i=\tau+1}^{T}(\hat{P}_i - P_i) - \sum_{i=t+1}^{T}(\hat{P}_i - P_i)\right] - \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i) \right\| \\
&\leq \frac{\tau}{t}\left\| \frac{1}{\tau}\sum_{i=1}^{\tau}(\hat{P}_i - P_i) \right\| + \frac{T-\tau}{t}\left\| \frac{1}{T-\tau}\sum_{i=\tau+1}^{T}(\hat{P}_i - P_i) \right\| \\
&\qquad + \frac{T-t}{t}\left\| \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i) \right\| + \left\| \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i) \right\| \\
&= \frac{\tau}{t}\left\| \frac{1}{\tau}\sum_{i=1}^{\tau}(\hat{P}_i - P_i) \right\| + \frac{T-\tau}{t}\left\| \frac{1}{T-\tau}\sum_{i=\tau+1}^{T}(\hat{P}_i - P_i) \right\| \\
&\qquad + \frac{T}{t}\left\| \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i) \right\|.
\end{aligned}$$

Therefore, in either case, we have, by using (5.2) for any $t \in [\alpha_0 T, \alpha_1 T]$, that

$$|\phi(t) - \bar{\phi}(t)| \leq \frac{2\epsilon}{\min\{\alpha_0, 1-\alpha_1\}},$$

with probability at least $1 - 3\delta_{\alpha_0,\alpha_1}(n, T, \epsilon)$. Taking an union bound over t, we get the desired uniform concentration. □

Proof of Theorem 5.4.1. This is an immediate consequence of Lemmas 5.6.1 and 5.6.2. □

Now we derive a couple of concentration results like (5.2), when $\hat{P} = A$.

Proposition 5.6.1 (Concentration in operator norm)**.** *Suppose that the A_i's are i.i.d. Bernoulli random matrices with $\mathbb{E}(A_i) = P$. Then*

$$\mathbb{P}\left(\frac{1}{n\rho_n}\|\frac{1}{t}\sum_{i=1}^{t}(A_i - P)\|_{op} > \epsilon\right) \leq 2n\exp\left(\frac{-Cn\rho_n t\epsilon^2}{1+\epsilon t}\right), \tag{5.7}$$

where $C > 0$ is a universal constant.

Proof. We imitate the proof of Theorem 3.1 of [64]. Note that we can decompose

$$\frac{1}{t}\sum_{i}^{t}(A_i - P) = \sum_{k\leq l} X_{kl},$$

where $X_{kl} = (\frac{1}{t}\sum_{i=1}^{t}(A_i)_{kl} - P_{kl})E_{kl}$, with

$$E_{ij} = \begin{cases} e_k e_l^\top + e_l e_k^\top & \text{for } k \neq l \\ e_k e_k^T & \text{for } k = l. \end{cases}$$

Then $\|X_{kl}\|_{op} \leq \|E_{kl}\|_{op} \leq 1$, and a quick calculation reveals that

$$\sum_{k\leq l}\mathbb{E}X_{kl}^2 \leq \max_k(\sum_l P_{kl})/t = O\left(\frac{n\rho_n}{t}\right).$$

Therefore, using matrix Bernstein inequality (e.g., Corollary 7.1 of [64]), we conclude that

$$\mathbb{P}\left(\|\frac{1}{t}\sum_{i=1}^{t}(A_i - P)\|_{op} > \epsilon'\right) \leq 2n\exp\left(-\frac{C\epsilon'^2}{\frac{n\rho_n}{t} + \epsilon'}\right),$$

for some absolute constant $C > 0$. Now choosing $\epsilon' = n\rho_n\epsilon$ gives the desired bound. □

Proof of Theorem 5.4.2. This follows from Proposition 5.6.1 and Theorem 5.4.1. □

Proposition 5.6.2 (Concentration in Frobenius norm)**.** *Suppose that the A_i's are i.i.d. Bernoulli random matrices with $\mathbb{E}(A_i) = P$. Then*

$$\mathbb{P}\left(\frac{1}{n^2\rho_n^2}\|\frac{1}{t}\sum_{i=1}^{t}(A_i - P)\|_F^2 > \frac{c}{t\rho_n}\right) \leq 2\exp(-Cn^2\rho_n^2), \tag{5.8}$$

where $c, C > 0$ are universal constants.

Proof. Let $\Delta_{kl} = \frac{1}{t}\sum_{i=1}^{t}(A_i - P)_{kl}$. Note that Δ_{kl} is a centered sub-Gaussian random variable with variance proxy $\frac{1}{4t}$. Therefore Δ_{kl}^2 is sub-exponential with sub-exponential norm $\asymp \frac{1}{t}$. Note that

$$\|\frac{1}{t}\sum_{i=1}^{t}(A_i - P)\|_F^2 = 2\sum_{k<l}\Delta_{kl}^2.$$

Now, by sub-exponential concentration (see, e.g., Proposition 5.16 of [78]), we have

$$\mathbb{P}(|\sum_{k<l}(\Delta_{kl}^2 - \mathbb{E}\Delta_{kl}^2)| > \epsilon) \leq 2\exp(-C\min(\frac{\epsilon^2 t^2}{n^2}, \epsilon t)).$$

Note that

$$\sum_{k<l}\mathbb{E}\Delta_{kl}^2 = \sum_{k<l}\frac{P_{kl}(1-P_{kl})}{t} \leq \frac{cn^2\rho_n}{2t},$$

for some global constant $c > 0$. Therefore

$$\mathbb{P}(|\sum_{k<l}\Delta_{kl}^2| > \frac{cn^2\rho_n}{2t} + \epsilon) \leq 2\exp(-C\min(\frac{\epsilon^2 t^2}{n^2}, \epsilon t)).$$

Choosing $\epsilon' = \frac{cn^2\rho_n}{2t}$ we get,

$$\mathbb{P}\left(\frac{1}{n^2\rho_n^2}|\sum_{k<l}\Delta_{kl}^2| > \frac{c}{t\rho_n}\right) \leq 2\exp(-Cn^2\rho_n^2),$$

which is equivalent to the desired bound. □

Proof of Theorem 5.4.3. This follows from Proposition 5.6.2 and Theorem 5.4.1. □

We will now prove an analogue of Lemma 5.6.2 in Regime 2.

Lemma 5.6.3 (Uniform concentration of $\phi(t)$ around $\bar{\phi}(t)$)**.** *Assume* (5.5)*. Then we have*

$$\mathbb{P}(\max_{1\leq t\leq T-1}|\phi(t) - \bar{\phi}(t)| > \epsilon) \leq T\delta(n,\epsilon).$$

Proof. We note that

$$\begin{aligned}
|\phi(t) - \bar{\phi}(t)| &\leq \frac{1}{n\rho_n}\|G_t - \bar{G}_t\| \\
&= \frac{1}{n\rho_n}\left\|\frac{1}{t}\sum_{i=1}^{t}(\hat{P}_i - P_i) - \frac{1}{T-t}\sum_{i=t+1}^{T}(\hat{P}_i - P_i)\right\| \\
&\leq \frac{1}{n\rho_n}\left[\frac{1}{t}\sum_{i=1}^{t}\|\hat{P}_i - P_i\| + \frac{1}{T-t}\sum_{i=t+1}^{T}\|\hat{P}_i - P_i\|\right] \\
&\leq \max_{1\leq i\leq T}\frac{1}{n\rho_n}\|\hat{P}_i - P_i\|.
\end{aligned}$$

From this the desired result follows by union bound. □

Proof of Theorem 5.4.4. This is immediate from Lemmas 5.6.1 and 5.6.3. □

5.7 Discussion

To summarize, we have considered the offline changepoint detection problem in the context of network data. We have theoretically analyzed the CUSUM statistic under the single changepoint model, and established consistency of the resulting estimate by establishing concentration bounds involving as parameters the number of networks, the (common) size of the networks and their (common) sparsity. We have shown that in this high dimensional problem one can beat the classical rate of estimation in fixed dimensional problems. We have analyzed the MIT reality mining data and the US senate roll call data as applications. There are many directions for future work — the most important ones are (i) deriving concentration bounds for dependent time series of networks to extend our analysis, (ii) analyzing the multiple changepoint case, (iii) proving optimality results, (iv) deriving second order asymptotics to calibrate tests for changepoints, and (v) tackling the online version of the problem.

Appendix A

Appendix to Chapter 2

A.1 PACE-corollaries

Quantitative versions of Corollary 2.3.3

We first consider $(1+\epsilon)$-approximate adjacency spectral clustering (ASP) of [43] as $\mathcal{A}$. We quote a slightly modified version of Corollary 3.2 of [43].

Lemma A.1.1 ([43]). *Let $c_0, \epsilon, r > 0$. Consider an adjacency matrix A generated from the simple blockmodel (2.8) where $\alpha_n \geq c_0 \log n/n$. If $\hat{Z}$ is the output of the $(1+\epsilon)$-approximate adjacency spectral clustering algorithm applied to A, then there exists an absolute constant $c = c(c_0, r)$ such that with probability at least $1 - n^{-r}$,*

$$\frac{1}{2}\delta(\hat{Z}, Z) \leq \min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^2\alpha_n}\frac{\pi_{\max}}{\pi_{\min}^2 n}, 1\right\}. \tag{A.1}$$

Corollary A.1.1 ($(1+\epsilon)$-approximate adjacency spectral clustering with random m-subgraphs). *Assume the setting of Lemma A.1.1. Let $r, r' > 0$. We have*

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \leq \frac{8m}{\theta(m-1)}\left[\min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^2\alpha_n}\frac{\pi_{\max}}{\pi_{\min}^2 m}C_{n,m,r'}, 1\right\} + \frac{2K}{n^{r'}} + \frac{1}{m^r}\right] + \pi_{\max} \times e^{-(1-\theta)^2 Tp/2}. \tag{A.2}$$

Here the quantity $C_{n,m,r'} \to 1$, if $\frac{\pi_{\min}^2 m}{\pi_{\max}\log n} \to \infty$.

The proof of Corollary A.1.1 follows from Corollary 2.3.1 and an estimate for $\mathbb{E}\delta(\hat{Z}_S, Z_S)$ given in (A.13), which is obtained using Lemma A.1.1. In order for the first term of (A.2) to go to zero we need $\frac{K\pi_{\max}}{\lambda^2 m\alpha_n\pi_{\min}^2} = o(1)$, i.e. $m \gg \frac{K\pi_{\max}}{\lambda^2\alpha_n\pi_{\min}^2}$. Thus, for balanced block sizes (i.e. $\pi_{\max}, \pi_{\min} \asymp \frac{1}{K}$), we need to have $m \gg \frac{K^2}{\lambda^2\alpha_n}$. So, qualitatively, for large K or small

α_n or a small separation between the blocks, m has to be large, which is natural to expect. In particular, for fixed K and λ, this shows that we need subgraphs of size $m \gg nd_n^{-1}$, and $T \gg \frac{n^2}{m^2}$ many of them to achieve consistency (here the average degree $d_n \asymp n\alpha_n$). Let $T = \frac{n^2}{m^2} r_n$ and $m = \frac{n}{d_n} s_n$, where both $r_n, s_n \to \infty$. Let us see what computational gain we get from this. Spectral clustering on the full graph has complexity $O(n^3)$, while the complexity of PACE with spectral clustering is

$$O(Tm^3) = O\left(\frac{n^2}{m^2} r_n m^3\right) = O\left(n^2 m r_n\right) = O\left(\frac{n^3}{d_n} r_n s_n\right).$$

So if $d_n = \Theta(n^\alpha)$, then the complexity would be $O(n^{3-\alpha} r_n s_n)$, which is essentially $O(n^{3-\alpha})$. When $d_n = \Theta(\log n)$ the gain is small.

Note, however, that for a parallel implementation, with N_c processor cores, we may get a significant boost in running time, at least in terms of constants; the running time would be $O\left(\frac{n^3}{N_c d_n} r_n s_n\right)$.

Corollary A.1.2 ($(1+\epsilon)$-approximate adjacency spectral clustering with ego subgraphs)**.** *Assume the setting of Lemma A.1.1. Let $r, r' > 0$. We have*

$$\mathbb{E}\tilde{\delta}(\hat{C}, C) \le \frac{32(1+\theta)^2}{\lambda^2}\left[\min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^4\alpha_n^2}\frac{\pi_{\max}}{\pi_{\min}^2 n} D_{n,B,r'}, 1\right\} + \frac{2K}{n^{r'}} + \frac{16}{n\lambda^2\alpha_n} + \left(\frac{4}{n\lambda\alpha_n}\right)^r\right] + \Delta, \tag{A.3}$$

where

$$\Delta \le \frac{4}{\lambda^2\alpha_n^2} \times \exp\left(-\frac{n\lambda^2\alpha_n}{16}\right) + \frac{4}{n^2\lambda^2\alpha_n^2} \times \exp\left(-\frac{\theta^2 n\alpha_n}{6}\right) + \pi_{\max} \times \left[2\exp\left(-\frac{n\lambda^4\alpha_n^2}{16}\right) + \exp\left(-\frac{T\lambda^2\alpha_n^2}{16}\right)\right].$$

Here the quantity $D_{n,B,r'} \to 1$, if $\frac{\pi_{\min}^2\lambda^2 n\alpha_n}{\pi_{\max}\log n} \to \infty$.

Corollary A.1.2 follows from Corollary 2.3.2 and an estimate for $\mathbb{E}\delta(\hat{Z}_S, Z_S)\mathbf{1}_{(|S|\ge m_\star)}$ given in (A.14), which is obtained using Lemma A.1.1. For the right hand side in (A.3) to go to zero (assuming K fixed, balanced block sizes), we need $\min\{n\alpha_n^2, T\alpha_n^2\} \to \infty$. In terms of average degree this means that we need $d_n \gg \sqrt{n}$, and $T \gg \frac{n^2}{d_n^2}$. That with ego neighborhoods we can not go down to $d_n = \Theta(\log n)$ is not surprising, since these ego networks are rather sparse in this case. One needs to use larger neighborhoods. Anyway, writing $d_n = \sqrt{n} r_n$, $T = \frac{n^2}{d_n^2} s_n$, where both $r_n, s_n \to \infty$, the complexity of adjacency spectral clustering, in this case becomes $O(Td_n^3) = O(n^2 d_n r_n s_n)$ and with N_c processing units gets further down to $O(\frac{n^2 d_n}{N_c} r_n s_n)$.

We conclude this section with an illustration of PACE with random m-subgraphs using SDP as the algorithm $\mathcal{A}$. We shall use the setting of Theorem 1.3 of [34] for the illustration, stated here with slightly different notation. Let SDP-GV denote the following SDP [34, SDP (1.10)]

$$\begin{aligned}&\text{maximize } \langle A, X\rangle\\ &\text{subject to } X \succeq 0, X \geq 0, \operatorname{diag}(X) = I_n, \mathbf{1}^\top X \mathbf{1} = \mathbf{1}^\top C \mathbf{1}.\end{aligned}$$

Lemma A.1.2 ([34], Theorem 1.3)**.** *Consider an SBM with* $\min_k B_{kk} \geq a/n$, $\max_{k\neq k'} B_{kk'} \leq b/n$, *where* $a > b$. *Also let the expected variance of all edges* $\frac{2}{n(n-1)} \sum_{1\leq k\leq k'\leq K} B_{kk'}(1 - B_{kk'})n_{kk'} = \frac{g}{n}$, *where* $n_{kk'}$ *denotes the number of pairs of vertices, one from community* k, *the other from community* k'. *Fix an accuracy* $\epsilon > 0$. *If* $g \geq 9$ *and* $(a-b)^2 \geq 484\epsilon^{-2}g$, *then any solution* $\hat{X}$ *of SDP-GV satisfies*

$$\frac{1}{n^2}\|\hat{X} - C\|_F^2 \leq \epsilon.$$

Corollary A.1.3 (SDP with random m-subgraphs)**.** *Consider the setting of Lemma A.1.2. Let* $c > 1 > c' > 0$. *and set* $\bar{g}_1 = \frac{c_1(n-1)g}{m-1}$ *and* $\bar{g}_2 = \frac{c_2 mg}{n}$, *where* $c_1 = c^2\left(1 - \frac{n-cm}{cm(n\pi_{\min}-1)}\right)$ *and* $c_2 = (c')^2\left(1 - \frac{n-c'm}{c'm(n\pi_{\max}-1)}\right)$. *Fix an accuracy* $\epsilon > 0$. *Assume that* $(a-b)^2 \geq 484\epsilon^{-2}\bar{g}_1$, *and that* $\bar{g}_2 \geq 9$. *Then we have*

$$\begin{aligned}\mathbb{E}\tilde{\delta}(\hat{C}, C) \leq \frac{4m}{\theta(m-1)} \times (\epsilon + e^3 5^{-m} + Ke^{-(1-c')^2 m\pi_{\min}/4}\\ + Ke^{-(c-1)^2 m\pi_{\min}/4}) + \frac{1}{2} \times e^{-(1-\theta)^2 Tp/2}.\end{aligned} \tag{A.4}$$

For the simple two parameter blockmodel $B = \frac{1}{n}((a-b)I + b\mathbf{1}\mathbf{1}^\top)$ with equal community sizes, we have $g \asymp \frac{a+(K-1)b}{K} \asymp d_n$, the average degree of the nodes (note that $d_n = \frac{a+(K-1)b}{K} - \frac{a}{n}$). The assumptions of Corollary A.1.3 are satisfied when

$$m = \Omega\left(\max\left\{\frac{n}{d_n}, \frac{nd_n}{\epsilon^2(a-b)^2}\right\}\right).$$

This is exactly similar to what we saw for spectral clustering (take $a = n\alpha_n$, and $b = n\alpha_n(1-\lambda)$). In particular, when the average degree $d_n = \Theta(n^\alpha)$, and $a - b = \Theta(n^\alpha)$, we need $m = \Omega(n^{1-\alpha}/\epsilon^2)$ and $T \gg n^{2\alpha}/\epsilon^4$ for PACE to succeed. However, in the bounded degree regime, the advantage is negligible, only from a potentially smaller constant, because we need $m = \Omega(n)$. Again, from our numerical results, we expect that with h-hop subgraphs, PACE will perform much better.

A.2 GALE-corollaries

We will now illustrate Theorem 2.3.2 with several algorithms $\mathcal{A}$. We begin with a result on $(1+\epsilon)$-approximate adjacency spectral clustering.

Corollary A.2.1 ($(1+\epsilon)$-approximate adjacency spectral clustering with GALE)**.** *Assume the setting of Lemma A.1.1. Let $0 < \theta < 1$. Let $r, r', r'', r''' > 0$. Let $m = \Omega_{r,r',\theta}\left(\sqrt{\frac{n \log n}{\pi_{\min}}}\right)$, $T = \Omega_{r,r',\theta}(n \log n/m)$, and $\tau = \frac{\theta T m}{n}$. Then we have, with probability at least $1 - \frac{T}{m^{r''}} - \frac{2TK}{n^{r'''}} - O\left(\frac{1}{n^{r'}}\right)$, that*

$$\max_{\mathcal{T}} \max_{(x_1,\ldots,x_J)} \delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) \leq \frac{2}{\theta}\left[\min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^2\alpha_n}\frac{\pi_{\max}}{\pi_{\min}^2 m} C_{n,m,r'''}, 1\right\}\right] + O\left(\frac{1}{n^r}\right), \tag{A.5}$$

where the constant $C_{n,m,r'''}$ is the same as in Corollary 2.3.1

We see that the first term is exactly same as the first term in Corollary 2.3.1. This, for balanced graphs, again imposes the condition $m \gg \frac{K^2}{\lambda^2\alpha_n}$. In particular, if $K = \Theta(1)$ and we are in a dense well separated regime, with $\lambda = \Theta(1)$, $\alpha_n = \Omega(1/\sqrt{n})$, then we need $m = \Omega(\sqrt{n \log n})$. If $K = \Theta(1)$, $\lambda = \Theta(1)$ and $\alpha_n = \Theta(\log n/n)$, then we need $m \gg n/\log n$. In both cases, we need $T = \Omega(n \log n/m)$. Thus, in the regime where average degree is like $\log n$, there is still some computational advantage for very large networks (also factoring in parallelizability); however, for moderately sized networks, GALE may not lead to much computational advantage.

Now we present an exact recovery result with SDP as the base algorithm $\mathcal{A}$. We shall use a (slightly rephrased) result[†] from [82] on an SDP which they call SDP-λ. Let $\kappa := \pi_{\max}/\pi_{\min}$. Also let $\boldsymbol{\pi}$ denote the vector of the cluster proportions $(\pi_1, \ldots, \pi_K)$.

Lemma A.2.1 (Theorem 2 of [82])**.** *Let $r > 0$. Then the optimal solution of the SDP-λ is given by $Z\,\mathrm{diag}(n\boldsymbol{\pi})^{-1}Z^\top$, with probability at least $1 - O((n\pi_{\min})^{-r})$, if*

$$\min_k (B_{kk} - \max_{\ell \neq k} B_{k\ell}) = \tilde{\Omega}_r\left(\kappa \max_k \sqrt{\frac{\max(B_{kk}, K \max_{\ell \neq k} B_{k\ell})}{n\pi_k}}\right). \tag{A.6}$$

Assuming that any subsequent clustering of the exactly recovered scaled clustering matrix $Z\,\mathrm{diag}(n\boldsymbol{\pi})^{-1}Z^\top$ gives the exact clustering Z back (for example, our distance based naive algorithm NaiveCluster[‡] will do this), we have the following corollary.

Corollary A.2.2 (SDP with GALE, exact recovery)**.** *Assume the setting of Lemma A.2.1. Let $0 < \theta < 1$. Let $r, r', r'', r''' > 0$. Let $m = \Omega_{r,r',\theta}\left(\sqrt{\frac{n \log n}{\pi_{\min}}}\right)$, $T = \Omega_{r,r',\theta}(n \log n/m)$, and $\tau = \frac{\theta T m}{n}$. Then, as long as the separation condition in (A.6) holds with m replacing n, we have, with probability at least $1 - O\left(\frac{T}{(m\pi_{\min})^{r''}} + \frac{TK}{n^{r'''}} + \frac{1}{n^{r'}}\right)$, that*

$$\max_{\mathcal{T}} \max_{(x_1,\ldots,x_J)} \delta(\hat{Z}^{\mathsf{GALE}}_{\mathcal{T},(x_1,\ldots,x_J)}, Z) = O\left(\frac{1}{n^r}\right), \tag{A.7}$$

[†]We are not using Lemma A.1.2 as it only shows that the solution of SDP-GV has small norm difference from the ideal clustering matrix, but does not relate this directly to misclustering error.

[‡]detailed in Section A.3

Note that, in the above bound r can taken to be greater than 1. This means that, with high probability, the proportion of misclustered nodes is less than $1/n$ and hence zero, leading to exact recovery. As for computational complexity, note that the separation condition (A.6), with n replaced by m, restricts how small m can be. Consider the simple SBM (2.8) with balanced block sizes for concreteness. In this case, the separation condition essentially dictates, as in the case of spectral clustering, that $m \gg \frac{K^2}{\lambda^2 \alpha_n}$. Thus the remarks made earlier on how large m or T should be chosen apply here as well.

As discussed earlier in Section 2.2, even a naive implementation of GALE will only result in an $O(n^{3/2})$ running time in addition to the time ($\eta_{m,T}$) required to cluster the T random m-subgraphs, whereas a more careful implementation will only add a time to $\eta_{m,T}$ that is nearly linear in T. Since SDPs are notoriously time intensive to solve, this gives us a big saving.

A.3 Details of DGCluster

Here we detail our distance based greedy algorithm DGCluster. The idea behind DGCluster is to note that, if i and j are in the same community, then $\|C_{i\star} - C_{j\star}\| = 0$, and otherwise $\|C_{i\star} - C_{j\star}\| = \sqrt{|\sigma(i)| + |\sigma(j)|} = \Theta(\sqrt{2n/K})$ (when the communities are balanced). Thus we expect to be able to cluster the vertices using $d_{ij} = \|\hat{C}_{proj,i\star} - \hat{C}_{proj,j\star}\|$, namely by starting with a root vertex r_1 and for some threshold γ putting all vertices j satisfying $d_{r_1 j} \leq \gamma$ in the same cluster as r_1, and then picking another root vertex r_2 from the remaining set, and putting all vertices j in the remaining set that satisfy $d_{r_2 j} \leq \gamma$ in the same cluster as r_2, and so on. Here a root vertex r_i may be chosen as one of the vertices with the highest degree in the remaining set (or according to a degree-weighted random sampling scheme), to give importance to highly connected vertices. We also note that depending on the threshold γ the number of blocks we get can vary. In practice, we will start with small γ (yielding a large number of communities), and stop at the smallest γ that gives us $\geq K$ blocks. If we get more than K blocks, we merge, in succession, pairs of blocks having the largest intersection in $\hat{C}$ (relative to their sizes) until we get exactly K blocks. A rule of thumb would be to start with $\gamma = c\sqrt{2n/K}$ with c small and then gradually increase c.

Algorithm 10 A distance based greedy clustering algorithm: DGCluster

Input: C, C', K. Output: $\sigma = \mathsf{DGCluster}(C, C', K)$, a clustering based on distances between the rows of C', with merging guidance from C, if necessary.
Set $\theta = \sqrt{2n/K}$. $c = c_{min} = 0, \delta = 0.01$.
Set $K_{now} = n$;
while flag = TRUE **do**
    $c \leftarrow c + \delta$.
    $\sigma_{temp} \leftarrow \mathsf{NaiveCluster}(C', c\theta)$.
    $K_{now} = \max_i \sigma_{temp}(i)$.
    if $K_{now} \geq K$ **then**
        flag $\leftarrow$ TRUE.
        $\sigma \leftarrow \sigma_{temp}$.
    else
        flag $\leftarrow$ FALSE.
$K_{now} \leftarrow \max_i \sigma(i)$.
while $K_{now} > K$ **do**
    Let $\Gamma_i = \sigma^{-1}(\{i\}), i = 1, \ldots, K_{now}$.
    Compute the (upper triangle of the) matrix R, where

$$R_{ij} \leftarrow \frac{\sum_{k \in \Gamma_i, l \in \Gamma_j} C_{kl}}{|\Gamma_i||\Gamma_j|}.$$

    Pick $(i_\star, j_\star) \in \arg\max_{i,j} R_{ij}$.
    $\sigma \leftarrow \mathsf{Merge}(\sigma, i_\star, j_\star)$.

Algorithm 10 in turn makes use of the following two algorithms.

Algorithm 11 A naive clustering algorithm: NaiveCluster

Input: C, γ : a threshold. Output: $\sigma = \mathsf{NaiveCluster}(C, \gamma)$, a clustering based on distances between the rows of C.
Set `Unassigned` $= \{1, \ldots, n\}$, $b = 1$, $\sigma \equiv 0$.
while `Unassigned` $\neq \emptyset$ **do**
    $i \leftarrow$ a random (uniform or degree-weighted, etc.) index in `Unassigned`.
    $\sigma(i) \leftarrow b$.
    `Unassigned` $\leftarrow$ `Unassigned` $\setminus \{i\}$.
    for $j \in$ `Unassigned` **do**
        Compute $d_{ij} = \|\hat{C}_{i\star} - \hat{C}_{j\star}\|$
        if $d_{ij} \leq \gamma$ **then**
            $\sigma(j) \leftarrow b$
            `Unassigned` $\leftarrow$ `Unassigned` $\setminus \{j\}$.
    $b \leftarrow b + 1$.

Algorithm 12 Merge

Input: σ, a, b. Output: $\sigma = \mathsf{Merge}(\sigma, a, b)$, a clustering with blocks a and b merged
$u \leftarrow \min\{a, b\}, v \leftarrow \max\{a, b\}$
for $i = 1, \ldots, n$ **do**
    if $\sigma(i) = v$ **then**
        $\sigma(i) \leftarrow u$.
    else if $\sigma(i) > v$ **then**
        $\sigma(i) \leftarrow \sigma(i) - 1$.

A.4 Miscellaneous proofs

Proof of Proposition 2.3.1. Since both Z, Z' are $0, 1$-valued, we can safely replace the count by Frobenius norm squared, i.e.

$$\delta(Z, Z') = \inf_{Q \text{ perm.}} \frac{1}{n} \|ZQ - Z'\|_F^2.$$

Now, note that $(ZQ)(ZQ)^\top = ZZ^\top$ for all permutation matrices Q. Thus

$$\begin{aligned}
\|C - C'\|_F &= \|(ZQ)(ZQ)^\top - Z'Z'^\top\|_F \\
&= \|(ZQ - Z')(ZQ)^\top + Z'((ZQ)^\top - Z'^\top\|_F \\
&\leq \|(ZQ - Z')(ZQ)^\top\|_F + \|Z'((ZQ)^\top - Z'^\top\|_F \\
&\leq \|ZQ - Z'\|_F \|ZQ^\top\|_2 + \|Z'\|_2 \|(ZQ)^\top - Z'^\top\|_F.
\end{aligned}$$

But $\|Z'\|_2^2$ is the maximum eigenvalue of $Z'^\top Z'$ which is diagonal with its maximum diagonal entry being the size of the largest cluster under Z'. Thus $\|Z'\|_2^2$ equals the size of the largest cluster under Z' and so is trivially upper bounded by n. Same goes for $\|ZQ^\top\|_2^2$. Therefore we get

$$\|C - C'\|_F \leq 2\sqrt{n}\|ZQ - Z'\|_F.$$

Squaring this, and taking infimum over all permutation matrices Q in the right hand side, we obtain the claimed inequality. □

Lemma A.4.1 (Thresholding PACE)**.** *Let $\hat{C}_\eta := [\hat{C} > \eta]$. We have*

$$\mathbb{E}\tilde{\delta}(\hat{C}_\eta, C) \leq \frac{\mathbb{E}\tilde{\delta}(\hat{C}_\eta, C)}{\min\{\eta^2, (1-\eta)^2\}}. \tag{A.8}$$

In particular, for $\eta = 1/2$, we have

$$\mathbb{E}\tilde{\delta}(\hat{C}_{1/2}, C) \leq 4\mathbb{E}\tilde{\delta}(\hat{C}, C).$$

Proof. Note that $|(\hat{C}_\eta)_{ij} - C_{ij}| \sim \text{Ber}(q_{ij})$, where

$$q_{ij} = C_{ij}\mathbb{P}(\hat{C}_{ij} \leq \eta) + (1 - C_{ij})\mathbb{P}(\hat{C}_{ij} > \eta).$$

Thus $\mathbb{E}((\hat{C}_\eta)_{ij} - C_{ij})^2 = q_{ij}$. Now

$$\mathbb{E}(\hat{C}_{ij} - C_{ij})^2 = \mathbb{E}(\hat{C}_{ij} - C_{ij})^2\mathbf{1}_{\{\hat{C}_{ij} \leq \eta\}} + \mathbb{E}(\hat{C}_{ij} - C_{ij})^2\mathbf{1}_{\{\hat{C}_{ij} \leq \eta\}}.$$

Therefore

$$\mathbb{E}(\hat{C}_{ij} - C_{ij})^2 \geq \begin{cases} (1-\eta)^2\mathbb{P}(\hat{C}_{ij} \leq \eta) = (1-\eta)^2 q_{ij}, & \text{if } C_{ij} = 1 \\ \eta^2\mathbb{P}(\hat{C}_{ij} > \eta) = \eta^2 q_{ij}, & \text{if } C_{ij} = 0. \end{cases}$$

This means

$$\mathbb{E}((\hat{C}_\eta)_{ij} - C_{ij})^2 \leq \frac{\mathbb{E}(\hat{C}_{ij} - C_{ij})^2}{\min\{\eta^2, (1-\eta)^2\}},$$

which, on summing over i, j, gives us the desired bound. □

A similar thresholding result is true for GALE.

Lemma A.4.2 (Rounding GALE)**.** *Consider an estimated cluster membership matrix $\hat{Z} \in [0,1]^{n\times K}$ and the true cluster membership matrix $Z \in \{0,1\}^{n\times K}$. Then $\|round(\hat{Z}) - Z\|_1 \leq 2\|\hat{Z} - Z\|_1$.*

Proof. Let $S := \{(i,k) \in [n] \times [K] : |\hat{Z}_{ik} - Z_{ik}| \geq 1/2\}$. Then by Markov's inequality, we have $|S| \leq 2\|\hat{Z} - Z\|_1$. Note that for $(i,k) \in [n] \times [K] \setminus S$, $\text{round}(\hat{Z}_{ik}) = Z_{ik}$. Thus

$$\|\text{round}(\hat{Z}) - Z\|_1 = \sum_{(i,k)\in S} |\text{round}(\hat{Z}_{ik}) - Z_{ik}| \leq |S| \leq 2\|\hat{Z} - Z\|_1$$

□

We shall need the following Bernstein-type concentration result for hypergeometric random variables.

Lemma A.4.3 (Corollary 1 of [33] (restated here using slightly different notation))**.** *Consider a Hypergeometric(k, ℓ, L) random variable H. Let $\mu = \ell/L$, $\sigma^2 = \mu(1-\mu)$, $f = (k-1)/(L-1)$. Then for $\lambda > 0$,*

$$\mathbb{P}\left(\sqrt{k}\left(\frac{H}{k} - \mu\right) > \lambda\right) \leq \exp\left(-\frac{\lambda^2/2}{\sigma^2(1-f) + \frac{\lambda}{3\sqrt{k}}}\right).$$

Let us deduce from this two convenient Chernoff type bounds on the upper and lower tail of a hypergeometric variable.

Lemma A.4.4. *Consider a random variable* $H \sim \mathsf{Hypergeometric}(k, \ell, L)$. *Then for* $0 < \epsilon < 1$, *we have*

$$\begin{aligned}\max\{\mathbb{P}\left(H < (1-\epsilon)\mathbb{E}H\right), \mathbb{P}\left(H > (1+\epsilon)\mathbb{E}H\right)\} &\leq \exp\left(-\frac{\epsilon^2 \mathbb{E}H}{2(1+\epsilon/3)}\right) \\ &\leq \exp\left(-\frac{\epsilon^2 \mathbb{E}H}{4}\right).\end{aligned}$$

Proof. Note that $k - H \sim \mathsf{Hypergeometric}(k, L-\ell, L)$. Using Lemma A.4.3 we get

$$\begin{aligned}\mathbb{P}\left(\frac{H}{k} - \frac{\ell}{L} < -\frac{\lambda}{\sqrt{k}}\right) &= \mathbb{P}\left(\sqrt{k}\left(\frac{k-H}{k} - \frac{L-\ell}{L}\right) > \lambda\right) \\ &\leq \exp\left(-\frac{\lambda^2/2}{\frac{\ell}{L} + \frac{\lambda}{3\sqrt{k}}}\right).\end{aligned}$$

Taking $\frac{\lambda}{\sqrt{k}} = \frac{\ell}{L}\epsilon$, we get

$$\mathbb{P}\left(H < \frac{k\ell}{L}(1-\epsilon)\right) \leq \exp\left(-\frac{\ell^3 k \epsilon^2/(2L^2)}{(1+\epsilon/3)\ell/(L)}\right) = \exp\left(-\frac{\epsilon^2 k \ell}{2(1+\epsilon/3)L}\right).$$

This gives us the desired bound for the lower tail, the upper tail can be handled similarly. □

Analysis of PACE under stochastic blockmodel

Recall that we need to know how an algorithm performs on a randomly selected subgraph under our subgraph selection procedure. Here we will discuss how one can obtain such guarantees under the stochastic blockmodel. This will require us to understand the behavior of sizes of different communities in a (randomly) chosen subgraph.

Community sizes in random m-subgraphs

(The results of this subsection do not depend on any modeling assumption.) Let S be a random m-subgraph, and set $m/n = q$. Then if $(n_1^{(S)}, \ldots, n_K^{(S)})$ is the cluster size vector for Z_S, then clearly $n_k^{(S)} \sim \mathsf{Hypergeometric}(m, n_k, n)$ and therefore, by Lemma A.4.4,

$$\begin{aligned}\mathbb{P}(n_k^{(S)} \leq n_{\min} q - \Delta) &\leq \exp\left(-\frac{((n_k - n_{\min})q + \Delta)^2}{4 n_k q}\right) \\ &\leq \exp\left(-\frac{\Delta^2}{4 n_{\max} q}\right).\end{aligned}$$

Therefore, by union bound,

$$\begin{aligned}\mathbb{P}(n_{\min}^{(S)} \le n_{\min}q - \Delta) &\le \sum_k \mathbb{P}(n_k^{(S)} \le n_{\min}q - \Delta) \\ &\le K \exp\left(-\frac{\Delta^2}{4n_{\max}q}\right).\end{aligned}$$

Choosing $\Delta = \sqrt{4r' n_{\max} q \log n}$, where $r' > 0$, we see that with probability at least $1 - \frac{K}{n^{r'}}$ we have

$$n_{\min}^{(S)} \ge n_{\min}q - \sqrt{4r' n_{\max} q \log n}. \tag{A.9}$$

Similarly, we can show that

$$\mathbb{P}(n_{\max}^{(S)} \ge n_{\max}q + \Delta) \le K \exp\left(-\frac{\Delta^2}{4n_{\max}q}\right),$$

and then, taking $\Delta = \sqrt{4r' n_{\max} q \log n}$, conclude that with probability at least $1 - \frac{K}{n^{r'}}$

$$n_{\max}^{(S)} \le n_{\max}q + \sqrt{4r' n_{\max} q \log n}. \tag{A.10}$$

Community sizes in 1-hop ego neighborhoods

Let now S be a randomly chosen 1-hop ego neighborhood. Note that the size of the k-th block in this neighborhood satisfies

$$n_k^{(S)} = \sum_{j\,:\,\sigma(j)=k} X_j,$$

where $X_j = \mathbf{1}_{\{j \in S\}}$.

Now it is not hard to see that the X_j's are independent conditional R, the root of S. We also have that $\mathbb{E}(X_j \,|\, R) = B_{\sigma(j)\sigma(R)}\mathbf{1}(j \ne R)$. It follows that

$$\mathbb{E}(n_k^{(S)} \,|\, R) = \sum_{j\,:\,\sigma(j)=k} \mathbb{E}(X_j \,|\, R) = \sum_{j\,:\,\sigma(j)=k} B_{\sigma(j)\sigma(R)}\mathbf{1}_{\{j \ne R\}},$$

which means that

$$(n_{\min} - 1)B_* \le (n_k - 1)B_\star \le \mathbb{E}(n_k^{(S)} \,|\, R) \le n_k B_\# \le n_{\max} B_\#.$$

Therefore, by Chernoff's inequality,

$$\begin{aligned}\mathbb{P}(n_k^{(S)} \le (n_{\min} - 1)B_\star - \Delta \,|\, R) &\le \exp\left(-\frac{(\mathbb{E}(n_k^{(S)} \,|\, R) - n_{\min}B_\star + \Delta)^2}{2\mathbb{E}(n_k^{(S)} \,|\, S)}\right) \\ &\le \exp\left(-\frac{\Delta^2}{2n_{\max}B_\#}\right).\end{aligned}$$

Since the right hand side does not depend on R, we can take expectations of both sides with respect to R to get

$$\mathbb{P}(n_k^{(S)} \leq (n_{\min} - 1)B_\star - \Delta) \leq \exp\left(-\frac{\Delta^2}{2n_{\max}B_\#}\right).$$

This implies, by an application of the union bound, that

$$\mathbb{P}(n_{\min}^{(S)} \leq (n_{\min} - 1)B_\star - \Delta) \leq K\exp\left(-\frac{\Delta^2}{2n_{\max}B_\#}\right).$$

Choosing, $\Delta = \sqrt{2r'n_{\max}B_\# \log n}$ we conclude, with probability at least $1 - \frac{K}{n^{r'}}$, that

$$n_{\min}^{(S)} \geq (n_{\min} - 1)B_\star - \sqrt{2r'n_{\max}B_\# \log n}. \tag{A.11}$$

One can prove similarly that

$$\mathbb{P}(n_{\max}^{(S)} \geq n_{\max}B_\# + \Delta) \leq K\exp\left(-\frac{\Delta^2}{3n_{\max}B_\#}\right),$$

and then, taking $\Delta = \sqrt{3r'n_{\max}B_\# \log n}$, conclude, with probability at least $1 - \frac{K}{n^{r'}}$, that

$$n_{\max}^{(S)} \leq n_{\max}B_\# + \sqrt{3r'n_{\max}B_\# \log n}. \tag{A.12}$$

Analysis of adjacency spectral clustering

We shall use the community size estimates from the previous subsection along with Lemma A.1.1.

Random m-subgraphs: From (A.9) and (A.10) we have, with probability at least $1 - \frac{2K}{n^{r'}}$, that

$$\frac{n_{\max}^{(S)}}{(n_{\min}^{(S)})^2} \leq \frac{n_{\max}q + \sqrt{4r'n_{\max}q\log n}}{(n_{\min}q - \sqrt{4r'n_{\max}q\log n})^2} \leq \frac{n_{\max}}{n_{\min}^2 q}C_{n,m,r'},$$

where

$$C_{n,m,r'} = \left(1 + \sqrt{\frac{4r'\log n}{n_{\max}q}}\right)\left(1 - \sqrt{\frac{4r'n_{\max}\log n}{n_{\min}^2 q}}\right)^{-2}.$$

Note that $C_{n,m,r'}$ stays bounded, e.g., by 6, if $\frac{n_{\min}^2 q}{n_{\max}\log n} \geq 16r'$. In fact, it approaches 1 as $\frac{n_{\min}^2 q}{n_{\max}\log n} \to \infty$. Thus, from (A.1), with probability at least $1 - \frac{2K}{n^{r'}}$ over randomness in S and with probability at least $1 - \frac{1}{m^r}$ over randomness in A_S, we have

$$\frac{1}{2}\delta(\hat{Z}_S, Z_S) \leq \min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^2\alpha_n}\frac{n_{\max}}{n_{\min}^2 q}C_{n,m,r'}, 1\right\}.$$

We conclude that

$$\frac{1}{2}\mathbb{E}\delta(\hat{Z}_S, Z_S) \le \min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^2\alpha_n}\frac{n_{\max}}{n_{\min}^2 q}C_{n,m,r'}, 1\right\} + \frac{2K}{n^{r'}} + \frac{1}{m^r}. \tag{A.13}$$

1-hop ego neighborhoods: By (A.1), given S (non-empty), with probability at least $1-\frac{1}{|S|^r}$ over the randomness in A_S, we have

$$\frac{1}{2}\delta(\hat{Z}_S, Z_S) \le \min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^2\alpha_n}\frac{n_{\max}^{(S)}}{(n_{\min}^{(S)})^2}, 1\right\} =: J_S.$$

So

$$\frac{1}{2}\mathbb{E}[\delta(\hat{Z}_S, Z_S)\,|\,S] \le J_S + \frac{1}{|S|^r}.$$

Therefore

$$\frac{1}{2}\mathbb{E}\delta(\hat{Z}_S, Z_S)\mathbf{1}_{(|S|\ge m_\star)} \le \mathbb{E}J_S\mathbf{1}_{(|S|\ge m_\star)} + \mathbb{E}\left(\frac{\mathbf{1}_{(|S|\ge m_\star)}}{|S|^r}\right).$$

From (A.11) and (A.12) we have, with probability at least $1-\frac{2K}{n^{r'}}$, that

$$\begin{aligned}\frac{n_{\max}^{(S)}}{(n_{\min}^{(S)})^2} &\le \frac{n_{\max}B_\# + \sqrt{3r' n_{\max}B_\# \log n}}{((n_{\min}-1)B_\star - \sqrt{2r' n_{\max}B_\# \log n})^2} \\ &\le \frac{n_{\max}B_\#}{n_{\min}^2 B_\star^2} D_{n,B,r'} = \frac{n_{\max}}{n_{\min}^2\lambda^2\alpha_n} D_{n,B,r'},\end{aligned}$$

where

$$\begin{aligned}D_{n,B,r'} &= \left(1+\sqrt{\frac{3r'\log n}{n_{\max}B_\#}}\right)\left(1-\frac{1}{n_{\min}}-\sqrt{\frac{2r' n_{\max}B_\#\log n}{n_{\min}^2 B_\star^2}}\right)^{-2} \\ &= \left(1+\sqrt{\frac{3r'\log n}{n_{\max}\alpha_n}}\right)\left(1-\frac{1}{n_{\min}}-\sqrt{\frac{2r' n_{\max}\log n}{n_{\min}^2\lambda^2\alpha_n}}\right)^{-2}.\end{aligned}$$

Note that $D_{n,B,r'}$ stays bounded, e.g., by $16(1+\sqrt{\frac{3}{8}})$, if $\frac{n_{\min}^2\lambda^2\alpha_n}{n_{\max}\log n} \ge 8r'$. In fact, it approaches 1 as $\frac{n_{\min}^2\lambda^2\alpha_n}{n_{\max}\log n} \to \infty$. Thus

$$\mathbb{E}J_S\mathbf{1}_{(|S|\ge m_\star)} \le \min\left\{c^{-1}(2+\epsilon)\frac{K}{\lambda^4\alpha_n^2}\frac{n_{\max}}{n_{\min}^2}D_{n,B,r'}, 1\right\} + \frac{2K}{n^{r'}}.$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\left(\frac{\mathbf{1}_{(|S|\geq m_\star)}}{|S|^r}\right) &\leq \mathbb{E}\left(\frac{\mathbf{1}_{(|S|\geq 1)}}{|S|^r}\right) \\
&= \mathbb{E}\left(\frac{\mathbf{1}_{(\frac{(n-1)B_\star}{2}>|S|\geq 1)}}{|S|^r}\right) + \mathbb{E}\left(\frac{\mathbf{1}_{(|S|\geq \frac{(n-1)B_\star}{2})}}{|S|^r}\right) \\
&\leq \mathbb{P}\left(|S| < \frac{(n-1)B_\star}{2}\right) + \left(\frac{2}{(n-1)B_\star}\right)^r \\
&\leq \exp\left(-\frac{nB_\star^2}{16B_\#}\right) + \left(\frac{2}{(n-1)B_\star}\right)^r \qquad \text{(by (2.17))} \\
&\leq \frac{16B_\#}{nB_\star^2} + \left(\frac{4}{nB_\star}\right)^r = \frac{16}{n\lambda^2\alpha_n} + \left(\frac{4}{n\lambda\alpha_n}\right)^r.
\end{aligned}$$

So, finally, we have

$$\begin{aligned}
\frac{1}{2}\mathbb{E}\delta(\hat{Z}_S, Z_S)\mathbf{1}_{(|S|\geq m_\star)} \leq \min\left\{c^{-1}\frac{K}{\lambda^4\alpha_n^2}\frac{n_{\max}}{n_{\min}^2}D_{n,B,r'}, 1\right\} + \frac{2K}{n^{r'}} \\
+ \frac{16}{n\lambda^2\alpha_n} + \left(\frac{4}{n\lambda\alpha_n}\right)^r. \qquad \text{(A.14)}
\end{aligned}$$

Analysis of SDP

In the setting of Corollary A.1.3, for a random m-subgraph S we have $\min_k B_{kk} \geq \frac{a}{n} = \frac{\tilde{a}}{m}, \max_{k\neq k'} B_{kk'} \leq \frac{b}{n} = \frac{\tilde{b}}{m}$ where $\tilde{a} = \frac{am}{n}$, $\tilde{b} = \frac{bm}{n}$. Let $n_{kk'}^{(S)}$ denote the number of pairs of vertices in the subgraph S such that one of them is from community k and the other is from community k'. Now, by Lemma A.4.4,

$$\mathbb{P}(n_k^{(S)} \geq cn_k m/n) \leq e^{-(c-1)^2 m\pi_{\min}/4}.$$

So, with probability at least $1 - Ke^{-(c-1)^2 m\pi_{\min}/4}$, we have, for all $1 \leq k \leq k' \leq K$, that

$$n_{kk'}^{(S)} = \begin{cases} n_k^{(S)} n_{k'}^{(S)} \leq \frac{c^2 n_k n_k' m^2}{n^2} \leq \frac{c_1 n_{kk'} m^2}{n^2}, & \text{for } k < k', \\ \frac{1}{2}n_k^{(S)}(n_k^{(S)} - 1) \leq \frac{cn_k m}{2n}(\frac{cn_k m}{n} - 1) \leq \frac{c_1 n_{kk} m^2}{n^2}, & \text{for } k = k', \end{cases}$$

where $c_1 = c^2\left(1 + \frac{\frac{cm}{n}-1}{\frac{cm}{n}(n\pi_{\min}-1)}\right)$. Similarly, we can show, with probability at least $1 - Ke^{-(1-c')^2 m\pi_{\min}/4}$, that

$$n_{kk'}^{(S)} \geq \begin{cases} \frac{c_2 n_{kk'} m^2}{n^2}, & \text{for } k < k', \\ \frac{c_2 n_{kk} m^2}{n^2}, & \text{for } k = k', \end{cases}$$

where $c_2 = (c')^2\left(1 + \frac{\frac{c'm}{n}-1}{\frac{c'm}{n}(n\pi_{\max}-1)}\right)$. Let

$$\mathcal{G} = \{S \mid c_2 n_{kk'} m^2/n^2 \leq n_{kk'}^{(S)} \leq c_1 n_{kk'} m^2/n^2, \text{ for all } 1 \leq k \leq k' \leq K\}.$$

Given S, the expected variance of edges in S is $\frac{2}{m(m-1)}\sum_{1\le k\le k'\le K} B_{kk'}(1-B_{kk'})n_{kk'}^{(S)} =: \frac{\tilde{g}}{m}$, say. Then, if $S \in \mathcal{G}$, we have

$$\frac{c_2 g}{n} \le \frac{c_2 m(n-1)}{(m-1)n} \times \frac{g}{n} \le \frac{\tilde{g}}{m} \le \frac{c_1 m(n-1)}{(m-1)n} \times \frac{g}{n} = \frac{m}{n^2} \times \bar{g}.$$

and so, using our assumptions on a, b, g, we get that $\tilde{g} \ge 9$ and $\tilde{a}, \tilde{b}$ satisfy $(\tilde{a}-\tilde{b})^2 \ge 484\epsilon^{-2}\frac{m^2}{n^2}\bar{g} \ge 484\epsilon^{-2}\tilde{g}$. Therefore, using Lemma A.1.2 on the subgraph S, we conclude that, given $S \in \mathcal{G}$, we have

$$\frac{1}{m^2}\|\hat{C}^{(S)} - \hat{C}^{(S)}\|_F^2 \le \epsilon,$$

with probability at least $1 - e^3 5^{-m}$, where $\hat{C}^{(S)}$ is a solution of SDP-GV on the subgraph S. As $\mathbb{P}(S \notin \mathcal{G}) \le Ke^{-(c-1)^2 m\pi_{\min}/4} + Ke^{-(1-c')^2 m\pi_{\min}/4}$, we conclude that

$$\mathbb{E}\tilde{\delta}(\hat{C}^{(S)}, C^{(S)}) \le \epsilon + e^3 5^{-m} + Ke^{-(c-1)^2 m\pi_{\min}/4} + Ke^{-(1-c')^2 m\pi_{\min}/4}.$$

Proofs of some results for GALE

In this section we collect the proofs of the auxiliary results we presented in Section 2.5.

Proof of Lemma 2.5.3. Note that the N_i's are i.i.d. $\mathsf{Binomial}(T; m/n)$ random variables and hence, by Chernoff's inequality, we have,

$$\mathbb{P}(N_i < \tau) \le e^{-(1-\theta)^2 Tm/2n}. \tag{A.15}$$

Another use of Chernoff gives, for $\delta > 0$,

$$\mathbb{P}\left(\sum_i \mathbf{1}_{\{N_i<\tau\}} > n\mathbb{P}(N_1<\tau)(1+\delta)\right) \le \exp(-\delta^2 n\mathbb{P}(N_1<\tau)/3).$$

Setting $\delta = \sqrt{\frac{3r\log n}{n\mathbb{P}(N_1<\tau)}}$, we conclude that with probability at least $1 - \frac{1}{n^r}$,

$$\begin{aligned}\sum_i \mathbf{1}_{\{N_i<\tau\}} &\le n\mathbb{P}(N_1<\tau) + \sqrt{3rn\mathbb{P}(N_1<\tau)\log n} \\ &\le ne^{-(1-\theta)^2 Tm/2n}\left(1 + \sqrt{\frac{3r\log n}{ne^{-(1-\theta)^2 Tm/2n}}}\right).\end{aligned}$$

□

Proof of Lemma 2.5.4. We will first show that for three different nodes a, b, c in $\mathcal{S}_{m,T}$ the overlap variables Y_{ab} and Y_{ac} are independent. This is an immediate consequence of the following two facts:

(i) Y_{ab} and $y^{(a)}$ are independent which follows from the observation that for any m-subset S of $[n]$, $Y_{ab} \mid y^{(a)} = \mathbf{1}_S \sim \mathsf{Hypergeometric}(m, m, n)$.

(ii) Given $y^{(a)}$, the overlaps Y_{ab} and Y_{ac} are independent, which follows from the fact that $y^{(b)}$ and $y^{(c)}$ are independent.

From the discussion above, $Y_{ab} \sim \mathsf{Hypergeometric}(m, m, n)$. Using Lemma A.4.4 we get, with $\epsilon = 1/2$, we get that

$$\mathbb{P}(a \sim b) = \mathbb{P}\left(Y_{ab} \geq \left\lceil \frac{m^2}{2n} \right\rceil\right) \geq 1 - \exp\left(-\frac{m^2}{16n}\right).$$

□

Proof of Lemma 2.5.6. Note first that $n_k^{(S)} \mid Y_{ab} \sim \mathsf{Hypergeometric}(Y_{ab}, n\pi_k, n)$, which follows from the fact that given Y_{ab}, $S_a \cap S_b$ is distributed as a uniform Y_{ab}-subset of $[n]$*. Now, Lemma A.4.4 gives us

$$\mathbb{P}\left(n_k^{(S)} < Y_{ab}\pi_k \left(1 - \sqrt{\frac{4r \log n}{Y_{ab}\pi_k}}\right) \Big| Y_{ab}\right) \leq \frac{1}{n^r}.$$

Therefore the same bound holds for the unconditional probability. Another application of Lemma A.4.4 for $Y_{ab} \sim \mathsf{Hypergeometric}(m, m, n)$ gives us

$$\mathbb{P}\left(Y_{ab} < \frac{m^2}{n}\left(1 - \sqrt{\frac{4rn \log n}{m^2}}\right)\right) \leq \frac{1}{n^r}.$$

Using these two bounds, we conclude that with probability at least $1 - \frac{2}{n^r}$, we have that

$$n_k^{(S)} \geq \frac{m^2\pi_k}{n}\left(1 - O\left(\sqrt{\frac{n \log n}{m^2\pi_k}}\right)\right),$$

as long as $\frac{m^2\pi_k}{n \log n}$ is large enough ($\frac{m^2\pi_k}{n \log n} \geq 20r$ suffices to make the RHS above ≥ 0). □

*Thanks to Satyaki Mukherjee for this observation, which makes the proof considerably shorter than our original approach based on concentration inequalities for sums of dependent Bernoulli random variables.

Bibliography

[1] L. A. Adamic and N. Glance. "The political blogosphere and the 2004 US election: divided they blog". In: *Proceedings of the 3rd international workshop on Link discovery*. ACM. 2005, pp. 36–43.

[2] E. M. Airoldi, T. B. Costa, and S. H. Chan. "Stochastic blockmodel approximation of a graphon: Theory and consistent estimation". In: *Advances in Neural Information Processing Systems*. 2013, pp. 692–700.

[3] E. M. Airoldi et al. "Mixed membership stochastic blockmodels". In: *Journal of Machine Learning Research* 9.Sep (2008), pp. 1981–2014.

[4] E. M. Airoldi et al. "Mixed membership stochastic blockmodels". In: *Advances in Neural Information Processing Systems*. 2009, pp. 33–40.

[5] D. A. Aldous. "Representations for partially exchangeable arrays of random variables". In: *Journal of Multivariate Analysis* 11.4 (1981), pp. 581–598.

[6] S. Aliakbary et al. "Distance metric learning for complex networks: Towards size-independent comparison of network structures". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.2 (2015), p. 023111.

[7] A. A. Amini and E. Levina. "On semidefinite relaxations for the block model". In: *arXiv preprint arXiv:1406.5647* (2014).

[8] A. A. Amini et al. "Pseudo-likelihood methods for community detection in large sparse networks". In: *The Annals of Statistics* 41.4 (2013), pp. 2097–2122.

[9] A. Anandkumar et al. "A tensor approach to learning mixed membership community models". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 2239–2312.

[10] J. Bai. "Estimation of a change point in multiple regression models". In: *Review of Economics and Statistics* 79.4 (1997), pp. 551–563.

[11] S. Bhattacharyya and P. J. Bickel. "Community detection in networks using graph distance". In: *arXiv preprint arXiv:1401.3915* (2014).

[12] S. Bhattacharyya and P. J. Bickel. "Spectral clustering and block models: A review and a new algorithm". In: *Statistical Analysis for High-Dimensional Data*. Springer, 2016, pp. 67–90.

[13] P. J. Bickel and A. Chen. "A nonparametric view of network models and Newman–Girvan and other modularities". In: *Proceedings of the National Academy of Sciences* 106.50 (2009), pp. 21068–21073.

[14] N. Binkiewicz, J. T. Vogelstein, and K. Rohe. "Covariate assisted spectral clustering". In: *arXiv preprint arXiv:1411.2158* (2014).

[15] D. M. Blei and J. D. Lafferty. "A correlated topic model of science". In: *The Annals of Applied Statistics* (2007), pp. 17–35.

[16] D. M. Blei and J. D. Lafferty. "Dynamic topic models". In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 113–120.

[17] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[18] B. Bollobás, S. Janson, and O. Riordan. "The phase transition in inhomogeneous random graphs". In: *Random Structures & Algorithms* 31.1 (2007), pp. 3–122.

[19] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[20] E. Brodsky and B. S. Darkhovsky. *Nonparametric methods in change point problems*. Vol. 243. Springer Science & Business Media, 2013.

[21] T. T. Cai and X. Li. "Robust and computationally feasible community detection in the presence of arbitrary outlier nodes". In: *The Annals of Statistics* 43.3 (2015), pp. 1027–1059.

[22] J. Cape, M. Tang, and C. E. Priebe. "The Kato–Temple inequality and eigenvalue concentration with applications to graph inference". In: *Electronic Journal of Statistics* 11.2 (2017), pp. 3954–3978.

[23] E. Carlen. "Trace inequalities and quantum entropy: an introductory course". In: *Entropy and the quantum* 529 (2010), pp. 73–140.

[24] M. S. Charikar. "Similarity Estimation Techniques from Rounding Algorithms". In: *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*. Montreal, Quebec, Canada, 2002, pp. 380–388.

[25] S. Chatterjee. "Matrix estimation by Universal Singular Value Thresholding". In: *Ann. Statist.* 43.1 (Feb. 2015), pp. 177–214.

[26] H. Chen and N. Zhang. "Graph-based change-point detection". In: *Ann. Statist.* 43.1 (Feb. 2015), pp. 139–176.

[27] N. Eagle and A. S. Pentland. "Reality mining: sensing complex social systems". In: *Personal and ubiquitous computing* 10.4 (2006), pp. 255–268.

[28] C. Gao, Y. Lu, and H. H. Zhou. "Rate-optimal graphon estimation". In: *Ann. Statist.* 43.6 (Dec. 2015), pp. 2624–2652.

[29] E. N. Gilbert. "Random graphs". In: *The Annals of Mathematical Statistics* 30.4 (1959), pp. 1141–1144.

[30] C. E. Ginestet et al. "Hypothesis Testing For Network Data in Functional Neuroimaging". In: *arXiv preprint arXiv:1407.5525* (2014).

[31] M. A. Girshick and H. Rubin. "A Bayes approach to a quality control model". In: *The Annals of mathematical statistics* (1952), pp. 114–125.

[32] P. K. Gopalan and D. M. Blei. "Efficient discovery of overlapping communities in massive networks". In: *Proceedings of the National Academy of Sciences* 110.36 (2013), pp. 14534–14539.

[33] E. Greene and J. A. Wellner. "Exponential bounds for the hypergeometric distribution". In: *Bernoulli* 23.3 (Aug. 2017), pp. 1911–1950.

[34] O. Guédon and R. Vershynin. "Community detection in sparse networks via Grothendieck's inequality". In: *Probability Theory and Related Fields* (2015), pp. 1–25.

[35] Z. Harchaoui, E. Moulines, and F. R. Bach. "Kernel change-point analysis". In: *Advances in neural information processing systems*. 2009, pp. 609–616.

[36] J. A. Hartigan. "Direct clustering of a data matrix". In: *Journal of the American Statistical Association* 67.337 (1972), pp. 123–129.

[37] P. W. Holland, K. B. Laskey, and S. Leinhardt. "Stochastic blockmodels: First steps". In: *Social networks* 5.2 (1983), pp. 109–137.

[38] D. N. Hoover. "Relations on probability spaces and arrays of random variables". In: *Technical report, Institute of Advanced Study, Princeton.* (1979).

[39] Barry James, Kang Ling James, and David Siegmund. "Asymptotic approximations for likelihood ratio tests and confidence regions for a change-point in the mean of a multivariate normal distribution". In: *Statistica Sinica* (1992), pp. 69–90.

[40] B. Karrer and M. E. J. Newman. "Stochastic blockmodels and community structure in networks". In: *Physical Review E* 83.1 (2011), p. 016107.

[41] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics (NRL)* 2.1-2 (1955), pp. 83–97.

[42] C. M. Le, E. Levina, and R. Vershynin. "Sparse random graphs: regularization and concentration of the Laplacian". In: *arXiv preprint arXiv:1502.03049* (2015).

[43] J. Lei and A. Rinaldo. "Consistency of spectral clustering in stochastic block models". In: *The Annals of Statistics* 43.1 (2015), pp. 215–237.

[44] J. B. Lewis et al. *Voteview: Congressional Roll-Call Votes Database*. 2017. URL: https://voteview.com/.

[45] T. Li, E. Levina, and J. Zhu. "Network cross-validation by edge sampling". In: *arXiv preprint arXiv:1612.04717* (2016).

[46] A. Lung-Yut-Fong, C. Lévy-Leduc, and O. Cappé. "Homogeneity and change-point detection tests for multivariate data using rank statistics". In: *arXiv preprint arXiv:1107.1971* (2011).

[47] P. Mahadevan et al. "Systematic Topology Analysis and Generation Using Degree Correlations". In: *SIGCOMM Comput. Commun. Rev.* 36.4 (Aug. 2006), pp. 135–146.

[48] X. Mao, P. Sarkar, and D. Chakrabarti. "Estimating Mixed Memberships with Sharp Eigenvector Deviations". In: *arXiv preprint arXiv:1709.00407* (2017).

[49] P.-A. G. Maugis, S. C. Olhede, and P. J. Wolfe. "Topology reveals universal features for network comparison". In: *arXiv preprint arXiv:1705.05677* (2017).

[50] J. J. McAuley and J. Leskovec. "Learning to Discover Social Circles in Ego Networks." In: *NIPS*. Vol. 2012. 2012, pp. 548–56.

[51] F. McSherry. "Spectral partitioning of random graphs". In: *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*. IEEE. 2001, pp. 529–537.

[52] J. Moody and P. J. Mucha. "Portrait of political party polarization". In: *Network Science* 1.1 (2013), pp. 119–121.

[53] E. Mossel, J. Neeman, and A. Sly. "A proof of the block model threshold conjecture". In: *Combinatorica* (2013), pp. 1–44.

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

[55] S. S. Mukherjee, P. Sarkar, and P. J. Bickel. "Two provably consistent divide and conquer clustering algorithms for large networks". In: *arXiv preprint arXiv:1708.05573* (2017).

[56] S. S. Mukherjee, P. Sarkar, and L. Lin. "On clustering network-valued data". In: *Advances in Neural Information Processing Systems*. 2017, pp. 7074–7084.

[57] M. Newman. *Networks: an introduction*. Oxford university press, 2010.

[58] M. E. J. Newman. "Assortative mixing in networks". In: *Phys. Rev. Lett.* 89.20 (2002), p. 208701.

[59] M. E. J. Newman. "The Structure and Function of Complex Networks". In: *SIAM review* 45.2 (2003), pp. 167–256.

[60] M. E. J. Newman and A. Clauset. "Structure and inference in annotated networks". In: *arXiv preprint arXiv:1507.04001* (2015).

[61] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Physical review E* 69.2 (2004), p. 026113.

[62] K. Nowicki and T. A. B. Snijders. "Estimation and Prediction for Stochastic Block-structures". In: *Journal of the American Statistical Association* 96.455 (Sept. 2001), pp. 1077–1087. ISSN: 0162-1459.

[63] S. C. Olhede and P. J. Wolfe. "Network histograms and universality of blockmodel approximation". In: *Proceedings of the National Academy of Sciences of the Unites States of America* 111.41 (2014), pp. 14722–14727.

[64] R. I. Oliveira. "Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges". In: *arXiv preprint arXiv:0911.0600* (2009).

[65] E. S. Page. "Continuous inspection schemes". In: *Biometrika* 41.1/2 (1954), pp. 100–115.

[66] E. S. Page. "On problems in which a change in a parameter occurs at an unknown point". In: *Biometrika* 44.1/2 (1957), pp. 248–252.

[67] Y. Park, C. E. Priebe, and A. Youssef. "Anomaly detection in time series of graphs using fusion of graph invariants". In: *IEEE journal of selected topics in signal processing* 7.1 (2013), pp. 67–75.

[68] L. Peel and A. Clauset. "Detecting Change Points in the Large-Scale Structure of Evolving Networks." In: *AAAI*. 2015, pp. 2914–2920.

[69] T. Qin and K. Rohe. "Regularized spectral clustering under the degree-corrected stochastic blockmodel". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3120–3128.

[70] S. Ranshous et al. "Anomaly detection in dynamic networks: a survey". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 7.3 (2015), pp. 223–247.

[71] K. Rohe, S. Chatterjee, and B. Yu. "Spectral clustering and the high-dimensional stochastic blockmodel". In: *Ann. Statist.* 39.4 (2011), pp. 1878–1915.

[72] S. Roy, Y. Atchadé, and G. Michailidis. "Change point estimation in high dimensional Markov random-field models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.4 (2017), pp. 1187–1206.

[73] N. Shervashidze et al. "Efficient Graphlet Kernels for Large Graph Comparison". In: *JMLR Workshop and Conference Proceedings Volume 5: AISTATS 2009*. Max-Planck-Gesellschaft. Cambridge, MA, USA: MIT Press, Apr. 2009, pp. 488–495.

[74] David Siegmund, Benjamin Yakir, and Nancy R Zhang. "Detecting simultaneous variant intervals in aligned sequences". In: *The Annals of Applied Statistics* (2011), pp. 645–668.

[75] T. A. B. Snijders and K. Nowicki. "Estimation and prediction for stochastic blockmodels for graphs with latent block structure". In: *Journal of Classification* 14.1 (1997), pp. 75–100.

[76] MS Srivastava and Keith J Worsley. "Likelihood ratio tests for a change in the multivariate normal mean". In: *Journal of the American Statistical Association* 81.393 (1986), pp. 199–204.

[77] M. Tang, D. L. Sussman, and C. E. Priebe. "Universally consistent vertex classification for latent positions graphs". In: *The Annals of Statistics* 41.3 (2013), pp. 1406–1430.

[78] R. Vershynin. “Introduction to the non-asymptotic analysis of random matrices”. In: *arXiv preprint arXiv:1011.3027* (2010).

[79] S. V. N. Vishwanathan et al. “Graph Kernels”. In: *J. Mach. Learn. Res.* 11 (Aug. 2010), pp. 1201–1242.

[80] Z. Xu et al. “A model-based approach to attributed graph clustering”. In: *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. ACM. 2012, pp. 505–516.

[81] B. Yan and P. Sarkar. “Convex relaxation for community detection with covariates”. In: *arXiv preprint arXiv:1606.01869* (2016).

[82] B. Yan, P. Sarkar, and X. Cheng. “Exact Recovery of Number of Blocks in Blockmodels”. In: *arXiv preprint arXiv:1705.08580* (2017).

[83] Y. Yu, T. Wang, and R. J. Samworth. “A useful variant of the Davis–Kahan theorem for statisticians”. In: *Biometrika* 102.2 (2015), pp. 315–323.

[84] Nancy R Zhang et al. “Detecting simultaneous changepoints in multiple sequences”. In: *Biometrika* 97.3 (2010), pp. 631–645.

[85] Y. Zhang, E. Levina, and J. Zhu. “Community Detection in Networks with Node Features”. In: *arXiv preprint arXiv:1509.01173* (2015).

[86] Y. Zhang, E. Levina, and J. Zhu. “Estimating network edge probabilities by neighborhood smoothing”. In: *arXiv preprint arXiv:1509.08588* (2015).

[87] Y. Zhao, E. Levina, and J. Zhu. “Consistency of community detection in networks under degree-corrected stochastic block models”. In: *The Annals of Statistics* 40.4 (2012), pp. 2266–2292.

[88] S. Zhou and R. J. Mondragón. “The rich club phenomenon in the Internet topology”. In: *IEEE Communications Letters* 8.3 (2004), pp. 180–182.