2020

Estimation and Clustering in Block Models

Majid Noroozi
University of Central Florida

ESTIMATION AND CLUSTERING IN BLOCK MODELS

by

MAJID NOROOZI
M.S. University of Central Florida, 2016
M.S. K.N.Toosi University of Technology, 2009
B.S. Damghan University, 2006

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the College of Sciences
at the University of Central Florida
Orlando, Florida

Summer Term
2020

Major Professor: Marianna Pensky

ABSTRACT

Networks with community structure arise in many fields such as social science, biological science, and computer science. Stochastic block models are popular tools to describe such networks. For this reason, in this dissertation which is composed of two parts we explore some stochastic block models and the relationship between them.

In the first part of the dissertation, we study the Popularity Adjusted Block Model (PABM) and introduce its sparse case, the Sparse Popularity Adjusted Block Model (SPABM). The SPABM is the only existing block model which allows to set some probabilities of connections to zero. For both the PABM and the SPABM, we produce the estimators of the probability matrix in the case of an arbitrary number of communities which possibly grows with a number of nodes in the network and is not assumed to be known. One of our main contributions is application of the Sparse Subspace Clustering (SSC) to partitioning the network into communities, the approach that is well known in Computer Vision but, to the best of our knowledge, has not been used for clustering network data.

There is a variety of block models such as the Stochastic Block Model (SBM) and the Degree Corrected Block Model (DCBM) and the PABM. However, while this variety leads to a range of choices, the block models do not have a nested structure, in addition the DCBM requires identifiability assumptions for its fitting. There is also a substantial jump in the number of parameters from the DCBM to the PABM. Therefore, in the second part of the dissertation, we explore the relationship between the existing block models. We suggest a set of conditions on the DCBM that leads to a nested structure in block models, with the Erdős-Rényi model being the simplest and the PABM the most complex. Moreover, we introduce the Heterogeneous Block Model (HBM) that is more complicated than DCBM but has fewer

unknown parameters than the PABM, thus bridging the gap between the DCBM and the PABM. The HBM is based on partitioning the network into the mega-communities that, in turn, are subdivided into communities, where the communities are distinguished by the average connection probabilities between them while the mega-communities are determined by the heterogeneity of the probabilities of connections. This results in formulation of a hierarchy of block model which does not rely on arbitrary identifiability conditions, treats the SBM, the DCBM and the PABM as its particular cases with specific parameter values, and also allows a multitude of versions that are more complicated than DCBM but have fewer unknown parameters than the PABM. The latter enables one to carry out clustering and estimation without preliminary testing which of the block models is really true.

The theories in this dissertation are supplemented by simulation studies and real data examples.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Marianna Pensky, for her constant support, guidance, and patience throughout this project. I would also like to thank Dr. Zixia Song, Dr. Hassan Foroosh, and Dr. Teng Zhang, for serving on my committee. Finally, many thanks to my family and friends for all the support you have shown me through this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Over the past decade there has been an explosion of network data, that is, measurements that are either of or from a system conceptualized as a network, from different fields of science. For this reason, statistical network analysis has become a major field of research, with applications as diverse as sociology, biology, genetics, ecology, information technology to name a few. Examples of networks include protein-protein interaction networks, human brain functional networks, social networks found on Facebook, Twitter and dating websites, academic paper co-authorship and citation networks, etc. Theoretically, a network is considered as a graph often defined in terms of nodes and edges. In the statistical literature, a graph is often defined in terms of the nodes and the corresponding measurements on pairs of nodes which can be represented, for instance, as a binary adjacency matrix in a setting where we are only concerned with encoding presence or absence of edges between pairs of nodes. Nodes in the network may represent individuals, organizations, or some other kind of unit of study. Edges correspond to types of links, relationships, or interactions between the units, and they may be directed or undirected. For undirected graphs the adjacency matrix is symmetric. Networks can be modeled in a variety of ways, however, in the last decade stochastic block models attracted more and more attention due to their ability to summarize data in a compact and intuitive way and uncover low-dimensional structures that fully describe a given network. An overview of statistical modeling of random graphs can be found in, e.g., [23] and [32].

Consider an undirected network with n nodes and no self-loops and multiple edges. Let $A \in \{0,1\}^{n\times n}$ be the symmetric adjacency matrix of the network with $A_{i,j} = 1$ if there is a

connection between nodes i and j, and $A_{i,j} = 0$ otherwise. We assume that

$$A_{i,j} \sim \text{Bernoulli}(P_{i,j}), \quad 1 \leq i \leq j \leq n, \tag{1.1}$$

where $A_{i,j}$ are conditionally independent given $P_{i,j}$ and $A_{i,j} = A_{j,i}$, $P_{i,j} = P_{j,i}$ for $i > j$.

The classical Erdős-Rényi [16] random graph model assumes that the edges in a random graph are drawn independently with an equal probability, does not allow community structures and is too simplistic for applications. While the model boosted research in the area, it was very simplistic and could not adequately describe the networks that appear in real life. In particular, one of the main goals of a network modeling is to partition the nodes into the communities that, in some sense, exhibit similar modes of behavior.

The block models assume that each node in the network belongs to one of K distinct blocks or communities $\mathcal{N}_k$, $k = 1, \cdots, K$. Let z denote the vector of community assignment, with $z_i = k$ if the node i belongs to the community k. Then, the probability of connection between node $i \in \mathcal{N}_k$ and node $j \in \mathcal{N}_l$ depends on the pair of blocks (k, l) to which nodes (i, j) belong. One can also consider a corresponding *membership* (or *clustering*) matrix $Z \in \{0, 1\}^{n \times K}$ such that $Z_{i,k} = 1$ iff $i \in \mathcal{N}_k$, $i = 1, \ldots, n$.

The simplest random graph model for networks with community structure is the Stochastic Block Model (SBM) [40], [1], [20]. Under the K-block SBM, all nodes are partitioned into communities $\mathcal{N}_k$, $k = 1, \ldots, K$, and the probability of connection between nodes is completely defined by the communities to which they belong: $P_{i,j} = B_{z(i),z(j)}$ where $B_{k,l}$ is the probability of connection between communities k and l, and $z : \{1, ..., n\} \to \{1, ..., K\}$ is a clustering function. In particular, any nodes from the same community have the same degree distribution and the same expected degree. The Erdős-Rényi model can be viewed as the

SBM with only one community $K = 1$.

Since the real-life networks usually contain a very small number of high-degree nodes while the rest of the nodes have very low degrees, the SBM fails to explain the structure of many networks that occur in practice. The Degree-Corrected Block Model (DCBM), introduced by Karrer and Newman (2011) addresses this deficiency by allowing these probabilities to be multiplied by the node-dependent weights. Under the DCBM, the elements of matrix P are modeled as

$$P_{i,j} = h_i B_{z(i),z(j)} h_j, \quad i, j = 1, \ldots, n, \tag{1.2}$$

where $h = [h_1, h_2, ..., h_n]$ is a vector of the degree parameters of the nodes, and B is the $(K \times K)$ matrix of baseline interaction between communities. Matrix B and vector h in (1.2) are defined up to a scalar factor, which is usually fixed via the so called *identifiability* condition, that can be imposed in a variety of ways. For example, Karrer and Newman [29] enforce a constraint of the form

$$\sum_{i \in \mathcal{N}_k} h_i = 1, \quad k = 1, ..., K. \tag{1.3}$$

A network feature that is closely associated with community structure is the popularity of nodes across communities defined as the number of edges between a specific node and a specific community. While the DCBM allows to correctly detect the communities, and accurately fits the total degree by enforcing the node-specific degree parameters, it enforces the node popularity to be uniformly proportional to the node degree. Hence, the DCBM fails to model node popularities in a flexible and realistic way. For this reason, recently, Sengupta and Chen (2018) [49] introduced the Popularity Adjusted Stochastic Block Model (PABM) which models the probability of a connection between nodes as a product of popularity

parameters that depend on the communities to which the nodes belong as well as on the pair of nodes themselves. In particular, in PABM

$$P_{i,j} = V_{i,z_j} V_{j,z_i}, \tag{1.4}$$

where $V_{i,k}$, $1 \leq i \leq n$, $1 \leq k \leq K$, are the popularity scaling parameters and $0 \leq P_{i,j} \leq 1$ for any i and j. Sengupta and Chen [49] introduced the notion of popularity of node i in community k as $\mu_{i,k} = \sum_{j \in \mathcal{N}_k} P_{i,j}$. They noted that the ratio of popularities of the nodes $(i, j) \in \mathcal{N}_k$ in the same community k is equal to one for the SBM, is independent of community k (a function of i and j only) in DCBM but can vary between nodes and communities for the PABM, thus, allowing a more flexible modeling of connection probabilities. The authors showed that PABM generalizes both the SBM and the DCBM, suggested the quasi-maximum likelihood type procedure for estimation and clustering and demonstrated the improvement achieved through this new methodology.

The flexibility of PABM, however, is not limited to modeling the popularity parameters of the nodes. In order to better understand the model, consider a rearranged version $P(Z, K)$ of matrix P where its first n_1 rows correspond to nodes from class 1, the next n_2 rows correspond to nodes from class 2 and the last n_K rows correspond to nodes from class K. Denote the (k, l)-th block of matrix $P(Z, K)$ by $P^{(k,l)}(Z, K)$. Since sub-matrix $P^{(k,l)}(Z, K) \in [0, 1]^{n_k \times n_l}$ corresponds to pairs of nodes in communities (k, l) respectively, one obtains from (1.4) that $P^{(k,l)}_{i,j} = V_{i_k,l} V_{j_l,k}$ where i_k is the i-th element in $\mathcal{N}_k$ and j_l is the j-th element in $\mathcal{N}_l$. Thus, matrices $P^{(k,l)}(Z, K)$ are rank-one matrices with the unique singular vectors generating them. Indeed, consider vectors $\Lambda^{(k,l)}$ with elements $\Lambda^{(k,l)}_i = V_{i_k,l}$, where $i = 1, \ldots, n_k$ and $i_k \in \mathcal{N}_k$. Then, equation (1.4) implies that

$$P^{(k,l)}(Z, K) = \Lambda^{(k,l)} \left[\Lambda^{(l,k)}\right]^T. \tag{1.5}$$

Moreover, it follows from (1.4) and (1.5) that $P^{(k,l)}(Z,K) = [P^{(l,k)}(Z,K)]^T$ and that each pair of blocks (k,l) involves a unique combination of vectors $\Lambda^{(l,k)}$:

$$P(Z,K) = \begin{bmatrix} \Lambda^{(1,1)}(\Lambda^{(1,1)})^T & \Lambda^{(1,2)}(\Lambda^{(2,1)})^T & \dots & \Lambda^{(1,K)}(\Lambda^{(K,1)})^T \\ \Lambda^{(2,1)}(\Lambda^{(1,2)})^T & \Lambda^{(2,2)}(\Lambda^{(2,2)})^T & \dots & \Lambda^{(2,K)}(\Lambda^{(K,2)})^T \\ \vdots & \vdots & \dots & \vdots \\ \Lambda^{(K,1)}(\Lambda^{(1,K)})^T & \Lambda^{(K,2)}(\Lambda^{(2,K)})^T & \dots & \Lambda^{(K,K)}(\Lambda^{(K,K)})^T \end{bmatrix}$$

where

$$\Lambda = \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} & \dots & \Lambda^{(1,K)} \\ \Lambda^{(2,1)} & \Lambda^{(2,2)} & \dots & \Lambda^{(2,K)} \\ \vdots & \vdots & \dots & \vdots \\ \Lambda^{(K,1)} & \Lambda^{(K,2)} & \dots & \Lambda^{(K,K)} \end{bmatrix} \tag{1.6}$$

The latter implies that matrix $P(Z,K)$ is formed by arbitrary rank one blocks and hence $rank(P(Z,K)) = rank(P)$ can take any value between K and K^2. In comparison, all other block models restrict the rank of P to be exactly K. This is true not only for the SBM and DCBM discussed above but also for their generalizations such as the Mixed Membership models (see, e.g., [4] and [11]) and the Degree Corrected Mixed Membership (DCMM) (see, e.g., [25]). Hence, the PABM allows for much more flexible spectral structure than any other block model above.

This flexibility makes the PABM an attractive choice for modeling networks that appear in biological sciences. Indeed, while social networks exhibit assortative behavior due to the human tendency of forming strong associations, the biological networks tend to be more diverse. For this reason, PABM tends to be a useful tool for modeling such networks.

However, while the PABM model is extremely valuable, the statistical inference in [49] has

been incomplete. In particular, the authors considered only the case of a small finite number of communities K; they provided only asymptotic consistency results as $n \to \infty$ without any error bounds when n is finite; their clustering procedure was tailored to the case of a small K, therefore, all simulations and real data examples in [49] only tackled the case of $K = 2$.

In this dissertation, we address some of those deficiencies and advance the theory of the PABM. Specifically, this dissertation makes the following contributions:

1. In contrast to [49], we consider the PABM with an arbitrary number of communities which possibly grows with a number of nodes in the network and is not assumed to be known.

2. We argue that the main appeal of the PABM is the flexibility of the spectral properties of the graph and replace the estimators in [49] that are based on averaging over the communities by more accurate counterparts based on low rank matrix approximations.

3. While Sengupta and Chen [49] only proved convergence of the estimation and clustering errors to zero as the number of nodes grows, we derive non-asymptotic upper bounds for those errors when the number of communities is arbitrary. In particular, we produce an upper bound for the estimation error of the matrix of the connection probabilities and provide a condition that guarantees that the proportion of misclassified nodes is bounded above by a specified quantity. All results in this dissertation are non-asymptotic and are valid for any combination of parameters.

4. We use the accuracy of approximation of the adjacency matrix for various number of communities, to identify the number of communities in the network.

5. We suggest to use the Sparse Subspace Clustering (SSC) approach to partition the network into communities. While the SSC is widely used in computer vision, to the

best of our knowledge, it has not been used for clustering network data. The advantage of the SSC procedure (in comparison with the Extreme Point algorithm applied in [49]) is that it has several well studied versions and can carry out clustering not only for the PABM but also for the SBM and DCBM.

6. Our simulation study as well as the real data examples handle various number of communities K between 2 and 6. In particular, we demonstrate the advantages of the PABM for modeling networks that appear in biological sciences.

The real life networks are usually sparse in a sense that a large number of nodes have small degrees. One of the shortcomings of both the SBM and the DCBM is that they do not allow to efficiently model sparsity in networks. Indeed, for the SBM, it is not realistic to assume that all nodes in a pair of communities have no connections, hence, in the SBM setting, one does not assume that the average block probabilities $B_{k,l} = 0$ for some k and l. The DCBM is not very different in this respect since setting any node–specific weight to zero will force the respective node to be totally disconnected from the network. For this reason, unlike in other numerous statistical settings, sparsity in block models is defined as a low maximum probability of connections between the nodes: $\max_{i,j} P_{i,j} \leq \rho(n)$ where $\rho(n) \ \to \ 0$ as $n \to \infty$ (see, e.g., [30], [35]). As a result, high degree nodes become very unlikely.

In addition to being unrealistic, the above definition of sparsity has other drawbacks. In particular, one has to estimate *every* probability of connections $B_{k,l}$, no matter how small it is, and, in many settings (see, e.g., [30]), in order to take advantage of the fact that $P_{i,j}$ are bounded above by $\rho(n)$, one needs to incorporate this unknown value into the estimation process.

On the contrary, the PABM setting allows some connection probabilities to be zero while keeping average connection probabilities between classes above certain level and the network

connected. This is possible only in the PABM context due to the flexible modeling of connection probabilities. The idea of setting some infinitesimally small probabilities of connections to zero is quite attractive. Indeed, it is well known that, when many of the elements of a vector or a matrix are identical zeros, identifying those zeros and estimating the rest of the elements leads to a smaller error than when this information is ignored. Similarly, allowing structural sparsity (i.e., setting connection probabilities to zero rather than to a very small positive number) not only leads to better understanding of network topology but leads to more precise estimation of the probability matrix P.

In the context of PABM, setting $\Lambda_i^{(k,l)} = 0$ simply means that that node i in class k is not active ("popular") in class l. This, nevertheless, does not prevent this node from having high probability of connection with nodes in another class. Setting some elements of vectors $\Lambda^{(k,l)}$ to zero will merely lead to some of the rows (columns) of sub-matrices $P^{(k,l)}(Z, K)$ being zero. Moreover, since $A_{i,j}$ are Bernoulli variables with the means $P_{i,j}$, those zeros are fairly easy to identify since $P_{i,j} = 0$ leads to $A_{i,j} = 0$.

Having several types of block models introduces a variety of choices, but also leads to some significant drawbacks. Specifically, although the block models can be viewed as progressively more elaborate with the Erdős-Rényi being the simplest and the PABM the most complex, the simpler models are not necessarily particular cases of the more sophisticated ones. Indeed, with the identifiability condition (1.3), the SBM matrix B will be different from the one in the DCBM formulation (1.2). For this reason, majority of authors carry out estimation and clustering under the assumption that the model which they use is indeed the correct one. There are only very few papers that study goodness of fit in block models and majority of them are concerned with either testing that there are no distinct communities ($K = 1$ in SBM or DCBM) [6], [19], [24], or testing the exact number of communities $K = K_0$ in the SBM [18], [34], [44]. To the best of our knowledge, [44] is the only paper testing the SBM

versus the DCBM, where the testing is carried out under rather restrictive assumptions. On the other hand, using the most flexible model, the PABM, may not always be the right choice since there is a substantial jump in complexity from the DCBM with $O(n+K^2)$ parameters to the PABM with $O(nK)$ parameters.

Therefore, formulation of a hierarchy of block model which does not rely on arbitrary identifiability conditions and treats the SBM, the DCBM and the PABM as its particular cases (with specific parameter values) provides a unified approach to block models. Moreover, the formulation allows a multitude of versions that are more complicated than DCBM but have fewer unknown parameters than the PABM. The aim of this construction is to treat all block models as a part of one paradigm and hence carry out estimation and clustering without preliminary testing to see which block model fits data at hand.

The rest of the dissertation is organized as follows.

Chapter 2 discusses estimation and clustering in PABM. Section 2.1 introduces notations used throughout Chapter 2. Section 2.2 formulates estimation and clustering as solutions of an optimization procedure. Section 2.3 derives upper bounds for estimation errors as well as sufficient conditions for the proportion of misclustered nodes to be bounded above by a pre-specified quantity ρ_n with a high probability. Section 2.4 deliberates about practical implementation of clustering and provides a simulation study and real data examples. In particular, Section 2.4.1 reviews the SSC and elaborates on what kind of SSC procedure we employ. Section 2.4.2 evaluates the performance of this method using synthetic networks with various values of K. Furthermore, it compares the performance of the SSC with the Extreme Point algorithm applied in [49] using the simulation example presented in [49] and shows the superiority of the former, especially when the homophily factor is small. Section 2.4.3 brings two examples of biological networks that we model using the PABM.

In Chapter 3 we introduce and analyze sparse PABM. After introducing notations in Section 3.1, we convey the structure of the probability matrix in Section 3.2. Section 3.3 formulates an optimization procedure for estimation and clustering. Furthermore, Section 3.4 suggests two possible expressions for the penalties and examines the support sets of the true and estimated probability matrices. Section 3.5 produces upper bounds on the estimation and clustering errors. Since the optimization procedure in Section 3.3 is NP-hard, Section 3.6 discusses implementation of the community detection via sparse subspace clustering. Sections 3.7.1 and 3.7.2 complement the theory with simulations on synthetic networks and real data examples.

Chapter 4 introduces the hierarchy of block models. In Section 4.1 we review the block models (SBM, DCBM, and PABM). Then, we introduce and formulate in Section 4.2 the heterogeneous stochastic block model (HBM). The optimization procedure for estimation and clustering is discussed in Section 4.3. Section 4.4 describes a computationally tractable clustering procedure for the implementation of clustering. The performance of the clustering procedure is evaluated on synthetic networks and real data examples in Section 4.5.

Finally, we devote Chapter 5 to a discussion of future work.

CHAPTER 2: ESTIMATION AND CLUSTERING IN PABM

2.1 Notation

For any two positive sequences $\{a_n\}$ and $\{b_n\}$, $a_n \asymp b_n$ means that there exists a constant $C > 0$ independent of n such that $C^{-1}a_n \leq b_n \leq Ca_n$ for any n. For any set Ω, denote cardinality of Ω by $|\Omega|$. For any numbers a and b, $a \wedge b = \min(a, b)$. For any vector $t \in \mathbb{R}^p$, denote its ℓ_2, ℓ_1, ℓ_0 and ℓ_∞ norms by, respectively, $\|t\|$, $\|t\|_1$, $\|t\|_0$ and $\|t\|_\infty$. Denote by 1_m the m-dimensional column vector with all components equal to one.

For any matrix A, denote its spectral and Frobenius norms by, respectively, $\|A\|_{op}$ and $\|A\|_F$. Let $\mathrm{vec}(A)$ be the vector obtained from matrix A by sequentially stacking its columns.

Denote by $\mathcal{M}_{n,K}$ a collection of clustering matrices $Z \in \{0,1\}^{n\times K}$ such that $Z_{i,k} = 1$ iff $i \in \mathcal{N}_k$, $i = 1, \ldots, n$, and $Z^TZ = \mathrm{diag}(n_1, \ldots, n_K)$ where $n_k = |\mathcal{N}_k|$ is the size of community k, where $k = 1, \ldots, K$. Denote by $\mathscr{P}_{Z,K} \in \{0,1\}^{n\times n}$ the permutation matrix corresponding to $Z \in \mathcal{M}_{n,K}$ that rearranges any matrix $B \in \mathbb{R}^{n,n}$, so that its first n_1 rows correspond to nodes from class 1, the next n_2 rows correspond to nodes from class 2 and the last n_K rows correspond to nodes from class K. Recall that $\mathscr{P}_{Z,K}$ is an orthogonal matrix with $\mathscr{P}_{Z,K}^{-1} = \mathscr{P}_{Z,K}^T$. For any $\mathscr{P}_{Z,K}$ and any matrix $B \in \mathbb{R}^{n\times n}$ denote the permuted matrix and its blocks by, respectively, $B(Z, K)$ and $B^{(k,l)}(Z, K)$, where $B^{(k,l)}(Z, K) \in \mathbb{R}^{n_k\times n_l}$, $k, l = 1, \ldots, K$, and

$$B(Z, K) = \mathscr{P}_{Z,K}^T B \mathscr{P}_{Z,K}, \qquad B = \mathscr{P}_{Z,K} B(Z, K) \mathscr{P}_{Z,K}^T. \tag{2.1}$$

Also, throughout this chapter, we use the star symbol to identify the true quantities. In

particular, we denote the true matrix of connection probabilities by P_*, the true number of classes by K_* and the true clustering matrix that partitions n nodes into K_* communities by Z_*.

2.2 Optimization procedure for estimation and clustering

In this section we consider estimation of the true probability matrix P_*. Consider block $P_*^{(k,l)}(Z_*, K_*)$ of the rearranged version $P_*(Z_*, K_*)$ of P_*. Let $\Lambda \equiv \Lambda(Z_*, K_*) \in [0,1]^{n \times K_*}$ be a block matrix with each column l partitioned into K_* blocks $\Lambda^{(k,l)} \equiv \Lambda^{(k,l)}(Z_*, K_*) \in [0,1]^{n_k}$. Then, due to (1.5), $P_*^{(k,l)}(Z_*, K_*)$ are rank-one matrices such that $P_*^{(k,l)}(Z_*, K_*) = [P_*^{(l,k)}(Z_*, K_*)]^T$ and that each pair of blocks (k,l) involves a unique combination of vectors $\Lambda^{(k,l)}$. The structures of matrices $P_*(Z_*, K_*)$, Λ and P_* are illustrated in Figure 2.1.

Observe that although matrices $P_*^{(k,l)}(Z_*, K_*)$ in (1.5) are well defined, vectors $\Lambda^{(k,l)}$ and $\Lambda^{(l,k)}$ can be determined only up to a multiplicative constant. In particular, under the constraint

$$1_{n_k}^T \Lambda^{(k,l)} = 1_{n_l}^T \Lambda^{(l,k)}, \tag{2.2}$$

Sengupta and Chen [49] obtained explicit expressions for vectors $\Lambda^{(k,l)}$ and $\Lambda^{(l,k)}$ in (1.5).

In reality, K_* and matrices Z_* and P_* are unknown and need to be recovered. If K_* were known, in order to estimate Z_* and P_*, one could permute the rows and the columns of the adjacency matrix A using permutation matrix $\mathscr{P}_{Z,K_*}$ obtaining matrix $A(Z, K_*) = \mathscr{P}_{Z,K_*}^T A \mathscr{P}_{Z,K_*}$ and then, following assumption (1.5), minimize some divergence measure between blocks of $A(Z, K_*)$ and the products $\Lambda^{(k,l)} [\Lambda^{(l,k)}]^T$. One of such measures is the Bregman divergence between $A(Z, K_*)$ and $\Lambda^{(k,l)} [\Lambda^{(l,k)}]^T$.

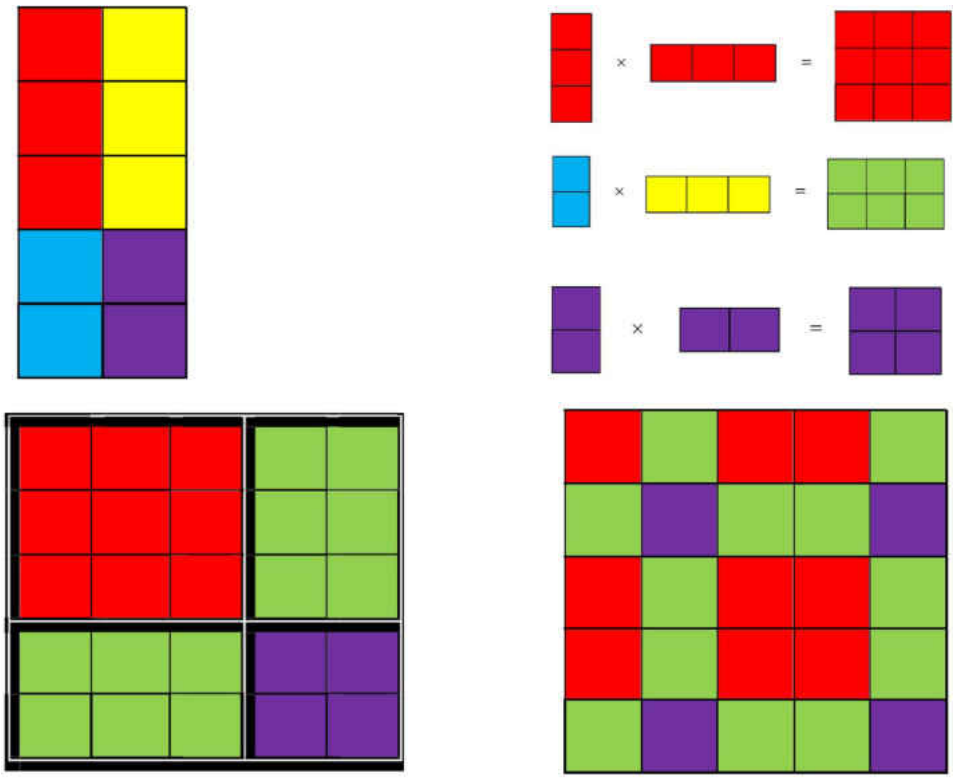

Figure 2.1: Matrices Λ, $P(Z,K)$ and P in the case of $n=5$ and $K=2$. Matrix Λ (top left): $\Lambda^{(1,1)}$ (red), $\Lambda^{(2,1)}$ (blue), $\Lambda^{(1,2)}$ (yellow), $\Lambda^{(2,2)}$ (violet). Assembling re-organized probability matrix $P(Z,K)$ (top right): $P^{(1,1)}(Z,K)$ (red), $P^{(2,1)}(Z,K)$ (green), $P^{(2,2)}(Z,K)$ (violet). Re-organized probability matrix $P(Z,K)$ (bottom left): $P^{(1,1)}(Z,K)$ (red), $P^{(2,1)}(Z,K)$ and $P^{(1,2)}(Z,K)$ (green), $P^{(2,2)}(Z,K)$ (violet). Probability matrix P (bottom right): nodes 1,3,4 are in community 1; nodes 2 and 5 are in community 2.

The Bregman divergence between vectors x and y associated with a continuously-differentiable, strictly convex function F is defined as

$$D_F(x,y) = F(x) - F(y) - \langle \nabla F(y), x - y \rangle$$

where $\nabla F(y)$ is the gradient of F with respect to y. The Bregman divergence between any matrices X and Y of the same dimension can be defined as the Bregman divergence

between their vectorized versions: $D_F(X,Y) = D_F(\mathrm{vec}(X), \mathrm{vec}(Y))$. It is well known that $D_F(X,Y) \geq 0$ for any X and Y and $D_F(X,Y) = 0$ iff $X = Y$. In particular, the Poisson log-likelihood maximization used in [49] corresponds to minimizing the Bregman divergence with

$$F(x) = \sum_i (x_i \ln x_i - x_i).$$

Under the assumption (1.5) and the constraint (2.2) of [49], the latter leads to maximization over $\Lambda^{(k,l)}$ and $Z \in \mathcal{M}_{n,K_*}$ of the following quantity

$$l(\Lambda|A) = -D_F(A,\Lambda) = \sum_{k,l=1}^{K_*} \sum_{i=1}^{n_k} \sum_{j=1}^{n_l} \left[A_{i,j}^{(k,l)} \ln \left(\Lambda_i^{(k,l)} \Lambda_j^{(l,k)} \right) - \left(\Lambda_i^{(k,l)} \Lambda_j^{(l,k)} \right) \right]. \tag{2.3}$$

where $A^{(k,l)}$ stands for $A^{(k,l)}(Z, K_*)$, the (k,l)-th block of matrix $A(Z, K_*)$. It is easy to see that the expression (2.3) coincides with the Poisson log-likelihood up to a term which depends on matrix A only, and is independent of P, Z and K_*. Maximization of (2.3) over Λ, under condition (2.2), for given Z and K_*, leads to the estimators of Λ obtained in [49]

$$\widehat{\Lambda}^{(k,l)} = \frac{A^{(k,l)}(Z,K_*)1_{n_l}}{\sqrt{1_{n_k}^T A^{(k,l)}(Z,K_*)1_{n_l}}}; \quad \widehat{\Lambda}^{(l,k)} = \frac{(A^{(k,l)}(Z,K_*))^T 1_{n_k}}{\sqrt{1_{n_k}^T A^{(k,l)}(Z,K_*)1_{n_l}}}. \tag{2.4}$$

Afterwards, Sengupta and Chen [49] plug the estimators (2.4) into (2.3), thus, obtaining the likelihood modularity function which they further maximize in order to obtain community assignments.

Here, we use the Bregman divergence associated with the Euclidean distance ($F(x) = \|x\|^2$) which, for a given K, leads to the following optimization problem

$$(\widehat{\Lambda}, \hat{Z}) \in \underset{\Lambda, Z}{\mathrm{argmin}} \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(Z,K) - \Lambda^{(k,l)} [\Lambda^{(l,k)}]^T \right\|_F^2 \right\} \quad \text{s.t.} \quad A(Z,K) = \mathscr{P}_{Z,K}^T A \mathscr{P}_{Z,K}$$

Note that recovery of the components $\Lambda^{(k,l)}$ and $\Lambda^{(l,k)}$ of the products above relies on an identifiability condition of the type (2.2). Since these conditions can be imposed in a variety of ways, we denote $\Theta^{(k,l)} = \Lambda^{(k,l)}[\Lambda^{(l,k)}]^T$ and recover the uniquely defined rank one matrix $\Theta^{(k,l)}$. In addition, since the number of clusters K is unknown, we impose a penalty on K in order to safeguard against choosing too many clusters. Hence, we need to solve the following optimization problem

$$\begin{aligned}(\hat{\Theta}, \hat{Z}, \hat{K}) \in \quad & \underset{\Theta, Z, K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(Z,K) - \Theta^{(k,l)} \right\|_F^2 + \operatorname{Pen}(n,K) \right\} \\ & \text{s.t.} \quad A(Z,K) = \mathscr{P}_{Z,K}^T A \mathscr{P}_{Z,K}, \quad \operatorname{rank}(\Theta^{(k,l)}) = 1; \quad k,l = 1,2,\cdots,K. \end{aligned} \tag{2.5}$$

Here, $\hat{\Theta}$ is the block matrix with blocks $\hat{\Theta}^{(k,l)}$, $k,l = 1,\ldots,\hat{K}$ and $\operatorname{Pen}(n,K)$ will be defined later.

Observe that, if $\hat{Z}$ and $\hat{K}$ were known, the best solution of problem (2.5) would be given by the rank one approximations $\hat{\Theta}^{(k,l)}$ of matrices $A^{(k,l)}(\hat{Z},\hat{K})$

$$\hat{\Theta}^{(k,l)}(\hat{Z},\hat{K}) = \Pi_{\hat{u},\hat{v}} \left(A^{(k,l)}(\hat{Z},\hat{K}) \right) = \hat{\sigma}_1^{(k,l)} \hat{u}^{(k,l)}(\hat{Z},\hat{K}) (\hat{v}^{(k,l)}(\hat{Z},\hat{K}))^T, \tag{2.6}$$

where $\hat{\sigma}_1^{(k,l)}$ are the largest singular values of matrices $A^{(k,l)}(\hat{Z},\hat{K})$); $\hat{u}^{(k,l)}(\hat{Z},\hat{K})$, $\hat{v}^{(k,l)}(\hat{Z},\hat{K})$ are the corresponding singular vectors, and $\Pi_{\hat{u},\hat{v}} \left(A^{(k,l)}(\hat{Z},\hat{K}) \right)$ is the rank one projection of matrix $A^{(k,l)}(\hat{Z},\hat{K})$. Plugging (2.6) into (2.5), we rewrite optimization problem (2.5) as

$$\begin{aligned}(\hat{Z}, \hat{K}) \in \quad & \underset{Z, K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(Z,K) - \Pi_{\hat{u},\hat{v}} \left(A^{(k,l)}(Z,K) \right) \right\|_F^2 + \operatorname{Pen}(n,K) \right\} \\ & \text{s.t.} \quad A(Z,K) = \mathscr{P}_{Z,K}^T A \mathscr{P}_{Z,K} \end{aligned} \tag{2.7}$$

In order to obtain $(\hat{Z},\hat{K})$, one needs to solve optimization problem (2.7) for every K, ob-

taining

$$\hat{Z}_K \in \underset{Z\in\mathcal{M}_{n,K}}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(Z,K) - \Pi_{\hat{u},\hat{v}} \left(A^{(k,l)}(Z,K) \right) \right\|_F^2 \right\} \tag{2.8}$$

and then find $\hat{K}$ as

$$\hat{K} \in \underset{K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(\hat{Z}_K,K) - \Pi_{\hat{u},\hat{v}} \left(A^{(k,l)}(\hat{Z}_K,K) \right) \right\|_F^2 + \operatorname{Pen}(n,K) \right\}. \tag{2.9}$$

Note that if the true number of clusters K_* were known, the penalty in (2.5) and (2.7) would be unnecessary.

Remark 2.2.1. Advantages of our estimation procedure. There are several advantages of the estimator (2.6) in comparison with estimators (2.4) of [49]. First, rather than obtaining estimators in (2.4) by averaging, we derive the rank one approximations of the unknown sub-matrices of probabilities which lead to the minimal error (see, e.g., [22]) even when some of the nodes are misclustered and, therefore, the matrices $P_*^{(k,l)}(\hat{Z},\hat{K})$ are not necessarily of rank one. Indeed, the estimators obtained by averaging are suboptimal since matrix P_* is contaminated with errors. Second, recoveries of the matrices $\Theta^{(k,l)}$ do not require any identifiability conditions that can be imposed in a variety of ways. Finally, estimators $\widehat{\Lambda}^{(k,k)}$ of vectors $\Lambda^{(k,k)}$ in (2.4) require the knowledge of the diagonal elements of matrix A that are not available. On the contrary, the rank one approximation of a matrix can be achieved in the presence of missing values (see, e.g., [30]).

Remark 2.2.2. The true community assignment. Sengupta and Chen [49] show that the likelihood modularity is maximized at the true community assignment provided the, so called, detectability condition holds: for any two distinct communities $\mathcal{N}_l$ and $\mathcal{N}_k$ and any two nodes, $j_1 \in \mathcal{N}_l$ and $j_2 \in \mathcal{N}_k$, the set $\{(P_*)_{i,j_1}/(P_*)_{i,j_2}\}_{i=1}^n$ assumes at least K_*+1 distinct values, where K_* is the true (known) number of clusters and P_* is the unknown true matrix of probabilities. In our case, the correct community assignment is a solution of the

optimization problem (2.8) if matrix P_* is a unique combination (up to permutations) of the K^2 rank one matrices. The latter is guaranteed if collections of vectors $\Lambda^{(k,1)}, \ldots, \Lambda^{(k,K_*)}$ are linearly independent for any $k = 1, \ldots, K_*$. Milder conditions can be found in [50].

2.3 The errors of estimation and clustering

In this section we evaluate the estimation and the clustering errors. We choose the penalty which, with high probability, exceeds the random errors. In particular, we denote

$$F_1(n, K) = C_1 nK + C_2 K^2 \ln(ne) + C_3(\ln n + n \ln K) \tag{2.10}$$

$$F_2(n, K) = 2 \ln n + 2n \ln K, \tag{2.11}$$

where C_1, C_2 and C_3 are absolute constants. Define the penalty of the form

$$\mathrm{Pen}(n, K) = \left(2 + 16\, \beta_1^{-1}\right) F_1(n, K) + \beta_2^{-1} F_2(n, K), \tag{2.12}$$

where positive parameters β_1 and β_2 are such that $\beta_1 + \beta_2 < 1$. Then, the following statement holds.

Theorem 2.3.1. *Let $(\hat{\Theta}, \hat{Z}, \hat{K})$ be a solution of optimization problem (2.5). Construct the estimator $\hat{P}$ of P_* of the form*

$$\hat{P} = \mathscr{P}_{\hat{Z},\hat{K}} \hat{\Theta}(\hat{Z}, \hat{K}) \mathscr{P}^T_{\hat{Z},\hat{K}} \tag{2.13}$$

where $\mathscr{P}_{\hat{Z},\hat{K}}$ is the permutation matrix corresponding to $(\hat{Z}, \hat{K})$. Then, for any $t > 0$ and

$\tilde{C} = 2\left(1 - \alpha_1 - 8\alpha_2\right)^{-1}\left(C_3 + 1/\alpha_1 + C_3/\alpha_2\right)$, *one has*

$$\mathbb{P}\left\{\frac{1}{n^2}\left\|\hat{P} - P_*\right\|_F^2 \leq \frac{Pen(n, K_*)}{\left(1 - \beta_1 - \beta_2\right) n^2} + \frac{\tilde{C}t}{n^2}\right\} \geq 1 - 3e^{-t}, \tag{2.14}$$

$$\frac{1}{n^2}\,\mathbb{E}\left\|\hat{P} - P_*\right\|_F^2 \leq \frac{Pen(n, K_*)}{\left(1 - \beta_1 - \beta_2\right) n^2} + \frac{3\tilde{C}}{n^2} \tag{2.15}$$

Remark 2.3.1. The penalty. By rearranging and combining the terms, the penalty in (2.12) can be written in the form

$$\mathrm{Pen}(n, K) = H_1 nK + H_2 K^2 \ln n + H_3 n \ln K, \tag{2.16}$$

where $H_i \equiv H_i(\beta_1, \beta_2, C_1, C_2, C_3)$, $i = 1, 2, 3$, and the estimation errors in (2.14) and (2.15) are proportional to the right hand side of (2.16). The first term in (2.16) corresponds to the error of estimating nK unknown entries of matrix Λ, the second term is associated with estimation of rank K^2 matrix while the last term is due to the clustering of n nodes into K communities. If K grows with n, i.e., $K = K(n) \to \infty$ as $n \to \infty$, then the first term in (2.16) dominates the other two terms. However, in the case of a fixed K, the first and the third terms grow at the same rate as $n \to \infty$. The second term is always of a smaller order provided $K(n)/n \to 0$.

In order to evaluate the clustering error, we assume that the true number of classes $K = K_*$ is known. Let $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix. Then $\hat{Z} \equiv \hat{Z}_K$ is a solution of the optimization problem (2.8). Note that if Z_* is the true clustering matrix and Z is any other clustering matrix, then the proportion of misclustered nodes can be evaluated as

$$\mathrm{Err}(Z, Z_*) = (2n)^{-1} \min_{\mathscr{P}_K \in \mathcal{P}_K} \|Z\mathscr{P}_K - Z_*\|_1 = (2n)^{-1} \min_{\mathscr{P}_K \in \mathcal{P}_K} \|Z\mathscr{P}_K - Z_*\|_F^2 \tag{2.17}$$

where $\mathcal{P}_K$ is the set of permutation matrices $\mathscr{P}_K : \{1, 2, \cdots, K\} \longrightarrow \{1, 2, \cdots, K\}$. Let

$$\Upsilon(Z_*, \rho) = \left\{ Z \in \mathcal{M}_{n,K} : (2n)^{-1} \min_{\mathscr{P}_K \in \mathcal{P}_K} \|Z\mathscr{P}_K - Z_*\|_1 \geq \rho \right\} \tag{2.18}$$

be the set of clustering matrices with the proportion of misclustered nodes being at least ρ, $\rho < 1$.

The success of clustering in (2.8) relies upon the fact that matrix P_* is a collection of K^2 rank one blocks, so that the operator and the Frobenius norms of each block are the same. On the other hand, if clustering were incorrect, the ranks of the blocks would increase which would lead to the discrepancy between their operator and Frobenius norms. In particular, the following statement is true.

Theorem 2.3.2. *Let $K = K_*$ be the true number of clusters and $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix. If for some $\alpha_1, \alpha_2, \rho_n \in (0, 1)$, one has*

$$\|P_*\|_F^2 - \frac{1+\alpha_2}{1-\alpha_1} \max_{Z \in \Upsilon(Z_*, \rho_n)} \sum_{k,l=1}^{K} \left\| P_*^{(k,l)}(Z) \right\|_{op}^2 \geq H[C_1 nK + C_2 K^2 \ln(ne) + C_3(n \ln K + t)], \tag{2.19}$$

then, with probability at least $1 - 2e^{-t}$, the proportion of the misclassified nodes is at most ρ_n. Here, $H \equiv H(\alpha_1, \alpha_2)$, is a function of α_1 and α_2 only.

2.4 Simulations and real data examples

2.4.1 Sparse subspace clustering

In Section 2.2, we obtained an estimator $\hat{Z}$ of the true clustering matrix Z_* as a solution of optimization problem (2.7). Minimization in (2.7) is somewhat similar to modularity

maximization in [7] or [59] in the sense that modularity maximization as well as minimization in (2.7) are NP-hard, and, hence, require some relaxation in order to obtain an implementable clustering solution.

In the case of the SBM and the DCBM, possible relaxations include semidefinite programming (see, e.g., [5] and references therein), variational methods ([10]) and spectral clustering and its versions (see, e.g., [28], [35] and [48] among others). Since in the case of PABM, columns of matrix P_* that correspond to nodes in the same class are neither identical, nor proportional, direct application of spectral clustering to matrix P_* does not deliver the partition of the nodes. However, it is easy to see that the columns of matrix P_* that correspond to nodes in the same class form a matrix with K rank-one blocks, hence, those columns lie in the subspace of the dimension at most K. Therefore, matrix P_* is constructed of K clusters of columns (rows) that lie in the union of K distinct subspaces, each of the dimension K. For this reason, the subspace clustering presents a technique for obtaining a fast and reliable solution of optimization problem (2.7) (or (2.8)).

Subspace clustering (see [54] for a review) has been widely used in computer vision and, for this reason, it is a very well studied and developed technique in comparison with the Extreme Points algorithm used in [49]. Subspace clustering is designed for separation of points that lie in the union of subspaces. Let $\{X_j \in \mathbb{R}^D\}_{j=1}^n$ be a given set of points drawn from an unknown union of $K \geqslant 1$ linear or affine subspaces $\{S_i\}_{i=1}^K$ of unknown dimensions $d_i = \dim(S_i)$, $0 < d_i < D$, $i = 1, ..., K$. In the case of linear subspaces, the subspaces can be described as

$$S_i = \{\boldsymbol{x} \in \mathbb{R}^D : \boldsymbol{x} = \boldsymbol{U}_i \boldsymbol{y}\}, \quad i = 1, ..., K$$

where $\boldsymbol{U}_i \in \mathbb{R}^{D \times d_i}$ is a basis for subspace S_i and $\boldsymbol{y} \in \mathbb{R}^{d_i}$ is a low-dimensional representation for point $\boldsymbol{x}$. The goal of subspace clustering is to find the number of subspaces K, their

dimensions $\{d_i\}_{i=1}^K$, the subspace bases $\{\boldsymbol{U}_i\}_{i=1}^K$, and the segmentation of the points according to the subspaces.

When there is only one subspace, the problem reduces to finding a basis $U \in \mathbb{R}^{D\times d}$, a low-dimensional representation $Y = [Y_1, ..., Y_n] \in \mathbb{R}^{d\times n}$, and the dimension d. This problem is known as Principal Component Analysis (PCA) [27] and can be solved in a simple way: (U, Y) can be obtained from the rank-d singular value decomposition (SVD) of the data matrix $Y = [X_1, ..., X_n] \in \mathbb{R}^{D\times n}$ as

$$U = \mathcal{U} \quad \text{and} \quad Y = \Sigma\mathcal{V}^T, \quad \text{where} \quad X = \mathcal{U}\Sigma\mathcal{V}^T,$$

and d can be obtained as $d = \text{rank}(X)$ with data without noise, or using model selection techniques when the data is noisy [27].

When $K > 1$, the subspace clustering problem becomes considerably more difficult because of a number of challenges. First, there is a strong connectivity between model estimation and data segmentation. Particularly, one could easily fit a single subspace to each group of points using standard PCA if the segmentation of the data were known. Conversely, one could easily find the data points that best fit each subspace if the subspace parameters were known. In practice, both problems need to be solved simultaneously since neither the segmentation of the data nor the subspace parameters are known. The second challenge is that the distribution of the data inside the subspaces is generally unknown. Third, the relative position of the subspaces can be arbitrary and the subspace clustering problem becomes significantly hard when two subspaces intersect or are very close. Fourth, the data can be corrupted by noise, outliers, missing entries, etc. Last, but not least, is the issue of model selection. In classical PCA, the dimension of the subspace is the only parameter, which can be found by searching for the subspace of smallest dimension that fits the data

with a given accuracy. In the case of having more than one subspace, one can fit the data with n different subspaces of dimension one, namely one subspace per data point, or with a single subspace of dimension D. Clearly, neither solution is satisfactory. The challenge is to find a model selection criteria that leads to a small number of subspaces of small dimension.

Several methods have been developed to address these challenges over the past few years. Algebraic methods ([8], [41], [55]), iterative methods ([9], [2], [52]), and spectral clustering based methods ([17], [38], [39], [51], [14], [15], [54]) are some of these methods.

Two algebraic algorithms for clustering noise free data drawn from multiple linear subspaces are matrix factorization-based algorithm [8] and Generalized PCA (GPCA) ([41], [55]). The first algorithm is applicable only to independent subspaces and is based on linear algebra, specifically matrix factorization. It obtains the segmentation of the data from a low-rank factorization of the data matrix X. Thus, it is a natural extension of PCA from one to multiple independent linear subspaces. The second one is applicable to any kind of subspaces and is based on polynomial algebra. It is an algebraic-geometric method for clustering data lying in (not necessarily independent) linear subspaces. The main idea behind GPCA is that one can fit a union of K subspaces with a set of polynomials of degree K, whose derivatives at a point give a vector normal to the subspace containing that point. Then, the segmentation of the data is obtained by grouping these normal vectors using possible techniques. These algorithms are designed for linear subspaces; however, in the case of noiseless data they can also be applied to affine subspaces by considering an affine subspace of dimension d in $\mathbb{R}^D$ as a linear subspace of dimension $d+1$ in $\mathbb{R}^{D+1}$. Also, while these algorithms are used under the assumption of noise free data, they provide good insights into the geometry and algebra of the subspace clustering problem. Moreover, they can be extended to handle moderate amounts of noise.

The performance of algebraic algorithms in the case of noisy data can be improved by using iterative methods. Intuitively, given an initial segmentation, one can fit a subspace to each group using classical PCA. Then, given a PCA model for each subspace, one can assign each data point to its closest subspace. By iterating these two steps until convergence, a refined estimate of the subspaces and of the segmentation can be obtained. This is the main idea behind the K-planes [9] and K-subspaces ([3], [52]) algorithms.

Spectral clustering algorithms are popular and widely used techniques for clustering high-dimensional data. These algorithms rely on construction of an affinity matrix $A \in \mathbb{R}^{n \times n}$, whose ij entry measures the similarity between points i and j. The similarity is typically measured based on some distance measures between the points. Preferably, $A_{ij} = 1$ if points i and j are in the same group and $A_{ij} = 0$ if points i and j are in different groups. Given A, the K-means algorithm is applied to the eigenvectors of a Laplacian matrix $L \in \mathbb{R}^{n \times n}$ formed from A to obtain the segmentation of the data. Specifically, K eigenvectors of L are chosen and stacked into a matrix and the K-means algorithm is then applied to the rows of this matrix. The affinity matrix A, the Laplacian $D - A$, where $D = \text{diag}(A\mathbf{1})$ and $\mathbf{1}$ is the vector of all 1's, or the normalized Laplacian $D^{-1/2}AD^{-1/2}$ are typical choices for the L. Typical choices for the eigenvectors are the top K eigenvectors of the affinity or the bottom K eigenvectors of the (normalized) Laplacian, where K is the number of groups. Defining a good affinity matrix is one of the main challenges in applying spectral clustering to the subspace clustering problem. This is because two points could be very close to each other, but lie in different subspaces, while they could be far from each other, but lie in the same subspace. Consequently, the typical distance-based affinity cannot be used. Spectral clustering based methods divide the problem in two steps. First, an affinity matrix is learned from the data. Second, the segmentation of the data is obtained by applying spectral clustering to this affinity matrix. Since the success of the spectral clustering algorithm is largely dependent

on constructing an informative affinity matrix, the first step is the most important. One of the solutions is to construct the affinity matrix using self-representation of the points with the expectation that a point is more likely to be presented as a linear combination of points in its own subspace rather than from a different one. A number of approaches such as Low Rank Representation (LRR) (see, e.g., [38], [39]) and Sparse Subspace Clustering (SSC) (see, e.g., [15], [14]) have been proposed for the solution of this problem in the past decade. LRR and SSC are very similar. LRR tries to find a low-rank representation, while SSC aims to find a sparse representation.

In this dissertation, we use SSC since it allows one to take advantage of the knowledge that, for a given K, columns of matrix P_* lie in the union of K distinct subspaces, each of the dimension at most K. If matrix P_* were known, the weight matrix W would be based on writing every data point as a sparse linear combination of all other points by minimizing the number of nonzero coefficients

$$\min_{W_j} \|W_j\|_0 \quad \text{s.t} \quad (P_*)_j = \sum_{k \neq j} W_{kj} (P_*)_k \tag{2.20}$$

where, for any matrix B, B_j is its j-th column. The affinity matrix of the SSC is the symmetrized version of the weight matrix W. If the subspaces are linearly independent, then the solution to the optimization problem (4.20) is such that $W_{k,j} \neq 0$ only if points k and j are in the same subspace. In the case of data contaminated by noise, the SSC algorithm does not attempt to write data as an exact linear combination of other points. Instead, SSC is based on the solution of the following optimization problem

$$\widehat{W_j} \in \operatorname*{argmin}_{W_j} \left\{ \|W_j\|_0 + \gamma \|A_j - AW_j\|_2^2 \quad \text{s.t.} \quad W_{jj} = 0 \right\}, \quad j = 1, ..., n, \tag{2.21}$$

where $\gamma > 0$ is a tuning parameter. Problem (2.21) can be rewritten in an equivalent form

as

$$\widehat{W}_j \in \underset{W_j}{\operatorname{argmin}} \left\{ \|A_j - AW_j\|_2^2 \quad \text{s.t.} \quad \|W_j\|_0 \leq L, \quad W_{jj} = 0 \right\}, \quad j = 1, ..., n, \tag{2.22}$$

where L is the maximum number of nonzero elements in each column of W; in our case $L = K$. We solve (2.22) using the Orthogonal Matching Pursuit (OMP) algorithm ([43], [58]) implemented in SPAMS Matlab toolbox (see [42]). Given $\widehat{W}$, the affinity matrix is defined as $|\widehat{W}| + |\widehat{W}^T|$ where, for any matrix B, matrix $|B|$ has absolute values of elements of B as its entries. The class assignment (clustering matrix) Z is then obtained by applying spectral clustering to $|\widehat{W}| + |\widehat{W}^T|$. We elaborate on the implementation of the SSC in Section 2.4.2.

2.4.2 Simulations on synthetic networks

In this section we evaluate the performance of our method using synthetic networks. We assume that the number of communities (clusters) K is known and for simplicity consider a perfectly balanced model with n/K nodes in each cluster. We generate each network from a random graph model with a symmetric probability matrix P given by the PABM model with a clustering matrix Z and a block matrix Λ.

Sengupta and Chen [49], in their simulations, considered networks with $K = 2$ communities of equal sizes and matrices Λ in (1.4) with elements $\Lambda_{i,r} = \alpha_i \sqrt{\frac{h}{1+h}}$ when node i lies in class r, and $\Lambda_{i,r} = \beta_i \sqrt{\frac{1}{1+h}}$ otherwise, where h is the homophily factor. The factors α_i and β_i were set to 0.8 for half of the nodes in each class and to 0.2 for another half at random, and h ranges between 1.5 and 4.0. Note that, although the data generated by the procedure above follows PABM, the probability matrix has constant blocks, for which the spectral clustering

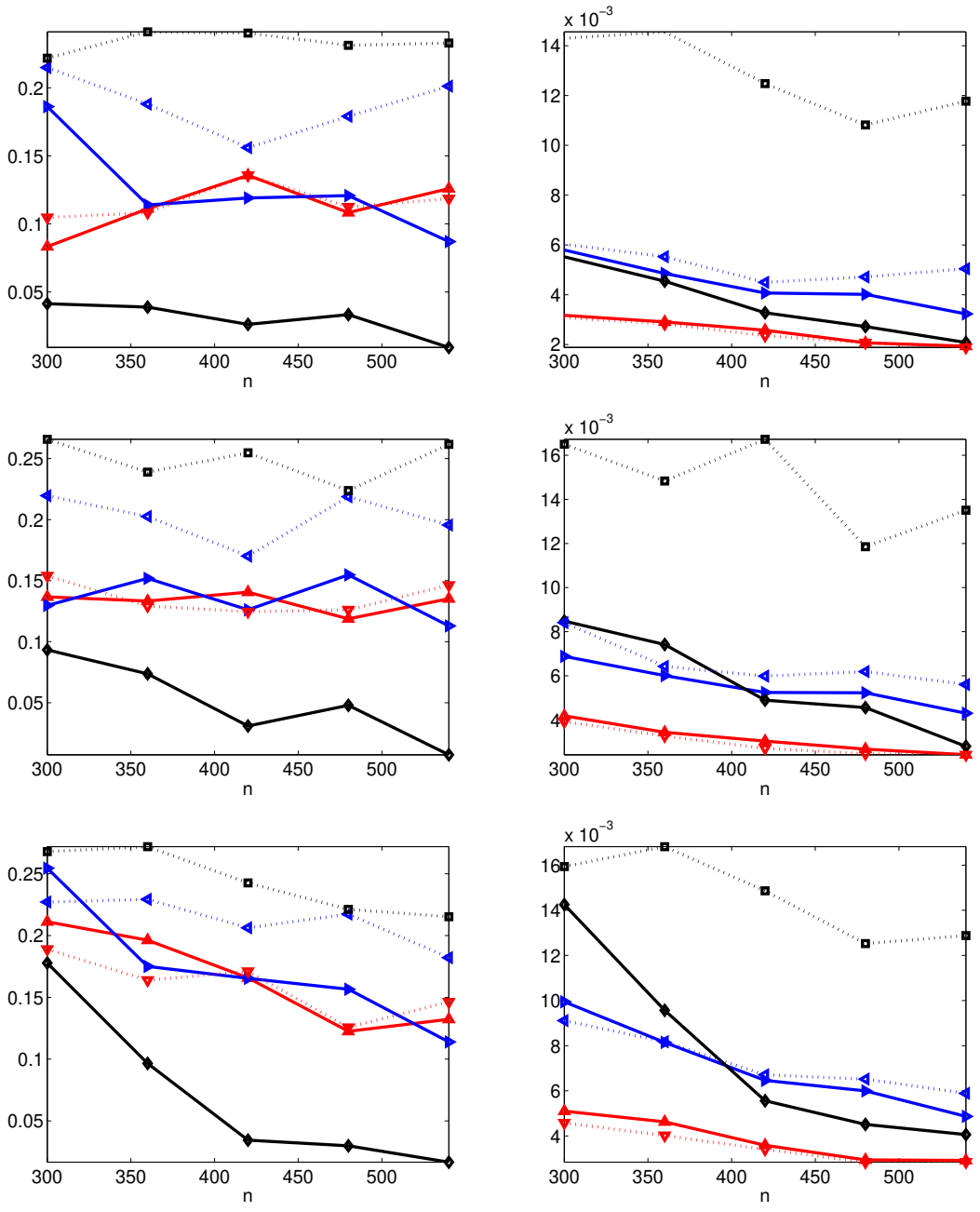

Figure 2.2: The clustering errors $\mathrm{Err}(\hat{Z}, Z)$ defined in (4.24) (left panels) and the estimation errors $n^{-2}\,\|\hat{P} - P\|_F^2$ (right panels) for $K = 3$ (top), $K = 4$ (middle) and $K = 5$ (bottom) clusters. The errors are evaluated over 50 simulation runs. The number of nodes ranges from $n = 300$ to $n = 540$ with the increments of 60. SSC results are represented by the solid lines; SC results are represented by the dash lines: $\omega = 0.5$ (red), $\omega = 0.7$ (blue) and $\omega = 0.9$ (black).

is known to deliver accurate results. In particular, the setting above leads to the SBM with four blocks. However, the spectral clustering incurs some difficulties as the probabilities of connections in every community become more diverse. In this dissertation, we ensure to generate networks that follow PABM with diverse probabilities of connections.

To generate a more diverse synthetic network, we start by producing a block matrix Λ in (1.6) with random entries between 0 and 1. We multiply the non-diagonal blocks of Λ by ω, $0 < \omega < 1$, to ensure that most nodes in the same community have larger probability of interactions. Then matrix $P(Z, K)$ with blocks $P_{Z,K}^{(k,l)} = \Lambda^{(k,l)}(\Lambda^{(l,k)})^T$, $k, l = 1, \ldots, K$, mostly has larger entries in the diagonal blocks than in the non-diagonal blocks. The parameter ω is the heterogeneity parameter. Indeed, if $\omega = 0$, the matrix P_* is strictly block-diagonal, while in the case of $\omega = 1$, there is no difference between diagonal and non-diagonal blocks. Next, we generate a random clustering matrix $Z \in \mathcal{M}_{n,K}$ corresponding to the case of equal community sizes and the permutation matrix $\mathscr{P}(Z, K)$ corresponding to the clustering matrix Z. Subsequently, we scramble rows and columns of $P(Z, K)$ to create the probability matrix $P = \mathscr{P}_{Z,K} P(Z, K) \mathscr{P}_{Z,K}^T$. Finally we generate the lower half of the adjacency matrix A as independent Bernoulli variables $A_{ij} \sim \text{Ber}(P_{ij})$, $i = 1, \ldots, n, j = 1, \ldots, i-1$, and set $A_{ij} = A_{ji}$ when $j > i$. In practice, the diagonal $\text{diag}(A)$ of matrix A is unavailable, so we estimate $\text{diag}(P)$ without its knowledge.

Sengupta and Chen [49] used the Extreme Points (EP) algorithm, introduced in [33], as a clustering procedure. For $K = 2$, the EP algorithm computes the two leading eigenvectors of the adjacency matrix A, and finds the candidate assignments associated with the extreme points of the projection of the cube $[-1, 1]^n$ onto the space spanned by the two leading eigenvectors of A. The technique is becoming problematic when K grows and the probabilities of connections are getting more diverse, hence, Sengupta and Chen [49] have only studied performances of estimation and clustering in the case of $K = 2$ and the choices of probability

matrix P described above. As we have mentioned before, these are the settings for which the spectral clustering procedure allows to identify the communities. Considering that we are interested in studying $K > 2$ and the more diverse probabilities of connections, we use the spectral clustering directly (SC thereafter) and compare its precision with the sparse subspace clustering (SSC) procedure.

Since the diagonal elements of matrix A are unavailable, we initially set $A_{ii} = 0$, $i = 1, ..., n$. We solve optimization problem (2.22) using the Orthogonal Matching Pursuit (OMP) algorithm. After matrix $\widehat{W}$ of weights is evaluated, we obtain the clustering matrix $\hat{Z}$ by applying spectral clustering to $|\widehat{W}| + |\widehat{W}^T|$, as it was described in Section 2.4.1. Given $\hat{Z}$, we generate matrix $A(\hat{Z}) = \mathscr{P}_{\hat{Z}}^T A \mathscr{P}_{\hat{Z}}$ with blocks $A^{(k,l)}(\hat{Z})$, $k, l = 1, \ldots, K$, and obtain $\hat{\Theta}^{(k,l)}(\hat{Z}, \hat{K})$ by using the rank one approximation for each of the blocks. Finally, we estimate matrix P by $\hat{P} = \hat{P}(\hat{Z}, \hat{K})$ using formula (2.13) with $\hat{K} = K$.

We compared the accuracy of SSC and SC methods in terms of the average estimation errors $n^{-2} \|\hat{P} - P\|_F^2$ and the average clustering errors $\mathrm{Err}(\hat{Z}, Z)$ defined in (2.17). Figure 2.2 shows results of these comparisons for $K = 3, 4$ and 5, respectively, and the number of nodes ranging from $n = 300$ to $n = 540$ with the increments of 60. The left panels display the clustering errors $\mathrm{Err}(\hat{Z}, Z)$ while the right ones exhibit the estimation errors $n^{-2} \|\hat{P} - P\|_F^2$, as functions of the number of nodes, for three different values of the parameter ω: $\omega = 0.5$, 0.7, and 0.9. Figure 2.2 confirms that the SSC is becoming more and more accurate in comparison with SC as ω grows. The latter is due to the fact that the SSC is more suitable for handling heterogeneous connections probabilities.

Figure 2.3 presents the results of comparison of the clustering errors of SSC and SC in the simulations settings of [49]. It is easy to see that, while for larger values of the homophily factor h both methods perform almost equally well, the accuracy of SC deteriorates as h

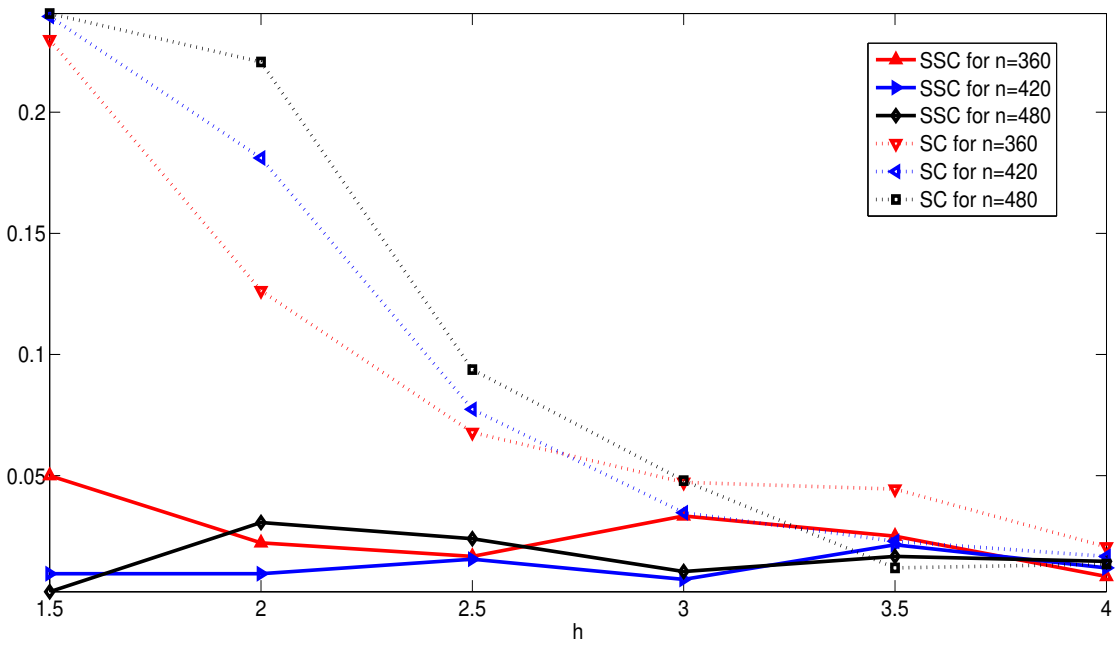

Figure 2.3: Clustering errors of SC and SSC for $K = 2$ clusters and $n = 360, 420$ and 480 nodes in the simulations setting of [49]. The homophily factor h ranges from 1.5 to 4 with increments of 0.5

is getting smaller, due to the fact that the differences between probabilities of connections within and between clusters become less significant. The latter shows that the SSC approach is beneficial for clustering in PABM model. Indeed, it delivers more accurate results than the SC when probabilities of connections are more diverse. On the other hand, SSC is still applicable when the PABM reduces to the SBM, although SC is more accurate in the case of the SBM since it does not require an additional step of evaluating the affinity matrix.

Remark 2.4.1. Spectral Clustering Versus Sparse Subspace Clustering. It is worth noting that when the matrix of probabilities P_* is close to being block diagonal, the spectral clustering can be still used for recovering community assignments, even if P_* does not follow the SBM. The latter is due to the fact that, in this situation, the graph can be well

approximated by a union of distinct connected components, and, therefore, SC allows to identify the true clusters. Moreover, in such situation, SC has an advantage of not requiring an additional step of self-representation, which is computationally costly and produces additional errors. On the other hand, as we shall see from examples below, when probabilities of connections become more heterogeneous, SSC turns to be more precise than SC. In addition, since PABM has more unknown parameters than SBM, its correct fitting requires sufficient number of nodes per class (see, e.g., [51]); otherwise, its accuracy declines.

Remark 2.4.2. Unknown number of clusters. In our previous simulations we treated the true number of clusters as a known quantity. However, we can actually use $\hat{P}$ to obtain an estimator $\hat{K}$ of K by solving, for every suitable K, the optimization problem (2.9), which can be equivalently rewritten as

$$\hat{K} = \underset{K}{\operatorname{argmin}}\{\|\hat{P} - A\|_F^2 + \operatorname{Pen}(n, K)\}. \tag{2.23}$$

The penalty $\operatorname{Pen}(n, K)$ defined in (2.12) is, however, motivated by the objective of setting it above the noise level with a very high probability. In our simulations, we also study the selection of an unknown K using somewhat smaller penalty

$$\operatorname{Pen}(n, K) = \rho(A) n K \sqrt{\ln n \, (\ln K)^3} \tag{2.24}$$

where $\rho(A)$ is the density of matrix A, the proportion of nonzero entries of A.

In order to assess the accuracy of $\hat{K}$ as an estimator of K, we evaluated $\hat{K}$ as a solution of optimization problem (2.23) with the penalty (2.24) in each of the previous simulations settings over 50 simulation runs. Table 2.1 presents the relative frequencies of the estimators $\hat{K}$ of K_* for K_* ranging from 3 to 6, $n = 420$ and $n = 540$ and $\omega = 0.5$, 0.7 and 0.9.

Table 2.1: The relative frequencies of the estimators $\hat{K}$ of K_* for K_* ranging from 3 to 6, $n = 420$ and $n = 540$ and $\omega = 0.5$, 0.7 and 0.9.

		n=420			n=540		
K_*	$\hat{K}$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$
3	2	0	0	0	0	0.02	0
	3	**0.76**	**0.80**	**0.90**	**0.66**	**0.76**	**0.92**
	4	0.24	0.16	0.10	0.24	0.16	0.08
	5	0	0.04	0	0.10	0.06	0
	6	0	0	0	0	0	0

		n=420			n=540		
K_*	$\hat{K}$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$
4	2	0	0	0	0	0	0
	3	0.06	0.14	0	0.04	0	0
	4	**0.64**	**0.66**	**0.96**	**0.8**	**0.76**	**0.96**
	5	0.28	0.16	0.04	0.12	0.22	0.04
	6	0.02	0.04	0	0.04	0.02	0

		n=420			n=540		
K_*	$\hat{K}$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$
5	2	0	0.02	0	0	0	0
	3	0.02	0	0.02	0	0.04	0
	4	0.14	0.16	0.04	0.12	0.1	0
	5	**0.64**	**0.66**	**0.82**	**0.76**	**0.74**	**0.96**
	6	0.2	0.16	0.12	0.12	0.12	0.04

		n=420			n=540		
K_*	$\hat{K}$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.9$
6	2	0	0.04	0	0	0	0
	3	0.06	0.18	0.02	0	0.06	0
	4	0.18	0.22	0.02	0.08	0.04	0
	5	0.28	0.22	0.08	0.20	0.26	0.06
	6	**0.48**	**0.34**	**0.88**	**0.72**	**0.64**	**0.94**

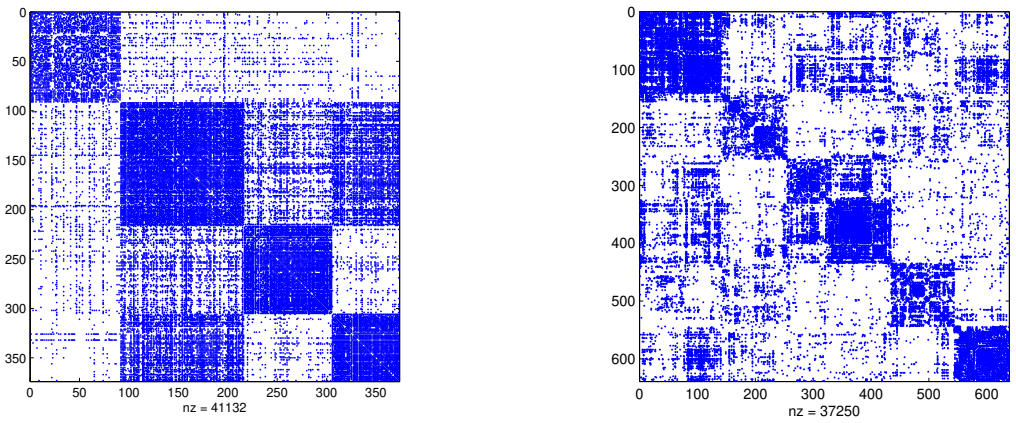

Figure 2.4: Adjacency matrices of the butterfly similarity network with 41132 nonzero entries and 4 clusters (left) and the brain network with 37250 nonzero entries and 6 clusters (right)

Table 2.1 confirms that for majority of settings, $\hat{K} = K_*$, the true number of clusters, with high probability. Moreover, the estimator $\hat{K}$ of K is more reliable for higher values of ω and larger number of nodes per cluster.

2.4.3 Real data examples

In this section, we report the performances of SSC and SC in studying real life networks. The social networks usually exhibit strong assortative behavior, the phenomenon which is possibly due to the tendency of humans to form strong associations. Perhaps, for this reason, the political blogs network, the British Twitter network, and the DBLP network which have been analyzed by Sengupta and Chen [49] have nearly block-diagonal adjacency matrices, so SC exhibits good performance in clustering of those networks (see Remark 2.4.1).

However, PABM provides a more accurate description of more diverse networks, in particular,

the networks that appear in biological sciences. Below, we consider a butterfly similarity network extracted from the Leeds Butterfly dataset described in [57]. Leeds Butterfly dataset contains fine-grained images of 832 butterfly species that belong to 10 different classes, with each class containing between 55 and 100 images. In this network, the nodes represent butterfly species and edges represent visual similarities between them. Visual similarities are evaluated on the basis of butterfly images and range from 0 to 1. We study a network by extracting the four largest classes as a simple graph with 373 nodes and 20566 edges. We draw an edge between the nodes if the visual similarity between those nodes is greater than zero. We carried out clustering of the nodes using the SSC and the SC and compared the clustering assignments of both methods with the true class specifications of the species. The SSC provides 89% accuracy while SC is correct only in 64% of cases. In addition, we applied formula (2.23) with K ranging from 2 to 6 and obtained the true number of clusters with 100% accuracy.

Figure 2.4 (left) shows the adjacency matrix of the graph (after clustering), which confirms that the network indeed follows the PABM. The latter is due to the fact that, since the phenotype of the species in the same class can vary, the SBM may not provide an adequate summary for the class similarities. Replacing the SBM by the DCBM does not solve the problem either, since it is unlikely that few butterflies are "more similar" to the others than the rest. On the other hand, the PABM allows some of the butterflies in one class to be "more similar" to species of another specific class than the other, thus, justifying application of the PABM.

As the second real network, we analyze a human brain functional network, measured using the resting-state functional MRI (fMRI). In particular we use the co-activation matrix described in [12] the brain connectivity dataset. In this dataset, the brain is partitioned into 638 distinct regions and a weighted graph is used to characterize the network topology. In our

analysis, we set all nonzero weights to one, obtaining the network with 18625 undirected edges. Since, for this network, the true clustering as well as the true number of clusters are unknown, we first applied formula (2.23) with K ranging from 2 to 10 to find the number of clusters obtaining $\hat{K} = 6$. This agrees with the assessment in [12] where the authors partitioned the network into 6 groups (if one considers the "rich-club" communities as separate clusters). Subsequently, we applied the SSC for partitioning the network into blocks and derived the estimator $\hat{P}$ of P_*. Figure 2.4 (right) shows the adjacency matrix of the graph after clustering. The true probability matrix P_* is unknown, we can only report that $n^{-2} \|\hat{P} - A\|_F^2 = 0.05$, which indicates high agreement between the two matrices.

CHAPTER 3: ESTIMATION AND CLUSTERING IN SPARSE PABM

3.1 Notation

Denote by $\Pi_J(X)$, the projection of a matrix $X : n \times m$ onto the set of matrices with non zero elements in the set $J = J_1 \times J_2 = \{(i, j) : i \in J_1,\ j \in J_2\}$. Denote by $\Pi_{(1)}(X)$ the best rank one approximation of matrix X and by $\Pi_{u,v}(X)$ the rank one projection of X onto pair of unit vectors u, v given by

$$\Pi_{u,v}(X) = (uu^T)X(vv^T). \tag{3.1}$$

Then, $\Pi_{(1)}(X) = \Pi_{u,v}(X)$ provided (u, v) is a pair of singular vectors of X corresponding to the largest singular value.

The notation in Chapter 2 also holds in this chapter.

3.2 The structure of the probability matrix

We consider the problem of estimation and clustering of the true matrix P_* of the probabilities of the connection between the nodes. Consider block $P_*^{(k,l)}(Z_*, K_*)$ of the rearranged version $P_*(Z_*, K_*)$ of P_*. Let $\Lambda_* \equiv \Lambda(Z_*, K_*) \in [0, 1]^{n \times K_*}$ be a block matrix with each column l partitioned into K_* blocks $\Lambda_*^{(k,l)} \equiv \Lambda_*^{(k,l)}(Z_*, K_*)$. Here, $\Lambda_*^{(k,l)} \in [0, 1]^{n_k}$ and $\Lambda_*^{(l,k)} \in [0, 1]^{n_l}$ are the column vectors and $P_*^{(k,l)}(Z_*, K_*)$ follows (1.5), i.e., $P_*^{(k,l)}(Z_*, K_*) = \Lambda_*^{(k,l)} [\Lambda_*^{(l,k)}]^T$. Hence, $P_*^{(k,l)}(Z_*, K_*)$ are rank-one matrices such that $P_*^{(k,l)}(Z_*, K_*) = [P_*^{(l,k)}(Z_*, K_*)]^T$ and that each pair of blocks $P_*^{(k,l)}$ and $P_*^{(l,k)}$, involves a unique combination of vectors $\Lambda_*^{(k,l)}$ and $\Lambda_*^{(l,k)}$, $k, l = 1, \ldots, K_*$.

Vectors $\Lambda_*^{(k,l)}$ and $\Lambda_*^{(l,k)}$ describe the heterogeneity of the connections of nodes in the pair of communities (k,l). While, on the average, those communities can be connected, some nodes in community k may have no interaction with nodes in community l or vice versa, so that some of the elements of vectors $\Lambda_*^{(k,l)}$ and $\Lambda_*^{(l,k)}$ can be identical zeros. Denote by $J_* \equiv J_*(Z_*, K_*) = \bigcup_{k,l=1}^{K} (J_*)_{k,l}$ the set of indices of all nonzero elements of matrix Λ_*, where

$$(J_*)_{k,l} \equiv (J_*)_{k,l}(Z_*, K_*) = \{i : \ (\Lambda_*)_i^{(k,l)} \neq 0\}, \quad J_*^{(k,l)} = (J_*)_{k,l} \times (J_*)_{l,k}, \tag{3.2}$$

are, respectively, the true support of vector $\Lambda_*^{(k,l)}$ and the set of all ordered pairs of indices (positions) of non-zero elements of sub-matrix $P_*^{(k,l)}(Z_*, K_*)$. Here, the elements of $(J_*)_{k,l}$ are enumerated by their corresponding rows in matrix Λ_*. Then,

$$(P_*)_{i,j}^{(k,l)}(Z_*, K_*) > 0 \quad \text{iff} \quad (i,j) \in J_*^{(k,l)}$$

and row i and column j of $P_*^{(k,l)}(Z_*, K_*)$ are equal to zero if $i \notin (J_*)_{k,l}$ or $j \notin (J_*)_{l,k}$.

Note that the set $J_* \equiv J_*(Z_*, K_*)$ relies upon the true clustering defined by K_* and Z_*. One can also consider sparsity sets $(\breve{J}_*)_{k,l} \equiv (\breve{J}_*)_{k,l}(Z, K)$ and $\breve{J}_{k,l} \equiv \breve{J}_{k,l}(Z, K)$ for an arbitrary K and matrix $Z \in \mathcal{M}_{n,K}$

$$(\breve{J}_*)_{k,l} = \{i : \ (P_*)_{i,j}^{(k,l)}(Z, K) \neq 0, \ j = 1, \ldots, n_l\}, \ \breve{J}_{k,l} = \{i : \ A_{i,j}^{(k,l)}(Z, K) \neq 0, \ j = 1, \ldots, n_l\}, \tag{3.3}$$

where the elements of $(\breve{J}_*)_{k,l}$ and $\breve{J}_{k,l}$ are enumerated by their corresponding rows in matrices P_* and A, respectively. Examples of the sets $(J_*)_{k,l}$, $(J_*)^{(k,l)}$, $(\breve{J}_*)_{k,l}$ and $(\breve{J}_*)^{k,l}$ are considered in Section 3.4.

For any sparsity sets $J_{k,l} \equiv J_{k,l}(Z,K)$, define, similarly to (3.2),

$$J = \bigcup_{k,l=1}^{K} J_{k,l} \quad \text{with} \quad J^{(k,l)} = J_{k,l} \times J_{l,k} \tag{3.4}$$

It follows from the definitions (3.3) and (3.4) that for any K, $Z \in \mathcal{M}_{n,K}$ and $k,l = 1,\dots,K$

$$\breve{J}_{k,l}(Z,K) \subseteq (\breve{J}_*)_{k,l}(Z,K) \quad \text{and} \quad \breve{J}(Z,K) \subseteq \breve{J}_*(Z,K). \tag{3.5}$$

3.3 Optimization procedure for estimation and clustering

Observe that although matrices $P_*^{(k,l)}(Z_*,K_*)$ and the sets $J_*^{(k,l)}$ are well defined, vectors $\Lambda_*^{(k,l)}$ and $\Lambda_*^{(l,k)}$ can be determined only up to a multiplicative constant. In order to avoid this ambiguity, denote $\Theta_*^{(k,l)} = \Lambda_*^{(k,l)}[\Lambda_*^{(l,k)}]^T$ and recover matrix Θ_* with the uniquely defined rank one blocks $\Theta_*^{(k,l)}$ and their supports $J_*^{(k,l)}$, $k,l = 1,\dots,K_*$. Then, one needs to solve the following optimization problem

$$\begin{aligned}(\hat{\Theta},\hat{Z},\hat{J},\hat{K}) \in \ & \underset{\Theta,Z,J,K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(Z,K) - \Theta^{(k,l)}(Z,J,K) \right\|_F^2 + \mathrm{Pen}(n,J,K) \right\} \\ & \text{s.t.} \quad A(Z,K) = \mathscr{P}_{Z,K}^T A \mathscr{P}_{Z,K},\ Z \in \mathcal{M}_{n,K}, \\ & \mathrm{supp}(\Theta^{(k,l)}) = J^{(k,l)} = J_{k,l} \times J_{l,k}, \quad \mathrm{rank}(\Theta^{(k,l)}) = 1, \quad k,l = 1,2,\cdots,K.\end{aligned} \tag{3.6}$$

Here, $\hat{\Theta}$ is the block matrix with blocks $\hat{\Theta}^{(k,l)}$, $k,l = 1,\dots,K$.

Observe that, if $\hat{Z}$, $\hat{J}$ and $\hat{K}$ were known, the best solution of problem (4.18) would be given by the best rank one approximations $\hat{\Theta}^{(k,l)}$ of matrices $A^{(k,l)}(\hat{Z},\hat{K})$ restricted to the sets $\hat{J}^{(k,l)}$

of indices of nonzero elements:

$$\hat{\Theta}^{(k,l)}(\hat{Z},\hat{J},\hat{K}) = \Pi_{(1)}\left(\Pi_{\hat{J}^{(k,l)}}(A^{(k,l)}(\hat{Z},\hat{K}))\right), \tag{3.7}$$

where $\Pi_{J^{(k,l)}}\left(A^{(k,l)}\right)$ is the projection of matrix $A^{(k,l)}$ onto the set of matrices with the support $J^{(k,l)}$ and $\Pi_{(1)}$ is the best rank one approximation of a matrix. Plugging (3.7) into (4.18), we rewrite optimization problem (4.18) as

$$(\hat{Z},\hat{J},\hat{K}) \in \underset{Z,J,K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \|A^{(k,l)}(Z,K) - \Pi_{(1)}[\Pi_{J^{(k,l)}}(A^{(k,l)}(Z,K))]\|_F^2 + \mathrm{Pen}(n,J,K) \right\} \tag{3.8}$$

$$\begin{aligned} \text{s.t.} \quad & A(Z,K) = \mathscr{P}_{Z,K}^T A \mathscr{P}_{Z,K}, \quad Z \in \mathcal{M}_{n,K}, \\ & J^{(k,l)} \equiv J^{(k,l)}(Z,K) = J_{k,l}(Z,K) \times J_{l,k}(Z,K). \end{aligned}$$

In practice, in order to obtain $(\hat{Z},\hat{J},\hat{K})$, one needs to solve optimization problem (3.8) for every K, obtaining

$$(\hat{Z}_K,\hat{J}_K) \in \underset{Z,J}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\|A^{(k,l)}(Z,K) - \Pi_{(1)}\left(\Pi_{J^{(k,l)}}(A^{(k,l)}(Z,K))\right)\right\|_F^2 + \mathrm{Pen}(n,J,K) \right\} \tag{3.9}$$

$$\begin{aligned} \text{s.t.} \quad & A(Z,K) = \mathscr{P}_{Z,K}^T A \mathscr{P}_{Z,K}, \quad Z_K \in \mathcal{M}_{n,K}, \\ & J^{(k,l)} \equiv J^{(k,l)}(Z,K) = J_{k,l}(Z,K) \times J_{l,k}(Z,K). \end{aligned}$$

and then find $\hat{K}$ as

$$\hat{K} \in \underset{K}{\operatorname{argmin}} \left\{ \sum_{k,l=1}^{K} \left\|A^{(k,l)}(\hat{Z}_K,K) - \Pi_{(1)}\left(\Pi_{\hat{J}_K^{(k,l)}}\left(A^{(k,l)}(\hat{Z}_K,K)\right)\right)\right\|_F^2 + \mathrm{Pen}(n,\hat{J}_K,K) \right\}. \tag{3.10}$$

3.4 The support of the probability matrix and the penalty

Consider solution of optimization problem (3.9) for a fixed value of K. If $\hat{Z}_K \in \mathcal{M}_{n,K}$ is a solution of (3.8), then

$$\begin{aligned} \hat{J}_K \in \ & \operatorname*{argmin}_J \left\{ \sum_{k,l=1}^{K} \left\| A^{(k,l)}(\hat{Z}_K, K) - \Pi_{(1)} \left(\Pi_{J^{(k,l)}} \left(A^{(k,l)}(\hat{Z}_K, K) \right) \right) \right\|_F^2 + \mathrm{Pen}(n, J, K) \right\} \\ & \text{s.t.} \quad A(\hat{Z}_K, K) = \mathscr{P}^T_{\hat{Z}_K, K} A \mathscr{P}_{\hat{Z}_K, K}, \quad J^{(k,l)} = J_{k,l} \times J_{l,k}, \quad J_{k,l} \equiv J_{k,l}(\hat{Z}_K, K). \end{aligned} \tag{3.11}$$

Observe that if the penalty term $\mathrm{Pen}(n, J, K)$ were not present in (3.11) or did not depend on set J, then one would have $\hat{J}_K = \breve{J}_K$ and $\hat{J}_K^{(k,l)} = \breve{J}_K^{(k,l)}$ where, by (3.3), $\breve{J}_K^{(k,l)}$ is the set of indices of nonzero rows and columns in $A^{(k,l)}(\hat{Z}_K, K)$. It is easy to see that

$$\begin{aligned} & \Pi_{\breve{J}^{(k,l)}} \left(A^{(k,l)}(\hat{Z}_K, K) \right) = A^{(k,l)}(\hat{Z}_K, K), \\ & \Pi_{(1)} \left(\Pi_{\breve{J}^{(k,l)}} \left(A^{(k,l)}(\hat{Z}_K, K) \right) \right) = \Pi_{(1)} \left(A^{(k,l)}(\hat{Z}_K, K) \right). \end{aligned}$$

Hence, even if sparsity is not specifically enforced (as it happens in [46] where the penalty depends on n and K only), one still obtains a sparse estimator $\hat{P}$ with the support $\hat{J}_K = \breve{J}_K$.

If the true number of clusters K_* and the true clustering matrix $Z_* \in \mathcal{M}_{n,K_*}$ were available, then the statement below shows that, under certain conditions, with high probability, sets $J_* \equiv J_*(Z_*, K_*)$ and $\breve{J}(Z_*, K_*)$ would coincide.

Lemma 3.4.1. *Let $K_*^2 \le n$ and the true matrix P_* be such that $(P_*)_{i,j} = 0$ or $(P_*)_{i,j} > \varpi(n, K_*)$. If the community sizes are balanced, i.e., the sizes of the true communities are no less than $\tilde{C}_0 n/K_*$ for some $\tilde{C}_0 \in (0, 1]$, and*

$$\varpi(n, K_*) \ge K_* \left(\sqrt{\ln n} + \sqrt{t} \right) \Big/ \left(\tilde{C}_0 \sqrt{2n} \right),$$

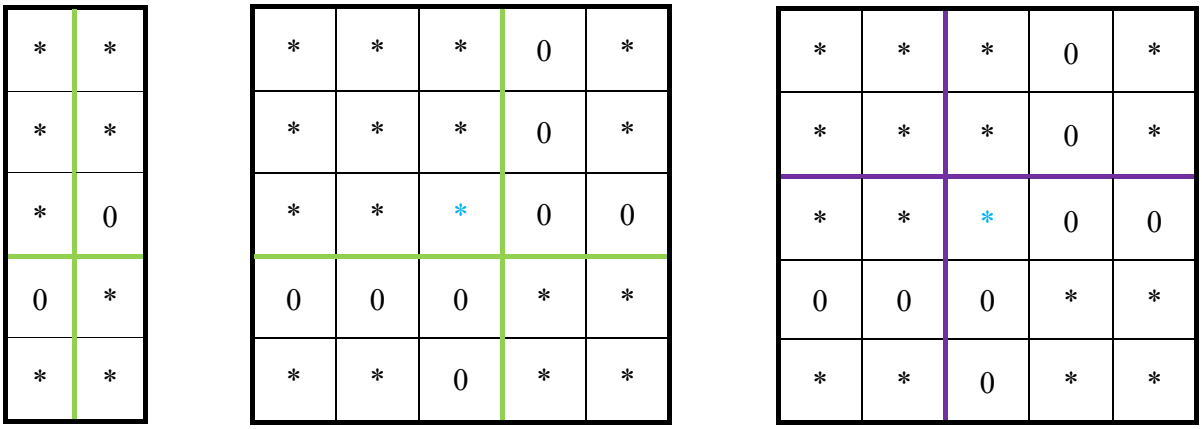

Figure 3.1: Zeros of the probability matrix with $n = 5$ and $K_* = 2$. Star symbols correspond to nonzero elements, the thick lines correspond to clustering assignments. Left panel: matrix Λ with $(J_*)_{1,1} = \{1, 2, 3\}$, $(J_*)_{2,1} = \{5\}$, $(J_*)_{1,2} = \{1, 2\}$ and $(J_*)_{2,2} = \{4, 5\}$. Middle panel: matrix $P_*(Z_*, K_*)$ with true clustering, $(\breve{J}_*)^c_{2,1}(Z_*) = \{4\}$ and $(\breve{J}_*)^c_{1,2}(Z_*) = \{3\}$, $\hat{P}_{i,j}(Z_*, K_*) = 0$ for $(i, j) \in \{(1, 4), (2, 4), (3, 4), (3, 5), (4, 1), (4, 2), (4, 3), (5, 3)\}$, so that, zero entries of the probability matrix are estimated by zeros. Right panel: matrix $P_*(\hat{Z}, K_*)$ with node 3 erroneously placed into community 2. The value of $(P_*)_{3,3}$ is nonzero. If $A_{3,3} = 0$, then $\breve{J}^c_{2,2}(\hat{Z}) = \{3\}$ and $\hat{P}_{i,j}(\hat{Z}, K_*) = 0$ for $(i, j) \in \{(1, 4), (2, 4), (3, 4), (3, 5), (4, 1), (4, 2), (4, 3), (5, 3)\}$, hence, zero entries of P_* are still estimated by the identical zeros. However, if $A_{3,3} = 1$, then zero elements $(P_*)_{3,4}$, $(P_*)_{3,5}$, $(P_*)_{4,3}$ and $(P_*)_{5,3}$ are estimated by positive values.

then, with probability at least $1 - e^{-t}$*, one has* $J_*(Z_*, K_*) = \breve{J}(Z_*, K_*)$.

Unfortunately, K_* and Z_* are unknown and, hence, $\hat{J}_K(Z, K) = \breve{J}_K(Z, K)$ may not always be the best estimator.

Consider, for example, the situation displayed in Figure 3.1 where $n = 5$, $K_* = 2$ and, under the true clustering, one has $n_1 = 3$ and $n_2 = 2$. Vectors $\Lambda_{2,1}$ and $\Lambda_{1,2}$ have one zero element each, so that $(J_*)_{1,1} = \{1, 2, 3\}$, $(J_*)_{2,1} = \{5\}$, $(J_*)_{1,2} = \{1, 2\}$ and $(J_*)_{2,2} = \{4, 5\}$ (left panel) leading to $(J_*)^{(1,1)} = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$,

$(J_*)^{(2,1)} = \{(5,1),(5,2)\}$, $(J_*)^{(1,2)} = \{(1,5),(2,5)\}$ and $(J_*)^{(2,2)} = \{(4,4),(4,5),(5,4),(5.5)\}$ (middle panel). With the true clustering (middle panel), $(\breve{J}_*)^c_{2,1}(Z_*) = \{4\}$ and $(\breve{J}_*)^c_{1,2}(Z_*) = \{3\}$, so that $\hat{P}_{i,j}(Z_*, K_*) = 0$ for $(i,j) \in \{(1,4),(2,4),(3,4),(3,5),(4,1),(4,2),(4,3),(5,3)\}$. Hence, zero entries of the probability matrix are estimated by zeros.

Consider now the situation where the third node has been erroneously placed into community 2 by clustering matrix $\hat{Z}$ (right panel). Then, we still have $(\breve{J}_*)^c_{2,1}(\hat{Z}) = \{4\}$, but $(\breve{J}_*)^c_{1,2}(\hat{Z})$ is an empty set. If $A_{3,3} = 0$, then $\breve{J}^c_{2,2}(\hat{Z}) = \{3\}$ and $\hat{P}_{i,j}(\hat{Z}, K_*) = 0$ for $(i,j) \in \{(1,4),(2,4),(3,4),(3,5),(4,1),(4,2),(4,3),(5,3)\}$, so that the zero entries of P_* are still estimated by the identical zeros. However, if $A_{3,3} = 1$, then zero elements $(P_*)_{3,4}$, $(P_*)_{3,5}$, $(P_*)_{4,3}$ and $(P_*)_{5,3}$ will be estimated by positive values.

For this reason, it is reasonable to introduce a penalty that will lead to trimming the support of $\hat{P}(Z,K)$.

We say that a penalty $Pen(n,J,K)$ is *separable* if for any K and any clustering matrix Z that partitions n nodes into K communities of sizes $n_k, k = 1,\ldots,K$, one can write

$$\mathrm{Pen}(n,J,K) = \mathrm{Pen}^{(0)}(n,J,K) + \mathrm{Pen}^{(1)}(n,K) \quad \text{with} \quad \mathrm{Pen}^{(0)}(n,J,K) = \sum_{l=1}^{K}\sum_{k=1}^{K} \mathscr{F}(|J_{k,l}|, n_k), \tag{3.12}$$

where $J_{k,l} \equiv J_{k,l}(Z,K)$. Otherwise, the penalty is *non-separable*.

Lemma 3.4.2. *Let $(\hat{Z}_K, \hat{J}_K)$ be the solution of the optimization problem (3.9). If $Pen(n,J,K)$ is separable and function $\mathscr{F}(j,m)$ in (3.12) is an increasing function of j for $0 \le j \le m$, then, for any $K < n$ and $k,l = 1,\ldots,K$, one has*

$$\hat{J}_{k,l}(\hat{Z}_K,K) \subseteq \breve{J}_{k,l}(\hat{Z}_K,K) \subseteq (\breve{J}_*)_{k,l}(\hat{Z}_K,K), \quad \hat{J}(\hat{Z}_K,K) \subseteq \breve{J}(\hat{Z}_K,K) \subseteq \breve{J}_*(\hat{Z}_K,K). \tag{3.13}$$

3.5 The errors of estimation and clustering

3.5.1 The penalty

In what follows, we consider the separable and the non-separable penalties of the form (3.12) with the common $\mathrm{Pen}^{(1)}(n, K)$, i.e.

$$\mathrm{Pen}^{(a)}(n, J, K) = \mathrm{Pen}^{(0,a)}(n, J, K) + \mathrm{Pen}^{(1)}(n, K), \tag{3.14}$$

where a =s for the separable penalty and a = ns for the nonseparable one, and

$$\begin{aligned} \mathrm{Pen}^{(0,s)}(n, J, K) &= \beta_1 \sum_{k,l=1}^{K} |J_{k,l}| \ln(n_k e/|J_{k,l}|) + \beta_2 K \sum_{k=1}^{K} \ln n_k & (3.15)\\ \mathrm{Pen}^{(0,ns)}(n, J, K) &= \beta_1 |J| \ln(nKe/|J|) + 2\beta_2 \ln n & (3.16)\\ \mathrm{Pen}^{(1)}(n, K) &= \beta_2 [n \ln K + \ln n]. & (3.17) \end{aligned}$$

Here, the separable penalty corresponds to $\mathscr{F}(|J_{k,l}|, n_k) = \beta_1 |J_{k,l}| \ln(n_k e/|J_{k,l}|) + \beta_2 \ln n_k$ and the exact expressions for β_1 and β_2 are given in Theorem 4.3.2 below.

In the next two sections, we shall provide upper bounds for the errors of the solution of optimization problem (4.18) with the separable or the non-separable penalty as well as upper bounds for the clustering error in the case of the separable penalty. While the separable penalty has some valuable properties (see Lemma 3.4.2), the non-separable penalty is much easier to interpret. Fortunately, as the statement below shows, under very nonrestrictive conditions, the penalties are within a constant factor of each other.

Lemma 3.5.1. *If $n \geq 8$ and $K \leq \sqrt{n/\ln n}$, then*

$$Pen^{(ns)}(n, J, K) < (2 + \beta_1/\beta_2)\, Pen^{(s)}(n, J, K) < 2\,(2 + \beta_1/\beta_2)\, Pen^{(ns)}(n, J, K). \tag{3.18}$$

3.5.2 The estimation errors

Theorem 3.5.1. *Let $(\hat{\Theta}, \hat{Z}, \hat{J}, \hat{K})$ be a solution of optimization problem (4.18) with the separable or non-separable penalty defined in (3.14). Construct the estimator $\hat{P}$ of P_* of the form*

$$\hat{P} = \mathscr{P}_{\hat{Z},\hat{K}} \hat{\Theta}(\hat{Z}, \hat{J}, \hat{K}) \mathscr{P}^T_{\hat{Z},\hat{K}} \tag{3.19}$$

where $\mathscr{P}_{\hat{Z},\hat{K}}$ is the permutation matrix corresponding to $(\hat{Z}, \hat{K})$. Let positive γ_1, γ_2 be such that $\gamma_1 + \gamma_2 < 1$ and β_1 and β_2 in (3.15)–(3.17) be given by

$$\beta_1 = \frac{2(C_1 + C_2)(8 + \gamma_1)}{\gamma_1} + \frac{2}{\gamma_2}, \quad \beta_2 = \frac{2C_2(8 + \gamma_1)}{\gamma_1} + \frac{2}{\gamma_2}, \tag{3.20}$$

where C_1 and C_2 are absolute constants. Then, for any $t > 0$, one has

$$\mathbb{P}\left\{ \frac{1}{n^2} \left\| \hat{P} - P_* \right\|_F^2 \leq \frac{Pen(n, J_*, K_*)}{n^2\,(1 - \gamma_1 - \gamma_2)} + \frac{\tilde{C}t}{n^2} \right\} \geq 1 - 3e^{-t}, \tag{3.21}$$

and,

$$\frac{1}{n^2}\, \mathbb{E} \left\| \hat{P} - P_* \right\|_F^2 \leq \frac{Pen(n, J_*, K_*)}{n^2\,(1 - \gamma_1 - \gamma_2)} + \frac{3\tilde{C}}{n^2} \tag{3.22}$$

where

$$\tilde{C} = 2\,{\gamma_1}^{-1}{\gamma_2}^{-1}(1 - \gamma_1 - 8\gamma_2)^{-1}\,(C_2\gamma_1\gamma_2 + \gamma_1 + 8C_2\gamma_2) \tag{3.23}$$

Observe that, due to Lemma 3.5.1, the separable and non-separable penalties are within

a constant factor of each other, so that Theorem 4.3.2 implies that the estimation error is proportional to $\mathrm{Pen}(n, J_*, K_*)$ where

$$\mathrm{Pen}(n, J, K) \asymp \mathrm{Pen}^{(ns)}(n, J, K) \asymp n \ln K + |J| \ln(nKe/|J|) + \ln n. \tag{3.24}$$

The first term in (3.24) is due to the clustering errors, the second term quantifies the difficulty of finding and estimating $|J|$ nonzero elements among nK elements of matrix $\Lambda \in [0,1]^{n\times K}$ while the $\ln n \asymp \ln(nK)$ term stands for the difficulty of finding the cardinality of the set $|J|$, and it is always dominated by the first two terms in (3.24).

Since each node has at least one community to which it is connected with a nonzero probability, one has $n \le |J| \le nK$. In the (non-sparse) PABM, $|J| = nK$ and the second term in (3.24) is always asymptotically larger, as $n \to \infty$, than the other two terms. In SPABM, the second term in (3.24) dominates the first term only if $K = 1$ or $|J|/n \to \infty$ as $n \to \infty$. However, if $K > 1$ and $|J| \asymp n$, then both terms are of the equal asymptotic order. If $K \to \infty$ and $|J| \asymp n$ as $n \to \infty$, then SPABM has the error $O(n \ln K)$ which is asymptotically smaller than $O(nK)$ error of PABM.

3.5.3 The clustering errors

In order to evaluate the clustering error, we assume that the true number of classes $K = K_*$ is known. Let $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix. Then $\hat{Z} \equiv \hat{Z}_K$ is a solution of the optimization problem (3.9). Note that if Z_* is the true clustering matrix and Z is any other clustering matrix, then the proportion of misclustered nodes can be evaluated as

$$\mathrm{Err}(Z, Z_*) = (2n)^{-1} \min_{\mathscr{P}_K \in \mathcal{P}_K} \|Z\mathscr{P}_K - Z_*\|_1 = (2n)^{-1} \min_{\mathscr{P}_K \in \mathcal{P}_K} \|Z\mathscr{P}_K - Z_*\|_F^2 \tag{3.25}$$

where $\mathcal{P}_K$ is the set of permutation matrices $\mathscr{P}_K : \{1, 2, \cdots, K\} \longrightarrow \{1, 2, \cdots, K\}$.

Theorem 3.5.2. *Let $K = K_*$ be the true number of clusters and $Z_* \in \mathcal{M}_{n,K_*}$ be the true clustering matrix and n_k be the true number of nodes in cluster $k = 1, \ldots, K$. Denote by $\gamma(Z_*, \rho_n)$ the set of clustering matrices with the proportion of at most ρ_n of the mis-clustered nodes. Let P_* and $J_* = J_*(P_*, Z_*)$ be, respectively, the true probability matrix and the true set J_*. If for some $\gamma_1, \gamma_2 > 0$ such that $\gamma_1 + \gamma_2 < 1$ and some $\tau \in (0, 1)$, one has*

$$\begin{aligned}
&\max_{\hat{Z} \in \gamma(Z_*, \rho_n)} \left\{ \sum_{k,l=1}^{K} \|P_*^{(k,l)}(\hat{Z})\|_{op}^2 - \frac{2C_1(\beta_1 - C_1 - C_2)}{(C_1 + C_2)\beta_1\gamma_2} K \sum_{k=1}^{K} \ln(\hat{n}_k) \right\} \\
&\leq \frac{(1-\tau)(\beta_1 - C_1 - C_2)}{\beta_1} \left[\|P_*\|_F^2 - 2(1+\sqrt{2})^2 \tau^{-1} \left(C_1 |J_*| + C_2 t \right) \right] \qquad (3.26) \\
&- (\beta_1 - C_1 - C_2) \left[\frac{C_2}{C_1 + C_2} \left(n \ln K + t \right) + \sum_{k,l=1}^{K} |(J_*)_{k,l}| \ln \left(\frac{n_k \, e}{|(J_*)_{k,l}|} \right) + \frac{\beta_2}{\beta_1} K \sum_{k=1}^{K} \ln(n_k) \right]
\end{aligned}$$

where β_1 and β_2 are defined in (3.20)*, then with probability at least $1 - 2\exp(-t)$, the proportion of mis-clustered nodes does not exceed ρ_n.*

3.6 Implementation of clustering

In Section 3.3, we obtained an estimator $\hat{Z}$ of the true clustering matrix Z_* as a solution of optimization problem (3.8). Minimization in (3.8) is somewhat similar to modularity maximization in [7] or [59] in the sense that modularity maximization as well as minimization in (3.8) are NP-hard, and, hence, require some relaxation in order to obtain an implementable clustering solution.

Since the SPABM is a special case of the PABM, the probability matrices in SPABM and PABM have the similar structure. In particular, matrix P_* is constructed of K clusters

of columns (rows) that lie in the union of K distinct subspaces, each of the dimension K. For this reason, the subspace clustering can be applied to obtain a solution of optimization problem (3.8) (or (3.9)).

As we discussed in Section 2.4.1, there are several methods to implement subspace clustering. Here, we use Sparse Subspace Clustering (SSC) since, similar to the PABM, it allows one to take advantage of the knowledge that, for a given K, columns of matrix P_* lie in the union of K distinct subspaces, each of the dimension at most K. If matrix P_* were known, the weight matrix W would be based on writing every data point as a sparse linear combination of all other points by solving the following optimization problem

$$\min_{W_j} \|W_j\|_1 \quad \text{s.t.} \quad (P_*)_j = \sum_{k \neq j} W_{kj}(P_*)_k \tag{3.27}$$

In the case of data contaminated by noise, the SSC algorithm does not attempt to write data as an exact linear combination of other points. Instead, SSC can be built upon the solution of the the elastic net problem

$$\widehat{W}_j \in \operatorname*{argmin}_{W_j} \left\{ \{\frac{1}{2}\|A_j - AW_j\|_2^2 + \gamma_1\|W_j\|_1 + \gamma_2\|W_j\|_2^2\} \quad \text{s.t.} \quad W_{jj} = 0 \right\}, \quad j = 1, ..., n, \tag{3.28}$$

where $\gamma_1, \gamma_2 > 0$ are tuning parameters. The quadratic term stabilizes the LASSO problem by making the problem strongly convex.

We solve (4.22) using the LARS algorithm [13] implemented in SPAMS Matlab toolbox (see [42]). Given $\widehat{W}$, the affinity matrix is defined as $|\widehat{W}| + |\widehat{W}^T|$ where, for any matrix B, matrix $|B|$ has absolute values of elements of B as its entries. The class assignment (clustering matrix) Z is then obtained by applying spectral clustering to $|\widehat{W}| + |\widehat{W}^T|$. We elaborate on the implementation of the SSC in Section 3.7.1.

3.7 Simulations and real data examples

3.7.1 Simulations on synthetic networks

In this section we evaluate the performance of our method using synthetic networks. We assume that the number of communities (clusters) K is known and for simplicity consider a perfectly balanced model with n/K nodes in each cluster. We generate each network from a random graph model with a symmetric probability matrix P given by the SPABM model with a clustering matrix Z and a block matrix Λ.

To generate synthetic networks, we start by producing a block matrix Λ in (1.6) with random entries between 0 and 1. We use a parameter σ as the proportion of nonzero entries in matrix Λ to control the sparsity of networks. To do that, we set $\lfloor nK\sigma \rfloor$ smallest non-diagonal entries of Λ to zero. Then we multiply the non-diagonal blocks of Λ by ω, $0 < \omega < 1$, to ensure that most nodes in the same community have larger probability of interactions. As a result, matrix $P(Z, K)$ with blocks $P^{(k,l)}(Z, K) = \Lambda^{(k,l)}(\Lambda^{(l,k)})^T$, $k, l = 1, \ldots, K$, has larger entries mostly in the diagonal blocks than in the non-diagonal blocks and some zero rows (columns) in the non-diagonal blocks. The parameter ω is the heterogeneity parameter. Indeed, if $\omega = 0$, the matrix P_* is strictly block-diagonal, while in the case of $\omega = 1$, there is no difference between entries in diagonal and nonzero entries in non-diagonal blocks. Next, we generate a random clustering matrix $Z \in \mathcal{M}_{n,K}$ corresponding to the case of equal community sizes and the permutation matrix $\mathscr{P}_{Z,K}$ corresponding to the clustering matrix Z. Subsequently, we scramble rows and columns of $P(Z, K)$ to create the probability matrix $P = \mathscr{P}_{Z,K} P(Z, K) \mathscr{P}_{Z,K}^T$. Finally we generate the lower half of the adjacency matrix A as independent Bernoulli variables $A_{ij} \sim \text{Ber}(P_{ij})$, $i = 1, \ldots, n, j = 1, \ldots, i-1$, and set $A_{ij} = A_{ji}$ when $j > i$. In practice, the diagonal elements of matrix A are unavailable, so we

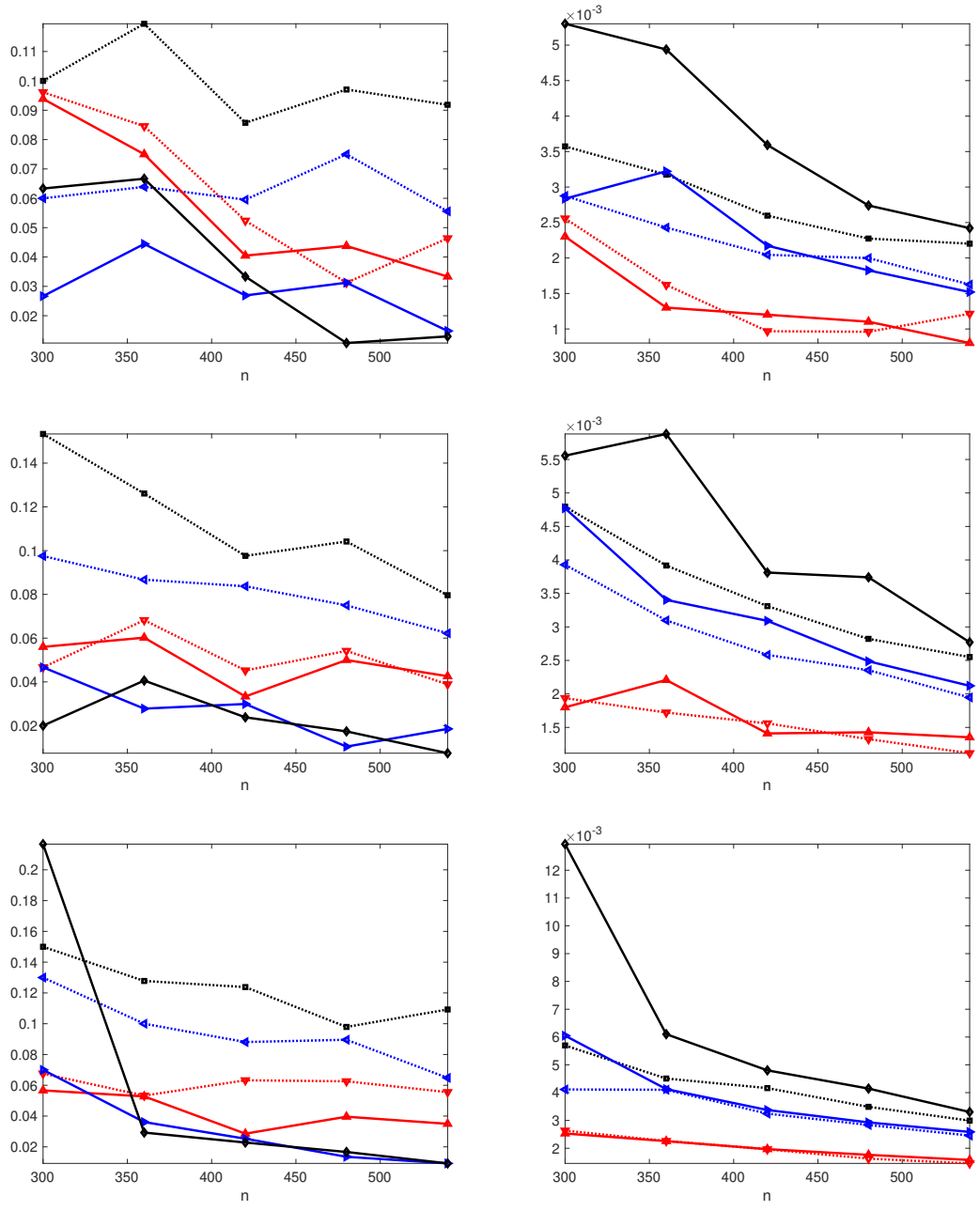

Figure 3.2: The clustering errors $\mathrm{Err}(\hat{Z}, Z)$ defined in (4.24) (left panels) and the estimation errors $n^{-2} \|\hat{P} - P\|_F^2$ (right panels) for $K = 4$ (top), $K = 5$ (middle) and $K = 6$ (bottom) clusters. The errors are evaluated over 50 simulation runs. The number of nodes ranges from $n = 300$ to $n = 540$ with the increments of 60. Dashed lines represent the results for $\omega = 0.5$ and solid lines represent the results for $\omega = 0.8$; $\sigma = 0.3$ (red), $\sigma = 0.5$ (blue) and $\sigma = 0.7$ (black).

estimate diag(P) without their knowledge.

Now we use SSC to find the clustering matrix $\hat{Z}$. Since the diagonal elements of matrix A are unavailable, we initially set $A_{ii} = 0$, $i = 1, ..., n$, and solve optimization problem (4.22) with $\gamma_1 = 30\rho(A)$ and $\gamma_2 = 125(1 - \rho(A))$, where where $\rho(A)$ is the density of matrix A, the proportion of nonzero entries of A. The values of γ_1 and γ_2 have been obtained empirically by testing on synthetic networks. After matrix $\widehat{W}$ of weights is evaluated, we obtain the clustering matrix $\hat{Z}$ by applying spectral clustering to $|\widehat{W}| + |\widehat{W}^T|$, as it was described in Section 3.6. In this chapter, we use the normalized cut algorithm [53] to perform spectral clustering. Given $\hat{Z}$, we generate matrix $A(\hat{Z}, K) = \mathscr{P}_{\hat{Z},K}^T A \mathscr{P}_{\hat{Z},K}$ with blocks $A^{(k,l)}(\hat{Z}, K)$, $k, l = 1, \ldots, K$, and obtain $\hat{\Theta}^{(k,l)}(\hat{Z}, K)$ by using the rank one approximation for each of the blocks. Finally, we estimate matrix P by $\hat{P} = \hat{P}(\hat{Z}, \hat{K})$ using formula (4.14) with $\hat{K} = K$.

Figure 4.3 represents the accuracy of SSC in terms of the average clustering errors $\mathrm{Err}(\hat{Z}, Z)$ defined in (4.24) and the average estimation errors $n^{-2} \|\hat{P} - P\|_F^2$ for $K = 4, 5$ and 6, respectively, and the number of nodes ranging from $n = 300$ to $n = 540$ with the increments of 60. The left panels display the clustering errors $\mathrm{Err}(\hat{Z}, Z)$ while the right ones exhibit the estimation errors $n^{-2} \|\hat{P} - P\|_F^2$, as functions of the number of nodes, for two different values of the parameter ω: $\omega = 0.5$ (dashed lines) and 0.8 (solid lines) and three different values of the parameter σ: $\sigma = 0.3$ (red lines), 0.5 (blue lines), and 0.7 (black lines). Figure 4.3 shows that as the sparsity increases, the estimation error decreases.

Our procedure does not estimate the set J explicitly. Instead, we set $\hat{J} = \breve{J} = \bigcup_{k,l=1}^K \breve{J}_{k,l}$ where $\breve{J}_{k,l}$ is defined in (3.3). Our next objective is to evaluate how accurate $\breve{J}$ is, as an estimator of J_*. While there are several ways for doing this, below we use two measures, the false positive rate ρ_{FP}, defined as the proportion of zero entries in P_* that are estimated by non-zeros in $\hat{P}$, and $\Delta_{FN} = \|P_*\|_F^{-1} \|X_*\|_F$, where $\|X_*\|_F$ is the Frobenius norm of nonzero

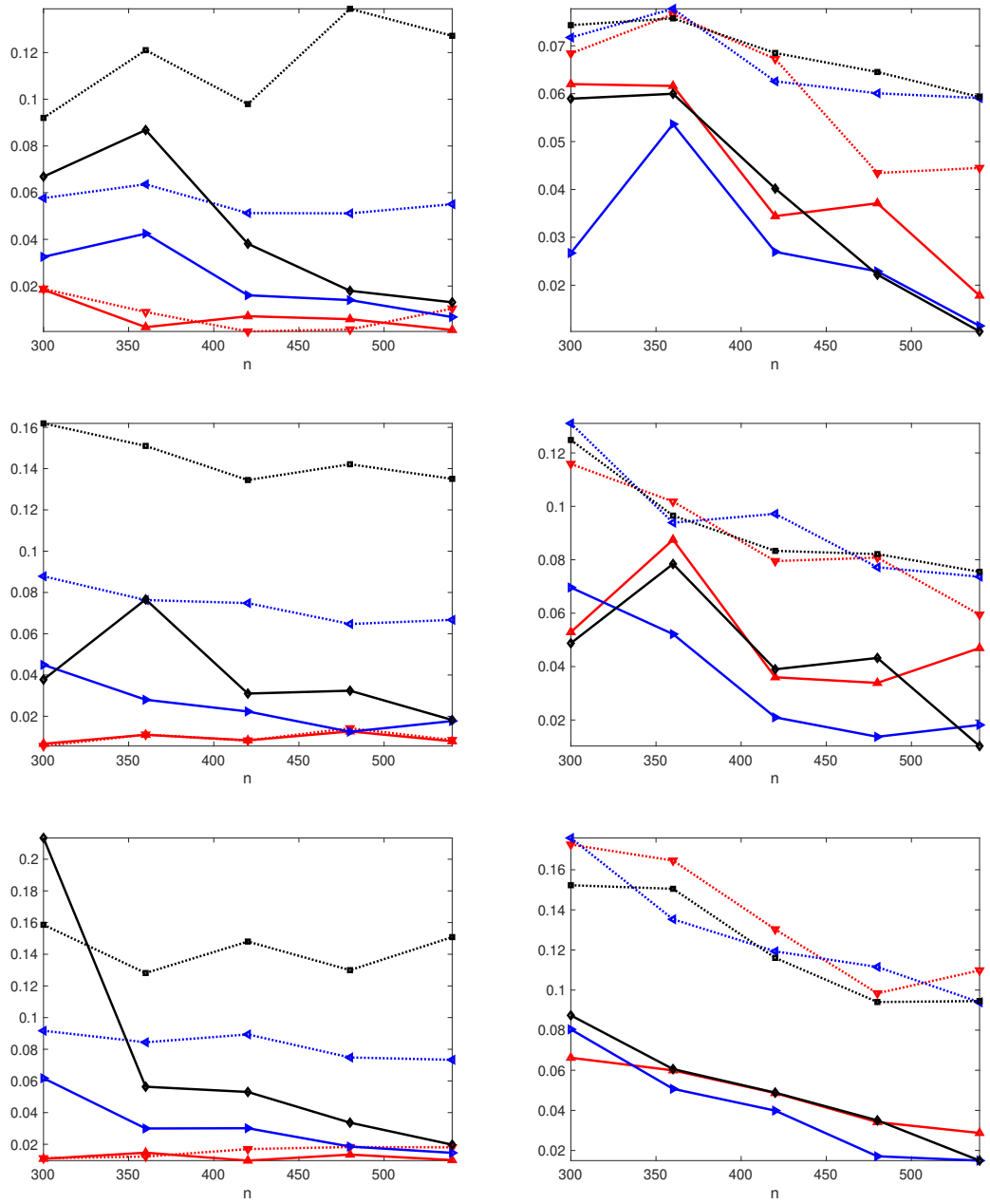

Figure 3.3: The false positive rates ρ_{FP} (left panels) and the rates Δ_{FN} (right panels) for $K = 4$ (top), $K = 5$ (middle) and $K = 6$ (bottom) clusters. The rates are evaluated over 50 simulation runs. The number of nodes ranges from $n = 300$ to $n = 540$ with the increments of 60. Dashed lines represent the results for $\omega = 0.5$ and solid lines represent the results for $\omega = 0.8$; $\sigma = 0.3$ (red), $\sigma = 0.5$ (blue) and $\sigma = 0.7$ (black).

entries in P_* that are estimated by zeros in $\hat{P}$. The reports on the accuracies of estimating J_* are presented in Figure 3.3. The left panels display ρ_{FP} while the right ones exhibit Δ_{FN}, as functions of the number of nodes for the same settings as in Figure 4.3.

Remark 3.7.1. Unknown number of clusters. In our previous simulations we treated the true number of clusters as a known quantity. However, we can actually use $\hat{P}$ to obtain an estimator $\hat{K}$ of K by solving, for every suitable K, the optimization problem (3.10), which can be equivalently rewritten as

$$\hat{K} = \underset{K}{\operatorname{argmin}}\{\|\hat{P} - A\|_F^2 + \operatorname{Pen}(n, J, K)\}. \tag{3.29}$$

The penalties $\operatorname{Pen}(n, J, K)$ defined in (3.14) are, however, motivated by the objective of setting it above the noise level with a very high probability. In our simulations, we also study the selection of an unknown K using an empirical version of this penalty

$$\operatorname{Pen}(n, J, K) = \rho(A) n K \sqrt{\ln n \, (\ln K)^3}. \tag{3.30}$$

In order to assess the accuracy of $\hat{K}$ as an estimator of K, we evaluated $\hat{K}$ as a solution of optimization problem (3.29) with the penalty (3.30) in each of the previous simulations settings over 50 simulation runs. Table 3.1 presents the relative frequencies of the estimators $\hat{K}$ of K_* for K_* ranging from 3 to 5, $n = 360$ and 480 and $\omega = 0.5$ and 0.8 and $\sigma = 0.4$, 0.6 and 0.8. Table 3.1 confirms that for majority of settings, $\hat{K} = K_*$, the true number of clusters, with high probability.

Table 3.1: The relative frequencies of the estimators $\hat{K}$ of K_* for K_* ranging from 3 to 5, $n = 360$ and 480 and $\omega = 0.5$ and 0.8 and $\sigma = 0.4$, 0.6 and 0.8.

		$n = 360$					
		$\omega = 0.5$			$\omega = 0.8$		
K_*	$\hat{K}$	$\sigma = 0.4$	$\sigma = 0.6$	$\sigma = 0.8$	$\sigma = 0.4$	$\sigma = 0.6$	$\sigma = 0.8$
3	2	0	0	0.02	0	0	0
	3	**0.58**	**0.60**	**0.58**	**0.58**	**0.76**	**0.88**
	4	0.26	0.28	0.32	0.28	0.18	0.12
	5	0.12	0.12	0.08	0.12	0.06	0
	6	0.04	0	0	0.02	0	0
4	2	0	0	0	0	0	0
	3	0	0.02	0.02	0	0	0.02
	4	**0.68**	**0.76**	**0.68**	**0.76**	**0.78**	**0.84**
	5	0.22	0.20	0.26	0.20	0.18	0.12
	6	0.10	0.02	0.04	0.04	0.04	0.02
5	2	0	0	0	0	0	0
	3	0.02	0	0	0	0	0
	4	0.04	0.10	0.30	0.02	0	0.14
	5	**0.74**	**0.78**	**0.50**	**0.84**	**0.94**	**0.78**
	6	0.20	0.12	0.20	0.14	0.06	0.08

		$n = 480$					
		$\omega = 0.5$			$\omega = 0.8$		
K_*	$\hat{K}$	$\sigma = 0.4$	$\sigma = 0.6$	$\sigma = 0.8$	$\sigma = 0.4$	$\sigma = 0.6$	$\sigma = 0.8$
3	2	0	0	0	0	0	0
	3	**0.66**	**0.52**	**0.58**	**0.56**	**0.72**	**0.86**
	4	0.28	0.34	0.28	0.28	0.20	0.10
	5	0.04	0.12	0.14	0.14	0.08	0.04
	6	0.02	0.02	0	0.02	0	0
4	2	0	0	0	0	0	0
	3	0.02	0	0.02	0	0	0
	4	**0.56**	**0.74**	**0.68**	**0.82**	**0.78**	**0.86**
	5	0.30	0.24	0.28	0.16	0.18	0.12
	6	0.12	0.02	0.02	0.02	0.04	0.02
5	2	0	0	0	0	0	0
	3	0	0	0.02	0	0	0
	4	0.06	0.04	0	0	0	0
	5	**0.72**	**0.86**	**0.84**	**0.86**	**0.88**	**0.90**
	6	0.22	0.10	0.14	0.14	0.12	0.10

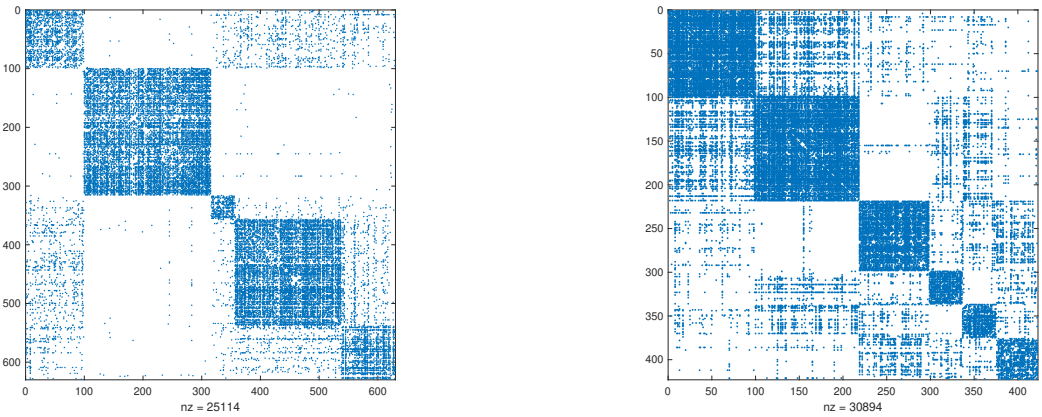

Figure 3.4: The adjacency matrices of the ego-network with 25114 nonzero entries and 5 clusters (left) and the brain network with 30894 nonzero entries and 6 clusters (right) after clustering

3.7.2 Real data examples

In this section, we report the performance of SSC and our estimation procedure when they are applied to two real life networks, an ego-network and a human brain network.

To study the ego-network, we use the dataset described comprehensively in [36]. An ego-network is a social network of a single person, with the exclusion of the person generating this network. Users of social networking sites are usually provided with a tool that allows them to organize their networks into categories, referred to, in [36], as *social circles*. Practically all major social networking cites provide such functionality, for example, "circles" on Google+, and "lists" on Facebook and Twitter. Examples of such circles include university classmates, sports team members, relatives, etc. Once circles are created by a user, they can be utilized, for example, for content filtering (e.g. to filter status updates posted by distant acquaintances) or for privacy (e.g., to hide personal information from coworkers).

Here, we attempt to recover social circles of an ego-network when only binary connection data is available. In particular, we formulate the problem of circle detection as a clustering problem on an individual ego-network. In principle, circles can overlap or a circle can be a subset of another circle, hence, as an example in this chapter, we study an ego-network with only few nodes overlap between the circles which does not affect the performance of the clustering method. Specifically, we study an ego-network from Facebook where user profiles are treated as nodes and a friendship between two user profiles is considered as an edge between them. Since a friendship is a mutual tie, the ego-network is undirected. The ego-network studied here, has 777 nodes with 17 circles, each circle containing between 2 to 225 nodes. For our study, we extract the five largest circles of the this network, obtaining a network with 629 nodes and 12557 edges. We carried out clustering of the nodes using the SSC and compared the clustering assignments of SSC with the true class assignments. The SSC provides 85% accuracy. In addition, we applied formula (3.29) with K ranging from 2 to 6 to the adjacency matrix with the randomly permuted rows (columns), obtaining the true number of clusters with 100% accuracy over 10 runs. Figure 4.4 shows the adjacency matrix of the graph after clustering (left), which confirms that the network indeed follows the SPABM. Indeed, the SPABM is a very appropriate model for this example since users display different degrees of connections to users in other circles, and, furthermore, the network is sparse, which justifies the application of the SPABM.

Our second example involves analyzing a human brain functional network, measured using the resting-state functional MRI (rsfMRI). We use the the brain connectivity dataset presented as a GroupAverage rsfMRI matrix described in [12]. In this dataset, the brain is partitioned into 638 distinct regions and a weighted graph is used to characterize the network topology. Nicolini *et al.* [45] developed a new Asymptotical Surprise method, which is applied for clustering the weighted graph. Asymptotical Surprise detects 47 communities

ranging from 1 to 133. Since the true clustering as well as the true number of clusters are unknown for this dataset, we treat the results of the Asymptotical Surprise as the ground truth.

In order to generate a binary network, we set all nonzero weights to one in the GroupAverage rsfMRI matrix, obtaining a network with 18625 undirected edges. For evaluating the performance of SSC on this network, we extract 6 largest communities derived by the Asymptotical Surprise, obtaining a network with 422 nodes and 15447 edges. Applying (3.29), with K ranging from 2 to 10, to the adjacency matrix with the randomly permuted rows (columns), we recovered the true number of clusters with 70% accuracy over 10 simulation runs. For this true number of communities, our version of the SSC detects the true communities with 94% accuracy. Figure 4.4 (right) shows the adjacency matrix of the network after clustering, showing that the network is very sparse. In addition, the SPABM provides a significantly tighter fit than the SBM with estimation errors $n^{-2}\,\|\hat{P}-A\|_F^2$ being 0.056 and 0.090, respectively, when $\hat{P}$ is estimated according to SPABM and SBM on the basis of the true clustering. Those considerations justify application of the SPABM to the data.

CHAPTER 4: THE HIERARCHY OF BLOCK MODELS

4.1 An overview of block models

Consider an undirected network with n nodes that are partitioned into K communities $\mathcal{N}_k$, $k = 1, \ldots, K$, by a clustering function $z : \{1, \ldots, n\} \to \{1, \ldots, K\}$ with the corresponding clustering matrix Z. Here, we shall deal only with the graphs where each node belongs to one and only one community, thus, leaving aside the mixed membership models [4], [26]. Denote by B the matrix of average connection probabilities between communities, so that for $k, l = 1, 2, \cdots, K$, one has

$$B_{k,l} = \frac{1}{n_k\, n_l} \sum_{i,j=1}^{n} P_{ij}\, I(z(i) = k)\, I(z(j) = l), \tag{4.1}$$

where n_k is the number of nodes in the community k.

In order to better understand the relationships between various block models, consider a rearranged version $P(Z)$ of matrix P where its first n_1 rows correspond to nodes from class 1, the next n_2 rows correspond to nodes from class 2 and the last n_K rows correspond to nodes from class K. Denote the (k_1, k_2)-th block of matrix $P(Z)$ by $P^{(k_1,k_2)}(Z)$. Then, the block models vary by how dissimilar matrices $P^{(k_1,k_2)}(Z)$ are. Indeed, under the SBM

$$P^{(k_1,k_2)}(Z) = B_{k_1,k_2} 1_{n_{k_1}} 1_{n_{k_2}}^T \tag{4.2}$$

where 1_k is the k-dimensional column vector with all elements equal to one. In the DCBM, there exists a vector $h \in \mathbb{R}_+^n$, with sub-vectors $h^{(k)} \in \mathbb{R}_+^{n_k}$, $k = 1, \ldots, K$, such that, for

$k_1, k_2 = 1, 2, \cdots, K$,

$$P^{(k_1,k_2)}(Z) = B_{k_1,k_2} h^{(k_1)} (h^{(k_2)})^T. \tag{4.3}$$

In the PABM, instead of one vector h, there are K vectors $\Lambda^{(1)}, \cdots, \Lambda^{(K)}$ with sub-vectors

$$\Lambda^{(k_1,k_2)} \in \mathbb{R}_+^{n_{k_1}}, \quad k_1, k_2 = 1, 2, \cdots, K. \tag{4.4}$$

In this case, vectors $\Lambda^{(k)}$ form the $(n \times K)$ matrix Λ with columns partitioned into sub-columns $\Lambda^{(k_1,k_2)}$, and

$$P^{(k_1,k_2)}(Z) = B_{k_1,k_2} \Lambda^{(k_1,k_2)} (\Lambda^{(k_2,k_1)})^T, \tag{4.5}$$

for every $k_1, k_2 = 1, 2, \cdots, K$. Hence, (4.2) and (4.3) coincide if $h \equiv 1_n$, and (4.5) reduces to (4.3) if all columns of matrix Λ are identical, i.e.

$$\Lambda^{(k_1,k_2)} \equiv h^{(k_1)}, \quad k_1, k_2 = 1, 2, \cdots, K. \tag{4.6}$$

Since in the DCBM there is only one vector h that models heterogeneity in probabilities of connections, the ratios $P_{i_1,j}/P_{i_2,j}$ of the probabilities of connections of two nodes, i_1 and i_2, that belong to the same community, are determined entirely by the nodes i_1 and i_2 and are independent of the community with which those nodes interact. On the other hand, for the PABM, each node has a different degree of popularity (interaction level) with respect to every other community, so that $P_{i_1,j_1}/P_{i_2,j_1} \neq P_{i_1,j_2}/P_{i_2,j_2}$ if nodes j_1 and j_2 belong to different communities. In the PABM, those variable popularities are described by the matrix $\Lambda \in [0,1]^{n \times K}$ which reduces to a single vector h in the case of the DCBM. One can easily imagine the situation where nodes do not exhibit different levels of activity with respect to every community but rather with respect to some groups of communities, "*mega-communities*", so that there are L, $1 \leq L \leq K$, different vectors $H^{(l)} \in \mathbb{R}_+^n, \quad l = 1, 2, \cdots, L$,

and each of columns Λ_k, $k = 1, 2, \cdots, K$, of matrix Λ is equal to one of vectors $H^{(l)}$. In other words, there exists a clustering function $c : \{1, ..., K\} \to \{1, ..., L\}$ with the corresponding clustering matrix C such that

$$\Lambda_k = H^{(l)},\ l = c(k),\ l = 1, \ldots, L,\ k = 1, \ldots, K.$$

We name the resulting model the *Heterogeneous Block Model* (HBM) to emphasize that, beyond the average connection probabilities of communities, the mega-communities are determined by the heterogeneity of the probabilities of connections.

4.2 The Heterogeneous Stochastic Block Model (HBM)

The HBM contains two types of communities, the regular communities that can be distinguished by the average probabilities of connections between them (like in the SBM or the DCBM) and the mega-communities that are described by the heterogeneity of probabilities of connections of individual nodes across the communities.

The idea of mega-communities is not entirely new. It was introduced in [56] and recently appeared in [37]. The difference between this chapter and the above cited publications is that in [56] and [37] the mega-communities are determined by intermediate results of the clustering algorithms while we define them on the basis of the heterogeneous patterns of the connection probabilities of nodes with respect to different communities.

For any M and $K \leq M$, denote by $\mathcal{M}_{M,K}$ the collection of all clustering matrices $Z \in \{0, 1\}^{M \times K}$ with the corresponding clustering function $z : \{1, \ldots, M\} \to \{1, \ldots, K\}$ such that $Z_{i,k} = 1$ iff $z(i) = k$, $i = 1, \ldots, M$. Then, $Z^T Z = \mathrm{diag}(n_1, \ldots, n_K)$ where n_k is the size of community k, $k = 1, \ldots, K$. The HBM, with K communities and $L \leq K$

mega-communities, is defined by two clustering matrices $Z \in \mathcal{M}_{n,K}$ and $C \in \mathcal{M}_{K,L}$ with corresponding clustering functions z and c that, respectively, partition the n nodes into K communities, and K communities into L mega-communities. If the l-th mega-community consists of K_l communities and the community sizes are n_k, then the total number of nodes in mega-community l is N_l, where

$$N_l = \sum_{k=1}^{K} n_k \, I(c(k) = l), \quad \sum_{l=1}^{L} K_l = K, \quad \sum_{l=1}^{L} N_l = n, \quad l = 1, \cdots, L. \tag{4.7}$$

The communities are characterized by their average connection probability matrix with elements B_{k_1,k_2}, $k_1, k_2 = 1, 2, \ldots, K$, defined in (4.1). In order to better understand the mega-communities, consider a permutation matrix $\mathscr{P}_{Z,C}$ that arranges nodes into communities consecutively, and orders communities so that the K_l blocks within the l-th mega-community are consecutive, $l = 1, 2, \ldots, L$. Recall that $\mathscr{P}_{Z,C}$ is an orthogonal matrix with $\mathscr{P}_{Z,C}^{-1} = \mathscr{P}_{Z,C}^{T}$ and denote

$$P(Z, C) = \mathscr{P}_{Z,C}^{T} P \mathscr{P}_{Z,C}, \quad P = \mathscr{P}_{Z,C} P(Z, C) \mathscr{P}_{Z,C}^{T}.$$

According to Z and C, matrix P is partitioned into K^2 blocks $P^{(k_1,k_2)}(Z, C) \in [0, 1]^{n_{k_1} \times n_{k_2}}$, $k_1, k_2 = 1, \ldots, K$, with the block-averages given by (4.1). In addition, blocks $P^{(k_1,k_2)}(Z, C)$ can be combined into the L^2 mega-blocks $\tilde{P}^{(l_1,l_2)}(Z, C) \in [0, 1]^{N_{l_1} \times N_{l_2}}$, corresponding to probabilities of connections between mega-communities l_1 and l_2, $l_1, l_2 = 1, \ldots, L$. Consider matrix $H \in \mathbb{R}_{+}^{n \times L}$ (Figure 4.1, top middle), where each column H_l, $l = 1, \ldots, L$, can be partitioned into K sub-vectors $h^{(k,l)} \in \mathbb{R}_{+}^{n_k}$ of lengths n_k, $k = 1, \ldots, K$. Those sub-vectors are combined into L mega sub-vectors $H^{(m,l)} \in \mathbb{R}_{+}^{N_m}$ of lengths N_m, $m = 1, \cdots, L$, according to matrix C, where N_m is defined in (4.7). Similarly, matrix $B \in [0, 1]^{K \times K}$ of block probabilities is partitioned into sub-matrices $B^{(l_1,l_2)} \in [0, 1]^{K_{l_1} \times K_{l_2}}$, $l_1, l_2 = 1, \cdots, L$. With these notations,

Figure 4.1: Matrices associated with the HBM with $K = 5$, $L = 2$, $K_1 = 3$, $K_2 = 2$. Bold lines identify mega-blocks. Top left: matrix B partitioned into blocks $B^{(l_1,l_2)}$. Top, middle: matrix H. Top right: matrix H with columns expressed via vectors $h^{(k,l)}$ and repeated: column 1- K_1 times; column 2 - K_2 times. Bottom: the probability matrix with K^2 blocks and L^2 mega-blocks.

for any $l_1, l_2 = 1, \cdots, L$, the (l_1, l_2)-th mega-block of P can be presented as

$$\tilde{P}^{(l_1,l_2)}(Z, C) = \left(H^{(l_1,l_2)}(H^{(l_2,l_1)})^T\right) \circ \left(J^{(l_1)} B^{(l_1,l_2)} (J^{(l_2)})^T\right), \tag{4.8}$$

where $A \circ B$ is the Hadamard product of A and B, and matrices $J^{(l)} \in \{0,1\}^{N_l \times K_l}$, $l =$

$1, \ldots, L$, are of the form

$$J^{(l)} = \begin{bmatrix} 1_{n_{k_1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & 1_{n_{k_2}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & 1_{n_{k_{K_l}}} \end{bmatrix}. \tag{4.9}$$

In order for the model to be identifiable, we impose the following assumptions:

A1. Matrix B is non-singular with $\lambda_{\min}(B) \geq \lambda_0 > 0$.

A2. For each $k = 1, \cdots, K$, vectors $H^{(k,l)}$, $l = 1, \cdots, L$, are linearly independent.

By rewriting (4.8) in an equivalent form, one can conclude that each of the mega-blocks $\tilde{P}^{(l_1,l_2)}(Z, C)$ (and, hence, $\tilde{P}^{(l_1,l_2)}$ if we scramble them to the original order) follows the (non-symmetric) DCBM model with $K_{l_1} \times K_{l_2}$ blocks. Specifically, for a pair of sub-vectors $H^{(l_1,l_2)} \in \mathbb{R}_+^{N_{l_1}}$ and $H^{(l_2,l_1)} \in \mathbb{R}_+^{N_{l_2}}$ of matrix H and a matrix $B^{(l_1,l_2)} \in [0,1]^{K_{l_1} \times K_{l_2}}$ containing average probabilities of connections for each pair of communities within the mega-community (l_1, l_2) one has

$$\tilde{P}^{(l_1,l_2)}(Z, C) = Q^{(l_1,l_2)} J^{(l_1)} B^{(l_1,l_2)} (J^{(l_2)})^T Q^{(l_2,l_1)}.$$

Here, $Q^{(l_1,l_2)} = \mathrm{diag}(H^{(l_1,l_2)})$ and the (k_1, k_2)-th block of P is given by

$$P^{(k_1,k_2)}(Z, C) = B_{k_1,k_2} h^{(k_1,l_2)} \left(h^{(k_2,l_1)}\right)^T, \tag{4.10}$$

where $l_i = c(k_i)$, $i = 1, 2$, and $h^{(k,l)} \in \mathbb{R}_+^{n_k}$ is a sub-vector of $H^{(m,l)}$ with $m = c(k)$. Observe that the formulation above imposes a natural scaling on the sub-vectors $h^{(k,l)}$ of H, since it follows from equations (4.1) and (4.10), that for any pair of communities (k_1, k_2) which

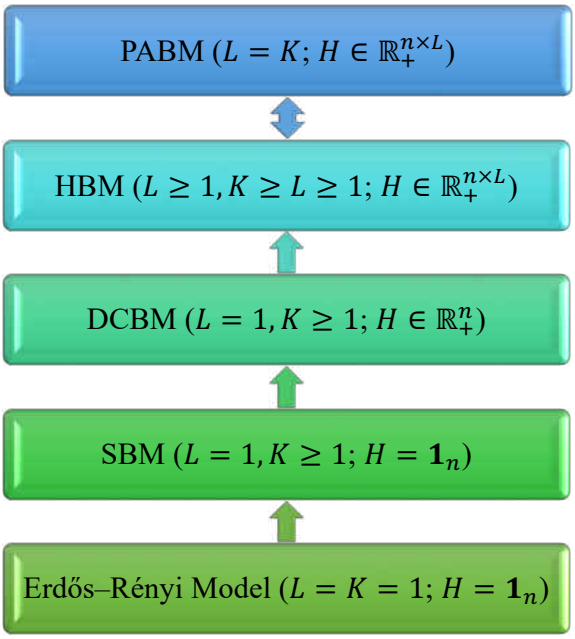

Figure 4.2: The hierarchy of block models

belong to a pair of mega-communities (l_1, l_2), one has

$$n_{k_1}\, n_{k_2}\; B_{k_1,k_2} = 1_{k_1}^T\, P^{(k_1,k_2)}(Z,C)\, 1_{k_2} = B_{k_1,k_2}\, \left(1_{k_1}^T h^{(k_1,l_2)}\right)\, \left(1_{k_2}^T\, h^{(k_2,l_1)}\right).$$

The latter implies that for any $k = 1, \ldots, K$ and $l = 1, \ldots, L$,

$$1_k^T h^{(k,l)} = n_k, \quad k = 1, \ldots, K,\ l = 1, \ldots, L. \tag{4.11}$$

Now, it is easy to see that all block models, the SBM, the DCBM and the PABM, can be viewed as particular cases of the HBM introduced above. Indeed, the DCBM is a particular case of the HBM with $L = 1$ while the PABM corresponds to the setting of $L = K$. Finally, due to (4.11), the SBM constitutes a particular case of the HBM with $L = 1$ and matrix H reduced to vector 1_n, the n-dimensional column vector with all entries equal to one.

Moreover, the absence of the community structure (whether in the SBM or the DCBM) is equivalent to $K = 1$, and implies that the HBM necessarily reduces to the DCBM.

4.3 Optimization procedure for estimation and clustering

Note that, in terms of the matrices $J^{(l)}$ defined in (4.9), the scaling conditions (4.11) appear as

$$(J^{(l)})^T\, Q^{(l,l')}\, J^{(l)} = (J^{(l)})^T\, J^{(l)},\ l, l' = 1, ..., L. \tag{4.12}$$

Let $\mathscr{P}_{\widehat{Z},\widehat{C}}$ be the permutation matrix corresponding to $\widehat{Z} \in \mathcal{M}_{n,\hat{K}}$ and $\widehat{C} \in \mathcal{M}_{\hat{K},\hat{L}}$. Consider the set $\Im(n, K, L)$ of matrices Θ with blocks $\Theta^{(l_1,l_2)} \in [0,1]^{N_{l_1} \times N_{l_2}}$, $l_1, l_2 = 1, ..., L$, such that

$$\begin{aligned}
&\Theta = \bigcup_{l_1,l_2} \Theta^{(l_1,l_2)},\ \Theta^{(l_1,l_2)} = Q^{(l_1,l_2)} J^{(l_1)} B^{(l_1,l_2)} (J^{(l_2)})^T Q^{(l_2,l_1)},\\
&B^{(l_1,l_2)} \in [0,1]^{K_{l_1} \times K_{l_2}},\ Q^{(l_1,l_2)} \in \mathcal{D}_{l_1},\\
&Z \in \mathcal{M}_{n,K},\ C \in \mathcal{M}_{K,L},\ l_1, l_2 = 1, ..., L,
\end{aligned} \tag{4.13}$$

where $\mathcal{D}_m$ the set of diagonal matrices with diagonals in $\mathbb{R}_+^m$ and conditions (4.7) and (4.12) hold. Then, it is easy to see that $P = \mathscr{P}_{Z,C}^T \Theta \mathscr{P}_{Z,C}$, so its estimator can be obtained as

$$\hat{P} = \mathscr{P}_{\widehat{Z},\widehat{C}}\, \widehat{\Theta}(\widehat{Z}, \widehat{C})\, \mathscr{P}_{\widehat{Z},\widehat{C}}^T. \tag{4.14}$$

Here, for given values of K and L, $(\widehat{Z}, \widehat{C}, \widehat{\Theta})$ is a solution of the following optimization problem

$$(\widehat{Z}, \widehat{C}, \widehat{\Theta}) \in \underset{Z,C,\Theta}{\operatorname{argmin}} \|A(Z, C) - \Theta\|_F^2 \tag{4.15}$$

subject to conditions $A(Z, C) = \mathscr{P}_{Z,C}^T A \mathscr{P}_{Z,C}$, (4.7), (4.12) and (4.13). In real life, however, the values of K and L are unknown and need to be incorporated into the optimization

problem by adding a penalty $\mathrm{Pen}(K, L)$ on K and L:

$$(\widehat{\Theta}, \widehat{Z}, \widehat{C}, \hat{K}, \hat{L}) \in \underset{Z,C,K,L,\Theta}{\operatorname{argmin}} \left\{ \|A(Z, C) - \Theta\|_F^2 + \mathrm{Pen}(K, L) \right\}, \tag{4.16}$$

where optimization is carried out subject to conditions $A(Z, C) = \mathscr{P}_{Z,C}^T A \mathscr{P}_{Z,C}$, (4.7), (4.12) and (4.13). After that, the estimator $\hat{P}$ of P_* can be obtained as (4.14).

In practice, one would need to solve optimization problem (4.15) for each $K = 1, ..., n$ and $L = 1, \ldots, K$, and then find the values $(\hat{K}, \hat{L})$ that minimize the right hand side in (4.16). After that, the estimator $\hat{P}$ of P is obtained as (4.14). Then, the following statement holds.

Theorem 4.3.1. *Let Assumptions* **A1** *and* **A2** *hold. Let* $(\widehat{\Theta}, \widehat{Z}, \widehat{C}, \hat{K}, \hat{L})$ *be a solution of optimization problem* (4.16) *subject to conditions* $A(Z, C) = \mathscr{P}_{Z,C}^T A \mathscr{P}_{Z,C}$, (4.7), (4.12) *and* (4.13) *with*

$$Pen(K, L) = C_1(nL + K^2) \ln n + C_2 n \ln K \tag{4.17}$$

where C_1 *and* C_2 *are absolute constants. Then, for the estimator* $\hat{P}$ *given by* (4.14)*, the true matrix* P_* *and any* K*,* L*,* $Z \in \mathcal{M}_{n,K}$*,* $C \in \mathcal{M}_{K,L}$ *and any matrix* $P = \mathscr{P}_{Z,C} \Theta \mathscr{P}_{Z,C}^T$ *with* $\Theta \in \Im(n, K, L)$*, one has*

$$\mathbb{P}\left\{ \|\hat{P} - P_*\|_F^2 \le 3\big[\|P - P_*\|_F^2 + Pen(K, L)\big] \right\} \ge 1 - (n^2 \log_2 n + 1) e^{-n/32},$$
$$\mathbb{E}\|\hat{P} - P_*\|_F^2 \le 3\big[\|P - P_*\|_F^2 + Pen(K, L)\big] + n^5 e^{-n/32}.$$

Solution of optimization problem (4.16) requires a search over the continuum of matrices Θ. In order to simplify the estimation, we consider a solution of a somewhat simpler optimization problem. It is easy to observe (see Figure 4.1) that each of the block columns of matrix P is a matrix of rank one and, given the clustering, it can be obtained by the rank one projection of

the respective adjacency sub-matrix. Denote the block columns of the re-arranged matrices P and A by $P^{(l,k)}(Z,C)$ and $A^{(l,k)}(Z,C)$. Then, the optimization problem appears as

$$(\widehat{Z},\widehat{C},\hat{K},\hat{L}) \in \underset{Z,C,K,L}{\operatorname{argmin}} \left\{ \sum_{l=1}^{L}\sum_{k=1}^{K} \left\| A^{(l,k)}(Z,C) - \Pi_{(1)}\left(A^{(l,k)}(Z,C)\right)\right\|_F^2 + \overline{\mathrm{Pen}}(K,L) \right\} \quad (4.18)$$
$$\text{s.t. } A(Z,C) = \mathscr{P}_{Z,C}^T A \mathscr{P}_{Z,C},$$

where $\Pi_{(1)}\left(A^{(l,k)}(Z,C)\right)$ is the rank one projection of the matrix $A^{(l,k)}(Z,C)$. Then, $\widehat{\Theta}$ is the block matrix with blocks $\widehat{\Theta}^{(l,k)} = \Pi_{(1)}\left(A^{(l,k)}(\widehat{Z},\widehat{C})\right)$, $l = 1,\cdots,\hat{L},\ k = 1,\cdots,\hat{K}$.

Theorem 4.3.2. *Let Assumptions* **A1** *and* **A2** *hold. Let* $(\widehat{\Theta},\widehat{Z},\widehat{C},\hat{K},\hat{L})$ *be a solution of optimization problem* (4.18) *with* $\overline{Pen}(K,L)$ *of the form*

$$\overline{Pen}(K,L) = \Psi_1 nK + \Psi_2 K^2 \ln n + \Psi_3 n \ln K, \quad (4.19)$$

where Ψ_1, Ψ_2, *and* Ψ_3 *are positive absolute constants. Then, for the estimator* $\hat{P}$ *of* P_* *given by* (4.14) *and any* $t > 0$, *one has*

$$\mathbb{P}\left\{ \left\|\hat{P} - P_*\right\|_F^2 \le \tilde{C}\left[\overline{Pen}(n,K_*,L_*) + t\right]\right\} \ge 1 - 3e^{-t},$$
$$\mathbb{E}\left\|\hat{P} - P_*\right\|_F^2 \le \tilde{C}\left[\overline{Pen}(n,K_*,L_*) + 3\right].$$

Here K_* *and* L_* *are the true number of communities and mega-communities and* $\tilde{C} = \tilde{C}(\Psi_1,\Psi_2,\Psi_3) > 0$ *is an absolute constant.*

Observe that Theorem 4.3.2 asserts smaller error rates if $K_*/L_* \ll \ln n$, i.e., if n is large.

4.4 Implementation of clustering

The optimization procedure in (4.16) is NP-hard. In this section, we describe a computationally tractable clustering procedure that can replace it. Since the model requires identification of mega-communities and communities, naturally, the clustering is carried out in two steps. First, we find the clustering matrix C that arranges the nodes into L mega-communities. Subsequently, we detect communities within each of the mega-communities, obtaining the clustering matrix Z.

In order to accomplish the first task, we use the fact that, for a given L, under Assumption **A2**, the columns of matrix P_* lie in the union of L distinct subspaces. Finding those subspaces can be carried out by the subspace clustering. Subspace clustering is widely used in, e.g., computer vision and is designed for separation of points that lie in the union of subspaces. While subspace clustering can be implemented by a variety of techniques, here we use spectral clustering based methods [15], [17], [38], [51]. In particular, we apply the Sparse Subspace Clustering (SSC) [15] which is based on representation of each of the vectors as a sparse linear combination of all other vectors, with the expectation that a vector is more likely to be represented as a linear combination of vectors in its own subspace rather than other subspaces.

If matrix P_* were known, the weight matrix W would be based on writing every data point as a sparse linear combination of all other points by minimizing the number of nonzero coefficients

$$\min_{W_j} \|W_j\|_0 \quad \text{s.t.} \quad (P_*)_j = \sum_{k \neq j} W_{k,j} (P_*)_k \tag{4.20}$$

where, for any matrix B, B_j is its j-th column. The affinity matrix of the SSC is the symmetrized version of the weight matrix W. Note that since, due to Assumption **A2**, the

Algorithm 1: The SSC procedure

Input: Adjacency matrix A, number of clusters k, tuning parameters γ_1, γ_2
Output: Clustering matrix C
Steps:
1: For $j = 1, ..., n$, find $\widehat{W}_j$ in (4.22)
2: Apply spectral clustering to the affinity matrix $|\widehat{W}| + |\widehat{W}^T|$ to find clustering matrix C

subspaces are linearly independent, the solution to the optimization problem (4.20) is W_j such that $W_{k,j} \neq 0$ only if points k and j are in the same subspace. Since the problem (4.20) is NP-hard, one usually solves its convex relaxation

$$\min_{W_j} \|W_j\|_1 \quad \text{s.t.} \quad (P_*)_j = \sum_{k \neq j} W_{k,j}(P_*)_k \tag{4.21}$$

In the case of data contaminated by noise, the SSC algorithm does not attempt to write data as an exact linear combination of other points and replaces (4.21) by penalized optimization. Here, we solve the elastic net problem

$$\widehat{W}_j \in \operatorname*{argmin}_{W_j} \left\{ \left[0.5 \|A_j - AW_j\|_2^2 + \gamma_1 \|W_j\|_1 + \gamma_2 \|W_j\|_2^2 \right] \text{ s.t. } W_{j,j} = 0 \right\}, \ j = 1, \ldots, n, \tag{4.22}$$

where $\gamma_1, \gamma_2 > 0$ are tuning parameters. The quadratic term stabilizes the LASSO problem by making the problem strongly convex. We solve (4.22) using the a fast version of the LARS algorithm implemented in SPAMS Matlab toolbox [42]. Given $\widehat{W}$, the clustering matrix C is then obtained by applying spectral clustering to the affinity matrix $|\widehat{W}| + |\widehat{W}^T|$, where, for any matrix B, matrix $|B|$ has absolute values of elements of B as its entries. Algorithm 1 summarizes the SSC procedure described above. Once the mega-communities are discovered, one needs to detect communities inside of each mega-community. Since each mega-community has been generated by a distinct column of H, it follows the non-

Algorithm 2: Spectral clustering with k-median

Input: Adjacency matrix $A \in \{0,1\}^{n\times n}$, number of clusters k
Output: Community assignment
Steps:
1: Find $\hat{P} = \Pi_{(k)}(A)$, the best rank k approximation of matrix A
2: For $j = 1, ..., n$, find $\tilde{P}_j = \hat{P}_j / \|\hat{P}_j\|_1$
3: Apply spectral clustering to $\tilde{P}$ to obtain community assignment

symmetric DCBM. One of the popular clustering methods for the DCBM is the weighted k-median algorithm used in [35] and [21]. Algorithm 2 follows [21]. For the known number of communities K, the algorithm starts with estimating the probability matrix P by the best rank K approximation of the adjacency matrix, obtaining $\hat{P} = UDU^T$, where $U \in \mathbb{R}^{n\times K}$ contains K leading eigenvectors and D is a diagonal matrix of top K eigenvalues. After that, the columns of $\hat{P}$ are normalized, leading to $\tilde{P}_i = \hat{P}_i / \|\hat{P}_i\|_1$, $i = 1, 2, \ldots, n$. Finally, the K-median spectral clustering is applied to $\tilde{P}$ to find the community assignment.

In the first step of clustering, we apply Algorithm 1 to the adjacency matrix A with $k = L$ to find L mega-communities defined by the clustering matrix C. In the second step, Algorithm 2 is applied to each of L mega-communities, obtained at the first step. Specifically, we apply Algorithm 2 with $k = K_l$ and $n = N_l$ to cluster the l-th mega-community, $l = 1, ..., L$. The union of these communities combined with the clustering matrix C, yields the clustering matrix Z.

4.5 Simulations and real data examples

4.5.1 Simulations on synthetic networks

In the experiments with synthetic data, we generate networks with n nodes, L mega-communities and K communities that fit the HBM. For simplicity, we consider perfectly balanced networks where the number of nodes in each community and mega-community are respectively n/K and n/L, and there are K/L communities in each mega-community. First, we generate L distinct n-dimensional random vectors with entries between 0 and 1. To this end, we generate a random vector $Y \in (0,1)^n$ and partition it into K blocks $Y^{(k)}$, $k = 1, ..., K$, of size n/K. The vector $\bar{h}^{(1)}$ is generated from Y by sorting each block of Y in ascending order. After that, we partition each of the K blocks, $\bar{h}^{(k,1)}$ of $\bar{h}^{(1)}$, into L sub-blocks $\bar{h}_i^{(k,1)}$, $i = 1, ..., L$, of equal size. To generate the k-th block $\bar{h}^{(k,2)}$ of $\bar{h}^{(2)}$, we reverse the order of entries in each sub-block $\bar{h}_i^{(k,1)}$ and rearrange them in descending order. The blocks $\bar{h}^{(k,s)}$ of subsequent vectors $\bar{h}^{(s)}$, $s = 3, ..., L$, are formed by re-arranging the order of sub-blocks $\bar{h}_i^{(k,2)}$ in each sub-vector $\bar{h}^{(k,2)}$. The L vectors $\bar{h}^{(l)}$, $l = 1, ..., L$, generated by this procedure have different patterns leading to detectable mega-communities. Subsequently, we scale the vectors as $H^{(k,l)} = (n/K)\,\bar{h}^{(k,l)}/\|\bar{h}^{(k,l)}\|_1$, $k = 1, ..., K$, $l = 1, ..., L$, obtaining matrix H. After that, we replicate K/L times each of the columns of H (Figure 4.1, top right) and denote the resulting matrix by $\tilde{H}$. Matrix B has entries

$$B_{k,l} = \tilde{B}_{k,l}\big((\tilde{H}_{max})_{k,l}\big)^{-2}, \quad k, l = 1, ..., K, \tag{4.23}$$

where $\tilde{B}$ is a $(K \times K)$ symmetric matrix with random entries between 0.35 and 1 to avoid very sparse networks, and the largest entries of each row (column) are on the diagonal.

Matrix $\tilde{H}_{max}$ is a $K \times K$ symmetric matrix defined as

$$(\tilde{H}_{max})_{k,l} = \max\left(\tilde{H}^{(k,l)}, \tilde{H}^{(l,k)}\right), \quad k,l = 1, ..., K,$$

where $\tilde{H}^{(k,l)}$ is the (k,l)-th block of matrix $\tilde{H}$. The term $\left((\tilde{H}_{max})_{k,l}\right)^{-2}$ in (4.23) guarantees that the entries of probability matrix $P(Z,C)$ do not exceed one. To control how assortative the network is, we multiply the off-diagonal entries of B by the parameter $\omega \in (0,1)$. The values of ω close to zero produce an almost block diagonal probability matrix $P(Z,C)$ while the values of ω close to one lead to $P(Z,C)$ with more diverse entries. We obtain the probability matrix $P(Z,C)$ as

$$P^{(k,l)}(Z,C) = B_{k,l}\, \tilde{H}^{(k,l)} \left(\tilde{H}^{(l,k)}\right)^T, \; k,l = 1, ..., K.$$

After that, to obtain the probability matrix P, we generate random clustering matrices $Z \in \mathcal{M}_{n,K}$ and $C \in \mathcal{M}_{K,L}$ and their corresponding $n \times n$ permutation matrices $\mathscr{P}(Z)$ and $\mathscr{P}(C)$, respectively. Subsequently, we set $\mathscr{P}_{Z,C} = \mathscr{P}(Z)\mathscr{P}(C)$ and obtain the probability matrix P as $P = \mathscr{P}_{Z,C} P(Z,C) (\mathscr{P}_{Z,C})^T$. Finally we generate the lower half of the adjacency matrix A as independent Bernoulli variables $A_{i,j} \sim \text{Bern}(P_{i,j})$, $i = 1, \ldots, n, j = 1, \ldots, i-1$, and set $A_{i,j} = A_{j,i}$ when $j > i$. In practice, the diagonal $\text{diag}(A)$ of matrix A is unavailable, so we estimate $\text{diag}(P)$ without its knowledge.

We apply Algorithm 1 to find the clustering matrix $\widehat{C}$. Since the diagonal elements of matrix A are unavailable, we initially set $A_{i,i} = 0$, $i = 1, ..., n$. We use $\gamma_1 = 30\rho(A)$ and $\gamma_2 = 125(1 - \rho(A))$ where $\rho(A)$ is the density of matrix A, the proportion of nonzero entries in A. The spectral clustering in step 2 of the Algorithm 1 is carried out by the normalized cut algorithm [53]. Once the mega-communities are obtained, we apply Algorithm 2 to detect communities inside each mega-community. The union of detected communities and

the clustering matrix $\widehat{C}$ yields the clustering matrix $\widehat{Z}$. Given $\widehat{Z}$ and $\widehat{C}$, we generate matrix $A(\widehat{Z},\widehat{C}) = \mathscr{P}^T_{\widehat{Z},\widehat{C}} A \mathscr{P}_{\widehat{Z},\widehat{C}}$ with blocks $A^{(k,l)}(\widehat{Z},\widehat{C})$, $k = 1,\ldots,K$, $l = 1,\ldots,L$, and obtain $\widehat{\Theta}^{(k,l)}(\widehat{Z},\widehat{C})$ by using the rank one projection for each of the blocks. Finally, we estimate matrix P by $\hat{P}$ given by formula (4.14).

We evaluated the accuracy of estimation and clustering in the setting above with $K = 6$, two values of L, $L = 2$ and $L = 3$, and the number of nodes ranging from $n = 180$ to $n = 720$ with the increments of 180. The proportion of misclustered nodes was evaluated as

$$\mathrm{Err}(Z,\widehat{Z}) = (2n)^{-1} \min_{\mathscr{P}_K \in \mathcal{P}_K} \|Z\mathscr{P}_K - \widehat{Z}\|_F \tag{4.24}$$

where $\mathcal{P}_K$ is the set of permutation matrices $\mathscr{P}_K : \{1,...,K\} \to \{1,...,K\}$. The accuracy of estimating the probability matrix P by $\hat{P}$ is measured as $n^{-2}\|\hat{P} - P\|_F^2$.

Figure 4.3 displays the accuracies of the two-step clustering procedure and the estimated probability matrix $\hat{P}$ in the above settings. We compare the results obtained by the two-step clustering procedure (solid lines) with the clustering results obtained by using only Algorithm 2 (dashed lines), where the post-clustering estimation is based on rank one approximations. The top panels present the clustering errors $\mathrm{Err}(\widehat{C},C)$, the middle ones show the clustering errors $\mathrm{Err}(\widehat{Z},Z)$, and the bottom panels exhibit the estimation errors $n^{-2}\|\hat{P} - P\|_F^2$, as functions of the number of nodes, for three different values of the parameter ω: $\omega = 0.35$ (red lines), 0.55 (blue lines), and 0.75 (black lines). One can see from Figure 4.3 that since mega-communities are detected first, the accuracy of detecting K communities (middle panels) depends on the precision of detecting L mega-communities (top panels). Furthermore, the estimation errors (bottom panels) in turn depend on the accuracy of detecting K communities (middle panels). Therefore, improved clustering precision leads to smaller estimation errors with finding the mega-communities being the key task.

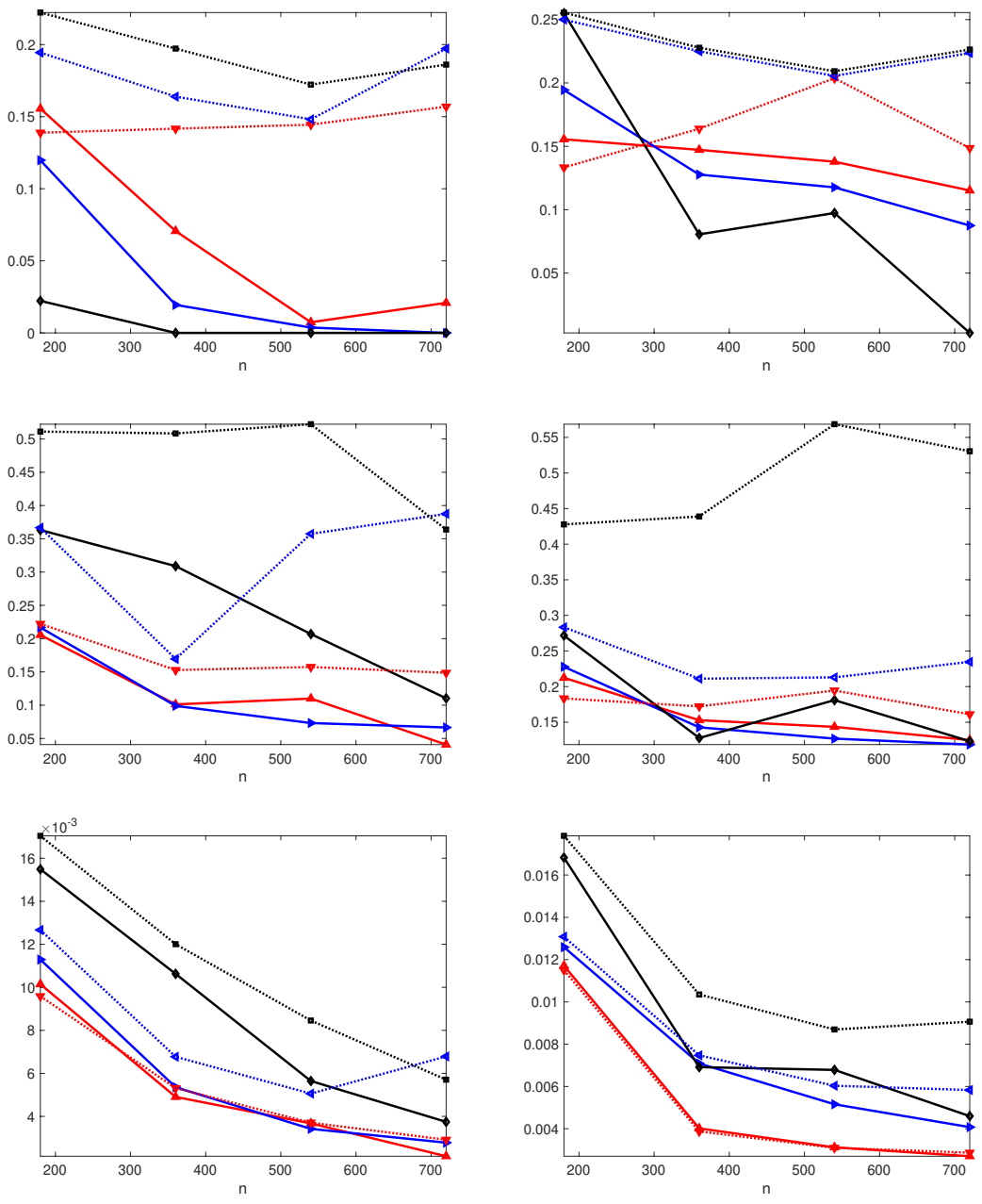

Figure 4.3: The clustering errors $\mathrm{Err}(\widehat{C}, C)$ (top panels) and $\mathrm{Err}(\widehat{Z}, Z)$ (middle panels) defined in (4.24) and the estimation errors $n^{-2}\,\|\hat{P} - P\|_F^2$ (bottom panels) for $K = 6$ communities and $L = 2$ (left) and $L = 3$ (right) mega-communities. The errors are evaluated over 50 simulation runs. The number of nodes ranges from $n = 180$ to $n = 720$ with the increments of 180. Dashed lines represent the results using Algorithm 2 for clustering and solid lines represent the results using the two-step clustering procedure; $\omega = 0.35$ (red), $\omega = 0.55$ (blue) and $\omega = 0.75$ (black).

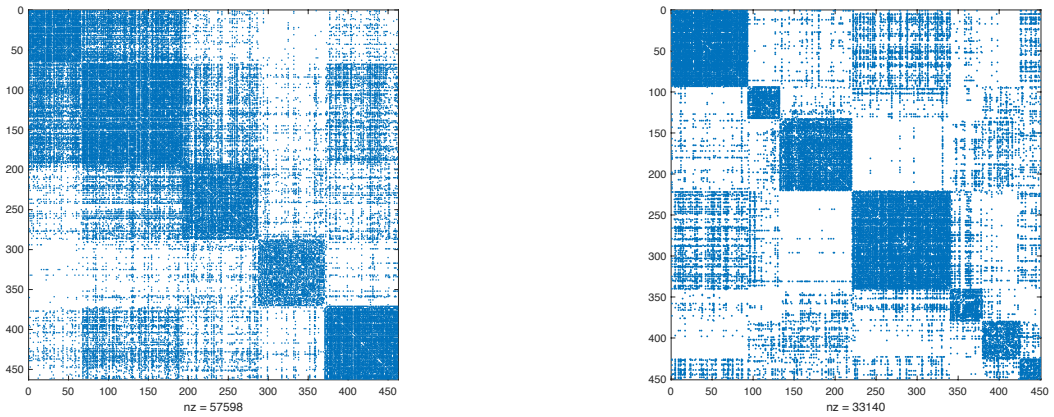

Figure 4.4: The adjacency matrices of the butterfly similarity network with 57598 nonzero entries and 5 clusters (left) and the brain network with 33140 nonzero entries and 7 clusters (right) after clustering

4.5.2 Real data examples

In this section, we describe application of the two-step clustering procedure of Section 4.4 to two real life networks, a butterfly similarity network and a human brain network.

We consider the butterfly similarity network extracted from the Leeds Butterfly dataset [57], which contains fine-grained images of 832 butterfly species that belong to 10 different classes, with each class containing between 55 and 100 images. In this network, the nodes represent butterfly species and edges represent visual similarities (ranging from 0 to 1) between them, evaluated on the basis of butterfly images. We extract the five largest classes and draw an edge between two nodes if the visual similarity between them is greater than zero, obtaining a simple graph with 462 nodes and 28799 edges. We carry out clustering of the nodes, employing the two-step clustering procedure, first finding $L = 4$ mega-communities by Algorithm 1, and then using Algorithm 2 to find communities within mega-communities.

We conclude that the first mega-community has two communities, while the other three mega-communities have one community each. We also applied Algorithms 1 and 2 separately for detection of five communities. Here, Algorithms 1 and 2 correspond, respectively, to the PABM and the DCBM settings with $K = 5$. Subsequently, we compare the clustering assignments with the true class specifications of the species. Algorithms 1 and 2 lead to 74% and 77% accuracy, respectively, while the two-step clustering procedure provides better 84% accuracy, thus, justifying the application of the HBM. The better results are due to the higher flexibility of the HBM.

The second example deals with analysis of a human brain functional network, based on the brain connectivity dataset, derived from the resting-state functional MRI (rsfMRI) [12]. In this dataset, the brain is partitioned into 638 distinct regions and a weighted graph is used to characterize the network topology. For a comparison, we use the Asymptotical Surprise method [45] which is applied for clustering the GroupAverage rsfMRI matrix in [12]. Asymptotical Surprise detects 47 communities with sizes ranging from 1 to 133. Since the true clustering as well as the true number of clusters are unknown for this dataset, we treat the results of the Asymptotical Surprise as the ground truth. In order to generate a binary network, we set all nonzero weights to one in the GroupAverage rsfMRI matrix, obtaining a network with 18625 undirected edges. For our study, we extract 7 largest communities derived by the Asymptotical Surprise, obtaining a network with 450 nodes and 16570 edges. Similarly to the previous example, we apply Algorithms 1 and 2 separately to detect seven communities, obtaining, respectively, 88% and 73% accuracy. We also use the the two-step clustering procedure above, detecting six mega-communities and seven communities, attaining 92% accuracy.

Figure 4.4 (right) shows the adjacency matrices of the butterfly similarity network (left) and the human brain network after clustering.

4.6 Discussion

The present chapter examines the hierarchy of block models with the purpose of treating all existing singular membership block models as a part of one formulation, which is free from arbitrary identifiability conditions. The blocks differ by the average probability of connections and can be combined into mega-blocks that have common heterogeneity patterns in the connection probabilities.

The hierarchical formulation proposed above (see Figure 4.2) can be utilized for a variety of purposes. Since the HBM treats all other block models as its particular cases, one can carry out estimation and clustering without assuming that a specific block model holds, by employing the HBM with K communities and L mega-communities, where both K and L are unknown. The values of K and L can later be derived on the basis of penalties. Furthermore, in the framework above, one can easily test one block model versus another. For instance, $L = K$ suggests the PABM while $L = 1$ implies the DCBM. If, additionally, $H = 1_n$, then DCBM reduces to SBM. Finally, one can see from Figure 4.2 that absence of distinct communities ($K = 1$) always leads to DCBM, which reduces to Erdős-Rényi model if $H = 1_n$.

CHAPTER 5: FUTURE WORK

The present dissertation deals with the analysis of a single stochastic network at a time. While it is a valid analysis, in many situations one needs to analyze multiple stochastic networks that are related to each other in some way. Examples of such networks include the brain networks of several individuals, the transportation networks with respect to various mode of transportation, or social networks with respect to different types of relationships. The models like this are called multilayer networks and recently attracted a lot of attention. The objectives in analysis of such networks usually involve assessment of the features that are common for all networks and are specific to some individual networks. In the context of the networks equipped with block models, one is interested in uncovering community structures that are common for all layers or groups of layers.

Specifically, we are planning to analyze brain network data of juvenile patients with the drug resistant epilepsy with the objective of uncovering speech related sub-networks. These sub-networks that may be affected by a surgical treatment which leads to subsequent speech deficiencies. The objective of the study is to increase the number of patients who can safely undergo a surgical treatment that is often the last resort for such patients.

There are also several other possible research areas related to the work in this dissertation. One obvious project is finding a practical way to estimate the true number of mega-communities and communities in HBM. Another natural area of exploration is studying weighted models which capture more information about networks. Some other avenues for future research on this topic include extending our methods and models to directed and dynamic settings which appear in many applications. One could also study the case that communities overlap.

APPENDIX : PROOFS

A.1 Proof of Theorem 4.3.1.

Let $\Xi = A - P_*$. We let $\mathscr{P}_{Z,C,K,L}$ denote the permutation matrix that arranges mega-blocks consecutively and also blocks all mega-blocks consecutively. For simplicity, let

$$\mathscr{P} \equiv \mathscr{P}_{Z,C,K,L},\ \mathscr{P}_* \equiv \mathscr{P}_{Z_*,C_*,K_*,L_*},\ \hat{\mathscr{P}} \equiv \mathscr{P}_{\hat{Z},\hat{C},\hat{K},\hat{L}}$$

For any matrix S, denote

$$S(Z,C,K,L) = \mathscr{P}^T_{Z,C,K,L} S \mathscr{P}_{Z,C,K,L} \tag{A.1}$$

Then, for any Z, C, K, and L:

$$\left\| \hat{\mathscr{P}}^T A \hat{\mathscr{P}} - \hat{\Theta}(\hat{Z},\hat{C},\hat{K},\hat{L}) \right\|_F^2 + \mathrm{Pen}(n,\hat{K},\hat{L}) \le \left\| \mathscr{P}^T A \mathscr{P} - \mathscr{P}^T P \mathscr{P} \right\|_F^2 + \mathrm{Pen}(n,K,L)$$

Therefore,

$$\left\| A - \hat{\mathscr{P}} \hat{\Theta}(\hat{Z},\hat{C},\hat{K},\hat{L}) \hat{\mathscr{P}}^T \right\|_F^2 + \mathrm{Pen}(n,\hat{K},\hat{L}) \le \|A - P\|_F^2 + \mathrm{Pen}(n,K,L)$$

or

$$\left\| A - \hat{P} \right\|_F^2 + \mathrm{Pen}(n,\hat{K},\hat{L}) \le \|A - P\|_F^2 + \mathrm{Pen}(n,K,L). \tag{A.2}$$

Subtracting and adding P_* in the norms in both sides of (A.2), we rewrite it as

$$\left\| \hat{P} - P_* \right\|_F^2 \le \|P - P_*\|_F^2 + 2\langle \Xi, \hat{P} - P \rangle + \mathrm{Pen}(n,K,L) - \mathrm{Pen}(n,\hat{K},\hat{L}). \tag{A.3}$$

Denote

$$P_0(K,L) = \inf_{P \in \Im(n,K,L)} \|P - P_*\|_F^2 ,$$

$$(K_0, L_0) = \inf_{K,L} \left\{ \|P_0(K,L) - P_*\|_F^2 + \mathrm{Pen}(n,K,L) \right\}.$$

Then, for $\hat{P} \equiv \hat{P}(\hat{K}, \hat{L})$ and $P_0 \equiv P_0(K_0, L_0)$, one has

$$\begin{aligned} \left\| \hat{P} - P_* \right\|_F^2 &\le \|P_0 - P_*\|_F^2 + 2\langle \Xi, P_* - P_0 \rangle \\ & 2\langle \Xi, \hat{P} - P_* \rangle + \mathrm{Pen}(n, K_0, L_0) - \mathrm{Pen}(n, \hat{K}, \hat{L}). \end{aligned} \tag{A.4}$$

Denote

$$\tau(n,K,L) = \quad n \ln K + K \ln L + (K^2 + 2nL) \ln (9nL) \tag{A.5}$$

and consider two sets Ω and Ω^c

$$\begin{aligned} \Omega &= \left\{ \omega : \left\| \hat{P} - P_* \right\|_F \ge C_0 2^{s_0} \sqrt{\tau(n, K_0, L_0)} \right\}, \\ \Omega^c &= \left\{ \omega : \left\| \hat{P} - P_* \right\|_F \le C_0 2^{s_0} \sqrt{\tau(n, K_0, L_0)} \right\} \end{aligned} \tag{A.6}$$

where s_0 is a constant. If $\omega \in \Omega^c$, then

$$\left\| \hat{P} - P_* \right\|_F^2 \le C_0^2 2^{2s_0} \tau(n, K_0, L_0) \tag{A.7}$$

Consider the case when $\omega \in \Omega$. It is known [31] that for any fixed matrix G, any $\alpha > 0$ and any $t > 0$ one has

$$\mathbb{P}\left\{ 2\langle \Xi, G \rangle \ge \alpha \|P_* - P_0\|_F^2 + 2t/\alpha \right\} \le e^{-t}. \tag{A.8}$$

Then, there exists a set $\tilde{\Omega}$ such that $P(\tilde{\Omega}_Z) \ge 1 - e^{-t}$ and for $w \in \tilde{\Omega}$

$$2\langle \Xi, P_* - P_0 \rangle \le \alpha \|P_* - P_0\|_F^2 + 2t/\alpha \tag{A.9}$$

Note that the set Ω can be partitioned as $\Omega = \bigcup_{K,L} \Omega_{K,L}$, where

$$\Omega_{K,L} = \left\{ \omega : \Big(\left\| \hat{P} - P_* \right\|_F \geq C_0 2^{s_0} \sqrt{\tau(n, K_0, L_0)} \Big) \cap (\hat{K} = K,\ \hat{L} = L) \right\} \tag{A.10}$$

with $\Omega_{K_1,L_1} \cap \Omega_{K_2,L_2} = \emptyset$ unless $K_1 = K_2$ and $L_1 = L_2$. Denote

$$\Delta(n, K, L) = C_0^2 C_2 \tau(n, K, L) + n, \tag{A.11}$$

where $\tau(n, K, L)$ is defined in (A.5). Then,

$$\begin{aligned} &\mathbb{P}\left\{ \left[2\langle \Xi, \hat{P}(n, \hat{K}, \hat{L}) - P_* \rangle - \frac{1}{2} \left\| \hat{P}(n, \hat{K}, \hat{L}) - P_* \right\|_F^2 - 2\Delta(n, \hat{K}, \hat{L}) \right] \geq 0 \right\} \\ &\leq \sum_{K=1}^{n} \sum_{L=1}^{K} \mathbb{P}\left\{ \sup_{\hat{P} \in \Omega_{K,L}} \left[2\langle \Xi, \hat{P} - P_* \rangle - \frac{1}{2} \left\| \hat{P} - P_* \right\|_F^2 - 2\Delta(n, K, L) \right] \geq 0 \right\} \end{aligned}$$

By Lemma A.3.3 in Section A.3, there exist sets $\tilde{\Omega}_{K,L} \subseteq \Omega_{K,L} \subset \Omega$ such that $\mathbb{P}(\tilde{\Omega}_{K,L}^c) \leq \log_2 n \cdot \exp\left(-n \cdot 2^{2s_0 - 7}\right)$ and, for $\omega \in \tilde{\Omega}_{K,L}$, one has

$$\left\{ 2\langle \Xi, \hat{P} - P_* \rangle \leq \frac{1}{2} \left\| \hat{P} - P_* \right\|_F^2 + 2\Delta(n, K, L) \right\} \cap \left\{ \hat{K} = K,\ \hat{L} = L \right\}$$

Denote

$$\tilde{\Omega} = \left(\bigcap_{K,L} \tilde{\Omega}_{K,L} \right) \cap \tilde{\Omega}_t \tag{A.12}$$

and observe that

$$\mathbb{P}\left(\tilde{\Omega}\right) \geq 1 - n^2 \log_2 n \cdot \exp\left(-n \cdot 2^{2s_0 - 7}\right) - e^{-t}.$$

Then, for $\omega \in \tilde{\Omega}$, one has

$$2\langle \Xi, \hat{P} - P_* \rangle \leq \frac{1}{2} \left\| \hat{P} - P_* \right\|_F^2 + 2\Delta(n, \hat{K}, \hat{L}) \tag{A.13}$$

and it follows from (A.9) with $\alpha = 1/2$ that

$$2\langle \Xi, P_* - P_0 \rangle \leq \frac{1}{2} \|P_* - P_0\|_F^2 + 4t \tag{A.14}$$

Plugging (A.13) and (A.14) into (A.4), obtain that for $\omega \in \tilde{\Omega}$ one has

$$\left\| \hat{P} - P_* \right\|_F^2 \leq \|P_0 - P_*\|_F^2 + \mathrm{Pen}(n, K_0, L_0) + \frac{1}{2} \left\| \hat{P} - P_* \right\|_F^2 + 2\Delta(n, \hat{K}, \hat{L}) + \frac{1}{2} \|P_* - P_0\|_F^2 + 4t - \mathrm{Pen}(n, \hat{K}, \hat{L})$$

Finally, setting

$$\mathrm{Pen}(n, K, L) = 2\Delta(n, K, L) = 2\left[C_0^2 \tau(n, K, L) + n\right],$$

obtain that for any $t > 0$, for $\omega \in \tilde{\Omega}$, one has

$$\left\| \hat{P} - P_* \right\|_F^2 \leq 3 \|P_0 - P_*\|_F^2 + 2\mathrm{Pen}(n, K_0, L_0) + 8t,$$

for any $\omega \in \Omega$. Now, for $\omega \in \Omega^c$, it follows from (A.7) that

$$\left\| \hat{P} - P_* \right\|_F^2 \leq C_0^2 2^{2s_0} \tau(n, K_0, L_0) \leq 2^{2s_0 - 1} \mathrm{Pen}(n, K_0, L_0)$$

Setting $s_0 = 1$ and $t = n/32$, obtain

$$\mathbb{P}\left\{ \left\| \hat{P} - P_* \right\|_F^2 \leq \left[3 \|P_0 - P_*\|_F^2 + 2\mathrm{Pen}(n, K_0, L_0) \right] + \frac{n}{4} \right\} \geq 1 - (n^2 \log_2 n + 1) e^{-\frac{n}{32}},$$

so that

$$\mathbb{P}\left\{ \left\| \hat{P} - P_* \right\|_F^2 \leq 3 \inf_{P \in \Im(n,K,L)} \left[\|P - P_*\|_F^2 + \mathrm{Pen}(n, K, L) \right] \right\} \geq 1 - (n^2 \log_2 n + 1) e^{-\frac{n}{32}}$$

Since $\left\|\hat{P} - P_*\right\|_F^2 \le n^2$, obtain

$$\mathbb{E}\left\|\hat{P} - P_*\right\|_F^2 \le 3 \min_{P \in \mathscr{M}(n,K,L)} \Big[\|P - P_*\|_F^2 + \mathrm{Pen}(n,K,L) \Big] + n^5 e^{-n/32}$$

A.2 Proof of Theorem 4.3.2.

Let

$$F_1(n,K,L) = C_1 nK + C_2 K^2 \ln(ne) + C_3(\ln n + (n+1)\ln K + K \ln L)$$

$$F_2(n,K,L) = 2\ln n + 2(n+1)\ln K + 2K \ln L,$$

where C_1, C_2, and C_3 are absolute constants. Denote $\Xi = A - P_*$ and recall that, given matrix P_*, entries $\Xi_{i,j} = A_{i,j} - (P_*)_{ij}$ of Ξ are the independent Bernoulli errors for $1 \le i \le j \le n$ and $A_{i,j} = A_{j,i}$. Then, following notation (A.1), for any Z, C, K, and L

$$\Xi(Z,C,K,L) = \mathscr{P}^T \Xi \mathscr{P}$$

$$P_*(Z,C,K,L) = \mathscr{P}^T P_* \mathscr{P},$$

where $\mathscr{P} \equiv \mathscr{P}_{Z,C,K,L}$. Then it follows from (4.18) that

$$\left\|\hat{\mathscr{P}}^T A \hat{\mathscr{P}} - \hat{\Theta}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 + \mathrm{Pen}(n,\hat{K},\hat{L}) \le \left\|\mathscr{P}_*^T A \mathscr{P}_* - \mathscr{P}_*^T P_* \mathscr{P}_*\right\|_F^2 + \mathrm{Pen}(n,K_*,L_*)$$

where $\mathscr{P}_* \equiv \mathscr{P}_{Z_*,C_*,K_*,L_*}$. Using the fact that permutation matrices are orthogonal, we can rewrite the previous inequality as

$$\left\|A - \hat{\mathscr{P}}\hat{\Theta}(\hat{Z},\hat{C},\hat{K},\hat{L})\hat{\mathscr{P}}^T\right\|_F^2 + \mathrm{Pen}(n,\hat{K},\hat{L}) \le \|A - P_*\|_F^2 + \mathrm{Pen}(n,K_*,L_*). \qquad \text{(A.15)}$$

Hence, (A.15) and (4.14) yield

$$\left\|A - \hat{P}\right\|_F^2 \leq \|A - P_*\|_F^2 + \text{Pen}(n, K_*, L_*) - \text{Pen}(n, \hat{K}, \hat{L}) \tag{A.16}$$

Subtracting and adding P_* in the norm of the left-hand side of (A.16), we rewrite (A.16) as

$$\left\|\hat{P} - P_*\right\|_F^2 \leq \Delta(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) + \text{Pen}(n, K_*, L_*) - \text{Pen}(n, \hat{K}, \hat{L}), \tag{A.17}$$

where

$$\Delta \equiv \Delta(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) = 2\text{Tr}\left[\Xi^T(\hat{P} - P_*)\right]. \tag{A.18}$$

Again, using orthogonality of the permutation matrices, we can rewrite

$$\Delta = 2\langle \Xi(\hat{Z}, \hat{C}, \hat{K}, \hat{L}), (\hat{\Theta}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) - P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}))\rangle,$$

where $\langle A, B\rangle = \text{Tr}(A^T B)$. Then, in the block form, Δ appears as

$$\Delta = \sum_{l=1}^{\hat{L}} \sum_{k=1}^{\hat{K}} \Delta^{(l,k)} \tag{A.19}$$

where

$$\Delta^{(l,k)} = 2\langle \Xi^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}), \Pi_{\hat{u},\hat{v}}(A^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L})) - P_*^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L})\rangle$$

and $\Pi_{\hat{u},\hat{v}}$ is defined in (A.52) of Lemma A.3.4.

Let $\tilde{u} = \tilde{u}^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L})$ and $\tilde{v} = \tilde{v}^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L})$ be the singular vectors of $P_*^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L})$

corresponding to its largest singular value. Then, according to Lemma A.3.4

$$\Pi_{\tilde{u},\tilde{v}}\left(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right) = \quad \tilde{u}^{(l,k)}(\tilde{u}^{(l,k)})^T P_*^{(l,k)} \tilde{v}^{(l,k)}(\tilde{v}^{(l,k)})^T \tag{A.20}$$

Recall that

$$\Pi_{\hat{u},\hat{v}}(A^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) = \quad \Pi_{\hat{u},\hat{v}}\left[P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}) + \Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right],$$

Then, $\Delta^{(l,k)}$ can be partitioned into the sums of three components

$$\Delta^{(l,k)} = \Delta_1^{(l,k)} + \Delta_2^{(l,k)} + \Delta_3^{(l,k)}, \qquad l = 1,2,\cdots,\hat{L}, \;\; k = 1,2,\cdots,\hat{K} \tag{A.21}$$

where

$$\Delta_1^{(l,k)} = 2\langle \Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}), \Pi_{\hat{u},\hat{v}}(\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}))\rangle \tag{A.22}$$

$$\Delta_2^{(l,k)} = 2\langle \Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}), \Pi_{\tilde{u},\tilde{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\rangle \tag{A.23}$$

$$\Delta_3^{(l,k)} = 2\langle \Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}), \Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - \Pi_{\tilde{u},\tilde{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}))\rangle \tag{A.24}$$

With some abuse of notations, for any matrix B, let $\Pi_{\tilde{u},\tilde{v}}\left(B(\hat{Z},\hat{C},\hat{K},\hat{L})\right)$ be the matrix

with blocks $\Pi_{\tilde{u},\tilde{v}}\left(B^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right)$ and $\Pi_{\hat{u},\hat{v}}\left(B(\hat{Z},\hat{C},\hat{K},\hat{L})\right)$ be the matrix with blocks

$$\Pi_{\hat{u},\hat{v}}\left(B^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right),\ l=1,2,\cdots,\hat{L},\ k=1,2,\cdots,\hat{K}$$

. Then, it follows from (A.21)–(A.24) that

$$\Delta=\Delta_1+\Delta_2+\Delta_3 \tag{A.25}$$

where

$$\Delta_1=2\langle(\Xi(\hat{Z},\hat{C},\hat{K},\hat{L}),\Pi_{\hat{u},\hat{v}}(\Xi(\hat{Z},\hat{C},\hat{K},\hat{L}))\rangle \tag{A.26}$$

$$\Delta_2=2\langle\Xi(\hat{Z},\hat{K}),\Pi_{\tilde{u},\tilde{v}}(P_*(\hat{Z},\hat{C},\hat{K},\hat{L}))-P_*(\hat{Z},\hat{C},\hat{K},\hat{L})\rangle \tag{A.27}$$

$$\Delta_3=2\langle\Xi(\hat{Z},\hat{C},\hat{K},\hat{L}),\Pi_{\hat{u},\hat{v}}(P_*(\hat{Z},\hat{C},\hat{K},\hat{L}))-\Pi_{\tilde{u},\tilde{v}}(P_*(\hat{Z},\hat{C},\hat{K},\hat{L}))\rangle \tag{A.28}$$

Observe that

$$\begin{aligned}\Delta_1^{(l,k)}&=2\langle\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}),\Pi_{\hat{u},\hat{v}}(\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}))\rangle\\&=2\left\|\Pi_{\hat{u},\hat{v}}(\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}))\right\|_F^2\\&\le 2\left\|\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_{op}^2.\end{aligned}$$

Now, fix t and let Ω_1 be the set where $\sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}}\left\|\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_{op}^2\le F_1(n,\hat{K},\hat{L})+C_3t$.

According to Lemma A.3.7,

$$\mathbb{P}(\Omega_1) \geq 1 - \exp(-t), \tag{A.29}$$

and, for $\omega \in \Omega_1$, one has

$$|\Delta_1| \leq 2 \sum_{l=1}^{\hat{L}} \sum_{k=1}^{\hat{K}} \left\| \Xi^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right\|_{op}^2 \leq 2F_1(n, \hat{K}, \hat{L}) + 2C_3 t \tag{A.30}$$

Now, consider Δ_2 given by (A.27). Note that

$$|\Delta_2| = 2 \left\| \Pi_{\tilde{u},\tilde{v}} \left(P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right) - P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right\|_F |\langle \Xi(\hat{Z}, \hat{C}, \hat{K}, \hat{L}), H_{\tilde{u},\tilde{v}}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \rangle| \tag{A.31}$$

where

$$H_{\tilde{u},\tilde{v}}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) = \frac{\Pi_{\tilde{u},\tilde{v}} \left(P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right) - P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L})}{\| \Pi_{\tilde{u},\tilde{v}} \left(P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right) - P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \|_F}$$

Since for any a, b, and $\alpha_1 > 0$, one has $2ab \leq \alpha_1 a^2 + b^2/\alpha_1$, obtain

$$\begin{aligned} |\Delta_2| \leq \alpha_1 & \left\| \Pi_{\tilde{u},\tilde{v}} \left(P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right) - P_*(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \right\|_F^2 + \\ & 1/\alpha_1 \, |\langle \, \Xi(\hat{Z}, \hat{C}, \hat{K}, \hat{L}), H_{\tilde{u},\tilde{v}}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \, \rangle|^2 \end{aligned} \tag{A.32}$$

Observe that if K, L, $Z \in \mathcal{M}_{n,K}$, and $C \in \mathcal{M}_{K,L}$ are fixed, then $H_{\tilde{u},\tilde{v}}(Z, C, K, L)$ is fixed and, for any K, L, Z, and C, one has $\|H_{\tilde{u},\tilde{v}}(Z, C, K, L)\|_F = 1$. Note also that, for fixed K, L, Z, and C, permuted matrix $\Xi(Z, C, K, L) \in [0, 1]^{n \times n}$ contains independent Bernoulli errors. It is well known that if ξ is a vector of independent Bernoulli errors and h is a unit vector, then, for any $x > 0$, Hoeffding's inequality yields

$$\mathbb{P}(|\xi^T h|^2 > x) \leq 2 \exp(-x/2)$$

Since

$$\langle \Xi(Z,C,K,L), H_{\tilde{u},\tilde{v}}(Z,C,K,L)\rangle = \ [\mathrm{vec}(\Xi(Z,C,K,L))]^T \mathrm{vec}(H_{\tilde{u},\tilde{v}}(Z,C,K,L)),$$

obtain for any fixed K, L, Z, and C:

$$\mathbb{P}\left(|\langle \Xi(Z,C,K,L), H_{\tilde{u},\tilde{v}}(Z,C,K,L)\rangle|^2 - x > 0\right) \le 2\exp(-x/2)$$

Now, applying the union bound, derive

$$\begin{aligned}
&\mathbb{P}\left(|\langle \Xi(\hat{Z},\hat{C},\hat{K},\hat{L}), H_{\tilde{u},\tilde{v}}(\hat{Z},\hat{C},\hat{K},\hat{L})\rangle|^2 - F_2(n,\hat{K},\hat{L}) > 2t\right) \\
&\le \mathbb{P}\Big[\max_{1\le K\le n}\max_{1\le L\le K}\max_{Z\in\mathcal{M}_{n,K}}\max_{C\in\mathcal{M}_{K,L}}(|\langle \Xi(Z,C,K,L), H_{\tilde{u},\tilde{v}}(Z,C,K,L)\rangle|^2 - F_2(n,K,L)) > 2t\Big] \qquad \text{(A.33)} \\
&\le 2nKK^nL^K \exp\{-F_2(n,K,L)/2 - t\} = 2\exp(-t),
\end{aligned}$$

where $F_2(n,K,L) = 2\ln n + 2(n+1)\ln K + 2K\ln L$. By Lemma A.3.5, one has

$$\begin{aligned}
&\left\|\Pi_{\tilde{u},\tilde{v}}\left(P_*(\hat{Z},\hat{C},\hat{K},\hat{L})\right) - P_*(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 \le \\
&\left\|\Pi_{\hat{u},\hat{v}}\left(P_*(\hat{Z},\hat{C},\hat{K},\hat{L})\right) - P_*(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 \le \left\|\hat{P} - P_*\right\|_F^2.
\end{aligned}$$

Denote the set on which (A.33) holds by Ω_2^C, so that

$$\mathbb{P}(\Omega_2) \ge 1 - 2\exp(-t). \qquad \text{(A.34)}$$

Then inequalities (A.32) and (A.33) imply that, for any $\alpha_1 > 0$, $t > 0$ and any $\omega \in \Omega_2$, one has

$$|\Delta_2| \le \alpha_1 \left\|\hat{P} - P_*\right\|_F^2 + 1/\alpha_1\, F_2(n,\hat{K},\hat{L}) + 2\,t/\alpha_1. \qquad \text{(A.35)}$$

Now consider Δ_3 defined in (A.28) with components (A.24). Note that matrices

$$\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - \Pi_{\tilde{u},\tilde{v}}\left(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right)$$

have rank at most two. Use the fact that (see, e.g., Giraud (2014), page 123)

$$\langle A, B\rangle \leq \|A\|_{(2,r)} \|B\|_{(2,r)} \leq 2\|A\|_{op} \|B\|_F, \quad r = \min\{\mathrm{rank}(A), \mathrm{rank}(B)\}. \tag{A.36}$$

Here $\|A\|_{(2,q)}$ is the Ky-Fan $(2,q)$ norm

$$\|A\|_{(2,q)}^2 = \sum_{j=1}^{q} \sigma_j^2(A) \leq \|A\|_F^2,$$

where $\sigma_j(A)$ are the singular values of A. Applying inequality (A.36) with $r = 2$ and taking into account that for any matrix A one has $\|A\|_{(2,2)}^2 \leq 2\|A\|_{op}^2$, derive

$$|\Delta_3^{(l,k)}| \leq 4\left\|\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_{op} \left\|\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}) - \Pi_{\tilde{u},\tilde{v}}\left(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right)\right\|_F.$$

Then, for any $\alpha_2 > 0$, obtain

$$\begin{aligned} |\Delta_3| \leq \sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}} |\Delta_3^{(l,k)}| \leq \frac{2}{\alpha_2}\sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}} \left\|\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_{op}^2 + \\ 2\alpha_2 \sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}} \left\|\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}) - \Pi_{\tilde{u},\tilde{v}}\left(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right)\right\|_F^2. \end{aligned} \tag{A.37}$$

Note that, by Lemma A.3.5,

$$\begin{aligned}
&\left\|\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - \Pi_{\tilde{u}\tilde{v}}\left(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right)\right\|_F^2 \\
\leq & 2\left\|\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 + \\
& 2\left\|\Pi_{\tilde{u},\tilde{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 \\
\leq & 4\left\|\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 \\
\leq & 4\left\|\Pi_{\hat{u},\hat{v}}(A^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 \\
= & 4\left\|\hat{\Theta}^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L}) - P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}}\left\|\Pi_{\hat{u},\hat{v}}(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})) - \Pi_{\tilde{u},\tilde{v}}\left(P_*^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right)\right\|_F^2 \leq \\
&4\left\|\hat{\Theta}(\hat{Z},\hat{C},\hat{K},\hat{L}) - P_*(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_F^2 = 4\left\|\hat{P} - P_*\right\|_F^2
\end{aligned} \tag{A.38}$$

Combine inequalities (A.37) and (A.38) and recall that

$$\sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}}\left\|\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_{op}^2 \leq F_1(n,\hat{K},\hat{L}) + C_3\,t$$

for $\omega \in \Omega_1$. Then, for any $\alpha_2 > 0$ and $\omega \in \Omega_1$, one has

$$|\Delta_3| \leq 8\alpha_2\left\|\hat{P} - P_*\right\|_F^2 + 2/\alpha_2 F_1(n,\hat{K},\hat{L}) + 2C_3\,t/\alpha_2. \tag{A.39}$$

Now, let $\Omega = \Omega_1 \cap \Omega_2$. Then, (A.29) and (A.34) imply that $\mathbb{P}(\Omega) \geq 1 - 3\exp(-t)$ and, for $\omega \in \Omega$, inequalities (A.30), (A.35) and (A.39) simultaneously hold. Hence, by (A.25), derive

that, for any $\omega \in \Omega$,

$$|\Delta| \leq (2 + 2/\alpha_2) F_1(n, \hat{K}, \hat{L})) + 1/\alpha_1 \, F_2(n, \hat{K}, \hat{L}) + \\ (\alpha_1 + 8\alpha_2) \left\| \hat{P} - P_* \right\|_F^2 + 2(C_3 + 1/\alpha_1 + C_3/\alpha_2) \, t.$$

Combination of the last inequality and (A.17) yields that, for $\alpha_1 + 8\alpha_2 < 1$ and any $\omega \in \Omega$,

$$(1 - \alpha_1 - 8\alpha_2) \left\| \hat{P} - P_* \right\|_F^2 \leq \left(2 + \frac{2}{\alpha_2} \right) F_1(n, \hat{K}, \hat{L}) + \\ \frac{1}{\alpha_1} F_2(n, \hat{K}, \hat{L}) + \mathrm{Pen}(n, K_*, L_*) - \mathrm{Pen}(n, \hat{K}, \hat{L}) \\ + 2(C_3 + 1/\alpha_1 + C_3/\alpha_2) \, t$$

Setting $\mathrm{Pen}(n, K, L) = (2+2/\alpha_2)F_1(n, K, L) + 1/\alpha_1 F_2(n, K, L)$ and dividing by $(1-\alpha_1-8\alpha_2)$, obtain that

$$\mathbb{P}\left\{ \left\| \hat{P} - P_* \right\|_F^2 \leq (1 - \alpha_1 - 8\alpha_2)^{-1} \, \mathrm{Pen}(n, K_*, L_*) + \tilde{C} \, t \right\} \geq 1 - 3e^{-t} \tag{A.40}$$

where

$$\tilde{C} = 2 \, (1 - \alpha_1 - 8\alpha_2)^{-1} \, (C_3 + 1/\alpha_1 + C_3/\alpha_2) \tag{A.41}$$

Moreover, note that for $\xi = \|\hat{P} - P_*\|_F^2 - (1-\beta_1-\beta_2)^{-1} \, \mathrm{Pen}(n, K_*, L_*)$, one has $\mathbb{E}\|\hat{P} - P_*\|_F^2 = (1 - \beta_1 - \beta_2)^{-1} \, \mathrm{Pen}(n, K_*, L_*) + \mathbb{E}\xi$,where

$$\mathbb{E}\xi \leq \int_0^\infty \mathbb{P}(\xi > z) dz = \tilde{C} \int_0^\infty \mathbb{P}(\xi > \tilde{C}t) dt \leq \tilde{C} \int_0^\infty 3 \, e^{-t} \, dt = 3\tilde{C},$$

By rearranging and combining the terms, the penalty $\mathrm{Pen}(n, K, L)$ can be written in the form (4.19) completing the proof.

A.3 Supplementary statements and their proofs

Lemma A.3.1. *The logarithm of the cardinality of a δ-net on the space of non-symmetric DCBMs of size $n_1 \times n_2$ with $K_1 \times K_2$ blocks is*

$$(K_1K_2 + n_1 + n_2)\ln\left(\frac{9}{\delta}\right) + \left(K_1K_2 + \frac{n_1+n_2}{2}\right)\ln(n_1n_2)$$

Proof. Let Z_1 and Z_2 be fixed. Then by rearranging Θ, rewrite it as $\Theta = Q_1BQ_2^T$, where B and Q_i, $i = 1, 2$, have the sizes $K_1 \times K_2$ and $n_i \times K_i$, respectively. Here, Q_i is of the form

$$Q_i = \begin{bmatrix} q_{i,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & q_{i,2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & q_{i,K_i} \end{bmatrix}, \tag{A.42}$$

We re-scale components of matrices Q_1, Q_2 and B, so that vectors $q_{i,j} \in \mathbb{R}_+^{n_{i,j}}$, $j = 1, \cdots, K_i$, $i = 1, 2$, have unit norms $\|q_{i,j}\|_2 = 1$, and $\sum_{j=1}^{K_i} n_{i,j} = n_i$. Let $\Theta^{(k_1,k_2)} \in \mathbb{R}^{n_{1,k_1} \times n_{2,k_2}}$ be the (k_1, k_2)-th block of Θ. Then,

$$\Theta^{(k_1,k_2)} = B_{k_1,k_2} q_{1,k_1} q_{2,k_2}^T$$

and

$$\left\|\Theta^{(k_1,k_2)}\right\|_F^2 = B_{k_1,k_2}^2 \left\|q_{1,k_1}\right\|_2^2 \left\|q_{2,k_2}\right\|_2^2 = B_{k_1,k_2}^2 \leq n_{k_1} \cdot n_{k_2},$$

due to $\left\|ab^T\right\|_F^2 \leq \|a\|_2^2 \|b\|_2^2$ (for any vectors a and b) and $\|\Theta\|_\infty \leq 1$. Hence, $B_{k_1,k_2} \leq \sqrt{n_{k_1} \cdot n_{k_2}} \leq \sqrt{n_1 \cdot n_2}$.

Let $\mathscr{D}_1(\delta_1)$, $\mathscr{D}_2(\delta_2)$, and $\mathscr{D}_B(\delta_B)$ be the δ_1, δ_2, and δ_B nets for Q_1, Q_2, and B, respectively.

The nets $\mathscr{D}_i(\delta_i)$ are essentially constructed for K_i vectors of length 1 in $\mathbb{R}^{n_{i,j}}$, hence, by [47]

$$\mathrm{card}(\mathscr{D}_i(\delta_i)) \leq \Pi_{j=1}^{K_i} (3/\delta_i)^{n_{i,j}} = (3/\delta_i)^{n_i}\,,\ i = 1, 2.$$

Let $b = \mathrm{vec}(B)$. Then, $b \in \mathbb{R}^{K_1 K_2}$ and $\|b\| \leq \sqrt{n_1 n_2}$ since

$$\|b\|^2 = \|B\|_F^2 = \sum_{k_1,k_2} B_{k_1,k_2}^2 = \sum_{k_1,k_2} n_{k_1} n_{k_2} = n_1 n_2.$$

Therefore,

$$\mathrm{card}(\mathscr{D}_B(\delta_B)) \leq \left(\frac{3 n_1 n_2}{\delta_B}\right)^{K_1 K_2}$$

Now, let us check what values of δ_1, δ_2, and δ_B result in a δ-net. Let $\Theta = Q_1 B Q_2^T$ and $\tilde{\Theta} = \tilde{Q}_1 \tilde{B} \tilde{Q}_2^T$. Then

$$\begin{aligned}
&\left\|\tilde{\Theta} - \Theta\right\|_F = \left\|\tilde{Q}_1 \tilde{B} \tilde{Q}_2^T - Q_1 B Q_2^T\right\|_F \leq \\
&\left\|(\tilde{Q}_1 - Q_1)\tilde{B}\tilde{Q}_2^T\right\|_F + \left\|Q_1(\tilde{B} - B)\tilde{Q}_2^T\right\|_F + \left\|Q_1 B (\tilde{Q}_2 - Q_2)^T\right\|_F
\end{aligned}$$

Note that

$$\|A_1 A_2\|_F \leq \min\left(\|A_1\|_F \|A_2\|_{op}, \|A_1\|_{op} \|A_2\|_F\right)$$

for any matrices A_1 and A_2, and that also

$$Q_i^T Q_i = \mathrm{diag}\left(\|q_{i,1}\|^2, \cdots, \|q_{i,K_i}\|^2\right) = I_{K_i},\ i = 1, 2.$$

Hence

$$\|Q_i\|_{op} = 1;\ \|Q_i\|_F = \sqrt{K_i},\ i = 1, 2.$$

Similarly, if $\tilde{Q}_i, Q_i \in \mathscr{D}_i(\delta_i)$, then

$$(\tilde{Q}_i - Q_i)^T(\tilde{Q}_i - Q_i) = \quad \mathrm{diag}\big(\|\tilde{q}_{i,1} - q_{i,1}\|^2, \cdots, \|\tilde{q}_{i,K_i} - q_{i,K_i}\|^2\big)$$

Thus

$$\left\|\tilde{Q}_i - Q_i\right\|_{op} = \delta_i; \;\; \left\|\tilde{Q}_i - Q_i\right\|_F \leq \sqrt{K_i}\delta_i, \; i = 1, 2.$$

Also, for $i = 1, 2$

$$\mathrm{Tr}(B^T Q_i^T Q_i B) = \|Q_i B\|_F^2 = \|B\|_F^2 = n_1 n_2.$$

Hence,

$$\begin{aligned}
\left\|\tilde{\Theta} - \Theta\right\|_F &\leq \left\|\tilde{Q}_1 - Q_1\right\|_{op} \left\|\tilde{B}\tilde{Q}_2^T\right\|_F \\
&+ \|Q_1 B\|_F \left\|\tilde{Q}_2 - Q_2\right\|_{op} + \|Q_1\|_{op} \left\|\tilde{B} - B\right\|_F \left\|\tilde{Q}_2\right\|_{op} \\
&= (\delta_1 + \delta_2)\sqrt{n_1 n_2} + \delta_B \leq \delta
\end{aligned}$$

Set $\delta_B = \frac{\delta}{3}$ and $\delta_1 = \delta_2 = \frac{\delta}{3\sqrt{n_1 n_2}}$. Then

$$\begin{aligned}
\mathrm{card}(\mathscr{D}_B(\delta_B)) &= \left(\frac{9 n_1 n_2}{\delta}\right)^{K_1 K_2}, \\
\mathrm{card}(\mathscr{D}_i(\delta_i)) &= \left(\frac{9\sqrt{n_1 n_2}}{\delta}\right)^{n_i},
\end{aligned}$$

which completes the proof.

Lemma A.3.2. *Consider the set of matrices P which can be transformed by a permutation matrix $\mathscr{P}_{Z,C}$ into a block matrix $\Theta \in \Im(n, K, L)$ where $\Im(n, K, L)$ is defined in (4.13). Let $\mathscr{Y}(\epsilon, n, K, L)$ be an ϵ-net on the set $\Im(n, K, L)$ and $|\mathscr{Y}(\epsilon, n, K, L)|$ be its cardinality. Then,*

for any K and L, $1 \le K \le n$, $1 \le L \le K$, one has

$$|\mathscr{Y}(\epsilon, n, K, L)| \le n \ln K + K \ln L + (K^2 + 2nL) \ln \left(\frac{9nL}{\epsilon} \right) \tag{A.43}$$

Proof. First construct nets on the set of matrices Z and C with the respective the cardinalities K^n and L^K. After that, validity of the lemma follows from Lemma A.3.1.

Lemma A.3.3. *Let $C_0^2 = 3009$, $C_2 = 1$, $s_0 > 0$ be an arbitrary constant and $\Omega_{K,L}$ be defined in* (A.10)*. Then,*

$$\mathbb{P}\Big\{ \sup_{\hat{P} \in \Omega_{K,L}} \Big[2\langle \Xi, \hat{P} - P_* \rangle - \frac{1}{2} \Big\| \hat{P} - P_* \Big\|_F^2 - 2\Delta(n, K, L) \Big] \ge 0 \Big\} \le \log_2 n \cdot \exp\big(- n \cdot 2^{2s_0 - 7} \big)$$

where $\Delta(n, K, L)$ is defined in (A.11)*.*

Proof. Consider sets

$$\begin{aligned} \chi_s(K, L) = \Big\{ & \exists Z, C : \ P(Z, C) \in \Im(n, K, L); \\ & C_0 2^s \sqrt{\tau(n, K_0, L_0)} \le \| P - P_* \|_F \le C_0 2^{s+1} \sqrt{\tau(n, K_0, L_0)} \Big\}, \end{aligned}$$

and

$$\mathcal{J}_s(K, L) = \Big\{ \exists Z, C : \ P(Z, C) \in \Im(n, K, L); \quad \| P - P_* \|_F \le C_0 2^s \sqrt{\tau(n, K_0, L_0)} \Big\}$$

Note that the set Ω can be partitioned as

$$\Omega = \bigcup_{K,L} \Omega_{K,L}$$

where $\Omega_{K,L}$ are defined in (A.10). Then

$$\mathbb{P}\Big\{\sup_{\hat{P}\in\Omega_{K,L}}\Big[\langle\Xi,\hat{P}-P_*\rangle-\frac{1}{4}\Big\|\hat{P}-P_*\Big\|_F^2-\Delta(n,K,L)\Big]\geq 0\Big\}\leq$$
$$\sum_{s=s_0}^{s_{\max}}\mathbb{P}\Big\{\sup_{\hat{P}\in\chi_s(K,L)}\Big[\langle\Xi,\hat{P}-P_*\rangle-\frac{1}{4}\Big\|\hat{P}-P_*\Big\|_F^2-\Delta(n,K,L)\Big]\geq 0\Big\}\leq$$
$$\sum_{s=s_0}^{s_{\max}}\mathbb{P}\Big\{\sup_{\hat{P}\in\chi_s(K,L)}\langle\Xi,\hat{P}-P_*\rangle\geq C_0^2 2^{2s-2}\tau(n,K_0,L_0)+\Delta(n,K,L)\Big\}\leq$$
$$\sum_{s=s_0}^{s_{\max}}\mathbb{P}\Big\{\sup_{\hat{P}\in\mathcal{J}_{s+1}(K,L)}\langle\Xi,\hat{P}-P_*\rangle\geq C_0^2 2^{2s-2}\tau(n,K_0,L_0)+\Delta(n,K,L)\Big\}$$

Here, $s_{max}\leq\log_2 n$ since $\Big\|\hat{P}-P_*\Big\|_F\leq n$.

Construct a 1-net $\mathscr{Y}_s(n,K,L)$ on the set of matrices in $\mathcal{J}_{s+1}(K,L)$ and observe that, for any $\hat{P}\in\mathcal{J}_s(K,L)$, there exists $\tilde{P}\in\mathscr{Y}_s(n,K,L)$ such that $\|\hat{P}-\tilde{P}\|_F\leq 1$. Then,

$$\sup_{\hat{P}\in\mathscr{Y}_{s+1}(n,K,L)}\langle\Xi,\hat{P}-P_*\rangle\leq$$
$$\max_{\tilde{P}\in\mathscr{Y}_s(n,K,L)}\big[\langle\Xi,\tilde{P}-P_*\rangle+\langle\Xi,\hat{P}-\tilde{P}\rangle\big]\leq$$
$$\max_{\tilde{P}\in\mathscr{Y}_s(n,K,L)}\langle\Xi,\tilde{P}-P_*\rangle+n$$

Hence,

$$\mathbb{P}\Big\{\sup_{\hat{P}\in\Omega_{K,L}}\Big[\langle\Xi,\hat{P}-P_*\rangle-\frac{1}{4}\Big\|\hat{P}-P_*\Big\|_F^2-\Delta(n,K,L)\Big]\geq 0\Big\}\leq$$
$$\sum_{s=s_0}^{s_{\max}}\mathbb{P}\Big\{\max_{\tilde{P}\in\mathscr{Y}_s(n,K,L)}\langle\Xi,\tilde{P}-P_*\rangle\geq C_0^2 2^{2s-2}\tau(n,K_0,L_0)+\Delta(n,K,L)-n\Big\}\leq$$
$$\sum_{s=s_0}^{s_{\max}}\sum_{\tilde{P}\in\mathscr{Y}_s(n,K,L)}\mathbb{P}\Big\{\langle\Xi,\tilde{P}-P_*\rangle\geq C_0^2 2^{2s-2}\tau(n,K_0,L_0)+\Delta(n,K,L)-n\Big\}$$

Below we shall use the following version of Bernstein inequality (see, e.g., [31]): if Ξ is a

matrix of independent Bernoulli errors and G is an arbitrary matrix of the same size, then for any $t > 0$ one has

$$\mathbb{P}\left\{\langle \Xi, G\rangle > t\right\} \leq \max\left(e^{-\frac{t^2}{4\|G\|_F^2}}, e^{-\frac{3t}{4\|G\|_\infty}}\right). \tag{A.44}$$

We apply (A.44) with $G = \tilde{P} - P_*$ and

$$t = C_0^2\Big[2^{2s-2}\tau(n, K_0, L_0) + C_2\tau(n, K, L)\Big]. \tag{A.45}$$

Then, $\|G\|_\infty = 1$ and $\|G\|^2 \leq C_0^2 2^{2s+2}\tau(n, K_0, L_0)$ due to $\tilde{P} \in \mathscr{Y}_s(n, K, L) \subseteq \mathcal{J}_{s+1}(K, L)$.

Denote

$$d_{K,L}^{(s)} = \max\left\{e^{-\frac{t^2}{4C_0^2 2^{2s+2}\,\tau(n,K_0,L_0)}}, e^{-\frac{3t}{4}}\right\} \tag{A.46}$$

$$d_{K,L} = \sum_{s=s_0}^{s_{\max}} d_{K,L}^{(s)} \cdot \exp\left\{\tau(n, K, L)\right\} \tag{A.47}$$

Obtain

$$\mathbb{P}\Big\{\sup_{\hat{P}\in\Omega_{K,L}}\Big[\langle \Xi, \hat{P} - P_*\rangle - \frac{1}{4}\Big\|\hat{P} - P_*\Big\|_F^2 - \Delta(n, K, L)\Big] \geq 0\Big\} \leq d_{K,L} \tag{A.48}$$

Observe that

$$\exp\Big\{-\frac{t^2}{4C_0^2 2^{2s+2}\tau(n, K_0, L_0)}\Big\} \geq \exp\Big\{-\frac{3t}{4}\Big\}$$

is equivalent to $t \leq 3C_0^2 2^{2s+2}\tau(n, K_0, L_0)$ which can be rewritten as

$$C_2\tau(n, K, L) \leq 47 \cdot 2^{2s-2}\tau(n, K_0, L_0) \tag{A.49}$$

Now, consider two cases: when (A.49) holds and when it does not.

Case 1: If (A.49) holds, then

$$d_{K,L}^{(s)} \le \exp\Big\{ - C_0^2\Big[2^{2s-8}\tau(n,K_0,L_0) + \frac{C_2^2\tau^2(n,K,L)}{2^{2s+4}\tau(n,K_0,L_0)}\Big]\Big\},$$

so that

$$\begin{aligned}
&d_{K,L}^{(s)} \exp\big\{\tau(n,K,L)\big\} \le \\
&\exp\Big\{ - \Big[C_0^2 2^{2s-8}\tau(n,K_0,L_0) - \frac{47\cdot 2^{2s-2}}{C_2}\tau(n,K_0,L_0)\Big]\Big\} \le \\
&\exp\Big\{ - \tau(n,K_0,L_0)\cdot 2^{2s_0-8}\Big[C_0^2 - \frac{47\cdot 64}{C_2}\Big]\Big\}.
\end{aligned}$$

Thus, it follows from (A.46) and (A.47) that

$$d_{K,L} \le \log_2 n \cdot \exp\Big\{-\tau(n,K_0,L_0)2^{2s_0-8}\tilde{C}\Big\} \tag{A.50}$$

where $\tilde{C} = (C_0^2C_2 - 47\cdot 64)/C_2$, provided $C_0C_2 \ge 47\cdot 64$.

Case 2: If (A.49) does not hold, then

$$\begin{aligned}
&d_{K,L}^{(s)} \le \\
&\exp\Big\{ - \frac{3C_0^2}{4}\Big[2^{2s-2}\tau(n,K_0,L_0) + C_2\tau(n,K,L)\Big]\Big\} \le \\
&\exp\Big\{ - \tau(n,K,L) - \tau(n,K,L)\big(\frac{3C_0^2C_2}{4} - 1\big)\Big\}
\end{aligned}$$

Hence, if $3C_0^2C_2 > 4$, then

$$d_{K,L} \le \log_2 n \cdot \exp\Big\{ - \tau(n,K,L)\Big(\frac{3C_0^2C_2 - 4}{4}\Big)\Big\}. \tag{A.51}$$

Combine (A.50) and (A.51) and observe that for $C_2 = 1$ and $C_0^2 = 47\cdot 64 + 1 = 3009$ inequalities $C_0C_2 \ge 47\cdot 64$ and $3C_0^2C_2 > 4$ hold. Then, due to $\tau(n,K,L) \ge 2n$, for any

(K, L)

$$d_{K,L} \leq \log_2 n \cdot \exp\Big\{ -2n \cdot 2^{2s_0-8} \Big\},$$

so that validity of the lemma follows from (A.48).

Lemma A.3.4. *For any matrices $A, B \in \mathbb{R}^{m\times n}$ and any unit vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$, let*

$$\Pi_{u,v}(A) = (uu^T)A(vv^T) \tag{A.52}$$

denote the projection of matrix A on the vectors (u, v). Then,

$$\langle \Pi_{u,v}(B), A - \Pi_{u,v}(A) \rangle = 0. \tag{A.53}$$

Furthermore, if we let $\hat{u}$ and $\hat{v}$ be the singular vectors of matrix A corresponding to its largest singular value σ, the best rank one approximation of A is given by

$$\Pi_{\hat{u},\hat{v}}(A) = (\hat{u}\hat{u}^T)A(\hat{v}\hat{v}^T) = \sigma\hat{u}\hat{v}^T. \tag{A.54}$$

Lemma A.3.5. *Let $(\hat{u}, \hat{v})$ and (u, v) denote the pairs of singular vectors of matrices A and P, respectively, corresponding to their largest singular values. Then,*

$$\|\Pi_{u,v}(P) - P\|_F \leq \|\Pi_{\hat{u},\hat{v}}(P) - P\|_F \leq \|\Pi_{\hat{u},\hat{v}}(A) - P\|_F \tag{A.55}$$

where $\Pi_{u,v}(\cdot)$ is defined in (A.52).

Proof. The first inequality in (A.55) is true because $\Pi_{u,v}(P)$ is the best rank one approxi-

mation of P. Now let $A = P + \Xi$. Then

$$\|\Pi_{\hat{u},\hat{v}}(A) - P\|_F^2 = \|\Pi_{\hat{u},\hat{v}}(P) - P + \Pi_{\hat{u},\hat{v}}(\Xi)\|_F^2 = \|\Pi_{\hat{u},\hat{v}}(P) - P\|_F^2 + \|\Pi_{\hat{u},\hat{v}}(\Xi)\|_F^2$$

which leads to the second inequality in (A.55).

Lemma A.3.6. *Let elements of matrix $\Xi \in (-1,1)^{n\times n}$ be independent Bernoulli errors and matrix Ξ be partitioned into KL sub-matrices $\Xi^{(l,k)}$, $l = 1, \cdots, L$, $k = 1, \cdots, K$. Then, for any $x > 0$*

$$\mathbb{P}\left\{\sum_{l=1}^{L}\sum_{k=1}^{K}\left\|\Xi^{(l,k)}\right\|_{op}^2 \le C_1 nK + C_2K^2\ln(ne) + C_3 x\right\} \ge 1 - \exp(-x), \tag{A.56}$$

where C_1, C_2 and C_3 are absolute constants independent of n,K, and L.

Proof. See [46] for the proof.

Lemma A.3.7. *For any $t > 0$,*

$$\mathbb{P}\left\{\sum_{l=1}^{\hat{L}}\sum_{k=1}^{\hat{K}}\left\|\Xi^{(l,k)}(\hat{Z},\hat{C},\hat{K},\hat{L})\right\|_{op}^2 - F_1(n,\hat{K},\hat{L}) \le C_3 t\right\} \ge 1 - \exp(-t). \tag{A.57}$$

where $F_1(n,K,L) = C_1nK + C_2K^2\ln(ne) + C_3(\ln n + (n+1)\ln K + K\ln L)$.

Proof. Using Lemma A.3.6, for any fixed K, L, $Z \in \mathcal{M}_{n,K}$, and $C \in \mathcal{M}_{K,L}$, we have

$$\mathbb{P}\left\{\sum_{l=1}^{L}\sum_{k=1}^{K}\left\|\Xi^{(l,k)}(Z,C,K,L)\right\|_{op}^2 - C_1nK - C_2K^2\ln(ne) - C_3x \ge 0\right\} \le \exp(-x).$$

Application of the union bound over $Z \in \mathcal{M}_{n,K}$, $C \in \mathcal{M}_{K,L}$, $K \in \{1,...,n\}$, and $L \in$

$\{1, ..., K\}$ and setting $x = t + \ln n + (n+1)\ln K + K \ln L$ yield

$$
\begin{aligned}
&\mathbb{P}\Bigg\{ \sum_{l=1}^{\hat{L}} \sum_{k=1}^{\hat{K}} \Big\| \Xi^{(l,k)}(\hat{Z}, \hat{C}, \hat{K}, \hat{L}) \Big\|_{op}^2 - F_1(n, \hat{K}, \hat{L}) \geq C_3 t \Bigg\} \\
&\leq \mathbb{P}\Bigg\{ \max_{1 \leq K \leq n} \max_{1 \leq L \leq K} \max_{Z \in \mathcal{M}_{n,K}} \max_{C \in \mathcal{M}_{K,L}} \Bigg(\sum_{l=1}^{L} \sum_{k=1}^{K} \Big\| \Xi^{(l,k)}(Z, C, K, L) \Big\|_{op}^2 - F_1(n, K, L) \Bigg) \geq C_3 t \Bigg\} \\
&\leq \sum_{i=1}^{n} \sum_{j=1}^{K} \sum_{Z \in \mathcal{M}_{n,K}} \sum_{C \in \mathcal{M}_{K,L}} \mathbb{P}\Bigg\{ \sum_{l=1}^{L} \sum_{k=1}^{K} \Big\| \Xi^{(l,k)}(Z, C, K, L) \Big\|_{op}^2 - F_1(n, K, L) \geq C_3 t \Bigg\} \\
&\leq nKK^n L^K \exp\Big\{ -t - \ln n - (n+1)\ln K - K \ln L \Big\} = \exp(-t),
\end{aligned}
$$

which completes the proof.

LIST OF REFERENCES

[1] Emmanuel Abbe, *Community detection and stochastic block models: Recent developments*, J. Mach. Learn. Res. **18** (2018), no. 177, 1–86.

[2] Pankaj K Agarwal and Nabil H Mustafa, *K-means projective clustering*, Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM, 2004, pp. 155–165.

[3] Pankaj K. Agarwal and Nabil H. Mustafa, *K-means projective clustering*, Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (New York, NY, USA), PODS '04, ACM, 2004, pp. 155–165.

[4] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing, *Mixed membership stochastic blockmodels*, J. Mach. Learn. Res. **9** (2008), 1981–2014.

[5] Arash A. Amini and Elizaveta Levina, *On semidefinite relaxations for the block model*, Ann. Statist. **46** (2018), no. 1, 149–179.

[6] Debapratim Banerjee and Zongming Ma, *Optimal hypothesis testing for stochastic block models with growing degrees*, 2017.

[7] Peter J. Bickel and Aiyou Chen, *A nonparametric view of network models and newman–girvan and other modularities*, Proceedings of the National Academy of Sciences **106** (2009), no. 50, 21068–21073.

[8] Terrance Boult and Lisa Gottesfeld Brown, *Factorization-based segmentation of motions*, 11 1991, pp. 179 – 186.

[9] P. S. Bradley and O. L. Mangasarian, *k-plane clustering*, J. of Global Optimization **16** (2000), no. 1, 23–32.

[10] Alain Celisse, Jean-Jacques Daudin, and Laurent Pierre, *Consistency of maximum-likelihood and variational estimators in the stochastic block model*, Electron. J. Statist. **6** (2012), 1847–1899.

[11] Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu, *High-dimensional mixed graphical models*, Journal of Computational and Graphical Statistics **26** (2017), no. 2, 367–378.

[12] Nicolas A Crossley, Andrea Mechelli, Petra E Vértes, Toby T Winton-Brown, Ameera X Patel, Cedric E Ginestet, Philip McGuire, and Edward T Bullmore, *Cognitive relevance of the community structure of the human brain functional coactivation network*, vol. 110, National Acad Sciences, 2013, pp. 11583–11588.

[13] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, *Least angle regression*, Annals of Statistics **32** (2004), 407–499.

[14] E. Elhamifar and R. Vidal, *Sparse subspace clustering*, 2009 IEEE Conference on Computer Vision and Pattern Recognition, June 2009, pp. 2790–2797.

[15] Ehsan Elhamifar and Rene Vidal, *Sparse subspace clustering: Algorithm, theory, and applications*, IEEE Trans. Pattern Anal. Mach. Intell. **35** (2013), no. 11, 2765–2781.

[16] P. Erdös and A. Rényi, *On random graphs i*, Publicationes Mathematicae Debrecen **6** (1959), 290.

[17] P. Favaro, R. Vidal, and A. Ravichandran, *A closed form solution to robust subspace estimation and clustering*, CVPR '11, IEEE Computer Society, 2011, pp. 1801–1807.

[18] Aditya Gangrade, Praveen Venkatesh, Bobak Nazer, and Venkatesh Saligrama, *Testing changes in communities for the stochastic block model*, 2018.

[19] Chao Gao and John Lafferty, *Testing for global network structure using small subgraph statistics*, 2017.

[20] Chao Gao, Zongming Ma, Anderson Y. Zhang, and Harrison H. Zhou, *Achieving optimal misclassification proportion in stochastic block models*, J. Mach. Learn. Res. **18** (2017), no. 1, 1980–2024.

[21] Chao Gao, Zongming Ma, Anderson Y Zhang, Harrison H Zhou, et al., *Community detection in degree-corrected block models*, The Annals of Statistics **46** (2018), no. 5, 2153–2185.

[22] Christophe Giraud, *Introduction to high-dimensional statistics*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability, CRC Press, Hoboken, NJ, 2015.

[23] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi, *A survey of statistical network models*, Foundations and Trends® in Machine Learning **2** (2010), no. 2, 129–233.

[24] Jiashun Jin, Zheng Ke, and Shengming Luo, *Network global testing by counting graphlets*, Proceedings of the 35th International Conference on Machine Learning (Stockholmsmässan, Stockholm Sweden) (Jennifer Dy and Andreas Krause, eds.), Proceedings of Machine Learning Research, vol. 80, PMLR, 10–15 Jul 2018, pp. 2333–2341.

[25] Jiashun Jin, Zheng Tracy Ke, and Shengming Luo, *Estimating network memberships by simplex vertex hunting*, arXiv e-prints (2017), arXiv:1708.07852.

[26] Jiashun Jin, Zheng Tracy Ke, and Shengming Luo, *Estimating network memberships by simplex vertex hunting*, 2017.

[27] Ian T Jolliffe, *Principal components in regression analysis*, Principal component analysis, Springer, 1986, pp. 129–155.

[28] Antony Joseph and Bin Yu, *Impact of regularization on spectral clustering*, Ann. Statist. **44** (2016), no. 4, 1765–1791.

[29] Brian Karrer and Mark E. J. Newman, *Stochastic blockmodels and community structure in networks*, Physical review. E, Statistical, nonlinear, and soft matter physics **83 1 Pt 2** (2011), 016107.

[30] Olga Klopp, Karim Lounici, and Alexandre B. Tsybakov, *Robust matrix completion*, Probability Theory and Related Fields **169** (2017), no. 1, 523–564.

[31] Olga Klopp, Yu Lu, Alexandre B. Tsybakov, and Harrison H. Zhou, *Structured matrix estimation and completion*, Bernoulli **25** (2019), no. 4B, 3883–3911.

[32] Eric D. Kolaczyk, *Statistical analysis of network data: Methods and models*, 1st ed., Springer Publishing Company, 2009.

[33] Can M. Le, Elizaveta Levina, and Roman Vershynin, *Optimization via low-rank approximation for community detection in networks*, Ann. Statist. **44** (2016), no. 1, 373–400.

[34] Jing Lei, *A goodness-of-fit test for stochastic block models*, Ann. Statist. **44** (2016), no. 1, 401–424.

[35] Jing Lei and Alessandro Rinaldo, *Consistency of spectral clustering in stochastic block models*, Ann. Statist. **43** (2015), no. 1, 215–237.

[36] Jure Leskovec and Julian J Mcauley, *Learning to discover social circles in ego networks*, Advances in neural information processing systems, 2012, pp. 539–547.

[37] Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter J. Bickel, and Elizaveta Levina, *Hierarchical community detection by recursive partitioning*, 2018.

[38] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma, *Robust recovery of subspace structures by low-rank representation*, IEEE Trans. Pattern Anal. Mach. Intell. **35** (2013), no. 1, 171–184.

[39] Guangcan Liu, Zhouchen Lin, and Yong Yu, *Robust subspace segmentation by low-rank representation*, Proceedings of the 27th International Conference on International Conference on Machine Learning (USA), ICML'10, Omnipress, 2010, pp. 663–670.

[40] François Lorrain and Harrison C. White, *Structural equivalence of individuals in social networks*, The Journal of Mathematical Sociology **1** (1971), no. 1, 49–80.

[41] Yi Ma, Allen Y. Yang, Harm Derksen, and Robert Fossum, *Estimation of subspace arrangements with applications in modeling and segmenting mixed data*, SIAM Rev. **50** (2008), no. 3, 413–458.

[42] Julien Mairal, F Bach, J Ponce, G Sapiro, R Jenatton, and G Obozinski, *Spams: A sparse modeling software, v2.3*, URL http://spams-devel. gforge. inria. fr/downloads. html (2014).

[43] S.G. Mallat and Zhifeng Zhang, *Matching pursuits with time-frequency dictionaries*, Trans. Sig. Proc. **41** (1993), no. 12, 3397–3415.

[44] Rajarshi Mukherjee and Subhabrata Sen, *Testing degree corrections in stochastic block models*, 2017.

[45] Carlo Nicolini, Cécile Bordier, and Angelo Bifone, *Community detection in weighted brain connectivity networks beyond the resolution limit*, Neuroimage **146** (2017), 28–39.

[46] Majid Noroozi, Ramchandra Rimal, and Marianna Pensky, *Estimation and Clustering in Popularity Adjusted Stochastic Block Model*, arXiv e-prints (2019), arXiv:1902.00431.

[47] David Pollard, *Empirical processes: theory and applications*, NSF-CBMS regional conference series in probability and statistics, JSTOR, 1990, pp. i–86.

[48] Karl Rohe, Sourav Chatterjee, Bin Yu, et al., *Spectral clustering and the high-dimensional stochastic blockmodel*, Ann. Statist. **39** (2011), no. 4, 1878–1915.

[49] Srijan Sengupta and Yuguo Chen, *A block model for node popularity in networks with community structure*, Journal of the Royal Statistical Society Series B **80** (2018), no. 2, 365–386.

[50] Mahdi Soltanolkotabi and Emmanuel J. Candes, *A geometric analysis of subspace clustering with outliers*, Ann. Statist. **40** (2012), no. 4, 2195–2238.

[51] Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candes, *Robust subspace clustering*, Ann. Statist. **42** (2014), no. 2, 669–699.

[52] Paul Tseng, *Nearest q-flat to m points*, Journal of Optimization Theory and Applications **105** (2000), no. 1, 249–252.

[53] Deepak Verma and Marina Meila, *A comparison of spectral clustering algorithms*, University of Washington Tech Rep UWCSE030501 **1** (2003), 1–18.

[54] René Vidal, *Subspace clustering*, IEEE Signal Processing Magazine **28** (2011), no. 2, 52–68.

[55] Rene Vidal, Yi Ma, and Shankar Sastry, *Generalized principal component analysis (gpca)*, IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005), no. 12, 1945–1959.

[56] Ken Wakita and Toshiyuki Tsurumi, *Finding community structure in mega-scale social networks: [extended abstract]*, Proceedings of the 16th International Conference on World Wide Web (New York, NY, USA), WWW '07, ACM, 2007, pp. 1275–1276.

[57] Bo Wang, Armin Pourshafeie, Marinka Zitnik, Junjie Zhu, Carlos D. Bustamante, Serafim Batzoglou, and Jure Leskovec, *Network enhancement as a general method to denoise weighted biological networks*, Nature Communications **9** (2018), no. 1, 3108.

[58] Sanford Weisberg, *Applied linear regression*, vol. 528, John Wiley & Sons, 2005.

[59] Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al., *Consistency of community detection in networks under degree-corrected stochastic block models*, Ann. Statist. **40** (2012), no. 4, 2266–2292.