Topics in Network Analysis with Applications to Brain Connectomics

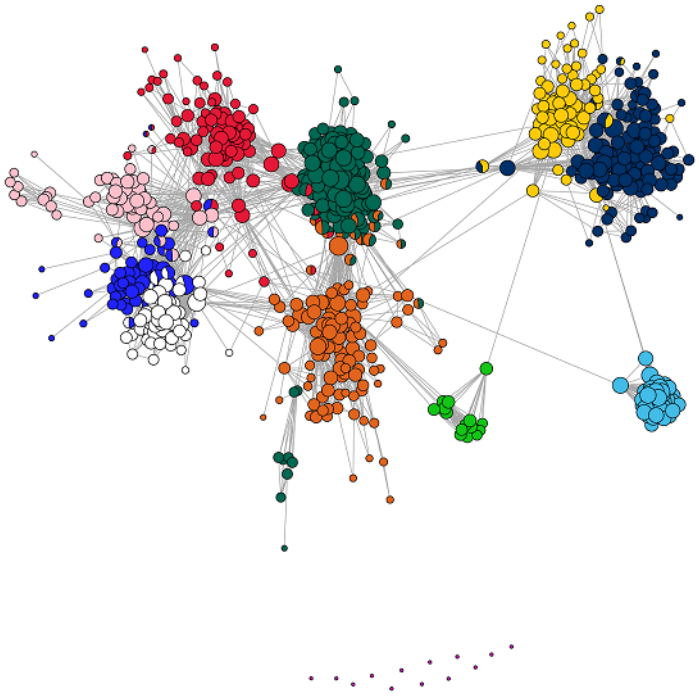
by

Jesús Daniel Arroyo Relión

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2018

Doctoral Committee:

Professor Elizaveta Levina, Chair
Associate Professor XuanLong Nguyen
Associate Professor Chandra Sripada
Professor Ji Zhu



Jesús Daniel Arroyo Relión
jarroyor@umich.edu
ORCID iD: 0000-0003-3071-9043

Dedicado a mamá, papá y Sarahí.

Acknowledgments

My Ph.D. studies were a great learning experience in both academic and personal terms. I owe most of what I learned to the interaction with many people, and I would like to acknowledge some of them here.

First of all, I am deeply grateful to my advisor, Professor Liza Levina, for her guidance and support during my studies. Through uncountable and invaluable discussions, suggestions, opportunities and advice, Liza helped me in shaping and developing my research interests and career. I am really thankful for her constant encouragement and patience.

I am also very thankful to the members of my committee, Professors Ji Zhu, Long Nguyen and Chandra Sripada. Their invaluable insights and discussions during my studies contributed to my research and professional development in many aspects. I would also like to thank Daniel Kessler and Professor Stephan Taylor for their insightful contributions to this thesis; I learned a lot about neuroimaging and statistics by collaborating with them.

I am grateful to all the faculty, staff and students in the Department of Statistics for providing a friendly and stimulating environment to develop my career. I am especially thankful to the current and past members of the Levina and Zhu research group for the inspiring discussions, presentations and comments, and in particular to Yuan Zhang for kindly sharing his code. I would also like to thank Elizabeth Hou, Mikhail Yurochkin and Alexander Giessing for the numerous discussions, collaborations and company inside and outside the school.

I am also thankful to many people in Ann Arbor for making my stay more enjoyable. I am very grateful to Laura for being caring, patient and supportive all the time. Many friends have enriched my experience in Michigan. I would like to especially thank Adrián, Andres, Ivan and Wilmer for their friendship in the past years.

Finally, I am really thankful to my mom, dad and sister Sarahí for their love and constant support throughout my life. I am glad of having my extended family and friends from México that are always able to listen and support me from far away.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF APPENDICES

ABSTRACT

Large complex network data have become common in many scientific domains, and require new statistical tools for discovering the underlying structures and features of interest. This thesis presents new methodology for network data analysis, with a focus on problems arising in the field of brain connectomics. Our overall goal is to learn parsimonious and interpretable network features, with computationally efficient and theoretically justified methods.

The first project in the thesis focuses on the problem of prediction with network covariates. This setting is motivated by neuroimaging applications, in which each subject has an associated brain network constructed from fMRI data, and the goal is to derive interpretable prediction rules for a phenotype of interest or a clinical outcome. Existing approaches to this problem typically either reduce the data to a small set of global network summaries, losing a lot of local information, or treat network edges as a "bag of features" and use standard statistical tools without accounting for the network nature of the data. We develop a method that uses all edge weights, while still effectively incorporating network structure by using a penalty that encourages sparsity in both the number of edges and the number of nodes used. We develop efficient optimization algorithms for implementing this method and show it achieves state-of-the-art accuracy on a dataset of schizophrenic patients and healthy controls while using a smaller and more readily interpretable set of features than methods which ignore network structure. We also establish theoretical performance guarantees.

Communities in networks are observed in many different domains, and in brain

networks they typically correspond to different regions of the brain responsible for different functions. In connectomic analyses, there are standard parcellations of the brain into such regions, typically obtained by applying clustering methods to brain connectomes of healthy subjects. However, there is now increasing evidence that these communities are dynamic, and when the goal is predicting a phenotype or distinguishing between different conditions, these static communities from an unrelated set of healthy subjects may not be the most useful for prediction. We present a method for supervised community detection, that is, a method that finds a partition of the network into communities that is most useful for predicting a particular response. We use a block-structured regularization and compute the solution with a combination of a spectral method and an ADMM optimization algorithm. The method performs well on both simulated and real brain networks, providing support for the idea of task-dependent brain regions.

The last part of the thesis focuses on the problem of community detection in the general network setting. Unlike in neuroimaging, statistical network analysis is typically applied to a single network, motivated by datasets from the social sciences. While community detection has been well studied, in practice nodes in a network often belong to more than one community, leading to the much harder problem of overlapping community detection. We propose a new approach for overlapping community detection based on sparse principal component analysis, and develop efficient algorithms that are able to accurately recover community memberships, provided each node does not belong to too many communities at once. The method has a very low computational cost relative to other methods available for this problem. We show asymptotic consistency of recovering community memberships by the new method, and good empirical performance on both simulated and real-world networks.

CHAPTER 1

Introduction

Network data analysis has received increasing interest in the recent years, as networks have become a popular data structure in many different fields such as social sciences, engineering, biology, chemistry and neuroscience. Networks are commonly used to represent data derived from complex systems, in which the nodes of the network correspond to the units of the system, and the edges encode interactions between these units. A network can represent relationships or interactions between agents in a social environment, physical connections in circuits and computer networks, protein-protein interactions in gene regulatory networks, bonds between atoms in chemical compounds, or the connectivity between different areas in the brain, to name some examples.

Technological advancements facilitate the collection of information, resulting in high-dimensional and complex datasets. New statistical methodologies that are able to provide parsimonious representations of underlying structures in the data can help to understand the scientific problem of interest. To deal with large amounts of information, computational efficient solutions are required. This thesis focuses on developing new methods for the analysis of network data in different problems, with special emphasis in applications to neuroimaging. Specific projects in this thesis are described below.

Prediction with network-valued covariates

Statistical analysis of samples of networks has received growing recent attention, particularly motivated by applications in neuroscience. Imaging techniques have made possible to study the brain at the population level, in order to characterize the activity and connectivity of the brain under different conditions, such as mental illneses, age-related changes or other subject-specific responses of interest. The structure of the

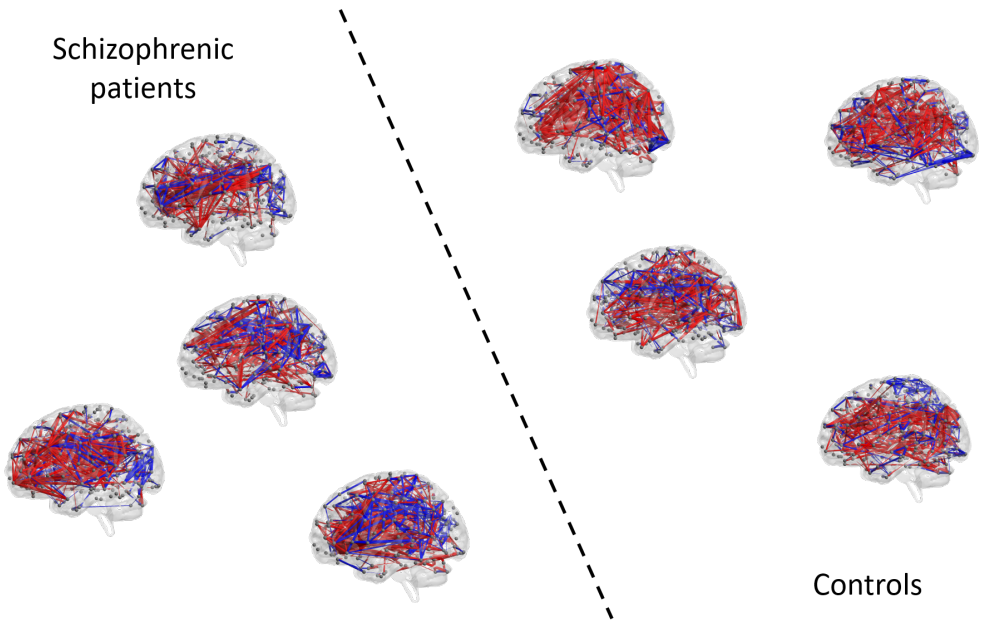


Figure 1.1: Supervised classification of brain networks.

brain is commonly represented with brain networks constructed from imaging data, in order to represent the connectivity between a predefined set of regions in the brain (Bullmore and Sporns, 2009).

Two generic approaches are usually performed when analyzing samples of networks. One approach consists in reducing networks to a set of global summaries, but this approach loses the ability to capture local information. In the second approach, multivariate data analysis tools are applied to vectorized adjacency matrices while ignoring the network structure of the data, but this can harm statistical power and interpretation. The first two chapters of the thesis aim to bridge the gap between these two approaches, by introducing new methods able to exploit the local information of the edges and the network structure of the data at the same time.

Network classification is the problem of predicting a response from a network-valued covariate (see Figure 1.1). This problem has applications in neuroscience, since brain networks are successful diagnostic biomarkers for certain mental illnesses such as schizophrenia. To understand the brain, the interest is not only to correctly classify a patient, but also to identify which abnormal connections might be predictive. In Chapter 2, we propose a classifier for network data. The network structure of the data is incorporated via convex regularizations that promote sparsity not only in the edges, which are the variables of interest in the problem, but also in the nodes. Node sparsity is obtained with a group lasso penalty, but the groups overlap, requiring a careful

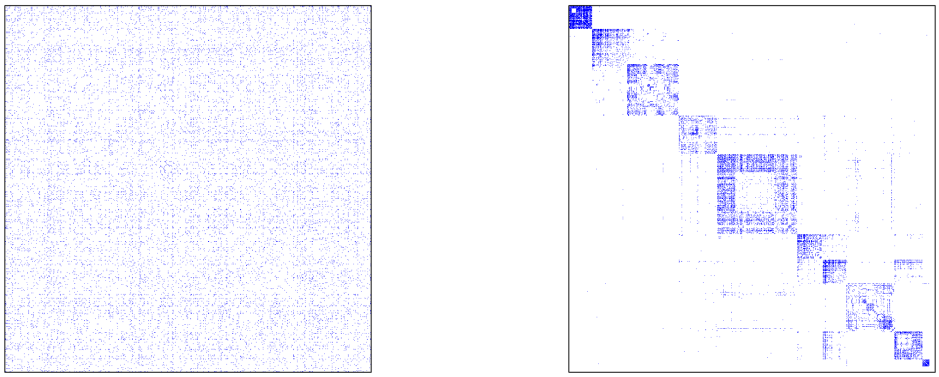


Figure 1.2: Adjacency matrix of a Facebook friendship network with the nodes permuted at random (left) and ordered according to their communities (right).

treatment in optimization and in theoretical analysis. We show that our penalty is able to correctly recover the predictive subnetwork for classification. Our method outperforms benchmarks in classification accuracy and simplifies the interpretation of the results by offering parsimonious solutions.

Supervised community detection in multiple networks

Community detection is the problem of clustering the nodes of a network into "coherent" groups. Many real-world networks from different fields show community structure, making community detection a problem of interest in network analysis.

In the analysis of large scale networks, communities provide a convenient way to simplify the structure of the data by partitioning the nodes of the network into a small number of groups that facilitate the analysis and interpretation, and allow to discover underlying structures of interest. Figure 1.2 shows the adjacency matrix of a Facebook ego-network, in which the nodes represent all the friends of a certain user, and the edges denote whether or not two individuals have a Facebook friendship relation. On the left side, the nodes are arranged in no particular order, and making sense of the data is extremely hard. After the nodes are ordered according to their community assignments (right side), some structure is revealed. The nodes are divided into coherent clusters, and edges appear in homogeneous blocks.

In brain networks, nodes organize into groups that co-activate during brain activ-

ity. The analysis and interpretation of brain studies is usually done at the community level, as it simplifies the interpretation by reducing the nodes to a number of larger units that have known brain functionality. Thus, being able to correctly identify meaningful clusters of nodes is a critical task in this setting.

A practical question when fitting communities to observed data is how to select the best model and evaluate the quality of the solution, as the task is usually unsupervised. In the context of prediction with network-valued data, however, communities can be used to improve classification by regularizing the solutions using this structure. Conversely, having "supervision" in the form of a response can guide the choice of the community assignments. In Chapter 3, we propose a block-structured regularization for problems with network-valued predictors. This regularization vastly reduces the number of coefficients to estimate and achieves the goal of assigning nodes to communities in a supervised way. Finding communities is in general a computationally hard problem, but we derive some efficient methods to obtain an approximate solution to the problem. The methods are evaluated in simulations and in real brain networks obtained from fMRI data, showing interpretable and accurate solutions.

Sparse overlapping community detection in a network

The problem of community detection has been extensively studied in the analysis of a single network. The stochastic block model (SBM) (Holland et al., 1983) provides a way to characterize community structure from a statistical perspective. This model is often too simplistic to apply to real-life networks, but several extensions have been proposed. Here, we focus in overlapping communities (Airoldi et al., 2009; Zhang et al., 2014), in which nodes are allowed to belong to more than one community at the same time.

Figure 1.3 shows a plot of the nodes and edges of the same Facebook ego network from Figure 1.2. If communities overlap, each node can be represented with a vector indicating the degree of association to each of the communities in the network. In the figure, a pie plot on each node represents the memberships, with the colors indicating the different community assignments. As always, parsimonious solutions have the advantage of interpretability, and thus solutions in which a node belongs to relatively few communities are preferable. In the extreme case, each node belongs to only one community as in the classic community detection problem.

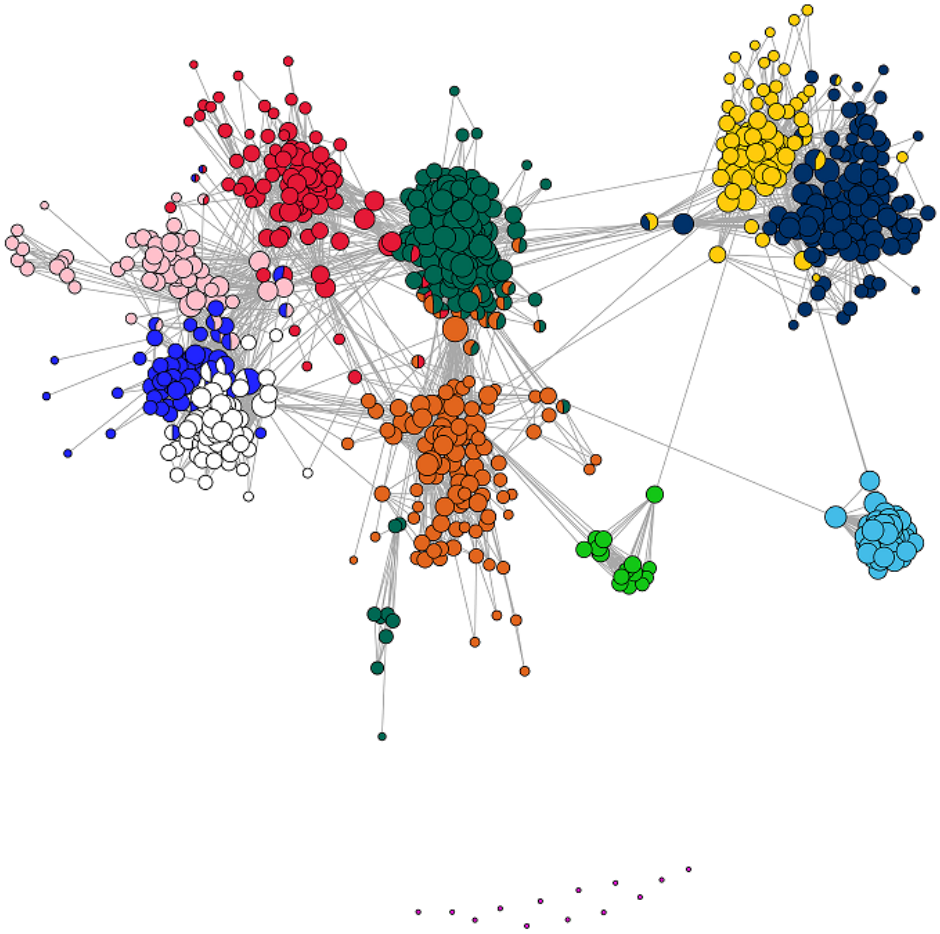


Figure 1.3: Overlapping communities of a Facebook ego network. Pie plots of each node indicate the degree of association to each community.

In Chapter 4, we develop efficient methods for fitting sparse overlapping communities. Our methodology is based on the observation that in statistical models for community detection, the expected adjacency matrix of the network can be characterized with a sparse eigenbasis. We propose a method based on sparse principal component analysis that is able to fit overlapping community memberships. The sparsity of the solution can be controlled with a tuning parameter. We show that the method is able to recover the correct sparsity pattern that corresponds to the community memberships for some statistical models. Many existing methods for community detection operate by performing a clustering procedure to the rows of the adjacency matrix leading eigenvectors. In contrast, by fitting an appropriate eigenbasis we directly estimate the memberships from the network, which allow us to obtain better statistical accuracy and computational efficiency. Our implementation easily handles networks with millions of nodes, and shows accurate results in simulated and real-world networks.

CHAPTER 2

Network classification with applications to brain connectomics

2.1 Introduction

Network data analysis has received a lot of attention in recent literature, especially unsupervised analysis of a single network which is thought of as generated from an exchangeable random graph model, for example Bickel and Chen (2009); Le et al. (2015); Zhang et al. (2016a); Gao et al. (2015) and many others. This setting is applicable to a number of real life scenarios, such as social networks, but there are situations where the network nodes are labeled and therefore not exchangeable, and/or more than one network is available for analysis, which have received relatively less attention. Here we focus on the setting motivated by brain connectomics studies, where a sample of networks is available from multiple populations of interest (for example, mentally ill patients and healthy controls). In this setting, each unit in the population (e.g., a patient) is represented by their own network, and the nodes (brain regions of interest) are labeled and shared across all networks, through a registration process that maps all individual brains onto a common atlas. There are many classical statistical inference questions one can ask in this setting, for example, how to compare different populations (Tang et al., 2017b,a). The question we foocus on in this paper is a classification problem: given a training sample of networks with labeled nodes drawn from multiple classes, the goal is to learn the rules for predicting the class of a given network, and just as importantly, interpret these rules.

Network methods are a popular tool in the neuroscience literature (Bullmore and Sporns, 2009; Bullmore and Bassett, 2011). A brain network represents connectivity between different locations of an individual's brain. How connectivity is defined varies with the type of imaging technology used and the conditions under which data were collected. In this paper, we focus on functional connectivity, which is a measure of

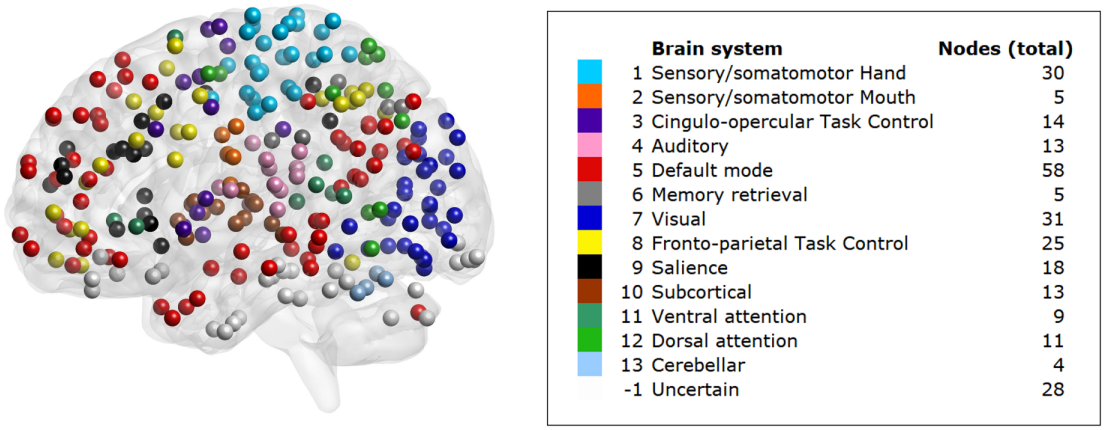


Figure 2.1: Regions of interest (ROIs) defined by Power et al. (2011), colored by brain systems, and the total number of nodes in each system.

statistical association between each pair of locations in the brain, construbted from functional magnetic resonance imaging (fMRI) data, although the methods we develop are applicable to any sample of weighted networks with labeled nodes. In fMRI studies, BOLD (blood oxygen-level dependent) signal, a known correlate of underlying neural activity, is measured at a sequence of time points at many spatial locations in the brain, known as voxels, resulting in a 4-dimensional data array, with three spatial dimensions and a time index. Brain networks constructed from fMRI data have been successfully used for various tasks, such as differentiating between certain illnesses, or between types of external stimuli (Bullmore and Sporns, 2009), and contain enough information to identify individual subjects (Finn et al., 2015). Extensive statistical literature has focused on the analysis of raw fMRI data (Lindquist et al., 2008; Zhou et al., 2013; Zhang et al., 2016b), usually aiming to characterize brain activation patterns obtained from task-based fMRI experiments. In this paper, we focus on resting-state fMRI data, where no particular task is performed and subjects are free to think about anything they want. Thus registesring the time dimension across different subjects is not possible. The connectivity network approach, which averages over the time dimension in computing a measure of dependence between different voxels, is thus a natural choice, and has been widely used with multiple types of neuroimaging data.

Two different datasets are analyzed in this paper, both of resting state fMRI studies on schizophrenic patients and healthy controls. One dataset, COBRE (about 70 each of scizophrenics and controls), is publicly available; another, which we will

refer to as UMich data (about 40 each of scizophenics and controls), was collected internally in the last author's lab. Having two datasets on the same disease allows us to cross-check models trained on one of them for classification on the other, which The raw data arrays undergo pre-processing and registration steps, discussed in detail in Section 2.6, along with additional details on data collection. To construct a brain network from fMRI measurements, a set of nodes is chosen, typically corresponding to regions of interests (ROIs) from some predefined parcellation. In our analysis we use the parcellation of (Power et al., 2011), which consists of 264 ROIs divided into 14 functional brain systems (see Figure 2.1). A connectivity measure is then computed for every pair of nodes, resulting in an adjacency matrix of size 264×264. Many choices of connectivity measures are available (Smith et al., 2013); perhaps the most commonly used one is the Pearson correlation coefficient between locations, computed by averaging over the time dimension. It has been argued that partial correlations are a better measure of connectivity (Varoquaux and Craddock, 2013; Narayan et al., 2015), but the choice depends on the final goal of analysis. In this paper we follow the vast majority of the connectomics literature and measure connectivity on each individual by using marginal correlations between the corresponding time series (see Figure 2.2). The correlations are then further rank-transformed and standardized; see Section 2.6 for details. These transformations are intended to deal with subject-to-subject variability and the global signal regression issue (Gotts et al., 2013). We observed that on our datasets, classification based on standardized ranks of marginal correlations outperformed classification based on other connectivity measures, such as marginal or partial correlations.

The problem of graph classification has been studied previously in other contexts, with a substantial literature motivated by the problem of classification of chemical compounds (Srinivasan et al., 1996; Helma et al., 2001), where graphs represent the compound's molecular structure. This setting is very different, with small networks of about 20 nodes on average, binary or categorical edges recorded with no noise, and different nodes corresponding to different networks, (Ketkar et al., 2009). Classification methods for chemical compounds is usually based on finding certain discriminative patterns in the graphs, like subgraphs or paths (Inokuchi et al., 2000; Gonzalez et al., 2000), and using them as features for training a standard classification method (Deshpande et al., 2005; Kudo et al., 2004; Fei and Huan, 2010). Computationally, finding these patterns is only possible on small binary networks.

Another type of methods is based on graph kernels (Gärtner et al., 2003; Vishwanathan et al., 2010), which define a similarity measure between two networks.

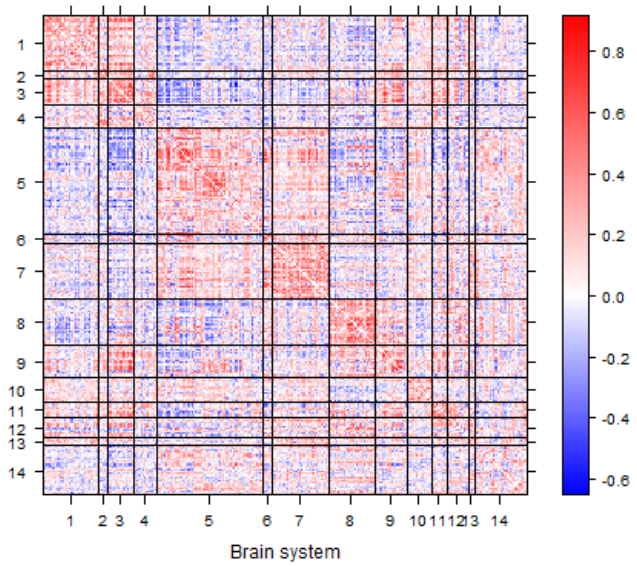


Figure 2.2: Adjacency matrix of a brain network from one of the subjects, showing the value of the Fisher z-transformed correlations between the nodes, with the 264 nodes grouped into 14 brain systems.

These kernels combined with support vector machines (SVMs) have been successfully used on small networks (Kashima et al., 2003; Borgwardt et al., 2005), but the curse of dimensionality makes local kernel methods unsuitable for large scale networks (Bengio and Monperrus, 2005). On our datasets, graph kernel methods did not perform better than random guessing.

In the context of classifying large-scale brain networks, two main approaches have been followed. One approach is to reduce the network to its global summary measures such as the average degree, clustering coefficient, or average path length (Bullmore and Sporns, 2009), and use those measures as features for training a classification method. Previous studies have reported significant differences on some of these network measures for groups of patients with certain brain diseases compared with healthy controls (Supekar et al., 2008; Liu et al., 2008), suggesting their usefulness as diagnostic biomarkers. However, global summary statistics collapse all local network information, which can harm the accuracy of classification and does not allow to identify local differences. In our data analysis, a method based on the network measures suggested in Prasad et al. (2015) performed poorly for classification (see Section 2.6).

An alternative approach to classification of large networks is to treat edge weights as a "bag of features", vectorizing the unique elements of the adjacency matrix and

ignoring the network nature of the data. This approach can leverage many existing classification methods for vectors, and provides an interpretation at the edge level if variable selection is applied (Richiardi et al., 2011; Craddock et al., 2009; Zhang et al., 2012). Spatial correlation between edges connecting neighboring nodes can be incorporated (Watanabe et al., 2014; Scott et al., 2015), although the effectiveness of this regularization will depend on the parcellation used to define nodes (Power et al., 2011). Alternatively, an individual test can be used for each edge to find significant differences between two populations, with a multiple testing correction and without constructing a classifier at all (Narayan et al., 2015). While these methods can deliver good predictions, their interpretability is limited to individual edge selection, which is less scientifically interesting than identifying differentiating nodes or regions, and they cannot account for network structure.

Taking the network structure into account can have benefits for both testing and classification settings. Some methods perform inference over groups of edges based on the community assignments of the nodes to which they are incident. For example, Sripada et al. (2014a,b) introduced *Network Contingency Analysis* which begins with massive univariate testing at each edge, and then counts the number of superthreshold connections in each *cell*, a group of edges that connect nodes in two functional systems. Nonparametric methods are then used to conduct inference on the count statistic for each cell, with multiple comparison correction for inference at the cell level. Power can be improved by applying a network-based multiple testing dependence correction (Zalesky et al., 2010). For classification, better interpretability and potentially accuracy can be obtained if we focus on understanding which brain regions or interactions between them are responsible for the differences. In somewhat related work, Vogelstein et al. (2013) proposed to look for a minimal set of nodes which best explains the difference, though that requires solving a combinatorial problem. Hypothesis testing on a type of graph average has also been proposed (Ginestet et al., 2017). Bayesian nonparametrics approaches for modeling populations of networks allow to test for local edge differences between the groups (Durante et al., 2017), but are computationally feasbile only for small networks.

Our goal in this paper is to develop a high-dimensional network classifier that uses all the individual edge weights but also respects the network structure of the data and produces more interpretable results. To achieve this goal, we use structured sparsity penalties to incorporate the network information by penalizing both the number of edges and the number of nodes selected. Although our main application here is classification of brain connectivity networks, our methods are applicable to

any weighted graphs with labeled nodes, and to general prediction problems, not just classification.

The rest of this paper is organized as follows. In Section 2.2, we introduce our classifier and the structured penalties. In Section 2.3 we show how to efficiently solve the resulting convex optimization problem by a proximal algorithm, each step of which is a further optimization problem which we solve by the alternating direction method of multipliers (ADMM). The performance of our method is evaluated and compared with other methods using simulations in Section 2.5. In Section 2.6, we analyze two brain connectivity datasets, each containing schizophrenic patients and healthy controls, and show that our regularization framework leads to state-of-the-art accuracy while providing interpretable results, some of which are consistent with previous findings and some are new. We conclude with a brief discussion in Section 2.7.

2.2 A framework for node selection in graph classification

2.2.1 A penalized graph classification approach

We start from setting up notation. All graphs we consider are defined on the same set of N labeled nodes. A graph can be represented with its adjacency matrix $A \in \mathbb{R}^{N\times N}$. We focus on graphs that are undirected ($A_{ij} = A_{ji}$) and contain no self-loops ($A_{ii} = 0$). These assumptions are not required for the derivations below, but they match the neuroimaging setting and simplify notation. Our goal is predicting a class label Y from the graph adjacency matrix A; in this paper we focus on the binary classification problem where Y takes values $\{-1, 1\}$, although extensions from binary to multi-class classification or real-valued responses are straightforward. Throughout this paper, we use $\|\cdot\|_p$ to denote the entry-wise ℓ_p norm,i.e., for a matrix $A \in \mathbb{R}^{m\times n}$, $\|A\|_p = \left(\sum_{i=1}^m \sum_{j=1}^n |A_{ij}|^p\right)^{1/p}$.

A standard general approach is to construct a linear classifier, which predicts the response Y from a linear combination of the elements of A, $\langle A, B\rangle = \mathrm{Tr}\left(B^T A\right)$, where we arrange the coefficients in a matrix $B \in \mathbb{R}^{N\times N}$ to emphasize the network nature of the predictors. We focus on linear classifiers here because variable selection is at least as important as prediction itself in the neuroimaging application, and setting coefficients to 0 is a natural way to achieve this. The coefficients are typically estimated from training data by minimizing an objective consisting of a loss function

plus a penalty. The penalty can be used to regularize the problem to make the estimator well-defined in high-dimensional problems, to select important predictors, and to impose structure, and many such penalties have been proposed, starting from the lasso (Tibshirani, 1996). Our focus is on designing a classifier in this framework that respects and utilizes the network nature of the predictors. In brain networks in particular, neuroscientists believe that edges are organized in subnetworks, also called brain systems (Power et al., 2011), that carry out specific functions, and certain subnetworks are important for prediction (Bullmore and Sporns, 2009), although different studies tend to implicate different regions (Fornito et al., 2012). Thus we aim to find subnetworks with good discriminative power, and hence select both the most informative nodes and edges. Although the methods we develop here can be used on small networks too, our main focus here is on the more challenging case of large brain networks (hundreds of nodes).

Let $\{(A^{(1)}, Y_1), \ldots, (A^{(n)}, Y_n)\}$ be the training sample of undirected adjacency matrices with their class labels, and let $Y = (Y_1, \ldots, Y_n)$. A generic linear classifier described above is computed by finding the coefficients B defined by

$$\hat{B} = \arg\min_{B \in \mathcal{B}} \left\{ \ell(B) + \Omega(B) \right\}, \tag{2.1}$$

where $\mathcal{B} = \left\{ B \in \mathbb{R}^{N \times N} : B = B^T, \mathrm{diag}(B) = 0 \right\}$, Ω is a penalty, and

$$\ell(B) = \frac{1}{n} \sum_{k=1}^{n} \tilde{\ell}(Y_k, A^{(k)}; B)$$

is a loss function evaluated on the training data. Our methodology can accommodate different choices of loss functions that can extend beyond classification problems (e.g., least squares or generalized linear models). The optimization algorithm presented in Section 2.3 can work with any convex and continuously differentiable loss function, and further assumptions are required for consistency (see Section 2.4). In this paper, for the purpose of classification we use the logistic loss function in the simulations and data analysis, which is defined as

$$\tilde{\ell}(Y, A; B, b) = \log\left(1 + \exp(-Y \langle A, B \rangle + b)\right) .$$

The threshold b is an additional parameter to be estimated.

To capture structural assumptions on important predictive edges, we focus on convex structured sparsity penalties (Bach et al., 2012) that encourage a small number

of active nodes, by which we mean nodes attached to at least one edge with a non-zero coefficient. One approach to finding a set of such nodes was proposed by Vogelstein et al. (2013), who called it a signal-subgraph, and proposed finding the minimal set of nodes (called signal vertices) which together are incident to all selected edges (but not every node connected to a selected edge is a signal vertex). Finding this set is a combinatorial optimization problem, and the set is not always uniquely defined. Instead, we focus on convex formulations that allow for efficient computation and encourage small active node sets indirectly.

Other convex penalties have been used for fMRI data as a way to enforce spatial smoothness in the solution (Grosenick et al., 2013; Xin et al., 2014; Hu and Allen, 2015). These methods assume that voxels are equally spaced in the brain, and neighboring voxels are highly correlated. In particular, Watanabe et al. (2014) proposed penalties for brain network classification using these spatial assumptions. Here, instead of enforcing a spatial regularization directly, we aim for a regularization that can be applied to any type of network data, and in particular to brain networks with coarse and/or uneven parcellations where enforcing spatial smoothness may not work as well. In any case, the flexibility of convex optimization algorithms allows one to easily incorporate additional spatially-informed penalties if needed.

2.2.2 Selecting nodes and edges through group lasso

To reflect the network structure of the predictors, we use a penalty that promotes a sparse classifier not only in the number of edges used, but also in the number of nodes. The group lasso penalty (Yuan and Lin, 2006) is designed to eliminate a group of variables simultaneously. Here we penalize the number of active nodes by treating all edges connected to one node as a group. Then eliminating this group (a row of coefficients in the matrix B) is equivalent to de-activating a node. The group penalty is defined as

$$\Omega_{\lambda,\rho}(B) = \lambda \left(\sum_{i=1}^{N} \|B_{(i)}\|_2 + \rho \|B\|_1 \right), \tag{2.2}$$

where $B_{(i)}$ is the i-th row of B (or equivalently, the i-th column), $\|B\|_1 = \sum_i \sum_j |B_{ij}|$ is the element-wise ℓ_1 norm of B and $\lambda, \rho \geq 0$ are tuning parameters. Note that the constraint $B = B^T$ makes the groups overlap, since a coefficient B_{ij} belongs to groups associated with the nodes i and j, and therefore, the edge between nodes i and j would be selected only if neither node was de-activated. The second term in the penalty $\rho\|B\|_1$ acts as the usual lasso penalty to promote sparsity inside the group (Friedman et al., 2010), allowing to select a subset of edges for an active node. Due

to the overlap in the groups, this lasso penalty is necessary in order to produce sparse solution. The constraint $\text{diag}(B) = 0$ in (2.1) is automatically enforced with this formulation.

Remark 2.1. An alternative to the constraint in the problem (2.1) is to optimize over the set

$$\tilde{\mathcal{B}} = \left\{ B \in \mathbb{R}^{N \times N}, \text{diag}(B) = 0 \right\}.$$

Without the symmetry constraint and assuming undirected graphs, the penalty (2.2) is equivalent to the overlapping group lasso formulation of Jacob et al. (2009). This formulation has some advantages. Since it gives group lasso without overlaps, the lasso penalty $\rho\|B\|_1$ is not required to obtain sparse solutions, and more efficient optimization algorithms exist for this case. This approach would loosely correspond to the idea of selecting signal nodes as in Vogelstein et al. (2013), in the sense that an edge can be selected if at least one of its nodes is selected, and the second node could be inactive. The downside is that each edge now corresponds to two different coefficients B_{ij} and B_{ji}, the problem encountered by all variable selection methods that ignore symmetry, such as Meinshausen and Bühlmann (2006). The standard solution for this problem, as suggested by Jacob et al. (2009), is to take the average of the coefficients. Intuitively, one would expect that the formulation using $\mathcal{B}$ would be better when the significant edges are incident to a small set of nodes, since both nodes have to be active for an edge to be selected, while using $\tilde{\mathcal{B}}$ may be better when for some nodes most of their edges are significant, creating "significant hubs". Since in our application we are primarily looking for discriminative brain subnetworks, we focus on the symmetrically constrained formulation for the rest of the paper. We also found that in practice this second formulation results in less accurate classifiers for the neuroimaging data discussed in Section 2.6.

Remark 2.2. The analogue to (2.2) for directed graphs would assign coefficients B_{ij} and B_{ij} to the same group, resulting in the penalty

$$\Psi_{\lambda,\rho}(B) = \lambda \left(\sum_{i=1}^{N} \sqrt{\sum_{j} \left(B_{ij}^2 + B_{ji}^2 \right)} + \rho \|B\|_1 \right), \tag{2.3}$$

where $B \in \tilde{\mathcal{B}}$. Alternatively, we can also use the formulation of Remark 2.1, by replicating the variables and estimating two matrices of coefficients, say $B^{(1)}$ and

$B^{(2)}$, with the penalty

$$\tilde{\Psi}_{\lambda,\rho}(B^{(1)}, B^{(2)}) = \lambda \left(\sum_{i=1}^{N} \sqrt{\sum_{j} \left(\left(B_{ij}^{(1)} \right)^2 + \left(B_{ji}^{(2)} \right)^2 \right)} + \rho \left(\|B^{(1)}\|_1 + \|B^{(2)}\|_1 \right) \right),$$

with $B^{(1)}, B^{(2)} \in \tilde{\mathcal{B}}$, and set the coefficients matrix to $B = \left(B^{(1)} + B^{(2)} \right)/2$. This formulation will again not directly select subnetworks as discussed in Remark 2.1.

Finally, for numerical stability we add an extra ridge penalty term $\frac{\gamma}{2}\|B\|_F^2 = \frac{\gamma}{2}\mathrm{Tr}\left(B^T B\right)$, with γ a small and fixed constant. There are several benefits of combining ridge and lasso penalties (see for example Zou and Hastie (2005)). The parameter γ can be potentially considered an additional tuning parameter, but here we only use a small fixed constant γ in order to avoid numerically degenerate solutions. In practice, the results are not sensitive to the exact value of γ.

Putting everything together, to fit our graph classifier, we solve the problem

$$\begin{aligned} (\hat{B}, \hat{b}) \;=\; & \underset{B \in \mathcal{B}, b \in \mathbb{R}}{\arg\min} \left\{ \frac{1}{n} \sum_{k=1}^{n} \log \left(1 + \exp(-Y_k \langle B, A^{(k)} \rangle + b \right) + \frac{\gamma}{2} \|B\|_F^2 \right. \\ & \left. + \lambda \left(\sum_{i=1}^{N} \|B_{(i)}\|_2 + \rho \|B\|_1 \right) \right\} \end{aligned} \tag{2.4}$$

for given values of λ and ρ, which will be chosen by cross-validation.

2.3 The optimization algorithm

Our optimization algorithm to solve the problem (2.4) combines two common approaches to convex optimization: proximal algorithms and alternating direction method of multipliers (ADMM). We use an accelerated version of the proximal algorithm (Beck and Teboulle, 2009) to solve the main problem (2.4). In each step, we need to calculate a proximal operator, which is a further convex optimization problem solved with the ADMM algorithm.

The main optimization difficulty comes from the overlapping groups. Some algorithms have been proposed for this case, including a subgradient descent method (Duchi and Singer, 2009), which has a slow rate of convergence, or a proximal algorithms based on smoothing the original problem (Yuan et al., 2011; Chen et al., 2012). Although smoothing yields fast algorithms, it is not clear that the sparsity pattern is preserved with those approximations. We follow an approach similar to Yuan et al. (2011) and Chen et al. (2012), but solve the proximal operator for the

penalty (2.2) directly using the ADMM method. This can potentially give a more accurate sparsity pattern, and the flexibility of the algorithm allows for additional penalties if desired, such as spatial smoothing similar to Watanabe et al. (2014) (see Remark 2.3).

The main problem (2.1) is solved with a proximal algorithm (see Parikh and Boyd (2013)). Recall that the proximal operator for a function f is defined as $\text{prox}_f(v) = \arg\min_x\{f(x) + \frac{1}{2}\|x - v\|_2^2\}$. Starting with an initial value $B^{(0)} \in \mathbb{R}^{N\times N}$, a proximal algorithm solves the optimization problem (2.1) by iteratively calculating the proximal operator of $\Omega = \Omega_{\lambda,\rho}$ for a descent direction of the differentiable loss function ℓ. We use an accelerated version of the algorithm (Beck and Teboulle, 2009), which for each $k = 2, \ldots,$ until convergence, performs the updates

$$\begin{aligned} W^{(k)} &= B^{(k-1)} + \frac{k-1}{k+2}\left(B^{(k-1)} - B^{(k-2)}\right) && (2.5) \\ B^{(k)} &= \text{prox}_{t^{(k)}\Omega}\left\{W^{(k)} - t^{(k)}\nabla\ell\left(W^{(k)}\right)\right\} \\ &= \underset{B\in\mathcal{B}}{\arg\min}\left\{\frac{1}{2}\left\|B - \left(W^{(k)} - t^{(k)}\nabla\ell(W^{(k)})\right)\right\|_2^2 + t^{(k)}\Omega(B)\right\}, && (2.6) \end{aligned}$$

where $\nabla\ell(W) \in \mathbb{R}^{N\times N}$ is the gradient of the loss function ℓ at W and $t^{(k)}, k = 1, 2, \ldots$ is a sequence of positive values. If $\nabla\ell$ is Lipschitz continuous, with L its Lipschitz constant, the sequence of values $\ell\left(B^{(k)}\right) + \Omega\left(B^{(k)}\right)$ converges to the optimal value at rate $O(1/k^2)$ if $t^{(k)} \in [0, 1/L)$. The value of $t^{(k)}$ can be chosen using a backtracking search (Beck and Teboulle, 2009), which decreases this value until the condition

$$\ell\left(B^{(k)}\right) \le \ell\left(W^{(k)}\right) + \left\langle\nabla\ell\left(W^{(k)}\right), B^{(k)} - W^{(k)}\right\rangle + \frac{1}{2t^{(k)}}\|B^{(k)} - W^{(k)}\|_2^2 \qquad (2.7)$$

is satisfied. This procedure ensures that step sizes $\{t^{(k)}\}$ become smaller as the algorithm progresses, until $t^{(k)} < 1/L$. In practice, L might be large, which can make the algorithm slow to converge. It has been observed in other sparse high-dimensional problems that search strategies for $t^{(k)}$ which allow for $t^{(k)} > 1/L$ when appropriate can actually speed up convergence (Scheinberg et al., 2014; Hastie et al., 2015). We use a strategy of this type, allowing $t^{(k)}$ to increase by a factor of $\alpha \ge 1$ if the relative improvement in the loss function on iteration k becomes small. We observed that this strategy significantly reduces the number of iterations until convergence. The entire procedure is summarized in Algorithm 2.1.

The logistic loss function of (2.4) has an extra parameter b. Rather than including it as an unpenalized coefficient for a constant covariate, we use block coordinate

Algorithm 2.1 Proximal algorithm for fitting graph classifier

Input: Training sample $\{(A^{(1)}, Y_1), \ldots, (A^{(n)}, Y_n)\}$; regularization parameters λ, ρ; step size constants $\alpha \geq 1, \delta \in (0,1), \eta > 0$; tolerance $\epsilon^{\text{PROX}} > 0$.

Initialize: Starting values $B^{(0)}$, $t^{(0)}$.

Iterate: for $k = 1, 2, \ldots$ until $\epsilon^{(k)} < \epsilon^{\text{PROX}}$)

1. Compute $W^{(k)}$ according to (2.5).
2. Compute $B^{(k)}$ by solving the proximal operator (2.6).
3. If condition (2.7) does not hold, decrease step size $t^{(k)} \leftarrow \delta t^{(k)}$ and return to 2.
4. Calculate loss improvement
$$\epsilon^{(k)} = \left(\ell\left(B^{(k-1)}\right) + \Omega\left(B^{(k-1)}\right)\right) - \left(\ell\left(B^{(k)}\right) + \Omega\left(B^{(k)}\right)\right).$$
5. If $|\epsilon^{(k)} - \epsilon^{(k-1)}|/\epsilon^{(k)} < \eta$, increase step size $t^{(k+1)} = \alpha t^{(k)}$, otherwise set $t^{(k+1)} = t^{(k)}$.

Output: $\hat{B} = B^{(k)}$.

descent and solve for b separately. This is convenient because the threshold b and the matrix of coefficients B may not be on the same scale. Thus, b can be updated by solving $b^{(k+1)} = \arg\min_{b \in \mathbb{R}} \ell\left(B^{(k)}, b\right)$, which is easy to compute via Netwon's method.

The proximal algorithm requires solving the proximal operator (2.6), which has no closed form solution for the penalty (2.2) under the symmetry constraint. Strategies based on smoothing this penalty have been proposed (Yuan et al., 2011; Chen et al., 2012). However, to allow for variable selection which results from non-differentiability of the penalty, we aim to solve the proximal operator directly using ADMM (see Boyd et al. (2011) for a review). Note that if the symmetric constraint is relaxed as in Remark 2.1, the proximal operator has a closed form solution (see Remark 2.4).

The ADMM works by introducing additional constraints and performing coordinate descent in the corresponding augmented Lagrangian function. Setting $Z = W^{(k)} - t^{(k)}\nabla\ell(W^{(k)})$ and $t = t^{(k)}$, and introducing the variables $Q, R \in \mathbb{R}^{N \times N}$, we can formulate (2.6) as a convex optimization problem

$$\begin{aligned} \min_{\tilde{B}, Q, R} \quad & \frac{1}{2}\|\tilde{B} - Z\|_2^2 + t\lambda\left(\sum_{i=1}^{N}\|Q_{(i)}\|_2 + \rho\|R\|_1\right) \\ \text{subject to} \quad & \tilde{B} = Q, \quad \tilde{B} = R, \quad \tilde{B} = \tilde{B}^T, \quad \text{diag}(\tilde{B}) = 0. \end{aligned} \tag{2.8}$$

The ADMM algorithm introduces the multipliers $U, V \in \mathbb{R}^{N \times N}$ and a penalty param-

eter $\mu > 0$ to perform gradient descent on the Lagrangian of (2.8), given by

$$\begin{aligned}\mathcal{L}_\mu(\tilde{B}, Q, R, U, V) &= \frac{1}{2}\left\|\tilde{B} - Z\right\|_2^2 + t\lambda\left(\sum_{i=1}^{N}\|Q_{(i)}\|_2 + \rho\|R\|_1\right) \\ &\quad + \left\langle U, \tilde{B} - Q\right\rangle + \left\langle V, \tilde{B} - R\right\rangle \\ &\quad + \frac{\mu}{2}\left(\left\|\tilde{B} - Q\right\|_2^2 + \left\|\tilde{B} - R\right\|_2^2 + \left\|\tilde{B} - R\right\|_2^2 + \left\|\tilde{B} - \tilde{B}^T\right\|_2^2\right) \quad (2.9)\end{aligned}$$

The value μ controls the gap between dual and primal feasibility. In practice, we observed that setting $\mu = 0.1$ gives a good balance between primal and dual feasibility, although other self-tuning methods are available (Parikh and Boyd, 2013). This function is optimized by coordinate descent, with each variable updated to minimize the value of $\mathcal{L}_\mu$ while all the other variables are fixed. This update has a closed form shown in Algorithm 2.2. These steps are performed until the algorithm converges within tolerance $\epsilon^{\text{ADMM}} > 0$. Note that ADMM will be performed in each iteration of the algorithm to solve (2.4) and thus tolerance ϵ^{ADMM} can be decreased as the algorithm progresses. On the other hand, performing only one iteration of algorithm (2.2) gives a similar algorithm to the one of Chen et al. (2012).

Remark 2.3. The ADMM makes it very easy to incorporate additional penalties. If Ψ is an new penalty, we can rewrite (2.8) by introducing an additional parameter $\tilde{Q}$ so it becomes

$$\begin{aligned}\min_{\tilde{B},Q,\tilde{Q},R} \quad & \frac{1}{2}\|\tilde{B} - Z^{(k)}\|_2^2 + t\lambda\left(\sum_{i=1}^{N}\|Q_{(i)}\|_2 + \rho\|R\|_1\right) + t\Psi(\tilde{Q}) \\ \text{subject to} \quad & \tilde{B} = Q, \quad \tilde{B} = \tilde{Q}, \quad \tilde{B} = R, \quad \tilde{B} = \tilde{B}^T, \quad \text{diag}(\tilde{B}) = 0.\end{aligned}$$

We can obtain the Lagrangian formulation (2.9) in a similar manner, and include new parameters in the ADMM updates, which can be performed efficiently as long as the proximal operator of Ψ has a closed form solution. This is in fact the case for some other penalties of interest, such as the GraphNet penalty (Grosenick et al., 2013; Watanabe et al., 2014), which can incorporate spatial location information.

Remark 2.4. The alternative formulation for the graph penalty given in Remark 2.1 corresponds to standard sparse group lasso (Friedman et al., 2010). In particular, we can still employ the proximal algorithms (2.5) and (2.6), but instead optimize over the set $\tilde{\mathcal{B}}$. Without the symmetric constraint on B, the overlap in the group lasso penalty dissapears, and this vastly simplifies the problem. Using Theorem 1 of Yuan

Algorithm 2.2 Proximal operator of the graph classifier by ADMM

Input: Z, ϵ^{ADMM}, μ.

Initialize: $\tilde{B}^{(0)} = Z$, $R^{(0)} = Z$, $Q^{(0)} = Z$, $U^{(0)} = 0_{N\times N}$, $V^{(0)} = 0_{N\times N}$.

Iterate: for $l = 1, 2, \ldots$ until convergence ($\epsilon^{(l)}_{\text{ADMM-p}} < \epsilon^{\text{ADMM}}$ and $\epsilon^{(l)}_{\text{ADMM-d}} < \epsilon^{\text{ADMM}}$)

1. Perform coordinate gradient descent on (2.9) by computing

$$\begin{aligned}
\tilde{B}^{(l)} &= \frac{1}{1+2\mu}\left(Z + \frac{1}{2}\mu\left(Q^{(l-1)} + Q^{(l-1)^T}\right) + \mu R^{(l-1)} - U^{(l-1)} - V^{(l-1)}\right) \\
Q^{(l)}_{(i)} &= \left(1 - \frac{t\lambda}{\mu\left\|\tilde{B}^{(l)}_{(i)} + \frac{1}{\mu}U^{(l-1)}_{(i)}\right\|_2}\right)_+ \left(\tilde{B}^{(l)}_{(i)} + \frac{1}{\mu}U^{(l-1)}_{(i)}\right), \qquad i = 1, \ldots, N \\
R^{(l)}_{ij} &= \left(1 - \frac{t\lambda\rho}{\mu\left|\tilde{B}^{(l)}_{ij} + \frac{1}{\mu}V^{(l-1)}_{ij}\right|}\right)_+ \left(\tilde{B}^{(l)}_{ij} + \frac{1}{\mu}V^{(l-1)}_{ij}\right), \quad i, j = 1, \ldots, N \\
U^{(l)} &= U^{(l-1)} + \mu\left(\tilde{B}^{(l)} - \frac{1}{2}\left(Q^{(l)} + Q^{(l)^T}\right)\right) \\
V^{(l)} &= V^{(l-1)} + \mu\left(\tilde{B}^{(l)} - R^{(l)}\right)
\end{aligned}$$

2. Update primal and dual residuals $\epsilon^{(l)}_{\text{ADMM-p}}$ and $\epsilon^{(l)}_{\text{ADMM-d}}$

$$\begin{aligned}
\epsilon^{(l)}_{\text{ADMM-p}} &= \mu\left(\|Q^{(l)} - Q^{(l-1)}\|_\infty + \|R^{(l)} - R^{(l-1)}\|_\infty\right), \\
\epsilon^{(l)}_{\text{ADMM-d}} &= \mu\left(\|\tilde{B}^{(l)} - Q^{(l)}\|_2 + \|\tilde{B}^{(l)} - R^{(l)}\|_2\right).
\end{aligned}$$

Output: $\tilde{B} = \tilde{B}^{(l)}$.

et al. (2011), the update for $B^{(k)}$ has a closed form solution given by

$$
\begin{aligned}
Y^{(k)} &= B^{(k-1)} + \frac{k-2}{k}\left(B^{(k-1)} - B^{(k-2)}\right) && (2.10)\\
Z_{ij}^{(k)} &= \left(1 - \frac{\lambda\rho}{\left\|Y_{ij}^{(k)} - t_k \nabla_{ij}\ell(Y^{(k)})\right\|_2}\right)_+ \left(Y_{ij}^{(k)} - t_k \nabla_{ij}\ell(Y^{(k)})\right) && (2.11)\\
B_{(i)}^{(k)} &= \left(1 - \frac{\lambda}{\left\|Z_{(i)}^{(k)}\right\|_2}\right)_+ \left(Z_{(i)}^{(k)}\right), \qquad i = 1, \ldots, N. && (2.12)
\end{aligned}
$$

2.4 Theory

In this section, we show that the solution of the penalized problem (2.2) can recover the correct subgraph corresponding to the set of non-zero coefficients, and give its rates of convergence. The theory is a consequence of the results of Lee et al. (2015) for establishing model selection consistency of regularized M-estimators under geometric decomposability (see Appendix for details). We present explicit conditions for our penalty to work well, which depend on the data as well as the tuning parameters.

Let $B^\star \subset \mathbb{R}^{N\times N}$ be the unknown parameter we seek to estimate, and we assume there is a set of active nodes $\mathcal{G} \subset \{1, \ldots, N\}$ with $|\mathcal{G}| = G$, so that $B_{ij}^\star = 0$ if $i \in \mathcal{G}^C$ or $j \in \mathcal{G}^C$. We allow some edge weights inside the subgraph defined by $\mathcal{G}$ to be zero, but we focus on whether the set $\mathcal{G}$ is correctly estimated by the set $\hat{\mathcal{G}}$ of active nodes in $\hat{B}$. Denote by $\mathcal{M} \subseteq \mathbb{R}^{N\times N}$ the set of matrices where the only non-zero coefficients appear in the active subgraph, that is,

$$
\mathcal{M} = \left\{B \in \mathbb{R}^{N\times N} \,\middle|\, B_{ij} = 0 \text{ for all } i \in \mathcal{G}^C \text{ or } j \in \mathcal{G}^C, B = B^T\right\} \tag{2.13}
$$

There are two main assumptions on the loss function ℓ required for consistent model selection in high-dimensional models (Lee et al., 2015). The first assumption is on the convexity of the loss function around $B^\star$, while the second assumption bounds the size of the entries in the loss Hessian between the variables in the active subgraph and the rest. Let the loss Hessian $\nabla^2\ell(B^\star) \in \mathbb{R}^{N\times N} \otimes \mathbb{R}^{N\times N}$ be defined by

$$
\nabla^2_{(i,j),(k,l)}\ell(B) = \frac{\partial^2\ell(B)}{\partial B_{ij}\partial B_{kl}},
$$

and define the matrix $H_{(i,j),\mathcal{G}} \in \mathbb{R}^{G\times G}$ with $(i,j) \in (\mathcal{G}\times\mathcal{G})^C$ such that

$$\left(H_{(i,j),\mathcal{G}}\right)_{k,l} = \mathrm{Tr}\left(\left(\nabla^2_{(i,j),(\mathcal{G},\mathcal{G})}\ell(B^\star)\right)\Lambda_{(k,l),(\cdot,\cdot)}\right), \qquad 1 \le k,l \le G,, \tag{2.14}$$

where $\Lambda \in \mathbb{R}^{G\times G}\otimes\mathbb{R}^{G\times G}$ is a tensor such that $\mathrm{Mat}(\Lambda)$ is a pseudoinverse of $\mathrm{Mat}\left(\nabla^2_{(\mathcal{G},\mathcal{G}),(\mathcal{G},\mathcal{G})}\ell(B^\star)\right)$, and Mat is the operation that unfolds the entries of a tensor Λ into a $G^2\times G^2$ matrix. The matrix $H_{(i,j),\mathcal{G}}$ measures how well the variable corresponding to the edge (i,j) can be represented by the variables in the active subgraph.

Assumption 2.1 (Restricted Strong Convexity)**.** *There exists a set $C \subset \mathbb{R}^{N\times N}$ with $B^\star \in C$, and constants $m > 0, \tilde{L} < \infty$ such that*

$$\begin{aligned}\sum_{i,j}\Delta_{i,j}\,Tr\left(\left(\nabla^2_{(i,j),(\cdot,\cdot)}\ell(B)\right)\Delta\right) &\ge m\|\Delta\|_2^2, \quad \textit{for all } B\in C\cap\mathcal{M}, \Delta\in C\cap\mathcal{M}\\ \|\nabla^2\ell(B)-\nabla^2\ell(B^\star)\|_2 &\le \tilde{L}\|B-B^\star\|_2, \quad \textit{for all } B\in C.\end{aligned}$$

Assumption 2.2 (Irrepresentability)**.** *There exists a constant $0 < \tau < 1$ such that*

$$\max_{i\in\mathcal{G}^C}\left\|\left(\sum_{k=1}^{G}\|(H_{(i,j),\mathcal{G}})_{k\cdot}\|_2\right)_{j=1}^{N}\right\|_2 = 1-\tau < 0.$$

This version of the irrepresentability condition corresponds to the one usually employed in group lasso penalties (Bach, 2008), but as we will see later, due to overlaps in the groups it further requires a lower bound on ρ to work for model selection.

The first two assumptions are stated directly as a function of the loss for a fixed design case, but they can be substituted with bounds in probability for the case of random designs. In order to obtain rates of convergence, we do require a distributional assumption on the first derivative of the loss. This assumption can be substituted with any bound on $\max_i \|\nabla\ell(B^\star)_{(i)}\|_2$ (see Lemma 3 in the Appendix).

Assumption 2.3 (Sub-Gaussian score function)**.** *Each pair in the sample (A,Y) is independent and comes from a distribution such that the entries of the matrix $\nabla\tilde{\ell}(Y,A;B^\star)$ are subgaussian. That is, for all $t > 0$ there is a constant $\sigma^2 > 0$ such that*

$$\max_{i,j}\mathbb{P}\left(\|\nabla_{ij}\tilde{\ell}\,(Y,A;B^\star)\,\|_\infty > t\right) \le 2\exp\left(-\frac{t^2}{\sigma^2}\right)\ .$$

With these assumptions, we establish consistency and correct model selection. The proof is given in the Appendix A.1.

Proposition 2.1. *Suppose Assumptions 2.1 and 2.3 hold.*

(a) Setting the penalty parameters as $\rho \geq 0$ *and* $\lambda = c_1\sqrt{\sigma^2\frac{\log N}{n}}\min(\sqrt{N}/(1+\rho), 1/\rho)$ *for some constant* $c_1 > 0$*, with probability at least* $1 - 2/N$ *the optimal solution of* (2.4) *is unique and satisfies*

$$\frac{1}{N^2}\|\hat{B} - B^\star\|_2^2 = O_P\left(\frac{\sigma^2 \log N}{n}\right). \tag{2.15}$$

(b) Suppose Assumption 2.2 also holds. If $n > c_2 G^2\sigma^2 \log N$ *for a constant* $c_2 > 0$*, setting the penalty parameters as* $\lambda = c_3\sqrt{\sigma^2\frac{\log N}{n}}\min(\sqrt{N}/(1+\rho), 1/\rho)$ *for some constant* $c_3 > 0$*, and*

$$\rho > \frac{1}{\tau} - \frac{1}{\sqrt{G}}\ , \tag{2.16}$$

then

$$\|\hat{B} - B^\star\|_2^2 = O_P\left(\frac{G^2 \log N}{n}\right), \tag{2.17}$$

$$\mathbb{P}\left(\hat{\mathcal{G}} \subseteq \mathcal{G}\right) = 1 - 2/N. \tag{2.18}$$

The part of the penalty associated with ρ causes the solution to be sparse. Due to the overlap in the groups, a small value of ρ will not result in zeros in the solution of the problem (2.4). The lower bound on ρ in (2.16) ensures that the irrepresentability condition of Lee et al. (2015) holds (see Lemma 2 in the Appendix).

The result (2.18) ensures that, with high probability, all edges estimated to have non-zero weights are contained in the active subgraph. To ensure that all active nodes are recovered, at least one edge corresponding to each active node needs to have a non-zero weight. A result similar to (2.18) can be obtained to guarantee recovery of all active nodes under a stronger assumption that the magnitude of the non-zero entries of $B^\star$ is bounded below by $|B^\star_{ij}| > c_4 G^2\lambda$ for a constant c_4.

2.5 Numerical results on simulated networks

In this section, we evaluate the performance of our method using synthetic networks. We are interested in assessing both the ability of the method to correctly identify

predictive edges and its classification accuracy, and comparisons to benchmarks. We compare the different methods' edge selection performace in simulations using area under the curve (AUC).

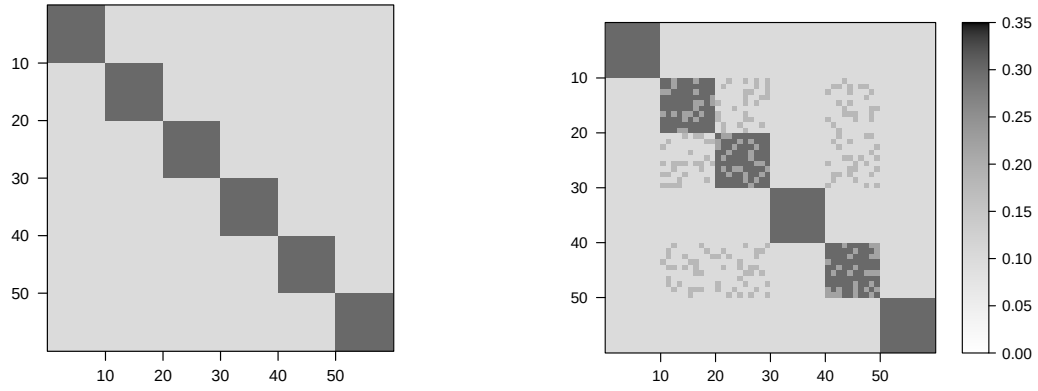
Brain connectomic networks are characterized by organization of nodes into communities (Bullmore and Sporns, 2009), in which nodes within the same community tends to have stronger connections than nodes belonging to different communities. In order to generate synthetic networks that mimic this property, we introduce community structure using the stochastic block model (SBM) (Holland et al., 1983). Before generating edges, we assign each node a community label, C_i, where $C_i \in \{1, \ldots, K\}$ for each $i = 1, \ldots, N$. The node assignments are the same for all networks in the population. Given the community labels, network edges are generated independently from a distribution that only depends on the community labels of the nodes associated with each edge. Since fMRI networks are real-valued networks, we generate edge weights from a Gaussian distribution, rather than the standard Bernoulli distribution normally used with the SBM. Specifically, we draw each A_{ij} independently from $N(\mu_{C_i C_j}, \sigma^2)$, with $\mu \in \mathbb{R}^{K \times K}$ defined by

$$\mu_{kl} = \begin{cases} 0.3, & \text{if } k = l, \\ 0.1 & \text{if } k \neq l, \end{cases}$$

and $\sigma^2 = 0.2$. These values were chosen to approximately match the distribution of edge weights in our datasets (see Section 2.6). We set the number of nodes $N = 60$, with $K = 6$ communities of size 10 each. We work with undirected networks, so the adjacency matrices are symmetric, with 1770 distinct edges.

To set up two different class distributions, we alter a set of edges selected at random. To construct this set of edges, we first select a number of communities (in our experiments, we use 2 and 3), and the set $\mathcal{G}$ of active nodes corresponds to the nodes on those communities. Then, with probability p, each edge in $\mathcal{G}$, is selected to belong to the set of differentiating edges $\mathcal{E}$, with weights sampled from $N(0.2, 0.2)$. Figure 2.3 shows example expected adjacency matrices for each class. We vary the size of the set $\mathcal{G}$ and the value of p to study the effect of the number of active nodes and the density of differentiating edges inside a subgraph. We then gen50 networks independently from each class, giving the total sample size $n = 100$.

Since we are interested is identifying predictive edges and nodes, we use the area under the curve (AUC) of the receiver operating characteristic (ROC) curve, for both edge and node selection. For each method, we calculate the ROC curve by varying



Class $Y = -1$ Class $Y = 1$

Figure 2.3: Expected adjacency matrices for each class. The second class ($Y = 1$) has altered edge weights on a subset of edges within the set of active nodes $\mathcal{G}$. Within the subgraph defined by $\mathcal{G}$, edge weights have been altered with probability 0.75.

the number of edges selected by changing its corresponding sparsity parameter. For a selection method $\mathcal{M}$ and a sparsity parameter η let $\hat{\mathcal{E}}(\mathcal{M}, \eta)$ be the set of edges selected by $\mathcal{M}$, and $\hat{\mathcal{G}}(\mathcal{M}, \eta)$ the set of active nodes corresponding to $\hat{\mathcal{E}}(\mathcal{M}, \eta)$. We calculate the edge false positive rate (EFPR) and edge true positive rate (ETPR) as

$$\text{EFPR}(\mathcal{M}, \eta) = \frac{\left|\hat{\mathcal{E}}(\mathcal{M}, \eta) \cap \mathcal{E}^C\right|}{|\mathcal{E}^C|}, \quad \text{ETPR}(\mathcal{M}, \eta) = \frac{\left|\hat{\mathcal{E}}(\mathcal{M}, \eta) \cap \mathcal{E}\right|}{|\mathcal{E}|}.$$

The node FPR and TPR are calculated similarly.

We also evaluate the prediction accuracy of the methods. For each method, we use 10-fold cross-validation to select the best tuning parameter using the training data, and then compute the test error on a different dataset simulated under the same settings. The AUC and test errors reported are averaged over 50 replications.

Methods for benchmark comparisons on simulated networks were selected based on their good performance on real data (see Section 2.6). For our method (GC), we vary the parameter ρ and compare results for two different values of λ, .05 (GC1) and 10^{-4} (GC2). For unstructured regularized logistic regression, we use the elastic net (Friedman et al., 2009), with a fixed $\alpha = 0.02$ (ENet). The performance of elastic net is not very sensitive to different values of α, but the number of variables that the method is able to select with large values is limited (including the case of $\alpha = 1$ that corresponds to the Lasso). A support vector machine with ℓ_1 penalty (Zhu et al., 2004; Becker et al., 2009) is also included (SVM) for comparison, and additionally

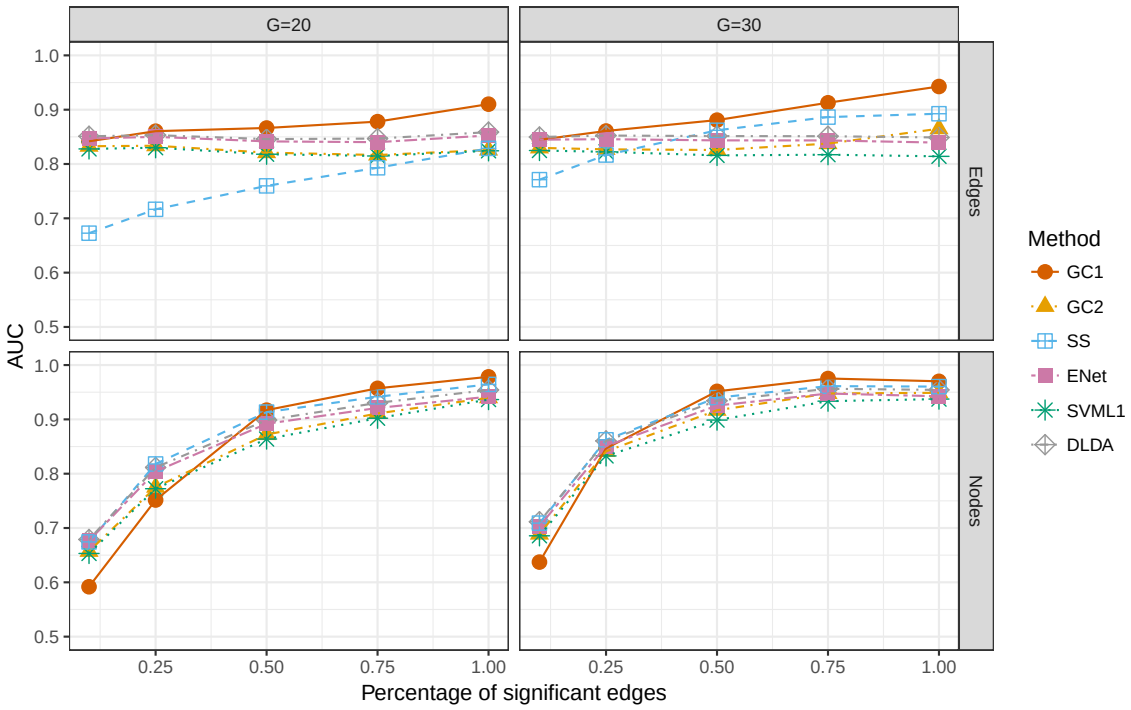


Figure 2.4: Variable selection performance of different methods in terms of edge AUC (top) and node AUC (bottom) as a function of the fraction of differentiating edges in the subgraph induced by the active node set G. Left: $|\mathcal{G}| = 20$; right: $|\mathcal{G}| = 30$ active nodes.

we evaluate the classification error of the original support vector machines (SVML2) (Cortes and Vapnik, 1995). For both SVMs, we use linear kernels, which performed better than nonlinear kernels. Diagonal linear discriminant analysis (DLDA) is also considered, with variables selected via independent screening using a t-test. Finally, we also compare with the signal-subgraph method (SS) (Vogelstein et al., 2013), the only other method that takes into account the network structure of the predictor variables. Note that the signal subgraph is designed for binary networks, so in order to apply it we thresholded the edges at the population median.

Figure 2.4 shows the values of the average AUC for selecting edges (top) and nodes (bottom). As the fraction of differentiating edges in the active node subgraph increases, methods that take into account network structure improve their edge AUC, since predictive nodes carry more information, while the edge AUC remains constant for unstructured methods (ENet, DLDA and SVML1). On node selection, all methods improve as the fraction of significant edges increases, but GC and SS have the largest gains. A similar trend is observed in classification error shown in Figure 2.5. All methods improve as the number of differentiating edges increases, but our method

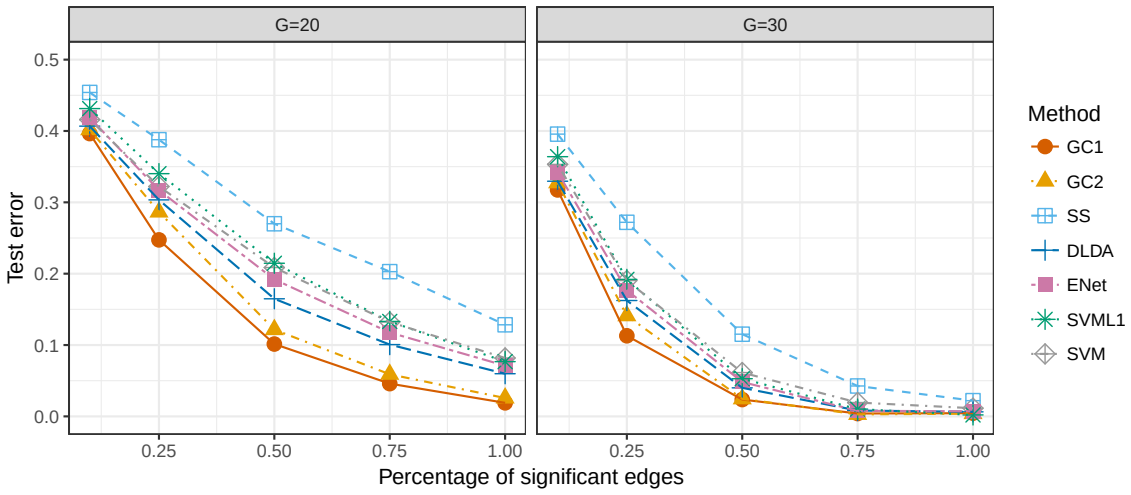


Figure 2.5: Classification error of different methods as a function of the fraction of differentiating edges in the subgraph induced by the active node set G. Left: $|\mathcal{G}| = 20$; right: $|\mathcal{G}| = 30$ active nodes.

has the best performance overall. Our method performed the best with the larger value of λ(GC1) on variable selection, particularly when the set of active nodes is smaller, but both values of λ give very good classification performance.

2.6 Application to schizophrenia data

We analyze the performance of the classifier on two different brain fMRI datasets, each containing schizophrenic patients and controls. The first dataset comes from the Center for Biomedical Research Excellence (COBRE). The second dataset, which we refer to as UMich data, is from the lab of Professor Stephan F. Taylor in the Department of Psychiatry at the University of Michigan. The code of our classifier and the processed connectomics datasets can be found at https://github.com/jesusdaniel/graphclass.

2.6.1 Subjects and imaging

The COBRE data

Raw anatomic and functional scans from 146 subjects (72 psychosis patients and 74 healthy control subjects) were downloaded from a public database (http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html). Four subjects coded as ambidextrous (2 patients, 2 controls) were excluded to yield 70 psychosis patients and 72

controls for analysis. To enter the COBRE dataset, subjects had a diagnosis of either schizophrenia or schizoaffective disorder and were without histories of neurological disorder, mental retardation, severe head trauma with more than 5 minutes loss of consciousness and substance abuse/dependence within the last 12 months.

In the primary sample, two schizophrenic (SCZ) subjects and one healthy control (HC) subject had insufficient voxels in the cerebrospinal fluid (CSF) segmentation on the CSF, and they were dropped from additional analyses. Two additional SCZ subjects were excluded for scrub ratios (see discussion of *scrubbing routine* in fMRI Data Analysis) greater than 0.6, leaving 38 SCZ subjects and 42 HC subjects for the analysis. In the replication sample, 15 psychosis patients and two control subjects were excluded for scrub ratios greater than 0.6; one patient was excluded with incomplete data, leaving 54 SCZ and 69 HC subjects for analysis (see Table 2.1).

A full description of the imaging parameters for the COBRE dataset is available online at the link provided above and in several related papers, see Calhoun et al. (2011); Hanlon et al. (2011); Mayer et al. (2013); Stephen et al. (2013).

The UMich data

Subjects were selected from experiments conducted by Professor Stephan F. Taylor at the University of Michigan between 2004 and 2011 for task-based fMRI studies, which included resting state scans. Forty-two stable outpatients were selected with DSM-IV schizophrenia or schizoaffective disorder (SCZ) (Association et al., 1994). Forty-three healthy comparison (HC) subjects, without a lifetime history of Axis I psychiatric disorders (First et al., 1995), were selected to approximate the age range, gender distribution and family education level of the patients. Prior to initial data collection, all subjects gave written, informed consent to participate in the protocol approved by the University of Michigan institutional review board (IRBMED).

MRI scanning occurred on a GE 3T Signa scanner (LX [8.3] release, General Electric Healthcare, Buckinghamshire, United Kingdom). Functional images were acquired with a T2*-weighted, reverse spiral acquisition sequence (gradient recalled echo, TE=30 msec, FA=90 degrees, field of view=22 cm, 40 slices, 3.0mm thick/0mm skip, equivalent to 64 x 64 voxel grid – yielding isotropic voxels 3 mm on edge). Because the data were acquired across different experiments, acquisition parameters differed slightly in the aggregate sample: 240 volumes @ TR=1500 msec (11 SCZ, 10 HC), 180 volumes @ TR=2000 msec (17 SCZ, 16 HC) and 240 volumes at 2000 msec (14 SCZ, 17 HC). Acquisitions were acquired in the resting state with eyes open and fixated on a large 'plus' sign projected on a monitor.

Dataset	# nodes	Status	# patients	Male/female	Average age (s.d.)
COBRE	263	Schizophrenic	54	48/6	35.4 (13.1)
		Control	70	48/22	35.1 (11.5)
UMich	264	Schizophrenic	39	29/10	40.7 (11.5)
		Control	40	28/12	36.8 (12.3)

Table 2.1: Summary statistics of the two datasets.

2.6.2 Pre-processing

We first performed standard pre-processing steps. All scans were slice-time corrected and realigned to the 10th image acquired during a scanning session (Jenkinson et al., 2002). Subsequent processing was performed with the Statistical Parametric Mapping SPM8 package (Wellcome Institute of Cognitive Neurology, London). Anatomic normalization was done with the VBM8 toolbox in SPM8, using the high resolution structural scans obtained for both datasets. Normalizing warps were applied to the co-registered, functional volumes, which were re-sliced and smoothed with an 8 mm isotropic Gaussian smoothing kernel. To assess and manage movement, we calculated the frame-wise displacement (FD) (Power et al., 2012), for all 6 parameters of rotation and translation. We used a *scrubbing routine* to censor any frame with FD > 0.5 mm from the regression analysis described below, yielding a scrub ratio for each subject. Three-compartment segmentation of the high-resolution structural image from the VBM8 normalization was applied to the functional time series to extract cerebral spinal volume (CSF) and white matter (WM) compartments, which were then subjected to a principal component analysis to identify the top 5 components in each (Behzadi et al., 2007), which should correspond to heart rate and respiratory effects on global signal (Chai et al., 2012). Multiple regressions were applied to the time series to remove the following nuisance effects: Linear trend, 6 motion parameters, their temporal derivatives, the quadratics of these 12 parameters, 5 components from the PCA of CSF, 5 components of PCA of WM, followed by band pass filtering from 0.01 – 0.1 Hz, and then motion scrubbing. For each 4D data set, time courses were then extracted from 10 mm diameter spheres based on the 264 sets of coordinates identified by Power et al. (2011). From these time series, a cross-correlation matrix of Pearson r-values was obtained and Fisher's R-to-Z transformation was applied for each of the 264 nodes with every other node (for COBRE dataset, node 75 is missing). Finally, for each individual, edge weights were assigned to be ranks of these score, with edge scores ranked separately for each subject, and then these values were centered and standardized across the individuals. Ranks have been used previously in brain connectomic studies to reduce the effect of potential outliers (Yan et al.,

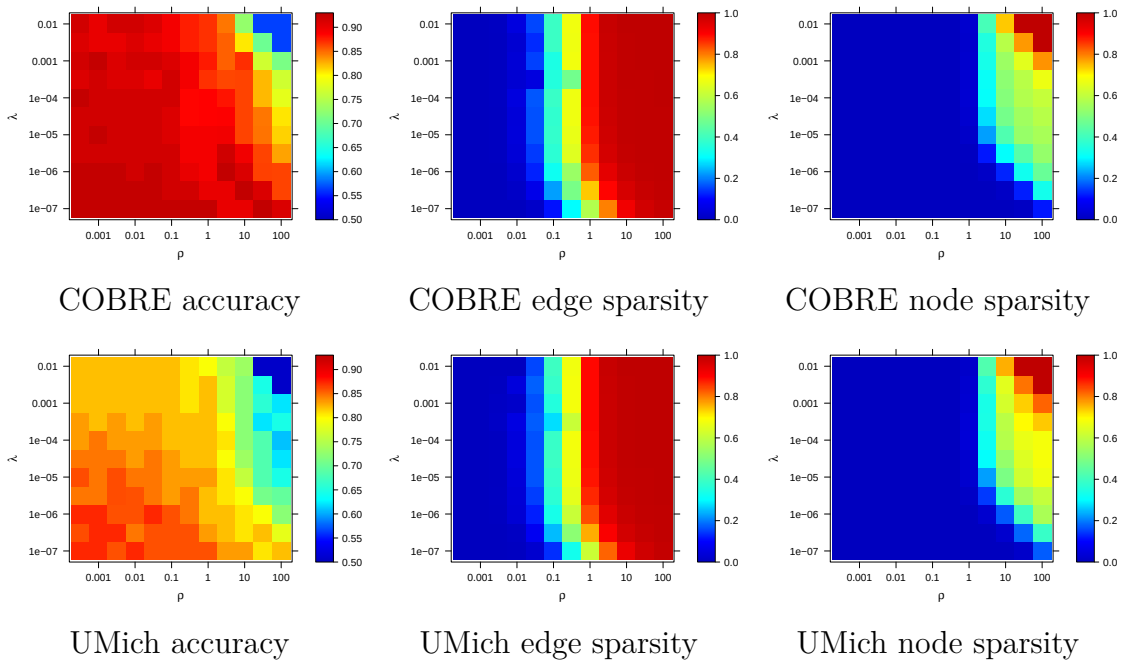
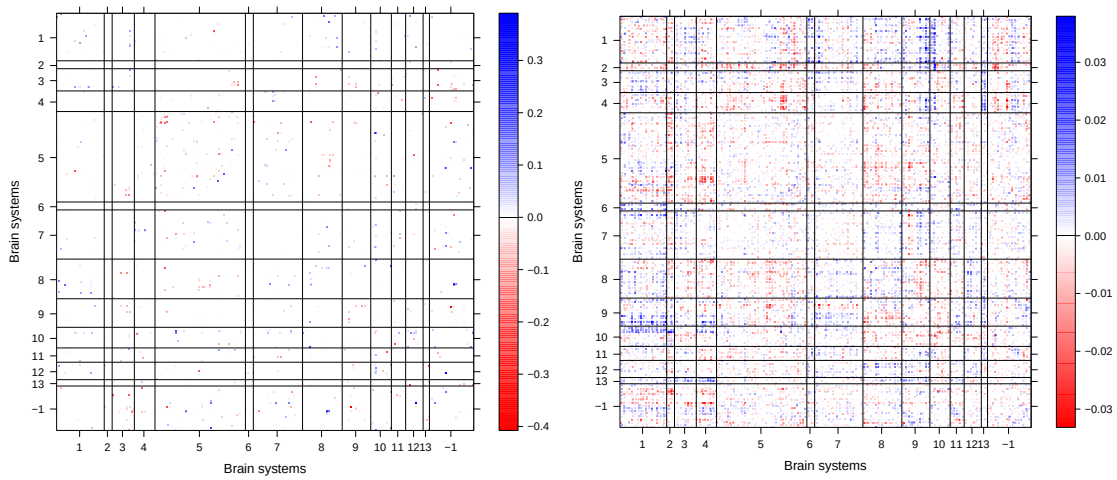


Figure 2.6: Cross-validated results for the two data sets. Classifcation accuracy (left), fraction of zero edge coefficients (middle), and fraction of inactive nodes (right).

2013); we observed that while ranks does not increase the classification accuracy significantly, they tend to produce sparser solutions with a similar accuracy to Pearson correlations.

2.6.3 Classification results

First, we evaluate our method's classification accuracy. We use a nested 10-fold cross-validation to choose tuning parameters and estimate the test accuracy. The classifier is trained for a range of values of λ and ρ, with $\lambda \in \{10^{-7}, 10^{-6.5}, \ldots, 10^{-2}\}$ and $\rho \in \{10^{-3}, 10^{-2.5}, \ldots, 10^{2}\}$. The value of γ in (2.4) is set to 10^{-5}; we observed that setting γ to a small value speeds up convergence without affecting the accuracy or sparsity of the solution. Figure 2.6 shows the average cross-validated accuracy, sparsity (fraction of zero coefficients) and node sparsity (fraction of inactive nodes), as a heat map over the grid of tuning parameter values. We observe that λ has little influence on sparsity, which is primarily controlled by ρ. Moreover, as Proposition 2.1 suggests, values of $\rho < 1$ do not result in node selection. As expected, accuracy generally decreases as the solution becomes sparser, which is not uncommon in high-dimensional settings (Hastie et al., 2015). However, we can still achieve excellent accuracy with a substantially reduced set of features. In the COBRE dataset, the



COBRE solution UMich solution

Figure 2.7: Fitted coefficients for COBRE and UMich datasets, with tuning parameters selected by the "one standard error rule". Positive coefficients corresponds to higher edge weights for schizophrenic patients.

best accuracy is obtained with only 1886 edges (5.4%) but almost all nodes are active (260). On the UMich data, 29733 edges (85.6%) achieve the best performance, and all nodes are active. Choosing parameters by cross-validation often tends to include too many noise variables (Meinshausen, 2007), as we also observed in simulations. A commonly used technique to report solutions that still achieve good accuracy with a substantially reduced set of features is the so-called "one-standard-error rule" (Hastie et al., 2015), in which one selects the most parsimonious classifier with cross-validation accuracy at most one standard error away from the best cross-validation accuracy. Figure 2.7 shows the solutions for each dataset obtained by this rule. Nodes are ordered by brain systems (see Figure 2.1). The fitted solution for COBRE has 549 non-zero coefficients (1.56%) and 217 active nodes (82.5%), while the UMich solution has 11748 non-zero entries (33.8%), and all nodes are active. Note that when many variables are selected, the magnitude of the coefficients becomes small due to the grouping effect of the penalty (Zou and Hastie, 2005).

We also compared our method to benchmarks (Table 2.2), using the same methods as in the previous section and training and evaluating all methods using with the same nested 10-fold cross-validation. For SVM, we tested different kernels, including graph aware kernels (Gärtner et al., 2003), but in most cases local kernel methods were no better than random guessing. We additionally included random forests and a method based on global and local network summaries previously proposed as features

for classifying brain data (Prasad et al., 2015). For the latter, because our dataset is much larger, we only considered global and node features proposed in Prasad et al. (2015), which resulted in about 30,000 features per individual, and omitted edge features. Watanabe et al. (2014) evaluated their classifiers on a different parcellation of the COBRE data, and we do not include their methods since they are based on the assumption of equally spaced nodes and cannot be directly applied to our data. Their reported accuracy of 71.9% and 73.5% for the COBRE data is substantially lower than our method, although the results are not directly comparable.

Results in Table 2.2 show that most methods performed better on the COBRE dataset than on the UMich dataset, which can be partially explained by the different sample sizes and possibly noise levels. Besides differences in sample size and demographic characteristics (Table 2.1), the COBRE dataset is more homogeneous as it was collected using identical acquisition parameters, whereas the UMich dataset was pooled across five different experiments spanning seven years.

Our method performs very well on both datasets, particularly among methods that perform variable selection. SVMs, which use the hinge loss, perform well too, and generally outperform methods using the logistic loss. Our penalty can be combined with any loss, so we could also include our penalty combined with hinge loss which might potentially improve classification accuracy, but we do not pursue this direction, for two reasons: one, our method is close to SVM + L1 as it is (better on COBRE, slightly worse on UMich but the difference is within noise levels), and because solutions based on logistic loss are generally considered more stable and preferable for variable selection Hastie et al. (2015). In Figure 2.8, we plot cross-validated classification accuracy of these methods as a function of the number of variables selected. For the COBRE data, as we have observed before, good accuracy can be achieved with a fairly small number of edges, and the noisier UMich data requires more edges. In all cases, our method uses fewer nodes than the others, as it is designed to do so.

Ultimately, assessing significance of the selected variables is necessary, which is in general a difficult task in high-dimensional settings and an active area of research (see for example Meinshausen and Bühlmann (2010); Van de Geer et al. (2014); Lockhart et al. (2014); Lee et al. (2016)). In brain connectomics, it is particularly challenging to identify significant variables because of small sample sizes (Button et al., 2013). Here we employ *stability selection* (Meinshausen and Bühlmann, 2010) which can be shown to control a type of false discovery rate by employing many rounds of random data splitting and calculating the probability of each variable being selected. Some versions of this method have been theoretically studied, and upper bounds on the

Method	Classification accuracy % (standard error) COBRE	UMich
With variable selection		
Our method (GC)	92.7 (2.6)	85.9 (3.6)
Elastic net	89.5 (1.8)	82.6 (4.7)
SVM-L1	87.9 (2.2)	86.2 (4.3)
Signal-subgraph	86.1 (3.3)	82.4 (3.3)
DLDA	84.6 (3.3)	73.4 (3.9)
Lasso	80.1 (5.6)	60.9 (5.6)
No variable selection		
SVM	93.5 (2.1)	89.8 (2.5)
Ridge penalty	91 (2.6)	80.9 (3.5)
Random forest	74.2 (2.6)	82.1 (3.9)
Network summaries	61.4 (3.1)	65 (7.2)

Table 2.2: Cross-validated accuracy (average and standard errors over 10 folds) for different methods.

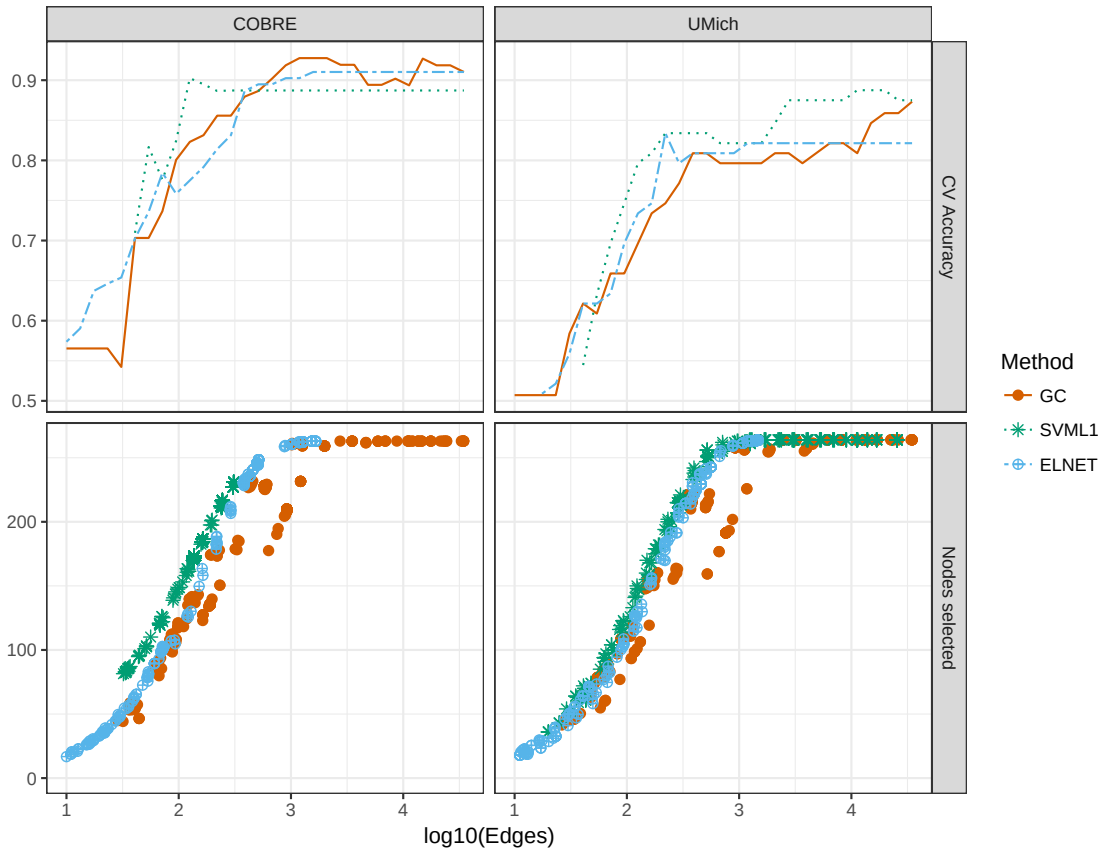


Figure 2.8: Cross-validated accuracy and number of nodes selected as a function of the number of edges used.

	COBRE				UMich		
	Edge	**Systems**	**Coefficient**		**Edge**	**Systems**	**Coefficient**
1	(208, 85)	(9, -1)	-0.187	1	(110, 207)	(5, 9)	-0.013
2	(260, 11)	(12, -1)	0.183	2	(255, 113)	(1, 5)	0.014
3	(194, 140)	(8, -1)	0.136	3	(33, 218)	(1, 9)	0.016
4	(52, 186)	(3, 8)	- 0.1	4	(46, 225)	(2, 10)	0.013
5	(160, 239)	(7, 11)	-0.082	5	(43, 90)	(2, 5)	-0.013
6	(120, 116)	(5, 5)	0.099	6	(23, 225)	(1, 10)	0.012
7	(57, 129)	(3, 5)	-0.128	7	(66, 118)	(4, 5)	-0.013
8	(24, 114)	(1, 5)	-0.148	8	(26, 145)	(1, 7)	0.013
9	(81, 179)	(5, 8)	-0.129	9	(186, 254)	(8, -1)	0.012
10	(193, 140)	(8, -1)	0.153	10	(15, 134)	(1, 6)	0.011
11	(178, 234)	(8, 10)	0.146	11	(76, 207)	(5, 9)	-0.012
12	(18, 194)	(1, 8)	0.116	12	(65, 84)	(4, -1)	-0.012
13	(215, 207)	(9, 9)	-0.076	13	(26, 122)	(1, 5)	0.012
14	(90, 224)	(5, 10)	0.123	14	(33, 145)	(1, 7)	0.012
15	(112, 253)	(5, -1)	0.136	15	(36, 224)	(1, 10)	0.011

Table 2.3: Edges with the top 15 largest selection probabilities from stability selection. The first column shows the pair of nodes making the edge, the second column the brain systems the nodes belong to in the Power parcellation, and the third column the fitted coefficient of the edge.

	Test data	
Training data	**COBRE**	**UMich**
COBRE	92.7 (2.6)	73.5 (3.4)
UMich	78.3 (3.0)	85.9 (3.6)

Table 2.4: Classification accuracy (cross-validation average and standard error) of the classifier fitted on one dataset and evaluated on the other. The intercept (the mean) is fitted on the test data and the accuracy is estimated using 10-fold cross-validation on the test data.

expected number of variables with a low selection probability that are included in the final solution (i.e., errors) have been derived under mild conditions (Meinshausen and Bühlmann, 2010; Shah and Samworth, 2013). We implemented the version of stability selection proposed by Shah and Samworth (2013), with values of λ and ρ obtained by cross-validation on the COBRE data, and by the "one standard error rule" on the UMich dataset, since stability selection is most relevant to sparse solutions. However, one of the advantages of stability selection is that it is not sensitive to the initial choice of tuning parameters, and changing tuning parameters only slightly alters the ordering of variables with the largest selection probabilities.

The edges with the 15 largest selection probabilities are reported in Table 2.3. Using the results of Shah and Samworth (2013) (equation 8), we estimated that the expected number of falsely selected variables (variables with a probability of selection

smaller than the estimated) is bounded by 6.1 for the COBRE dataset and 9.7 for the UMich data, which also suggests that results on the UMich data might be less reliable. While the two datasets yield somewhat different patterns of edge selection, it is notable that the default mode network (5) was often selected in both. This network has been consistently implicated in schizophrenia (Whitfield-Gabrieli et al., 2009; Öngür et al., 2010; Peeters et al., 2015), as well as other psychiatric disorders, possibly as a general marker of psychopathology (Broyd et al., 2009; Menon, 2011). In the COBRE dataset, edges were also selected from the fronto-parietal task control region (8), previously linked to schizophrenia (Bunney and Bunney, 2000; Fornito et al., 2012). These results coincide with the findings of Watanabe et al. (2014) on a different parcellation of the same data, which is an encouraging indication of robustness to the exact choice of node locations. Some of the variables with the highest estimated selection probabilities appear in the uncertain system (-1), in particular in the cell connecting it with salience system (9), which suggests that alternative parcellations that better characterize these regions may offer a better account of the schizophrenia-related changes. Additionally, sensory/somatomotor hand region (1) and salience system (9) also stand out in the UMich data, and these are networks that have also been implicated in schizophrenia (Dong et al., 2017).

While results in Table 2.3 do not fully coincide on the two datasets, there are clear commonalities. Table 2.4 compares classification accuracy when the classifier is trained on one dataset and tested on the other (with the exception of the intercept, since the datasets are not centered in the same way, which is fitted on a part of the test data, and the test error is then computed via 10-fold cross-validation). While the accuracy is lower than when the same dataset is used for training and testing, as one would expect, it is still reasonably good and in fact better than some of the benchmark methods even when they train and test on the same data. We again observe that the COBRE dataset is easier to classify.

Figure 2.9 shows the active nodes in the COBRE dataset (marked in green), corresponding to the endpoints of the edges listed in Table 2.3. We also identified a set of 25 nodes that are not selected in any of the sparse solutions with cross-validation accuracy within one standard error from the best solution (marked in purple). These consistently inactive nodes are mostly clustered in two anatomically coherent regions.

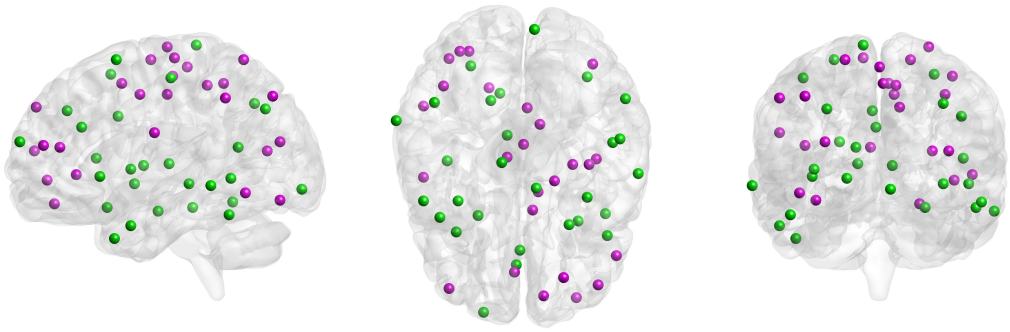


Figure 2.9: Nodes shown in green are endpoints of edges selected by stability selection shown in Table 2.3. Node shown in purple are nodes not selected by any of the sparse solutions within one standard error of the most accurate solution.

2.7 Discussion

We have presented a method for classifying graphs with labeled nodes, motivated by brain connectomics but generally applicable to any setting with such graphs. The distinct feature of our method is that it is graph-aware, aiming to select a sparse set of *both edges and nodes*, but it is general in the sense that it does not rely on the spatial structure of the brain. The method is computationally efficient since the regularization we use is convex.

The results we obtained on the schizophrenia data are generally in agreement with previous studies of schizophrenia. In particular, the default mode network has been consistently implicated in schizophrena and many other psychiatric disorders (Öngür et al., 2010; Broyd et al., 2009). While different networks were implicated by the two different datasets, we are still able to predict the disease status fairly accurately by training on one dataset and testing on the other. The differences between the two datasets may reflect real differences in samples collected at different sites and in different experiments, as significant pathophysiological heterogeneity occurs for all psychiatric diagnoses, or they may simply reflect type 2 errors.

Our methods work with a sample of networks with labelled nodes and associated responses. We acknowledge that in dealing with fMRI data many additional pre-processing steps are taken to arrive at this sample, which adds uncertainty and can potentially affect conclusions. We aimed to reduce the impact of some of these steps by using ranks which are more robust to global signal regression, and in practice multiple pre-processing pipelines can be used and compared to further validate results.

CHAPTER 3

A block structured regularization for prediction with network-valued covariates

3.1 Introduction

While the study of networks has traditionally been driven by social sciences applications and focused on understanding the structure of single network, neuroimaging applications have given rise to new methods for statistical analysis of multiple networks (Vogelstein et al., 2013; Ginestet et al., 2017; Narayan et al., 2015; Arroyo et al., 2017; Athreya et al., 2017). In neuroimaging, brain networks are constructed from raw imaging data, such as fMRI, to represent connectivity between a predefined set of nodes in the brain (Bullmore and Sporns, 2009), often referred to as regions of interest (ROIs). Collecting data from multiple subjects has made possible population level studies of the brain under different conditions, for instance, mental illness. Typically this is accomplished by using the network as a covariance, and predicting or testing differences in a phenotype, either a category like disease diagnosis or a quantitative measurement like attention levels.

Most of the previous work on this has followed one of two general approaches. One approach reduces the networks to global features that summarize the structure of the brain, such as average degree, centrality, etc., but these features do not capture local structure. The other approach applies standard prediction tools to vectorized adjacency matrices, treating all edge weights as individual features, but this fails to account for network structure, which can reduce both accuracy and interpretability. For a detailed discussion on this, see Section 2.1. Communities are a structure commonly found in networks from many domains A community is typically defined as a group of nodes that are more connected to each other than to the rest of the network. There are many methods available for detecting communities, including statistical models (Newman, 2010). Network models with communities such as stochastic block

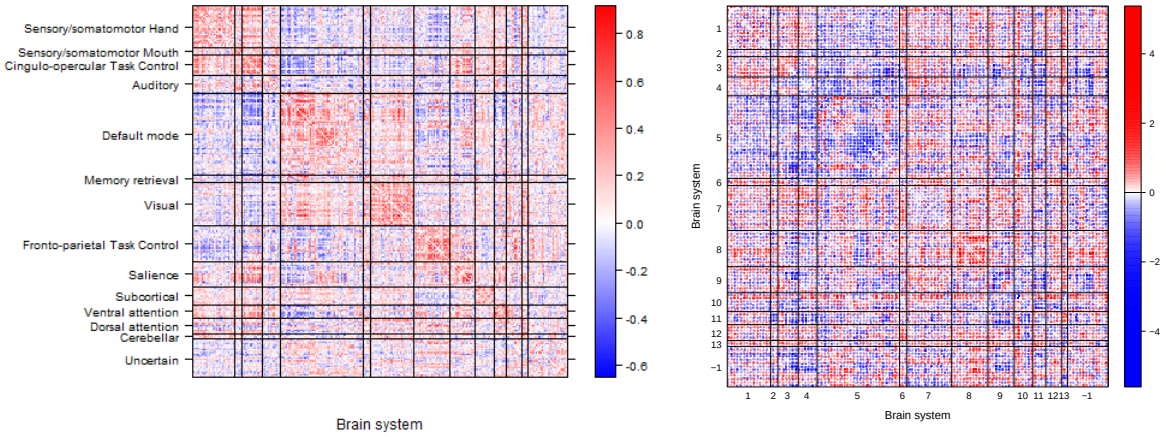


Figure 3.1: Left: A healthy subject connectivity matrix. Right: The t-statistics of edge-level differences between two samples of healthy and schizophrenic subjects. The data are from the COBRE dataset.

models can also provide a good approximation to more general statistical network models (Olhede and Wolfe, 2014; Amini and Levina, 2018).

Communities have been observed in brain networks, corresponding to functional brain systems (Fox et al., 2005; Chen et al., 2008; Bullmore and Sporns, 2009). The nodes in a community can be thought of as activating together during tasks and having similar functionality in the brain. Thus, each set of edges connecting two communities or brain systems, referred to as a *cell*, tends to have homogeneous connectivity levels. For example, this pattern appears in the left panel of Figure 3.1, which shows the fMRI brain network of a healthy subject from the COBRE data (`http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html`). In this example, the nodes of the network are defined according to the Power et al. parcellation (Power et al., 2011) which identifies 264 regions of interest (ROIs) in the brain. These nodes are divided into 14 communities, labeled in the left panel, which mostly correspond to known functional systems of the brain.

Analysis and interpretation of fMRI brain networks is usually done at the level of brain systems and cells, as voxels or ROIs are too granular and result in an excessively large number of edges to interpret. This approach is supported by the fact that the organization and connectivity of those brain systems is usually associated with subject-level phenotypes of interest, such as age, gender, or mental illness diagnosis (Meunier et al., 2009; Sripada et al., 2014b; Kessler et al., 2016). However, there is no universal agreement on exactly how to divide the brain into systems, and multiple parcellations are available (see Arslan et al. (2017) and references therein). These

parcellations are typically constructed either at the subject level by applying a community detection algorithm to a single network, or, more recently, at the sample level by estimating a common community structure for all the networks, typically from healthy subjects (Thirion et al., 2014). Several known parcellations have been obtained and mapped to known brain systems this way. However, this process does not take into consideration other variables associated with the subjects, or the fact that the functional systems may rearrange themselves depending on the task or condition (Smith et al., 2012; Fair et al., 2007; Sripada et al., 2014c)

When studying the association between the brain networks and some other subject-level variable, the choice of parcellation is important, as the resolution of some may not capture relevant associations. The right panel of Figure 3.1, shows the two sample t-test statistic values for each edge, for the difference in average connectivity between schizophrenic subjects and healthy controls from the COBRE data. Clearly, the cells of t-statistics are not as homogeneous as the connectivity values themselves, and some cells have both strongly positive and strongly negative groups, making it difficult to interpret the function of the cell as a whole. In particular, the default mode network (brain system 5) is a region that has been strongly associated with schizophrenia (Broyd et al., 2009), but interpreting this system as a complete unit based on this parcellation can be misleading, since it contains regions indicating both positive and negative effects.

In this chapter, we develop a new method that learns the most relevant community structure in the course of solving a prediction problems. We achieve this by enforcing a block-constant constraint on edge coefficients, in order to identify a grouping of nodes into clusters that gives the best prediction accuracy for the problem of interest, and has cells that really do behave homogenously. The solution is obtained with a combination of a spectral method and an efficient iterative optimization algorithm based on ADMM. We study the performance of our method in simulations, and apply it to schizophrenia prediction. Our method is able to obtain new sets of communities that give a parsimonious and interpretable solution with good prediction accuracy.

The rest of this chapter is organized as follows. In Section 3.2, we formulate a block regularization approach that enforces community structure in supervised prediction problems. Section 3.3 presents an algorithm to solve the corresponding optimization problem. In Section 3.4, we evaluate the performance of our methods in terms of recovering community structure and prediction accuracy on simulated data. Section 3.5 presents results for the COBRE dataset. We conclude with a discussion and future work in Section 3.6.

3.2 Supervised community detection

We start by setting up notation. Since the motivation comes from brain networks constructed from fMRI data, we focus on weighted undirected networks with no self loops, although our approach can be easily extended to other network settings.

We observe a sample of n networks with N labeled nodes that match across all networks $A^{(1)}, \ldots, A^{(n)}$ and their associated response vector $\mathcal{Y} = (Y_1, \ldots, Y_n)$, with $Y_i \in \mathbb{R}$, $i = 1, \ldots, n$. Each network here is represented by its weighted adjacency matrix $A^{(i)} \in \mathbb{R}^{N \times N}$, satisfying $A^{(i)} = (A^{(i)})^T$ and $\text{diag} A^{(i)} = 0$.

The inner product between two matrices $U, V \in \mathbb{R}^{N \times N}$ is denoted by $\langle U, V \rangle = \text{Tr}(V^T U)$. The entry-wise ℓ_p norm of a matrix $M \in \mathbb{R}^{N_1 \times N_2}$ is denoted by $\|M\|_p = \left(\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} M_{ij}^p \right)^{1/p}$; in particular, $\| \cdot \|_2 = \| \cdot \|_F$ is the Frobenius matrix norm.

We focus on linear methods for prediction, considering that one of our major goals is interpretation; however, the linear predictor can be replaced by another function as long as it is convex in the parameters of interest. For a given matrix A, the corresponding response Y will be predicted using a linear combination of the entries of A. Thus, we can define a matrix of coefficients $B \in \mathbb{R}^{N \times N}$, an intercept $b \in \mathbb{R}$ and a loss function ℓ such that

$$\ell(B) = \sum_{k=1}^{n} \tilde{\ell} \left(Y_k, \langle A^{(i)}, B \rangle + b \right), \tag{3.1}$$

in which $\tilde{\ell}$ is a prediction loss which can be chosen according to the problem of interest; in particular, this framework includes generalized linear models which can be used for binary or categorical responses. The entry B_{ij} of B is the coefficient for the edge (i, j), and since the networks are undirected with no self loops, we require $B = B^T$ and $\text{diag}(B) = 0$ for identifiability. In addition to the loss function, it is often convenient to include a penalty Ω_λ in the objective function. The penalty Ω_λ, with λ a tuning parameter that controls the amount of regularization, can be useful to make the solution uniquely defined in situations when the number of samples is small, or to enforce some structure in the solution. Popular choices for this penalty include the ridge, lasso penalty or elastic net (Friedman et al., 2009).

Remark 3.1. The parameter b in (3.1) is the intercept of the linear method and is important for accurate prediction, but since it can be removed in some situations by centering and is easy to optimize over if it is not removed, we omit the intercept in derivations that follow.

As discussed in the introduction, communities in brain networks we are inter-

ested in correspond to similar functionality, and thus it is reasonable to assume that edges within one network "cell" will have a similar effect on the response. This property can be explicitly encoded in the matrix of coefficients B. Suppose the nodes are partitioned into K groups $\mathcal{C}_1, \ldots, \mathcal{C}_K \subset \{1, \ldots, N\}$ such that $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ and $\bigcup_{k=1}^{K} \mathcal{C}_k = \{1, \ldots, N\}$. We assume that the values of B depend only on the community assignments of the nodes of the corresponding edge, so we can represent B using a $K \times K$ matrix C, such that $B_{ij} = C_{uv}$ if $i \in \mathcal{C}_u$ and $j \in \mathcal{C}_v$. Equivalently, define a binary membership matrix $Z \in \{0,1\}^{N \times K}$ such that $Z_{ik} = 1$ if $i \in \mathcal{C}_k$, and 0 otherwise. Then B can be written as

$$B = ZCZ^T. \tag{3.2}$$

This enforces equal coefficients for all the edges within one network cell (see Figure 3.2). This definition is analogous to the stochastic block model (SBM) (Holland et al., 1983), with the crucial difference that here B is not a matrix of edge probabilities, but the matrix of coefficients of a linear predictor.

Suppose for the moment that we are given a membership matrix Z. We can enforce cell-constant coefficients by adding a constaint on B to the optimization problem, solving

$$\begin{array}{ll} \min\limits_{C} & \ell(B) + \Omega(B) \\ \text{subject to} & B = ZCZ^T,\ C \in \mathbb{R}^{K \times K},\ C = C^T, \end{array} \tag{3.3}$$

or, using the fact $\left\langle A, ZCZ^T \right\rangle = \left\langle Z^T A Z, C \right\rangle$, we can restate the optimization problem in terms of C as

$$\hat{C} \;=\; \arg\min_{C} \left\{ \sum_{i=1}^{n} \ell\left(Y_i, \langle Z^T A^{(i)} Z, C \rangle + b\right) + \Omega(ZCZ^T) \right\}. \tag{3.4}$$

This effectively reduces the number of different coefficients from $N(N-1)/2$ to only $K(K+1)/2$, which allows for much easier interpretation of network cells.

For many choices of the penalty Ω_λ (for example, lasso or ridge) the optimization problem (3.4) is a standard prediction loss plus penalty problem with only $K(K+1)/2$ different parameters, which is easy to solve. Note that when the number of parameters $K(K+1)/2$ is large relative to n, some penalty Ω_λ is necessary in order to make the solution well defined.

Now let us return to the case of unknown Z. Our goal is to find a partition into communities that will give us the best prediction. Thus we need to jointly optimize

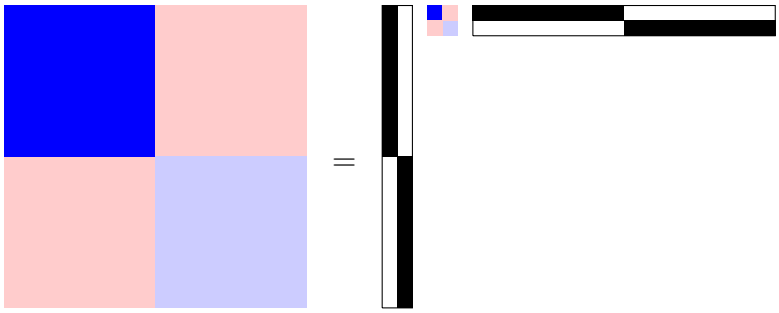


Figure 3.2: Factorization of matrix of coefficients B as ZCZ^T, with Z a membership matrix.

over Z and C, solving

$$\min_{Z,C} \left\{ \sum_{i=1}^{n} \ell\left(Y_i, \langle Z^T A^{(i)} Z, C\rangle + b\right) + \Omega(ZCZ^T) \right\} \tag{3.5}$$
$$\text{subject to } C \in \mathbb{R}^{K\times K},\ C = C^T$$
$$Z \in \{0,1\}^{N\times K}\ ,\ Z\mathbf{1}_K = \mathbf{1}_N.$$

The formulation (3.5) is aimed at the best community assignments for predicting a response, and enforcing block structure on the coefficients has the effect of grouping edges with similar predictive function into cells by clustering the associated nodes. Approaches for simultaneously predicting a response and clustering predictors have been proposed; for example, Bondell and Reich (2008) introduced a penalty that achieves this goal via fused lasso. Our goal, however, is not just clustering predictors (edges), it is partitioning the brain network into meaning regions, which requires clustering nodes.

The value of K in (3.5) plays the role of a tuning parameter that controls the amount of regularization. When $K = N$, there is no clustering. In practice, the value of K is unknown and can be chosen by cross-validation, as is commonly done with such tuning parameters. Alternatively, our method can be seen as a way of enforcing structure in the coefficients in order to simplify the interpretation with an approximately good solution, in the same way that unsupervised community detection is often used as an approximation to more general network distributions. Then the value of K can be chosen to match commonly used numbers of brain regions (typically 10-20), to obtain an interpretable solution and facilitate comparisons with existing parcelations.

Solving the optimization problem (3.5) is no longer easy, since optimizing over the membership matrix Z is a combinatorial problem. Next, we describe a computation-

ally feasible strategy to obtain an approximate solution.

3.3 An optimization algorithm for block structured regularized coefficients

Solving the problem (3.5) exactly is computationally infeasible, as the problem is NP-hard in Z. Instead, we propose an iterative optimization algorithm based on the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). Since the problem is not convex, having a good initial value is critical, even though in principle any membership matrix Z can be used to initialize the optimization algorithm. Some reasonable choices include starting from one of the previously published brain parcellations, on run an unsupervised community detection method on the networks, but these ignore the response $\mathcal{Y}$. We propose a spectral clustering algorithm that can give an approximate solution to the problem while taking the response into account, and can be used to initialize the ADMM optimization procedure; alternative initializations are also discussed.

3.3.1 Spectral clustering solution for the sum of squares loss

We start by introducing a spectral algorithm for solving the constrained problem (3.5) when the loss function is the sum of squared means. We assume for simplicity that the network matrices and the responses are centered, that is, $\bar{\mathcal{A}} = \frac{1}{n}\sum_{i=1}^{n} A^{(i)} = 0$ and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i = 0$. Then we can write the loss function as

$$\ell(B) = \frac{1}{2n}\sum_{i=1}^{n}\left(Y_i - \operatorname{Tr}\left(A^{(i)}B\right)\right)^2. \tag{3.6}$$

Denote by $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$ the $N \times N$ m matrix of coefficients for the simple linear regression of each edge (u, v) and the response, defined as

$$\left(\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}\right)_{uv} = \frac{\widehat{\operatorname{Cov}}(A_{uv}, Y)}{\widehat{\operatorname{Var}}(A_{uv})} = \frac{\sum_{i=1}^{n} Y_i A^{(i)}}{\sum_{i=1}^{n}(A^{(i)}_{uv})^2}. \tag{3.7}$$

Next, we perform spectral clustering on $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$. That is, we first compute the K leading eigenvectors of $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$, denoted by $V \in \mathbb{R}^{N\times K}$, and then cluster the rows of V using K-means, as summarized in Algorithm 3.1. Cluster assignments give a membership matrix $\hat{Z}^{(0)}$, which can be used either as a regularization in problem (3.3), or as an

initial value in Algorithm 3.2, introduced in the next section.

Algorithm 3.1 Spectral clustering solution for least squares loss

Input: Training sample $\{(A^{(1)}, Y_1), \ldots, (A^{(n)}, Y_n)\}$ centered and standardized; number of communities K.

1. Compute $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$ as in equation (3.7).
2. Compute the eigendecomposition of $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}} = Q\Lambda Q^T$, with Q an orthogonal matrix and Λ a diagonal matrix.
3. Set $V = (Q_{\cdot i_1} \cdots Q_{\cdot i_K})$ as the K leading eigenvectors of $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$, so $\Lambda_{i_1 i_1}, \ldots, \Lambda_{i_K i_K}$ are largest entries of Λ in absolute value.
4. Run k-means (or other clustering method) to cluster the rows of V into K groups. Let $\hat{\mathcal{C}}_1, \ldots, \hat{\mathcal{C}}_K$ be the indexes of those groups.
5. Form the membership matrix $\hat{Z}$ so that $\hat{Z}_{ik} = 1$ if $i \in \hat{\mathcal{C}}_k$ and 0 otherwise.

Output: $\hat{Z}^{(0)} = \hat{Z}$.

To justify the spectral clustering method, note that when the predictors are uncorrelated it is a well known fact that the least squares solution is given by $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$, so the loss function becomes

$$\check{\ell}(B) = \frac{1}{2} \left\| B - \Sigma^{\mathcal{A},\mathcal{Y}} \right\|_F^2 . \tag{3.8}$$

In general, it might be unrealistic to assume that the edges are uncorrelated. However, many methods constructed based on this assumption have surprisingly good performance in practice, even though this assumption might not hold (Bickel and Levina, 2004). Using the loss function (3.8), we approximate the solution for the least squares loss by solving the following optimization problem

$$\begin{aligned} \min_{Z,C} \quad & \tfrac{1}{2} \left\| ZCZ^T - \Sigma^{\mathcal{A},\mathcal{Y}} \right\|_F^2 \\ \text{subject to} \quad & Z \in \{0,1\}^{N \times K} \\ & Z\mathbf{1}_K = \mathbf{1}_N. \end{aligned} \tag{3.9}$$

This problem is still not convex, but its solution has been approximated with spectral clustering as in Algorithm 3.1 (Rohe et al., 2011; Chatterjee et al., 2015), so this is the approach we follow to obtain an approximate solution for the least squares loss function.

Remark 3.2. We also use Algorithm 3.1 when the loss function ℓ belongs to the family of generalized linear models (GLM). A popular optimization approach to fit a GLM is by iteratively reweighted least squares (IRLS) (Daubechies et al., 2010). This

method is based on iteratively solving a linear approximation to the loss function, which results in a weighted least squares problem. Algorithm 3.1 can be thought as a solution for the first step of IRLS with block constraints.

Remark 3.3. The sample covariance between the responses and the matrices $\Sigma^{\mathcal{A},\mathcal{Y}}$ provides a good estimate of $\hat{B}$ when the previous assumptions hold, so it is an appropriate and computationally cheap candidate for the spectral clustering algorithm. However, it is possible to substitute $\Sigma^{\mathcal{A},\mathcal{Y}}$ in the first step of Algorithm 3.1 with other solutions. In particular, it is appealing to use a an estimator $\tilde{B}$ that is approximately low rank, since this constraint is enforced in (3.5). A convex relaxations to a low-rank constraint in B was proposed by (Zhou and Li, 2014), which use a nuclear norm penalty to regularize the rank of the matrix. Thus, it is possible to use this estimator instead of $\Sigma^{\mathcal{A},\mathcal{Y}}$ in Algorithm 3.1. In our application, we did not observe a significant difference in the prediction error using these two choices, so we use $\hat{\Sigma}^{\mathcal{A},\mathcal{Y}}$ for data analysis.

3.3.2 Iterative optimization with ADMM

In this section, we propose a heuristic algorithm to approximately solve the optimization problem (3.5). Given Z, the problem is easy to solve as we described in Section 3.2, but finding the optimal Z is computationally infeasible. We propose an optimization strategy to approximate the solution, and later in Sections 3.4 and 3.5 we evaluate its performance.

The ADMM is a method for solving convex optimization problems with linear constraints. As we observed in Section 2.3, the ADMM is a flexible method for incorporating different types of penalty functions. Although the method is limited to linear constraints, some heuristics have been proposed to extend its applicability to more general settings with non-convex constraints (Diamond et al., 2016), showing a good numerical performance in those scenarios. We derive an iterative ADMM for solving (3.5). As we will see, the steps of the ADMM can be approximately solved in an efficient way.

Let $\mathcal{Z} = \left\{ Z \in \{0,1\}^{N\times K} \,|\, Z\mathbf{1}_K = \mathbf{1}_N \right\}$ be the set of valid membership matrices with K communities. Introducing a dual variable $V \in \mathbb{R}^{N\times N}$ and a variable W with block structure such that

$$W \in \mathcal{W} = \left\{ B \in \mathbb{R}^{N\times N} \,\middle|\, B = ZCZ^T, Z \in \mathcal{Z}, C \in \mathbb{R}^{K\times K} \right\},$$

the augmented Lagrangian of the problem can be written as

$$L_\rho(B, V, W) = \ell(B) + \Omega(B) + \langle V, B - W\rangle + \frac{\rho}{2}\|B - W\|_F^2, \tag{3.10}$$

with $\rho > 0$ a parameter of the optimization algorithm. Given some initial values $B^{(0)}, V^{(0)}$ and $W^{(0)}$, the ADMM consists of the following steps,

$$\begin{aligned} B^{(t)} &= \underset{B\in\mathbb{R}^{N\times N}}{\arg\min}\, L_\rho(B, V^{(t-1)}, W^{(t-1)}) && (3.11)\\ W^{(t)} &= \underset{W\in\mathcal{W}}{\arg\min}\, L_\rho(B^{(t)}, V^{(t-1)}, W) && (3.12)\\ V^{(t)} &= V^{(t-1)} + \rho\left(B^{(t)} - W^{(t)}\right). && (3.13) \end{aligned}$$

Equation (3.11) depends on the loss function ℓ and the penalty Ω, and can be expressed as

$$\begin{aligned} B^{(t)} &= \underset{B}{\arg\min}\left\{\ell(B) + \Omega(B) + \left\langle V^{(t-1)}, B - W^{(t-1)}\right\rangle + \frac{\rho}{2}\|B - W^{(t-1)}\|_F^2\right\}\\ &= \underset{B}{\arg\min}\left\{\ell(B) + \Omega(B) + \frac{\rho}{2}\left\|B - \left(W^{(t-1)} - \frac{1}{\rho}V^{(t-1)}\right)\right\|_F^2\right\} \end{aligned} \tag{3.14}$$

If ℓ and Ω are convex, as it is generally the case, then (3.11) is also convex, and in some cases it is possible to express it as a regularized regression problem with a ridge penalty. Step (3.12) requires to solve a non-convex combinatorial problem due to the membership matrix Z. This step can be expressed as

$$W^{(t)} = \underset{W\in\mathcal{W}}{\arg\min}\left\|W - \left(B^{(t)} + \frac{1}{\rho}V^{(t-1)}\right)\right\|_F^2. \tag{3.15}$$

This previous equation is equivalent to problem (3.9). We use the same strategy to approximately solve this problem by performing spectral clustering to the matrix $B + \frac{1}{\rho}V$ in order to obtain a membership matrix $Z^{(t)}$. Once we find $Z^{(t)}$, the solution of equation (3.15) is given by

$$W^{(t)} = Z^{(t)}(Z^{(t)^T}Z^{(t)})^{-1}\left(B^{(t)} + \frac{1}{\rho}V^{(t-1)}\right)(Z^{(t)^T}Z^{(t)})^{-1}Z^{(t)^T}. \tag{3.16}$$

Finally, equation (3.13) requires to update the dual variables. We iterate this algorithm until some convergence criteria is met. For example, we can use the convergence criteria employed in Algorithm 2.2 at Section 2.3. The steps of the ADMM algorithm

are summarized in Algorithm (3.2)

Algorithm 3.2 Iterative optimization with ADMM

Input: $\{(A^{(1)}, Y_1), \ldots, (A^{(n)}, Y_n)\}$, K, ρ, $Z^{(0)}$
Initialize: Set $W^{(0)}$ as the solution of problem (3.3) using $Z = Z^{(0)}$, and $V^{(0)} = 0_{N \times N}$.
Iterate: for $t = 1, 2, \ldots$ until convergence

1. Compute $B^{(t)}$ according to equation (3.14).
2. Update $W^{(t)}$ by spectral clustering
   (a) Set V be the K leading eigenvectors of $B^{(t)} + \frac{1}{\rho} V^{(t-1)}$.
   (b) Run K-means to cluster the rows of V into K groups, let $\hat{C}_1, \ldots, \hat{C}_K$ be those groups.
   (c) Set $Z^{(t)}$ as $Z_i^{(t)} k = 1$ if $i \in \hat{\mathcal{C}}_k$ and 0 otherwise.
   (d) Update $W^{(t)}$ as in equation (3.16).
3. Update $V^{(t)}$ according to equation (3.13).

Output: $\hat{B} = B^{(t)}$ and $\hat{Z} = Z^{(t)}$

The parameter ρ in Algorithm 3.2 controls the size of the primal and dual steps. A larger ρ will enforce $B^{(t)}$ to stay close to the value of $W^{(t-1)}$, and since this parameter depends on the community assignment $Z^{(t-1)}$ the community assignments will be less likely to change on each iteration. In convex optimization problems, the ADMM is guaranteed to converge to the optimal value for any $\rho > 0$. Here, if ρ is small, the algorithm might not converge, while for a large enough ρ, the algorithm will not move from the initial community assignment $Z^{(0)}$. In practice, we run Algorithm (3.2) for a set of different values of ρ, and choose the solution $\hat{B}$ that gives the best value in equation (3.5).

3.4 Numerical results on simulated networks

In this section, we use simulated data to evaluate the performance of our method on both predicting a response and recovering the community structure. We generate networks with $N = 40$ nodes with the community structure of a stochastic block model (SBM), resulting in $p = 780$ distinct edges, with nodes partitioned into $K = 4$ communities of equal size, and define the node community labels as $c_1, \cdots, c_{40} \in \{1, \ldots, 4\}$. In our main application edges have real-valued weights, so instead of using the Bernoulli distribution stipulated by SBM, we generate edge weights from a Gaussian distribution. Given a subject connectivity matrix $H^{(i)} \in \mathbb{R}^{K \times K}$, each edge

(u, v) of the network $A^{(i)}$, with $u > v$, is generated independently as

$$A_{uv}^{(i)} \sim N(H_{c_u c_v}^{(i)}, s^2), \tag{3.17}$$

with $s = 0.1$. The connectivity matrix $H^{(i)}$ for subject i is generated as

$$H^{(i)} = \begin{pmatrix} 0.3 + tU_i & 0.3 & 0.1 & 0.1 \\ 0.3 & 0.3 + tU_i & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.3 + tU_i & 0.3 \\ 0.1 & 0.1 & 0.3 & 0.3 + tU_i \end{pmatrix}, \tag{3.18}$$

with U_i a random variable uniformly distributed on $(-0.5, 0.5)$ and $t \in \mathbb{R}$ a parameter of the model. When $t = 0$, the networks have only two communities, and as t increases, the four communities become more distinguishable. Given a network $A^{(i)}$, the response Y_i is generated from the linear model,

$$Y_i = \langle A^{(i)}, B \rangle + \epsilon_i, \tag{3.19}$$

with $\epsilon_i \sim N(0, \sigma^2)$ i.i.d. noise. The matrix of coefficients B shares the community structure of the networks and is defined as

$$B_{uv} = \begin{cases} 1 & \text{if } c_u = c_v, \\ 0 & \text{otherwise.} \end{cases}$$

We fit the model on training data of size n (to be specified) by solving the optimization problem with the least squares loss function. No penalty is included in the objective function since the value of K is small compared to the sample size. We evaluate prediction performance by mean squared error (MSE) on test data. The parameter K is chosen by 5-fold cross-validation. Since the estimated K may be different from $K = 4$ used to generate the data, we measure community detection performance by the co-clustering error, which measures the proportion of nodes not assigned to the same community. Given two membership matrices Z and $\tilde{Z}$, the co-clustering error is defined as

$$E(Z, \tilde{Z}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |(ZZ^T - \tilde{Z}\tilde{Z}^T)_{ij}|.$$

We fit our method by the spectral clustering (Algorithm 3.1) used to provide an intial value to the ADMM (Algorithm 3.2). As a benchmark for prediction, we use lasso

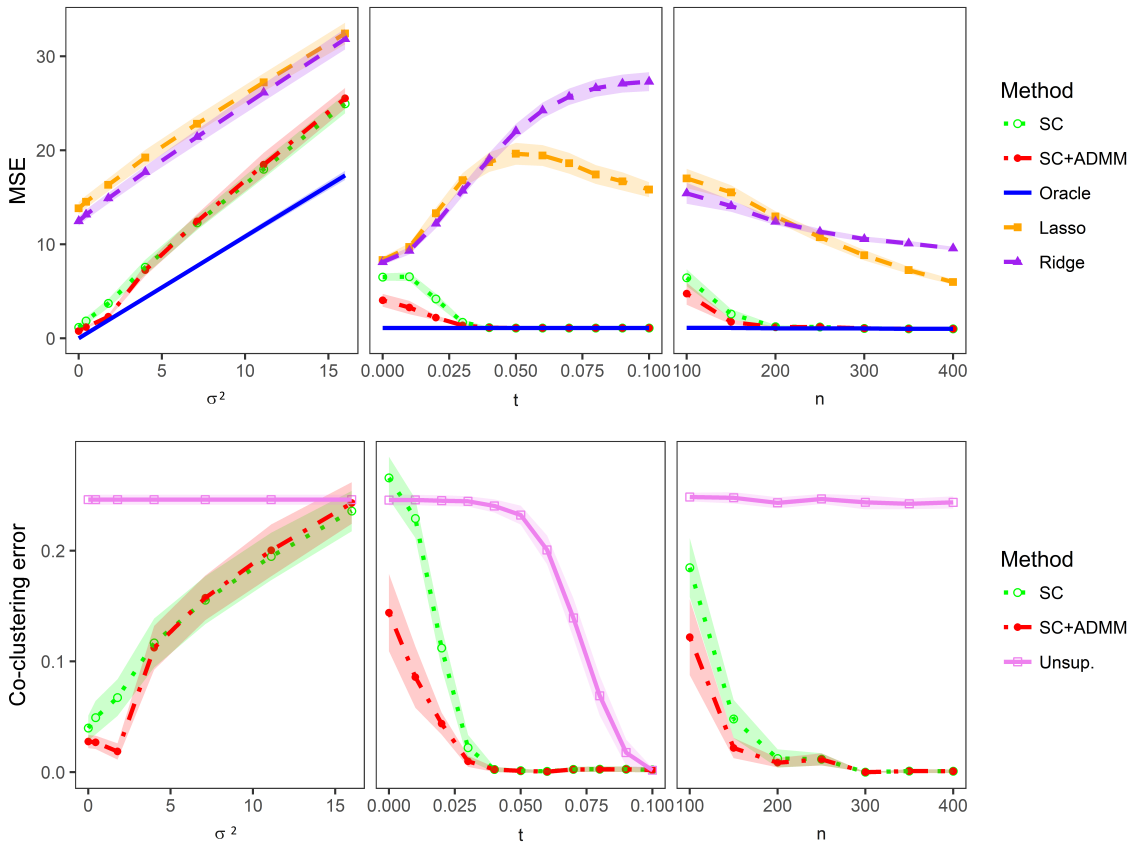


Figure 3.3: Results for prediction (top) and community detection (bottom). Each plot shows the corresponding error averaged over 50 replications as a function of one of the parameters σ, t, and n, keeping the other two fixed.

and ridge regression (Friedman et al., 2009), which are generic regularized linear regression methods. An oracle method is included as a reference, constructed by solving (3.3) using the true communities. Unsupervised community detection methods often combine sample networks by averaging, but these methods generally are designed for binary edges, and will fail in our setting since $\mathbb{E}A^{(i)}$ is a matrix with only two communities. Bhattacharyya and Chatterjee (2017) proposed a community detection method for samples of networks that can work in our setting, based on performing spectral clustering on the sum of the squared adjacency matrices $\tilde{A} = \sum_{i=1}^{n}(A^{(i)^2} - \mathrm{diag}(A^{(i)^2}))$. We use this method as a benchmark for community detection.

The difficulty of the problem is controlled by multiple parameters. We vary the noise level σ, the strength of the community structure (controlled by t), and the sample size n. In each experiment, we vary one of these parameters while keeping the other two constant. The constant values are set to $\sigma = 1$, $t = 0.025$ and $n = 150$. Each scenario is repeated 50 times.

Figure 3.3 shows the average performance of the methods as a function of model parameters. In general, all methods perform better when the noise level is lower, the community structure is stronger, and the sample size larger, as one would expect. Our method outperforms both lasso and ridge, since it takes advantage of the underlying structure. Note that the value of $\langle A^{(i)}, B\rangle$ changes with t, which explains why the MSE of lasso and ridge may not be monotone as a function of t. ADMM usually improves on the initial value provided by SC, and when the signal is strong enough, both methods recover the community structure correctly, performing as well as the oracle. Unsupervised community detection does not use the response values, and hence its performance does not depend on σ. Community detection becomes easier as t increases, but the unsupervised method requires a much larger t than our supervised community detection algorithm to achieve the same performance. The sample size has virtually no effect on unsupervised community detection, but supervised detection improves very quickly as the sample size grows.

3.5 Supervised community detection in fMRI brain networks

Here we apply the proposed method to classification of brain networks from healthy and schizophrenic subjects from the COBRE dataset. Diagnosing schizophrenia from fMRI data can be useful in clinical practice as behavioral data can be misleading in diagnosing such disorders (Campanella, 2015), and understanding which regions of the brain are implicated in schizophrenia is an important step towards developing new treatments.

The COBRE dataset includes 54 schizophrenic patients and 70 controls. For a description of the data and pre-processing steps, see Section 2.6. The Power parcellation (Power et al., 2011) was employed to define the regions of interest (ROIs), resulting in a total of $N = 263$ nodes in the brain (see left panel of Figure 3.4).

We train our method using logistic regression loss and the lasso penalty, needed since the sample size $n = 124$ only allows us to fit up to $K = 15$ communities without regularization. As is commonly done with logistic regression with a large number of predictors, we also add a small ridge penalty for numerical stability. The objective function is given by

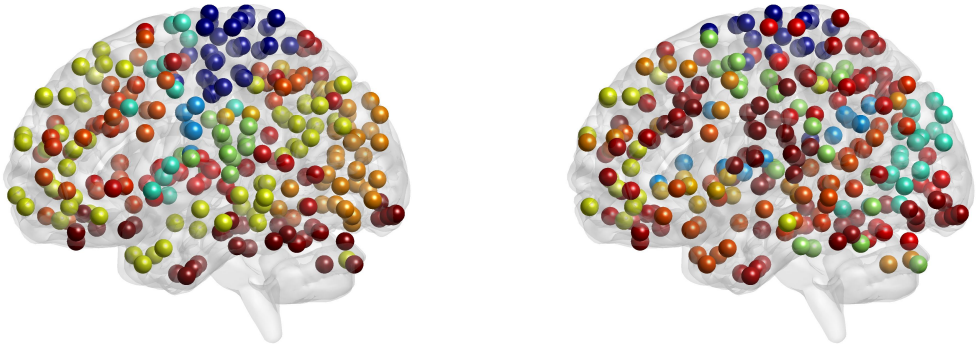
$$\ell(B) + \Omega_\lambda(B) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-Y_i(\langle A^{(i)}, B\rangle + b))\right) + \frac{\gamma}{2}\|B\|_F^2 \ + \ \lambda\|B\|_1.$$

The parameter λ in the lasso penalty controls sparsity of the solution and is selected by cross-validation. The value of γ is fixed at 10^{-5}.

The analysis and interpretation of brain connectomic studies is usually done at the level of brain systems, since nodes or edges are too granular for interpretation. In the Power parcellation, nodes are partitioned into 14 brain systems (see Figure 2.1) which are related to known functional areas in the brain (Power et al., 2011). The nodes labeled according to these communities are shown in Figure 3.5. We use these communities as a baseline by solving the constrained problem (3.3), and compare the performance of our supervised community detection method using the same number of $K = 14$ communities.

Figure 3.4 shows the 263 nodes of the data colored according to the parcellation by Power et al. (2011) (left panel) and the communities we found with our method with $K = 14$ (right panel). In both cases, we chose λ by cross-validation, and evaluated prediction accuracy using 10-fold nested cross-validation. The accuracy for each method is reported in Figure 3.4. Our supervised method uses the same number of parameters but has a noticeably higher accuracy. for easier interpretation of the communities we found, Figure 3.6 shows a separate plot for each community. These node clusters are mostly well concentrated in space, suggesting they may correspond to meaningful structural or functional regions. In Figure 3.7, a Sankey diagram shows a comparison of the new and the old community assignments. Many of the Power communities are partitioned into smaller communities with supervised community detection. Figure 3.8 shows the fitted coefficients ordered according to the supervised communities. Note that the communities F, H and I are mostly composed by nodes from the default mode network (community 5 in Power parcellation), but now we can observe cells with positive or negative coefficients within those communities.

Using 10-fold cross-validation, the average prediction error and number of non-zero different coefficients for a grid of values of λ and K are reported in Figure 3.9. Even when the number of communities is small, the accuracy of the supervised method is better than the baseline communities (Figure 3.4), and as K increases, the accuracy improves significantly. Comparing with other methods that we previously evaluated using the same data (Table 2.2), our method shows satisfactory accuracy with a small number of parameters that are highly interpretable, and its performance is better than methods like lasso or DLDA, which are only able to select individual edges.



Power brain systems
CV accuracy: 62%

Supervised community detection
CV accuracy: 79%

Figure 3.4: Baseline communities (Power et al., 2011) and communities found by our supervised community detection method.

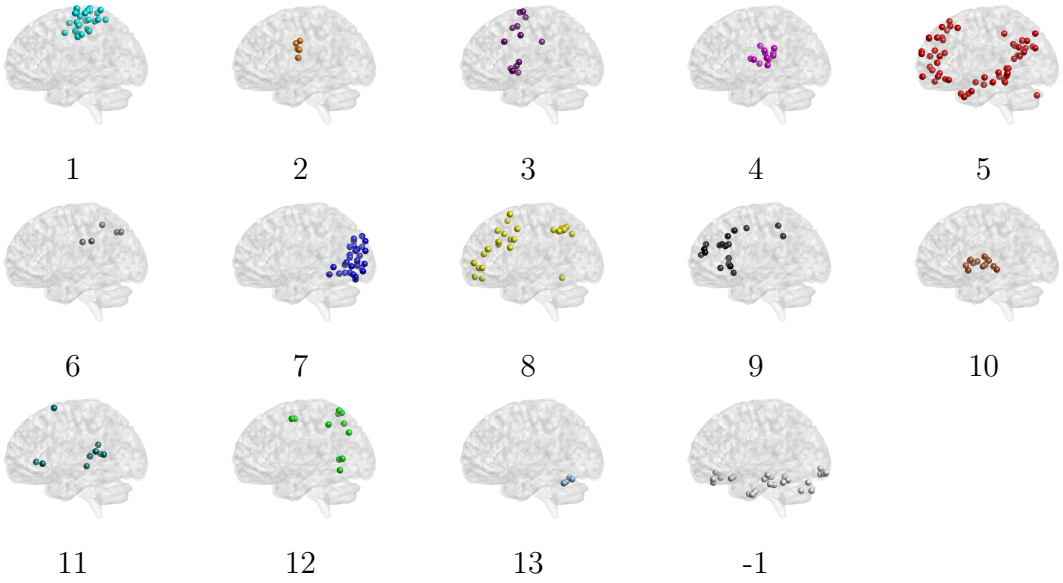


Figure 3.5: Individual communities proposed by Power et al. (2011).

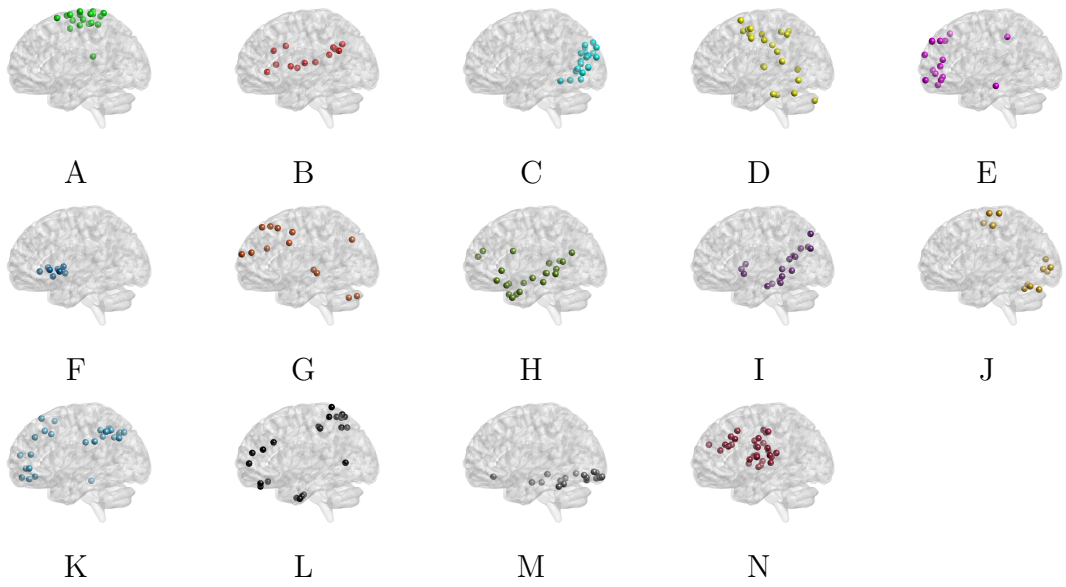


Figure 3.6: Individual communities found by the supervised community detection method.

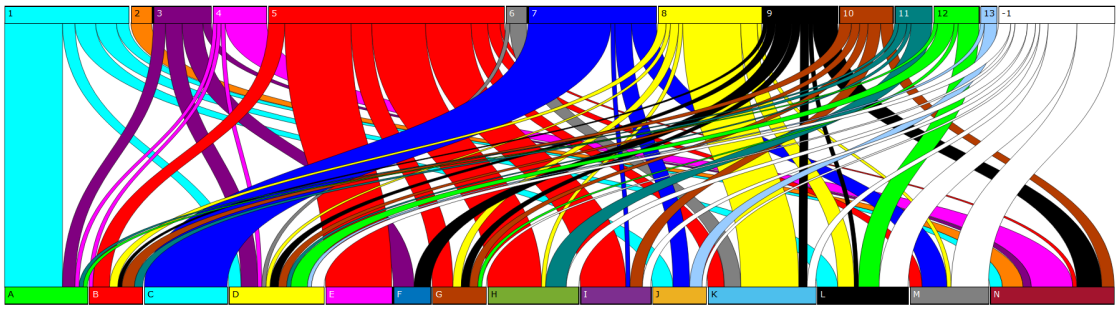


Figure 3.7: Sankey diagram of the node community assignment changes from the Power parcellation (top row) and the communities found by our method (bottom row).

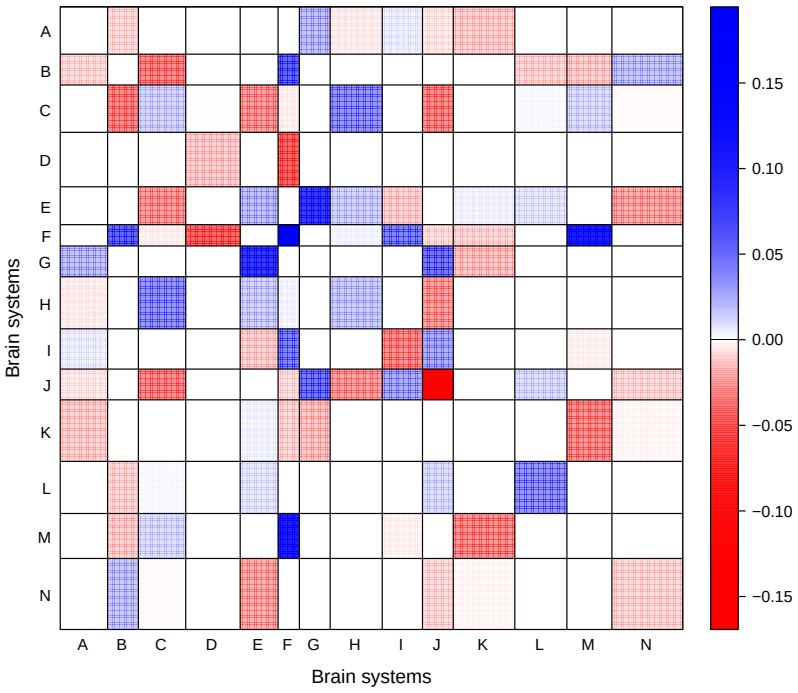


Figure 3.8: Matrix of fitted coefficients with the communities found by supervised community detection ($K = 14$).

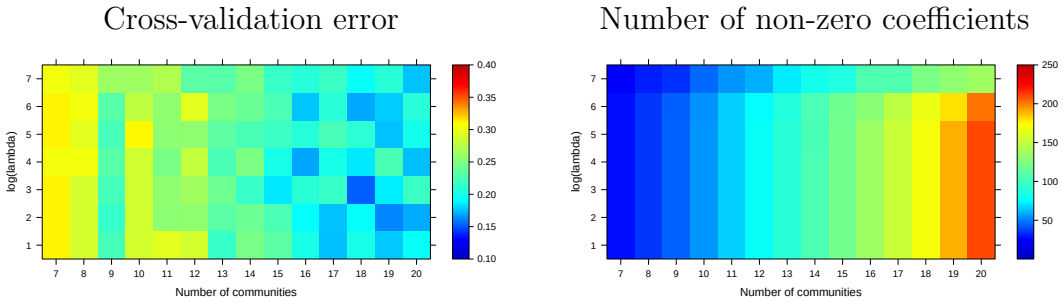


Figure 3.9: Average cross-validation error (left panel) and average number of non-zero different coefficients (right panel) for a grid of λ and K values.

3.6 Discussion

Finding communities in network is a much studied problem, but frequently there is no way to evaluate the success of the procedure – even when "ground truth" is available, it's often just another network covariate which may or may correspond to communities – and no clear way to extract useful information from the structure discovered beyond the communities themselves; this is a common problem for unsupervised tasks. In contrast, here we use community structure as a regularization tool in a prediction problem, allowing for clear evaluation and comparisons in terms of prediction error. However, good prediction is only one of our goals; having a sparse and interpretable solution is just as important from the scientific point of view. This is one reason that we imposed an equal coefficient constraint on all the edges within a cell rather than shrink them towards each other; another reason for this choice is a much simpler optimization problem. Imposing a shrinkage penalty within communities and then optimizing jointly over coefficients and communities, and tuning both K and the shrinkage penalty parameter, is in principle a valid approach, but with the tools currently available it would vastly increase the computation costs.

Our method performed well on neuroimaging data, providing solutions that are easier to interpret than those which treat each edge weight as a separate predictor. We hope to apply it in future work to a much wider range of datasets and study how much suggested parcellations and cells' functions differ between conditions, tasks, stages of development, etc. This can potentially lead to a paradigm shift in neuroimaging data analysis, where the parcellations are currently treated as fixed.

From statistical perspective, there is much more that can be done. We leave establishing statistical guarantees for our method, such as consistency of community detection and parameter estimation, for future work. More complex prediction rules, rather than linear functions, can be used to obtain more flexible classifiers; the method extends easily to methods such as polynomial regression, splines, generalized additive models, or anything else that fits coefficients to an expanded basis. We can further amend the loss function to not just evaluate the quality of prediction, but also the strength of discovered community structure; this would allow a balance between finding the most predictive communities and the strongest communities purely in the network sense, which may or may not be the most predictive. Finally, community structure can be used as an approximation to more general models, for example, smooth graphons such as those in Zhang et al. (2017).

Valid statistical inference for this approach is left for future work as well. While

there has been a lot of recent activity in high-dimensional post-selection inference (Van de Geer et al., 2014; Lee et al., 2016; Lockhart et al., 2014), we are in a much harder setting of grouped rather sparse coefficients, and the groups themselves are learned from data, unlike in typical group lasso problems (Yuan and Lin, 2006). There are currently no methods we are aware of that can handle this type of setting, but we will investigate whether existing methods can be modified or new ones developed in future work.

CHAPTER 4

Overlapping community detection using sparse principal component analysis

4.1 Introduction

Networks have become a popular representation of complex data that appear in different fields such as biology, physics, chemistry or the social sciences. In a network, units of a system are represented by nodes, and the interactions between them by edges. Thus, a network can encode the relationships between people in a social environment (Wasserman and Faust, 1994), connectivity between areas of the brain (Bullmore and Sporns, 2009) or interactions between proteins (Schlitt and Brazma, 2007). The constant technological advancements have improved our ability to collect and process information, leading to an explosion in the size and complexity of the data. In particular, this has increased the availability of network data. Nowadays, networks that scale from hundreds to millions of nodes are ubiquitous, opening statistical challenges to analyze such type of the data. Parsimonious models are required in order to being able to interpret the solutions, as well as efficient methods that can scale to large datasets.

Communities are a structure of interest in the analysis of networks, since in many real-world systems nodes tend to form groups with strong connectivity between the members (Girvan and Newman, 2002). Usually, communities are defined as clusters of nodes that have stronger connections to each other than to the rest of the network. Finding these communities allows to simplify the dimensionality of the data by reducing the nodes to a smaller number of units which often reflect structures that are meaningful for understanding the system of interest. In real-world networks, communities can represent functional areas of the brain (Schwarz et al., 2008; Power et al., 2011), political affinity in social networks (Adamic and Glance, 2005; Conover

et al., 2011; Latouche et al., 2011), research areas in citation networks (Ji et al., 2016), among many other examples.

The stochastic block model (SBM) (Holland et al., 1983) is a simple statistical model for community structure in a network with by now well understood theoretical guarantees (Bickel and Chen, 2009; Decelle et al., 2011; Mossel et al., 2015; Le et al., 2015; Gao et al., 2015). Under the SBM, a pair of nodes is connected with a probability that only depends on the community memberships of these nodes. Due to its simplicity, the model is not able to capture some aspects of real-world networks, but extensions have been proposed to incorporate aspects of interest such as *hubs* (Ball et al., 2011), or allow nodes to belong to more than one community (Airoldi et al., 2009; Latouche et al., 2011; Zhang et al., 2014).

In this paper, we focus on the estimation of sparse overlapping continuous memberships. Overlapping community models allow to characterize each node by a vector of memberships to the different communities in the network. In real-world networks, memberships are usually sparse, in the sense that most nodes belong to only one or few groups. In fact, the sparsest scenario in which nodes belong to only one community corresponds to the classic community detection setting, and its success in modeling and analyzing real-world networks in many different fields suggests the idea that the sparsity assumption in the overlapping memberships is reasonable.

Detecting overlapping communities involves identifying what are these communities, and at the same time assigning each node to one or multiple communities. Existing statistical models for overlapping community detection define the node memberships either as binary, in which a node is assigned or not to a community with a fixed degree of association (Latouche et al., 2011), or as a continuous membership that allows each node to have a different level of association to each community (Airoldi et al., 2009; Ball et al., 2011; Psorakis et al., 2011; Zhang et al., 2014). Binary memberships are a natural way to account for sparsity in the overlapping assignments, but the model is restrictive since community memberships cannot reflect different strengths of belonging to a community, and fitting those memberships can be computationally intensive. On the other hand, continuous assignments are not able to explicitly account for sparsity, and the resulting estimates usually assign most of the nodes to many or all communities. To obtain sparse memberships, a post-processing step is required, which can harm the estimation accuracy.

Here, we present a new approach for detecting overlapping communities in a network based on the estimation of an appropriate sparse basis for the principal subspace of a network adjacency matrix A. Statistical models for community detection usually

assume that the expected value W of the network adjacency matrix has a low rank structure, and the principal subspace of W contains the information required to identify the communities. Existing spectral methods for community detection exploit this fact by computing the leading eigenvectors of the adjacency matrix A or its Laplacian, and then apply some clustering technique to obtain the latent memberships (see for example Newman (2006); Rohe et al. (2011); Lyzinski et al. (2014); Zhang et al. (2014); Jin et al. (2017)). In contrast, we directly estimate an appropriate eigenbasis that contains the community information via sparse principal component analysis (Jolliffe et al., 2003; Zou et al., 2006; Ma, 2013). We present methods that can estimate a non-orthogonal sparse eigenbasis representing the node memberships. Our estimators are able to recover a membership matrix with the correct sparsity pattern in the assignments. Moreover, our methods have a low computational cost that is no larger than computing the leading eigenvectors of a matrix.

4.2 Detecting communities by sparse subspace estimation

4.2.1 Community detection with overlaps

Here, we focus in the study of a single unweighted network with n nodes. A network can be represented with a binary adjacency matrix A of size $n \times n$. We assume that the network is undirected with no self-loops, so A is a symmetric matrix with zeros on the diagonal, and $A_{ij} = 1$ indicates that there is a link between nodes i and j. Many popular statistical network models use a matrix $W \in \mathbb{R}^{n \times n}$ to encode the probability of the edges, which are independent Bernoulli random variables.

The SBM (Holland et al., 1983) is one of the earliest and most popular statistical models for community detection. The model partitions the nodes into K non-overlapping communities $\mathcal{C}_1, \ldots, \mathcal{C}_K \subset \{1, \ldots, N\}$, and the probability of link between two nodes just depends on the communities of this nodes. Given a probability matrix $B \in [0,1]^{K \times K}$, a network is distributed according to the SBM if for every pair of nodes i, j, $i > j$, the edge connecting i and j is an independent Bernoulli random variable with probability

$$P(A_{ij} = 1) = B_{uv},$$

whenever $i \in \mathcal{C}_u$ and $j \in \mathcal{C}_v$. Alternatively, the model can be described using a

membership matrix Z of size $n \times K$ indicating the community memberships, so $Z_{ik} = 1$ whenever $i \in \mathcal{C}_k$, and zero otherwise, so the expected value $W \in \mathbb{R}^{n \times n}$ can be expressed as

$$W = \mathbb{E}A = ZBZ^T.$$

The SBM provides a simple model that allows to characterize community structure but it is not able to capture some other properties of real networks. Multiple extensions to the SBM have been proposed to overcome those limitations, and here we focus in *overlapping communities*. In real networks, nodes can belong to more than one community at a time, and identifying all the memberships is a problem of interest. The overlapping continuous community assignment model (OCCAM) (Zhang et al., 2014) is a particular extension to the SBM that encompasses many other models for overlapping and non-overlapping community detection. Given a continuous membership matrix $Z \in \mathbb{R}^{n \times K}$ with $Z \geq 0$ and $\|Z_{i\cdot}\|_2 = 1$, a connectivity matrix $B \in \mathbb{R}^{K \times K}$, a vector of degree intensities $\theta \in \mathbb{R}^n$, and a parameter $\alpha > 0$ that controls the average degree, OCCAM defines the expected adjacency matrix of a network as

$$W = \mathbb{E}A = \alpha \Theta Z B Z^T \Theta, \tag{4.1}$$

where $\Theta \in \mathbb{R}^{n \times n}$ is a diagonal matrix such that $\text{diag}(\Theta) = \theta$. In OCCAM, nodes can belong to multiple communities at the same time. Each row of Z can have multiple or all the entries different from zero, indicating the communities to which the node belong. OCCAM also incorporates a vector of degree intensities θ which allows to model hub nodes in the network as in the degree-corrected SBM (Ball et al., 2011). The model is also related to the mixed-membership SBM (Airoldi et al., 2009; Jin et al., 2017), which is a hierarchical Bayesian version of overlapping community detection.

OCCAM contains many more parameters than the SBM, which are identifiable under some restrictions on Z, Θ and B (Zhang et al., 2014). In the next Theorem, we show that in general any low-rank model $W = \mathbb{E}A$ that has a valid membership matrix as the basis of the eigenspace is identifiable up to positive scaling and permutations. The proof is on the Appendix.

Proposition 4.1. *Let $W \in \mathbb{R}^{n \times n}$ be a symmetric matrix of rank K. Suppose that there exists a matrix $Z \in \mathbb{R}^{n \times K}$ that satisfies the next conditions:*

- *Z is nonnegative, i.e., $Z_{ik} \geq 0$ for all i, k.*
- *For each $k = 1, \ldots, K$ there exists at least one row i_k of Z such that $Z_{i_k k} > 0$*

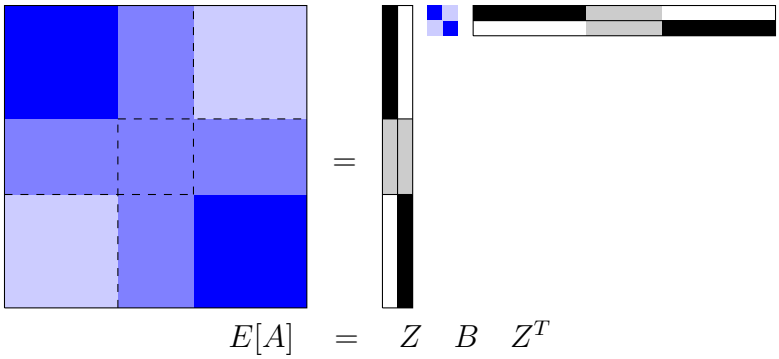


Figure 4.1: Representation of the expected value of a network adjacency matrix with an overlapping structure. Each column of Z corresponds to a community, and the non-zero entries on each row indicates the community that the corresponding node belongs to.

and $Z_{i_k j} = 0$ *for* $j \neq k$.

- Z *is a basis of the column space of* W*, that is,* $range(A) = range(Z)$.

If there is any other matrix $\tilde{Z} \in \mathbb{R}^{n \times K}$ *that satisfies the previous conditions, then* $\tilde{Z}PD = Z$*, with* P *a permutation matrix and* D *a diagonal matrix with* $diag(D) > 0$*, and therefore*

$$supp(Z) = supp(\tilde{Z}P),$$

with $supp(Z) = \{(i,j)|Z_{ij} \neq 0\}$ *the set of non-zeros of* Z.

Remark 4.1. When the model for W is OCCAM, Proposition 4.1 provides similar identifiability conditions than Theorem 2.1 of Zhang et al. (2014). Here, we do not require explicit conditions on B or Θ, but it is implicitly required that B has full rank. Our statement is slightly more general since we do not require that B is positive definite.

The previous result implies that if $W = \mathbb{E}A$ has rank K and there is a membership matrix Z that is a basis of the eigenspace of W, then the matrix Z is unique up to permutations and scaling factors, but the pattern of non-zeros remains the same. This fact allow us to characterize the community structure by looking at the non-zero values of each column of Z (see Figure 4.1). In the rest of this paper, any matrix Z that satisfies those conditions is referred as a *membership matrix*. Thus, if we know the expectation of the adjacency matrix W, the task of overlapping community detection can be solved by identifying any membership matrix that is an eigenbasis of W. This perspective motivates the methods we present here.

4.2.2 Community detection via sparse principal component analysis

Fitting statistical models for community detection requires in principle to solve a combinatorial problem, but a vast literature of computationally feasible algorithms has been developed, with the majority of them focusing in the non-overlapping case. Spectral methods are a popular approach to find Z due to its simplicity and computational speed. These methods usually compute the leading eigenvectors of A (or the Laplacian), which contain information about the communities, and then apply a clustering procedure to the rows of the eigenvectors in order to assign the nodes into communities (see for example Rohe et al. (2011); Lei et al. (2015); Zhang et al. (2014); Jin et al. (2017)). When communities overlap, the result is usually a continuous membership matrix Z, and an additional thresholding step is required to find sparse memberships. Thus, spectral algorithms require to perform different procedures in separate steps, and this can harm the accuracy of the final estimator. Moreover, some clustering techniques or other methods for identifying the communities can be computationally expensive (Zhang et al., 2014; Jin et al., 2017). In this paper, we propose an approach to directly estimate an appropriate eigenbasis of A that contains the information of the overlapping memberships. As we will see, this process can result in computationally efficient and accurate methods.

Principal component analysis (PCA) (Hotelling, 1933) is a popular dimensionality reduction technique. There are several ways to formally define PCA, but in order to motivate our analysis, we use the following formulation. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, PCA can be defined as the best approximation to A with a matrix $\hat{M}$ of rank K such that

$$\hat{M} = \underset{M:\mathrm{rank}(M)=K}{\arg\min} \|A - M\|_F^2. \tag{4.2}$$

The previous equation has a closed form solution. If $q_1, \ldots, q_K$ are the K leading eigenvectors of W, then for any matrix $V \in \mathbb{R}^{n\times K}$ that satisfies $\mathrm{range}(V) = \mathrm{ran}(\{q_1, \ldots, q_K\})$, the solution of (4.2) is given by

$$\hat{M} = V(V^TV)^{-1}V^TAV(V^TV)^{-1}V^T. \tag{4.3}$$

This matrix V is obviously not unique. In PCA, it is usually assumed that the eigenvectors are the basis of interest, as their orthogonality simplifies interpretation. In general, there is no particular reason to choose orthogonal vectors. In fact, in overlapping community models, the membership matrix Z is a basis of W but it is

not orthogonal. Although the principal space of A is still a good estimator for the principal space of W, using the eigenvectors of A to estimate Z might not be the optimal solution.

Sparse principal component analysis (SPCA) methods (Jolliffe et al., 2003; Zou et al., 2006) incorporate sparsity constraints or regularizations that promote some additional sparsity structure in the solution of (4.2). In high-dimensional scenarios, enforcing sparsity in the solution of PCA can improve the estimation when the data samples are scarce, or simplify the interpretation of the solutions. Our goal in this paper is related to SPCA since we are interested in estimating a sparse eigenbasis of W. Other connections between SPCA and community detection have been reported. Amini and Levina (2018) observed a relation between a convex relaxation of the MLE for non-overlapping community detection and a convex formulation of SPCA.

As the eigenvectors are usually the solution of interest in PCA, many SPCA methods have been proposed to estimate the eigenvectors of a matrix under sparsity assumptions (see for example Amini and Wainwright (2008); Johnstone and Lu (2009); Vu et al. (2013); Ma (2013)). Orthogonal iteration is classic method for estimating the eigenvectors of a matrix. Ma (2013) extended this method to estimate sparse eigenvectors by an iterative thresholding algorithm. The author studied this method under the spiked covariance model, showing a good statistical and computational performance. Although the iterative thresholding method is specific to the estimation of sparse eigenvectors, this framework can be adapted to more general settings in order to obtain a regularized basis for a matrix A. Starting from an initial matrix $Z^{(0)} \in \mathbb{R}^{n\times K}$, a general version of the algorithm of Ma (2013) consists on iteratively performing the next steps until convergence:

- Multiplication step:
$$U^{(t)} = AZ^{(t-1)}. \tag{4.4}$$

- Regularization step:
$$V^{(t)} = \mathcal{R}(U^{(t-1)}), \tag{4.5}$$
where $\mathcal{R} : \mathbb{R}^{n\times K} \to \mathbb{R}^{n\times K}$ is a regularization function.

- Numerical stability step:
$$Z^{(t)} = V^{(t)}\Pi^{(t)}, \tag{4.6}$$
where $\Pi^{(t)}$ is a $K \times K$ matrix that can depend on $V^{(t)}$.

Denote by $\hat{Z}$ the value of $Z^{(t)}$ at convergence, and by $\tilde{Q}$ the $n \times K$ matrix of the leading eigenvectors of A. Also, for a pair of full-rank matrices $U, V \in \mathbb{R}^{n\times K}$, define

the distance between the subspaces range(U) and range(V) by

$$\mathcal{L}(U, V) = \|U(U^T U)^{-1} U^T - V(V^T V)^{-1} V^T\|^2,$$

where $\|M\|$ is the spectral norm of a matrix $M \in \mathbb{B}^{n \times n}$. The previous algorithm provides a general framework for obtaining a regularized basis $\hat{Z}$ of the principal subspace of A that is close to range($\tilde{Q}$). On each iteration, the multiplication step (4.4) reduces the distance between the subspaces range($\tilde{Q}$) and range($Z^{(t)}$). Next, in the regularization step, some structure in the solution is enforced. If the distance between range($U^{(t)}$) and range($V^{(t)}$) is not too large (see Proposition 4.2), the distance between range($V^{(t)}$) and range($\tilde{Q}$) will be still under control. In Ma (2013), as the structure of interest is sparsity, a thresholding function is used as regularization. Finally, the numerical stability step is used to ensure identifiability. For example, the QR iteration algorithm uses a QR decomposition $Q^{(t)} R^{(t)} = V^{(t)}$ and sets $Z^{(t)} = V^{(t)} R^{(t)^{-1}}$, which is an orthogonal matrix.

The next proposition follows directly from Proposition 6.1 in Ma (2013), and provides conditions on the regularization step that control the distance between the subspace of $\hat{Z}$ and that of $\tilde{Q}$,

Proposition 4.2. *Let A be a $n \times n$ symmetric matrix with eigenvalues $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_K > \gamma_{K+1} \geq \ldots \geq \gamma_n$ and eigenvectors $q_1, \ldots, q_n$. Let $\tilde{Q} = (q_1 \cdots q_K)$ be the $n \times K$ matrix of the K leading eigenvectors. Suppose that the algorithm defined by the equations (4.4), (4.5) and (4.6) satisfies the following conditions in all the iterations*

- *The initial value $Z^{(0)}$ satisfies*

$$\mathcal{L}(Z^{(0)}, \tilde{Q}) \leq 1 - 4\left(\frac{\gamma_{K+1}}{\gamma_K + \gamma_{K+1}}\right)^2. \tag{4.7}$$

- *There is a real number ω with $0 < \omega < \frac{1}{2}\left(1 - \left|\frac{\gamma_{K+1}}{\gamma_K}\right|\right)^2$ for which the regularization step (4.5) satisfies*

$$\mathcal{L}(U^{(t)}, V^{(t)}) \leq \omega^2. \tag{4.8}$$

- *The matrix $Z^{(t)}$ is full rank*

If $\mathcal{L}(Z^{(t-1)}, \tilde{Q}) \leq \frac{4\omega^2}{(1-|\gamma_{K+1}/\gamma_K|)^2}$, then the value of $Z^{(t)}$ on the next iteration also satisfies

$$\mathcal{L}(Z^{(t)}, \tilde{Q}) \leq \frac{4\omega^2}{\left(1 - \left|\frac{\gamma_{K+1}}{\gamma_K}\right|\right)^2} \tag{4.9}$$

Otherwise,

$$\mathcal{L}(Z^{(t)}, \tilde{Q}) \leq \left(\frac{1 + |\gamma_{K+1}/\gamma_K|}{2}\right)^2 \mathcal{L}(Z^{(t-1)}, \tilde{Q}). \tag{4.10}$$

The previous proposition ensures that as long as the starting value is not extremely far from the eigenspace (equation (4.7)), the regularization does not severely alter the estimated basis $U^{(t)}$ at the current iteration (equation (4.8)), and $Z^{(t)}$ does not become a degenerate matrix, then the algorithm will converge to a solution $\hat{Z}$ that is not too far from the eigenspace of A. Equation (4.9) provides a bound on the approximation of the eigenspace of A by using $\hat{Z}$. This bound depends on the amount of regularization in (4.5) which is controlled by ω. Equation (4.10) measures the decrement of the the distance between $\text{ran}(Z^{(t)})$ and the eigenspace of A on each iteration. This decrement depends on the ratio of the eigenvalues γ_K and γ_{K+1}. In many situations, $\mathbb{E}A$ is a matrix of rank K, so this ratio will be generally small.

We use this framework to derive methods for estimating a membership matrix in the context of overlapping community detection.

4.2.2.1 Sparse eigenbasis estimation

The eigenvectors are usually the basis of interest in the estimation of principal subspaces, so many methods estimate an orthogonal basis of the principal subspace. However, our interest is in estimating a membership matrix that is not necessarily orthogonal. Here, we present an algorithm that converges to an arbitrary basis of the eigenspace. Dropping the orthogonality restriction makes the solution not identifiable in general, but by enforcing an appropriate level of regularization, it is possible to recover a basis with a membership matrix structure as in Theorem 4.1.

Orthogonal iteration is based on the fact that the multiplication step brings the matrix $Z^{(t)}$ close to the eigenspace of A, and the QR decomposition ensures that the solution is orthogonal and close to the eigenvectors. Iterative thresholding in SPCA (Ma, 2013) introduces a thresholding step and after some iterations, when the value of $Z^{(t)}$ is close to the eigenvectors, the role of the regularization becomes important, as it enforces some structure in the solution. Heuristically, the thresholding step in Ma (2013) works well because after the multiplication step, the columns of $AZ^{(t)}$ are still proportional to the columns of $Z^{(t)}$ when $Z^{(t)}$ is close to the eigenvectors, and hence applying a regularization function on them directly enforces some structure on the estimated eigenvectors. When $Z^{(t)}$ is not the leading eigenvectors, this is not necessarily the case since the values of $Z^{(t)}$ and $AZ^{(t)}$ might not be not directly related. The next proposition provides a relation between these two matrices. The

proof can be found on the Appendix.

Proposition 4.3. *Let A be a $n \times n$ symmetric matrix with eigendecomposition $A = Q\Lambda Q^T$ and $Q = [q_1 \cdots q_n]$. Suppose that $z_1, \ldots, z_K \in \mathbb{R}^n$ are vectors that satisfy $\|z_i\|_2 = 1$,*

$$span\{z_1, \ldots, z_K\} = span\{q_1, \ldots, q_K\}, \tag{4.11}$$

and $Z = [z_1 \cdots z_K]$ is a K rank matrix. Define $P = Z(Z^TZ)^{-1}$ and $\Gamma = P^TAP$ a $K \times K$ matrix. Therefore

$$AZ = Z\Gamma(Z^TZ).$$

The previous result suggest a way to construct an iterative thresholding algorithm for a general basis of the principal subspace. Suppose that for some t, $Z^{(t-1)}$ is close to the basis of interest. After the multiplication step $T^{(t)} = AZ^{(t-1)}$, we use Proposition 4.3 to introduce a step that "returns" $T^{(t)}$ to a value that is close to $Z^{(t-1)}$ in Frobenius norm but has the same range than $T^{(t)}$, by multiplying with some matrix Γ, so that $\text{ran}(T^{(t)}) = \text{ran}(T^{(t)}\Gamma)$, and hence $T^{(t)}$ is again close in Frobenius norm to the basis of interest. Defining the matrices $P^{(t)} = Z^{(t-1)}({Z^{(t-1)}}^{-1}Z^{(t-1)})$ and $\Gamma^{(t)} = {P^{(t)}}^TAP^{(t)}$, we perform a correction step

$$U^{(t)} = T^{(t)}\Gamma^{(t)}({Z^{(t-1)}}^TZ^{(t)})^{-1}.$$

Note that after this step, $\text{ran}(U^{(t)}) = \text{ran}(AZ^{(t)})$, so the information of the multiplication step is conserved, but in addition $U^{(t)}$ and $Z^{(t-1)}$ will be close to each other. After this, we apply the a regularization step to $U^{(t)}$. In particular, we use a thresholding function $\mathcal{S}$ and a threshold parameter $\lambda \in [0, 1)$ such that

$$\left(\mathcal{S}(U, \lambda)\right)_{ik} = \begin{cases} U_{ik} & \text{if } U_{ik} > \lambda \max_{j=1,\ldots,K} |U_{ij}|, \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

The function $\mathcal{S}$ performs hard-thresholding with different threshold for each row. In network models with a degree correction term, such as OCCAM, the rows of the membership matrix Z are proportional to the node degree correction term, but this degree is not related to the community structure. This is adjusted by using a different threshold level on each row that is proportional to the norm of the row. The threshold parameter λ controls the level of sparsity in the regularization step. As λ increases, more zeros are introduced in the solutions. Finally, a normalization step is performed for numerical stability. The algorithm also requires to specify a convergence criterion. We stop the algorithm after the relative difference in ℓ_2 norm between $Z^{(k)}$ and $Z^{(k-1)}$

is small,

$$\frac{\|Z^{(k)} - Z^{(k-1)}\|_2}{\|Z^{(k)}\|_2} < \epsilon.$$

These steps are summarized in Algorithm (4.1).

Algorithm 4.1 SPCA-eig: Sparse Eigenbasis Estimation

Input: Adjacency matrix A, eigenbasis dimension K, regularization parameter $\lambda \in [0, 1)$, initial estimator $Z^{(0)}$.

for $t = 1, \ldots$ until convergence **do**

Update $\Gamma^{(t)} = (Z^{(t-1)^T} Z^{(t-1)})^{-1} Z^{(t-1)^T} A Z^{(t-1)} (Z^{(t-1)^T} Z^{(t-1)})^{-1}$.

Multiplication step: $T^{(t)} = AZ^{(t-1)}$.

Identification step: $U^{(t)} = T^{(t)} \Gamma^{(t)^{-1}} (Z^{(t-1)^T} Z^{(t-1)})^{-1}$.

Thresholding: $Y^{(t)} = \mathcal{S}(U^{(t)}, \lambda)$.

Normalization: $Z^{(t)}_{\cdot,j} = \frac{1}{\|Y^{(t)}_{j\cdot}\|_2} Y^{(t)}_{\cdot,j}$, for $j = 1, \ldots, K$.

end for

return $\hat{Z}_\lambda = Z^{(t)}$ the value at convergence.

4.2.2.2 Community detection in networks with homogeneous degrees

Here, we present a second algorithm for sparse membership estimation. In addition to the sparsity in the memberships, here we also include an additional regularization step to remove the degree heterogeneity effect, so the matrix $Z^{(t)}$ has rows with constant norm $\|Z^{(t)}_{i\cdot}\|_1 = 1$. In practice we observed that this simplification gives very accurate results in terms of community detection. Note that after the multiplication step $T^{(t)} = AZ^{(t-1)}$, the columns of $T^{(t)}$ are proportional to the norm of the columns $Z^{(t-1)}$ which is in turn proportional to the estimated community sizes. Thus, in order to remove the effect of this parameter which is not meaningful for community detection, we divide over the norm of the columns of $Z^{(t-1)}$ $T^{(t)}$, so that $U^{(t)}_{\cdot j} = T^{(t)}_{\cdot j} / \|Z^{(t)}_{\cdot j}\|_1$ on each row. After this, we use the thresholding function (4.12) again in order to remove the small entries on the estimated basis. Finally, we use another regularization step to remove the effect of the node degree parameter, by normalizing each row. This steps are described in detail in Algorithm 4.2

The next theorem shows that in the case of the SBM, a matrix with the correct sparsity pattern is a fixed point of Algorithm 4.2

Theorem 4.1. Let A be a network generated from a stochastic blockmodel with K communities of sizes $C_1, \ldots, C_K$, membership matrix Z and connectivity matrix $B \in$

Algorithm 4.2 SPCA-CD: Community detection via sparse principal component analysis.

Input: Adjacency matrix A, number of communities K, regularization parameter $\lambda \in [0, 1)$, initial estimator $Z^{(0)}$.
for $t = 1, \ldots$ until convergence **do**
Multiplication step: $T^{(t)} = AZ^{(t-1)}$.
Column normalization: $U^{(t)}_{\cdot j} = \frac{1}{\|Z^{(t)}_{\cdot j}\|_1} T^{(t)}_{\cdot j}$, for $j = 1, \ldots, K$.
Thresholding: $V^{(t)} = \mathcal{S}(U^{(t)}, \lambda)$.
Row normalization: $Z^{(t)}_{i\cdot} = \frac{1}{\|V^{(t)}_{i\cdot}\|_1} V^{(t)}_{i\cdot}$, for $i = 1, \ldots, n$.
end for
return $\hat{Z}_\lambda = Z^{(t)}$ the value at convergence.

$[0, 1]^{K \times K}$ of the form

$$B_{rs} = \begin{cases} p, & \text{if } r = s, \\ q, & \text{if } r \neq s. \end{cases}$$

Suppose that for some $\lambda \in (0, 1)$,

$$\lambda p - q > \sqrt{\frac{\log(KN)}{C_{\min}}}, \tag{4.13}$$

with $C_{\min} = \min_{i=1,\ldots,K} C_i$, and there exists a constant $C^* > 0$ such that $C_{\max}/C_{\min} \leq C^*$. Then, Z is a stationary point of Algorithm 4.2 with probability at least $1 - N^c$, with $c > 0$ a constant that depends on λ.

4.2.3 Selection of threshold parameter

The methods we introduced depend on two parameters: the number of communities K and the threshold level λ. The parameter λ controls the sparsity of the membership matrix $\hat{Z}$. As λ increases, the membership solution $\hat{Z}_\lambda$ becomes sparser. In practice, looking to the path of solutions for different values of λ might be informative, as controlling the overlap size can result in different community assignments. On the other hand, it is important to select an appropriate value λ that provides a good fit to the data. We discuss two possible techniques for choosing this parameter, the Bayesian Information Criterion (BIC) and edge cross-validation (ECV) (Li et al., 2016). In this work, we assume that the number of communities is known in advance, but if this is not the case, multiple methods can be used to determine this number (Wang and Bickel, 2015; Le and Levina, 2015; Li et al., 2016).

4.2.3.1 Bayesian Information Criterion

The Bayesian Information Criterion (Schwarz et al., 1978) is a popular model selection method in statistics which balances the fit and parsimony of a model. Given a candidate estimator $\hat{Z}_\lambda$, the BIC is defined as

$$\mathrm{BIC}\left(\hat{Z}_\lambda\right) = -2\mathcal{L}(\hat{Z}_\lambda) + \mathrm{df}\left(\hat{Z}\right)\log(S), \tag{4.14}$$

where $\mathcal{L}(Z)$ is the value of the loglikelihood of the model at Z, $\mathrm{df}(Z)$ are the degrees of freedom of Z and S denotes the sample size. In a network with independent edges, $S = n(n-1)/2$. We can derive the other conditions if we assume that the model $W = \mathbb{E}A$ that generated A is OCCAM. Note that since the models we are comparing only change on the sparsity of Z, we can use as a proxy for the degrees of freedom the number of non-zeros in Z, that is $\mathrm{df}(Z) = \|\hat{Z}\|_0$. We additionally need an estimator $\hat{W}$ for the edge probabilities in the model. The membership matrix $\hat{Z}$ is an estimator of the eigenspace of W, and thus a natural estimator for W is the projection of A onto the subspace spanned by $\hat{Z}$. To obtain such projection, we need to find a matrix $\hat{B} \in \mathbb{R}$ that minimizes

$$\hat{B} = \arg\min_{B \in \mathbb{R}^{K\times K}} \|A - ZBZ^T\|_F^2.$$

By differentiating, it is easy to show that such $\hat{B}$ is given by

$$\hat{B} = \left(\hat{Z}^T\hat{Z}\right)^{-1} = \left((\hat{Z}_\lambda^\dagger)^T A \hat{Z}_\lambda^\dagger\right),$$

where $Z^\dagger = Z(Z^TZ)^{-1}$ is the pseudoinverse of Z. Hence, the estimator for W is given by

$$\hat{W}_\lambda = \hat{Z}_\lambda\left((\hat{Z}_\lambda^\dagger)^T A \hat{Z}_\lambda^\dagger\right)\hat{Z}_\lambda^T. \tag{4.15}$$

The loglikelihood of A for the model given by $\hat{Z}_\lambda$ can be estimated by

$$\mathcal{L}(\hat{Z}) = \sum_{i<j}\left(A_{ij}\log\left(\hat{W}_{ij}\right) + (1-A_{ij})\log\left(1-\hat{W}_{ij}\right)\right).$$

With these definitions, we calculate the BIC of a model given by $\hat{Z}_\lambda$ as

$$\mathrm{BIC}\left(\hat{Z}_\lambda\right) = -2\mathcal{L}(\hat{Z}_\lambda) + \|\hat{Z}_\lambda\|_0\log(n(n-1)/2), \tag{4.16}$$

and select a λ that minimizes the value of the equation given above.

4.2.3.2 Network cross-validation

Cross-validation (CV) is another popular approach for choosing a tuning parameter. When the data is a single network observed, CV is a challenging task, as splitting the data into folds does not result on sets of independent observations, or destroys the structure of the network itself.

In recent work, Li et al. (2016) developed a new cross-validation method for network data based on splitting the set of node pairs $\mathcal{N} = \{(i,j) : i,j \in \{1,\dots,N\}\}$ into L folds. For each fold l, the corresponding set of node pairs $\Omega_l \subset \mathcal{N}$ is retained, and the rest are used to fit the model. As some node pairs are not observed, the network is incomplete, but using a matrix completion algorithm the value of the node pairs in the hold out data can be estimated, and the resulting matrix $\hat{M}_l$ can be used to fit the model with each candidate tuning parameter. Then, the fit of the model can be measured in the node pairs Ω and a loss function evaluates the quality of this fit for each tuning parameter. The process is repeated for each of the folds, and the tuning parameter is selected based on the average loss. We use this procedure for choosing the threshold λ of our methods. Given a subset of node pairs Ω_l, we obtain the completed matrix $\hat{M}_l$ from A by using the procedure proposed in Li et al. (2016) based on the rank K truncated SVD. Then, for each candidate threshold λ, we fit the methods in order to get an eigenbasis $\hat{Z}_\lambda(\hat{M}_l)$, and we obtain an estimator $\hat{W}_\lambda(\hat{M})_l$ of W as in equation (4.15). With that estimator, we can compute the loss function on the hold out set as

$$\ell(A, \hat{W}_\lambda(\hat{M}_l); \Omega_l) = \frac{1}{|\Omega_l|} \sum_{(i,j)\in\Omega_l} (A_{ij} - \hat{W}_\lambda(\hat{M}_l)_{ij}). \tag{4.17}$$

We choose the tuning parameter λ that minimizes the average cross-validation loss $\ell(\lambda) = \frac{1}{L}\sum_{l=1}^{L} \ell(A, \hat{W}_\lambda(\hat{M}_l); \Omega_l)$.

In simulations, we observed that neither BIC or CV outperforms the other in choosing the correct model when the networks are generated from OCCAM, but in general both methods tend to select values that give a reasonable solution (see Section 4.3.2). On the other hand, CV is computationally, so BIC might be more convenient in some situations.

4.3 Simulations on synthetic networks

In this section, we use simulations to evaluate the performance of our methods, and compare them with other state of the art methods for overlapping community detection. In all scenarios, we generate networks from OCCAM, in which the edges of A are independent Bernoulli random variables, with expectation given in equation (4.1). We assume that each row Z_i of Z satisfies $\|Z_i\|_1 = 1$, so each node has the same expected degree. The difficulty of detecting overlapping communities is affected by multiple parameters in the generating model. We investigate the performance of the methods in different scenarios by varying the following properties.

a) Number of overlapping nodes. For a given percentage p, we select pn overlapping nodes. The rest of the nodes are assigned to only one community and distributed equally into all the communities. For most of the experiments we use $K = 3$ communities, and $1/4$ of the overlapping nodes are assigned to all communities with $Z_i = (1/3, 1/3, 1/3)$, while the rest are assigned to two communities j, k, with $Z_{ij} = Z_{ik} = 1/2$, equally distributing these nodes on all pairs (j, k). When $K > 3$, we only assign the overlapping nodes to two communities following the same process.

b) Connectivity between communities. We vary the ratio of the number of edges within and between communities by changing the non-diagonal elements of B, by parameterizing it as

$$B = (1 - \rho)I_{KK} + \rho 1_K 1_K^T,$$

and then varying $\rho \geq 0$ for a range of values. As ρ increases, the modularity of the network decreases, making community detection harder.

c) Average degree of the network. The average degree is controlled by α. For a given average degree d, we choose α such that the expected average degree $1_n^T \mathbb{E}A 1_n/n = \alpha(1_n^T \Theta Z B Z^T \Theta 1_n)/n$ is equal to d. Community detection is usually harder in sparse networks with low average degree (Le et al., 2015).

d) Node degree heterogeneity. We control the node degree by using different values in $\theta = \text{diag}(\Theta)$. In most simulations, we set $\theta = 1_n$, so all nodes have the same degree, but in some scenarios we also introduce hub nodes by setting $\theta_i = 5$ with probability 0.1.

e) Number of communities. We vary K, the number of communities in the network, and distribute the nodes equally in all the communities following the procedure described before. A larger number of communities can make the problem computationally more expensive.

In most simulation scenarios, we fix $n = 500$, and $K = 3$. All simulation settings are run 50 times, and the average result together with its 95% confidence band are reported.

Our main goal is to find the set of non-zero elements of the membership matrix. Many measures can be adopted to evaluate the quality of a solution. Here we use the *normalized variation of information* (NVI) introduced by Lancichinetti et al. (2009), which is specifically designed for problems with overlapping clusters. This measure is defined as follows. Given a pair of binary random vectors X, Y of size K, the normalized conditional entropy of X with respect to Y can be defined as

$$H_{norm}(X|Y) = \frac{1}{K}\sum_{k=1}^{K}\frac{H(X_k|Y_k)}{H(X_k)},$$

where $H(X_k)$ is the entropy of X_k and $H(X_k|Y_k)$ is the conditional entropy of X_k given Y_k, defined as

$$\begin{aligned} H(X_k) &= -P(X_k=0)\log P(X_k=0) - P(X_k=1)\log P(X_k=1) \quad (4.18)\\ H(X_k,Y_k) &= -\sum_{a=0}^{1}\sum_{b=0}^{1} P(X_k=a,Y_k=b)\log P(X_k=a,Y_k=b) \quad (4.19)\\ H(X_k|Y_k) &= H(X_k,Y_k) - H(Y_k), \end{aligned}$$

and the normalized variation of information between X and Y is defined as

$$N(X|Y) = 1 - \min_{\sigma}\frac{1}{2}\left(H_{norm}(\sigma(X)|Y) + H_{norm}(Y|\sigma(X))\right), \quad (4.20)$$

where σ is a permutation of the indexes to account for the fact that the binary assignments can be equivalent up to a permutation. When X and Y are independent, the NVI is equal to 0, and equals to 1 if $X = Y$. Now, for a given pair of membership matrices Z and $\tilde{Z}$ with binary entries, we can replace the probabilities in equations (4.18) and (4.19) with the sample versions using the rows of $\tilde{Z}$ and Z.

4.3.1 Choice of initial value

First, we evaluate the performance of our methods using different initialization values. Here, we simulate networks with the process described above, by fixing $n = 500$, $K = 3$, $d = 50$, and changing rho as well as the number of overlapping nodes. For both methods (SPCA-eig and SPCA-CD), we fit a path of solutions using a range of values of $\lambda = \{0.05, 0.1, \ldots, 0.95\}$, and report the solution with the highest NVI for each of the methods (note that here we are not performing any method for selecting λ). The initialization strategies we compare are the following.

- An overlapping community detection solution. We use the method to fit OCCAM proposed in Zhang et al. (2014).
- A non-overlapping community detection solution. We use SCORE (Jin, 2015), which is a spectral clustering method able to handle networks with heterogeneous degree.
- A random initialization, in which each node is randomly assigned to only one community. We run the algorithm by using five different random initializations, and the solution is chosen as the minimizer of the mean square error as in (4.2).

Figure 4.2 shows the performance of the initialization values in different scenarios. In general, all methods decrease their performance as the problem becomes harder, but when the initial solution is reasonably good (either by using an overlapping or non-overlapping community detection solution), the methods achieve its best performance. A random initialization also achieves good performance when the number of overlapping nodes is small. In general, a small threshold is required to identify the memberships of the overlapping nodes. When the threshold is small, a good initial value is important, since the methods might converge to eigenbasis that are not sparse. For the rest of our analysis, unless explicitly stated, we use the non-overlapping community detection solution (SCORE) to initialize the algorithm.

4.3.2 Tuning the threshold parameter

The tuning parameter λ controls the sparsity of the solution, and hence, the purity of the nodes. In practice, since the problem of community detection is unsupervised, it is often useful to look at different solutions with distinct levels of sparsity, so the path of solutions for different values of λ might be informative (see Section 4.4.1). However, it is also important to choose a value of λ that provides a good fit and a parsimonious

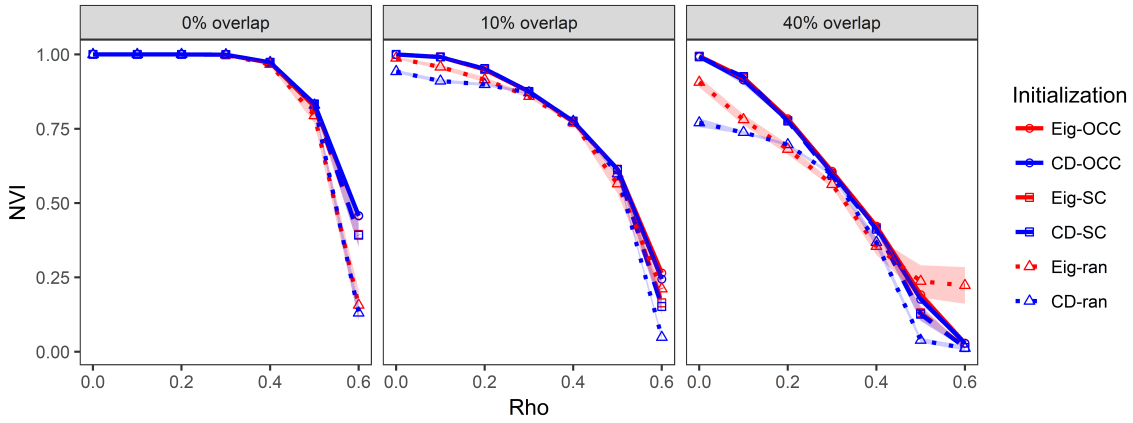


Figure 4.2: Performance of our methods measured by NVI using different initialization strategies (OCCAM, SCORE and a random initialization). The methods are evaluated on different scenarios, varying the connectivity between communities (x axis) and the size of the overlap (columns).

solution. Here, we evaluate the performance of the two strategies proposed in Section 4.2.3, BIC and CV in recovering the true set of non-zero memberships when networks are generated from OCCAM.

Figure 4.3 shows the results of the average performance measured by NVI of the two methods for tuning the threshold parameter. We observed that in general, BIC tends to select sparser solutions than CV. Hence, when the number of overlapping nodes is node large, so the true membership matrix is sparse, BIC shows a superior performance than CV, but when the overlap is large, CV usually performs better, specially for SPCA-CD. Since there is no clear advantage between the two methods in general, we use BIC in the following analysis, as BIC is computationally cheaper.

4.3.3 Comparison with existing methods

We compare the performance of several state of the art methods for overlapping community detection. We use the same simulation settings as in the previous section ($n = 500$ and $K = 3$), including sparser scenarios with $d = 20$, and networks with heterogeneous degree ($d = 50$ and 10% of hub nodes, as described at the beginning of the section).

We select a list of competitors based on their good performance reported in previous studies (see Zhang et al. (2014)), and include some other recent methods. We compare the fitting procedure of OCCAM (Zhang et al., 2014), the Ball-Karrer-Newman (BKN) model Ball et al. (2011), the overlapping stochastic blockmodel of Latouche et al. (2011) (OSBM), Bayesian non-negative matrix factorization (BNMF)

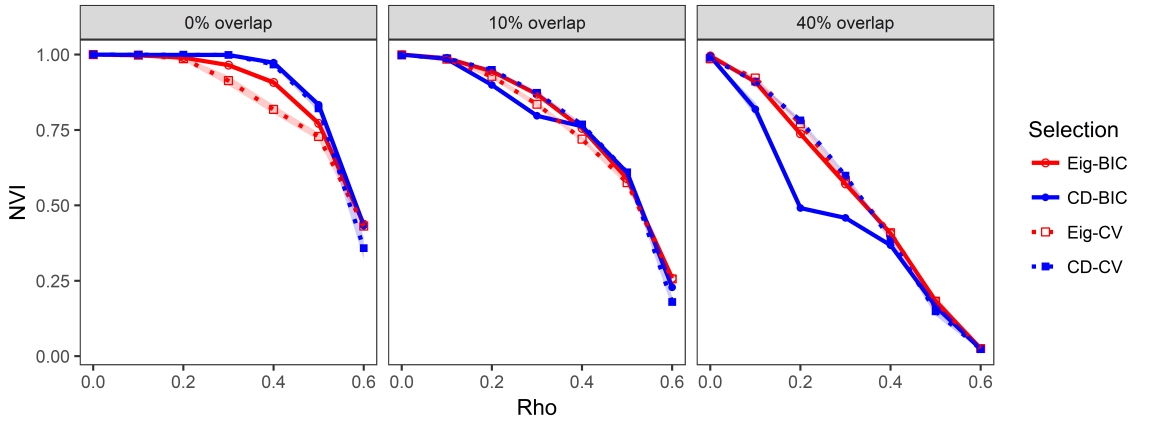


Figure 4.3: Performance of our methods measured by NVI for different parameter selection strategies (BIC and CV), on different scenarios. The methods are evaluated on different scenarios, varying the connectivity between communities (x axis) and the size of the overlap (columns).

by Psorakis et al. (2011), and the newly developed Mixed-SCORE (Jin et al., 2017). Some of these methods (OCCAM, BKN and Mixed-SCORE) return a continuous membership assignment, so we follow the approach of Zhang et al. (2014) and set to zero the values of the membership matrix $\hat{Z}$ that are smaller than $1/K$.

Figure 4.4 shows the average NVI of the methods as a function of ρ under different scenarios. Most methods show an excelent performance when $\rho = 0$, but as this parameter increases, the performance of all methods deteriorate. Our methods (SPCA-CD and SPCA-eig) generally achieve the best performance when the number of overlapping nodes is not large, and still achieve a competitive performance with 40% of overlapping nodes. OCCAM shows very good performance in general, which is reasonable since the networks were generated following that model. Mixed-SCORE has a good performance with no overlapping nodes, but the performance deteriorates fast as the overlap size increases. We should keep in mind that OCCAM and Mixed-SCORE are designed for estimating continuous memberships, and the threshold to obtain binary memberships might not be the optimal. When there are no overlapping nodes, many methods achieve a good performance. Note that even though classic algorithms for non-overlapping community detection can be used in this setting, the problem is more challenging here as the interest is in correctly assigning a node to no more than one community.

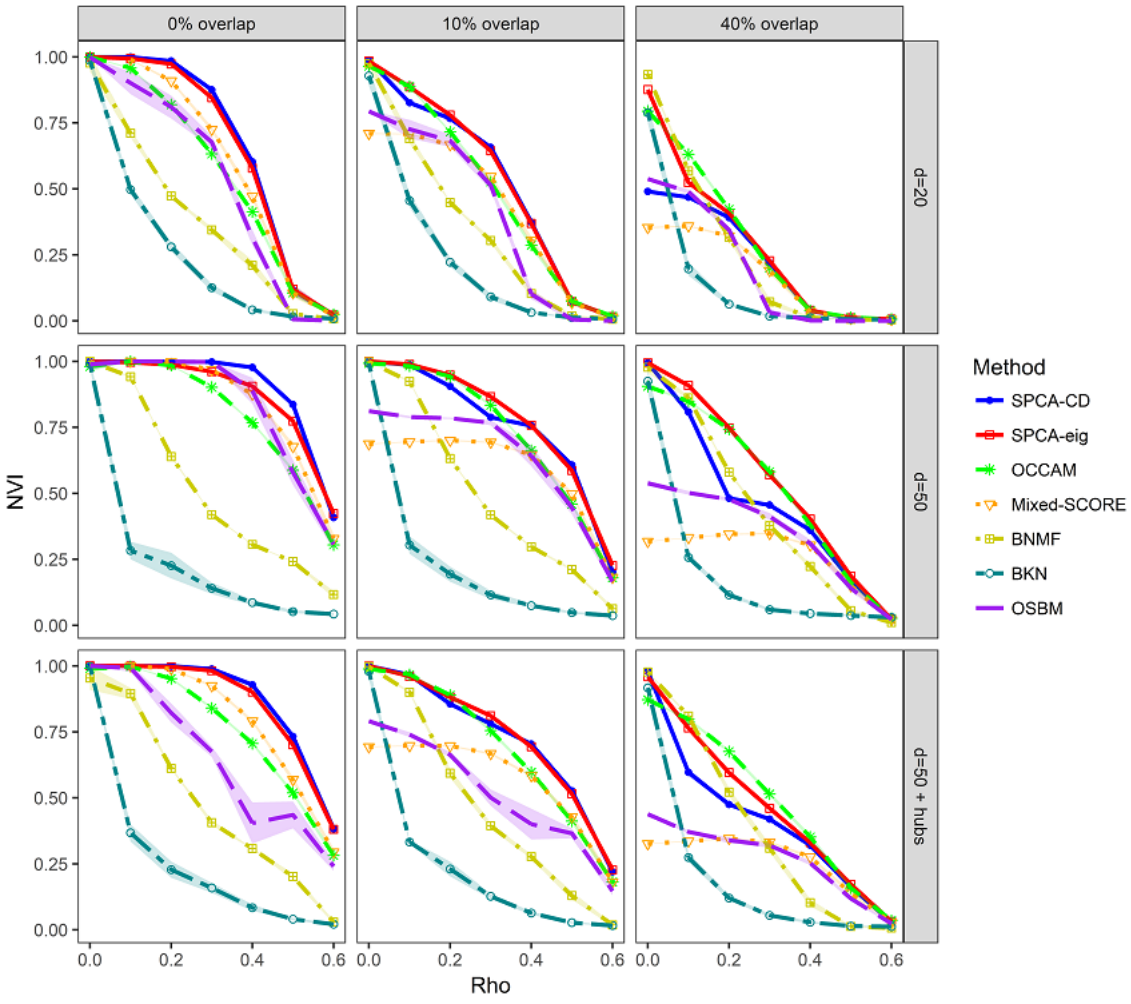


Figure 4.4: Performance of different methods for overlapping community detection measured by NVI. The methods are evaluated on different scenarios varying the ratio of edges between communities (x axis), the number of overlapping nodes (columns) and the node degree (rows).

4.3.4 Computational performance

We examine the performance of our methods in handling large networks. Spectral methods for overlapping and non-overlapping community detection are very popular, partly due to its scalability to large networks. The accuracy of those methods usually depends on the clustering algorithm, which in practice might require several restarts to improve their performance. In contrast, our methods based on sparse principal component analysis directly estimate the membership matrix without having to estimate the eigenvectors or perform a clustering step. Although our methods usually achieve the best performance when the threshold parameter is appropriately chosen, the algorithms provide reasonably good solutions using different values of λ.

Networks with different number of communities were simulated ($K = 3, 6$ and 10) but different sized of the network. The number of nodes was increased while keeping the average degree fixed to $d = 50$, with 10% overlapping nodes. Our methods usually obtain accurate solutions when λ is not so small, so for this purpose we fix $\lambda = 0.6$. We start SPCA-CD using a random membership matrix. SPCA-eig is more sensitive to the initial value, so we use the solution of SPCA-CD as starting point, but the running time of this algorithm is reported as the sum of both. We compare the performance of our methods with OCCAM, which uses a k-medians clustering to find the centroids of the overlapping communities. Since k-medians is a computationally expensive method and is not able to handle very large networks, we also report the performance of the solution obtained by replacing the clustering step with k-means. Additionally, we report the running time of calculating the K leading eigenvectors of the adjacency matrix, which is a starting step required by multiple spectral algorithms. All simulations are run using Matlab R2015a. The leading eigenvectors in OCCAM are computed using the standard Matlab function `eigs`$(\cdot, K)$.

The performance in terms of time and accuracy of different methods is shown in Figure 4.5. Our methods based on SPCA show a computational cost similar to calculating the K leading eigenvectors of the adjacency matrix, and when the number of communities is not large, our methods perform even faster. The original version of OCCAM based on k-medians clustering is limited in the size of networks it can handle, and when using k-means the computational cost is still larger than SPCA. Our methods produce solutions with great quality in all scenarios, while OCCAM deteriorates its performance when the number of communities increases. Note that in general the performance of all methods can be improved by using different random starting values, either for clustering in OCCAM or for initializing our methods,

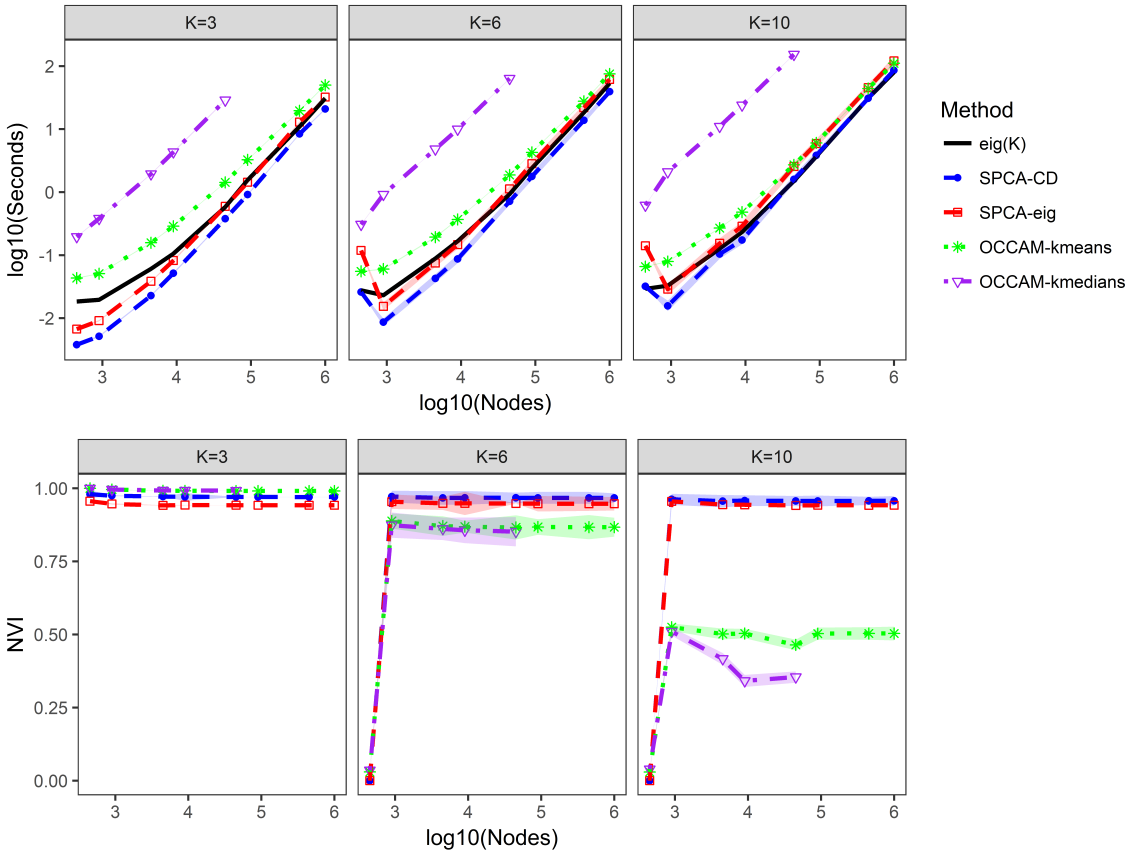


Figure 4.5: Performance of different methods in terms of running time (top row) and NVI (bottom row) as a function of the size of the network (x axis) for scenarios varying the number of communities (columns). We compare the performance of our methods (SPCA-CD and SPCA-eig), OCCAM with two different clustering procedures (k-means and k-medians), and the computational cost of calculating the K leading eigenvectors (eig(K)).

but this will increase the computational cost too. On the other hand, the threshold parameters used for our methods or for converting the solutions of OCCAM to sparse memberships might not be the optimal. However, the procedures we discussed for choosing a good threshold value are not able to scale for large networks, since computing the likelihood involves $O(n^2)$ parameters. A method for choosing a good threshold value in those settings is left as future work.

4.4 Evaluation on real-world networks

In this section, we evaluate the performance of our methods in some real-world networks. Zachary's karate club network (Zachary, 1977) is a classic example of a small

network with community structure, and we start with this data to illustrate the use and performance of the methods. Next, we evaluate the ability of the methods to recover community structure in the presence of hub nodes by fitting communities in the political blog network (Adamic and Glance, 2005), which is a popular benchmark to evaluate methods that can handle degree heterogeneity. Finally, we compare the performance of the method with other state of the art overlapping community detection algorithms in the ego network dataset (McAuley and Leskovec, 2012), which contain several social networks from Facebook, Twitter and Gplus in which nodes have been marked with a ground truth.

4.4.1 Zachary's karate club network

Zachary (1977) recorded the real-life interactions of 34 members of a karate club from a period of two years. During this period, the club split into two factions due to a conflict between the leaders. Not surprisingly, the edges of the network defined by the club members and their interactions reflect the structure of these two groups, as most of the connections appear between members of the same faction. These data become a popular example of community structure in a network, and we use it to illustrate our methods. Figure 4.6 shows a plot of the nodes and edges of the network, with the colors of the nodes corresponding to the club affiliations of the members.

We fit our methods to the karate club network. Using either BIC or CV to choose the threshold parameter, the selected solutions correspond to memberships composed by pure nodes only, and the estimated communities agree with the true club affiliations of the members. This is not surprising, as there are few edges connecting members of the two factions, but other methods for overlapping community detection assign some nodes to both communities. We also fit OCCAM and mixed-SCORE methods to this network. OCCAM assigns 17 nodes (50%) to both communities, and mixed-SCORE assigns 26 (76%). Thus, our methods are able to offer simpler solutions in this example.

As the threshold parameter λ changes, the methods select different memberships for the nodes. Both of our methods can identify community memberships, but SPCA-eig also provides information on the degree-correction parameter of the nodes. In Figure 4.7, we examine the effect of the threshold parameter λ for SPCA-eig. The plots show the path of the membership solutions for different values of λ. Each plot corresponds to one of the communities, and each trajectory represents the membership of one of the nodes as a function of λ. The colors indicate the faction of the

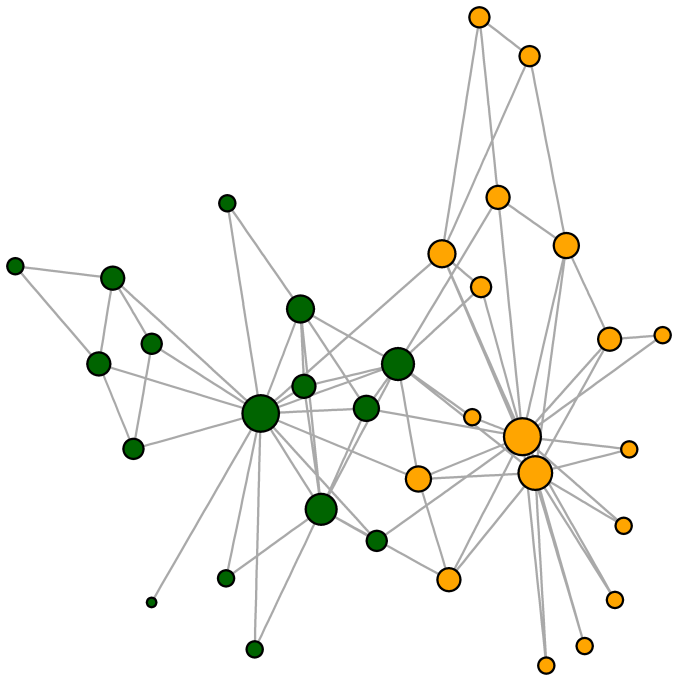


Figure 4.6: Zachary's karate club network, colored by club affiliation.

corresponding nodes, and the faction leaders are indicated with a dashed line. The value of the y axis indicates the association of the node to the corresponding community, and it is weighted by the degree-corrected parameter. On each community, the nodes with the larger values of y correspond to the faction leaders, which are connected to most of the nodes in the faction. For large values of λ, all nodes are assigned to only one community, but as λ decreases the membership matrix contains more non-zero values. Figure 4.6 shows the memberships for two different values of λ ($\lambda = 0.3$ and $\lambda = 0.4$). In both cases, overlapping nodes appear in the middle of both communities. Examining the solutions for different values of λ, we can identify some candidate nodes that might belong to both communities.

4.4.2 Political blog network

The political blog network (Adamic and Glance, 2005) represents the hyperlinks between 1490 political blogs during the time of the 2004 US presidential election. Blogs were manually labeled as liberal and conservative. It is expected that blogs corresponding to the same political view are going to have a similar connectivity pattern, but finding this structure is challenging due to the high degree heterogeneity (Jin,

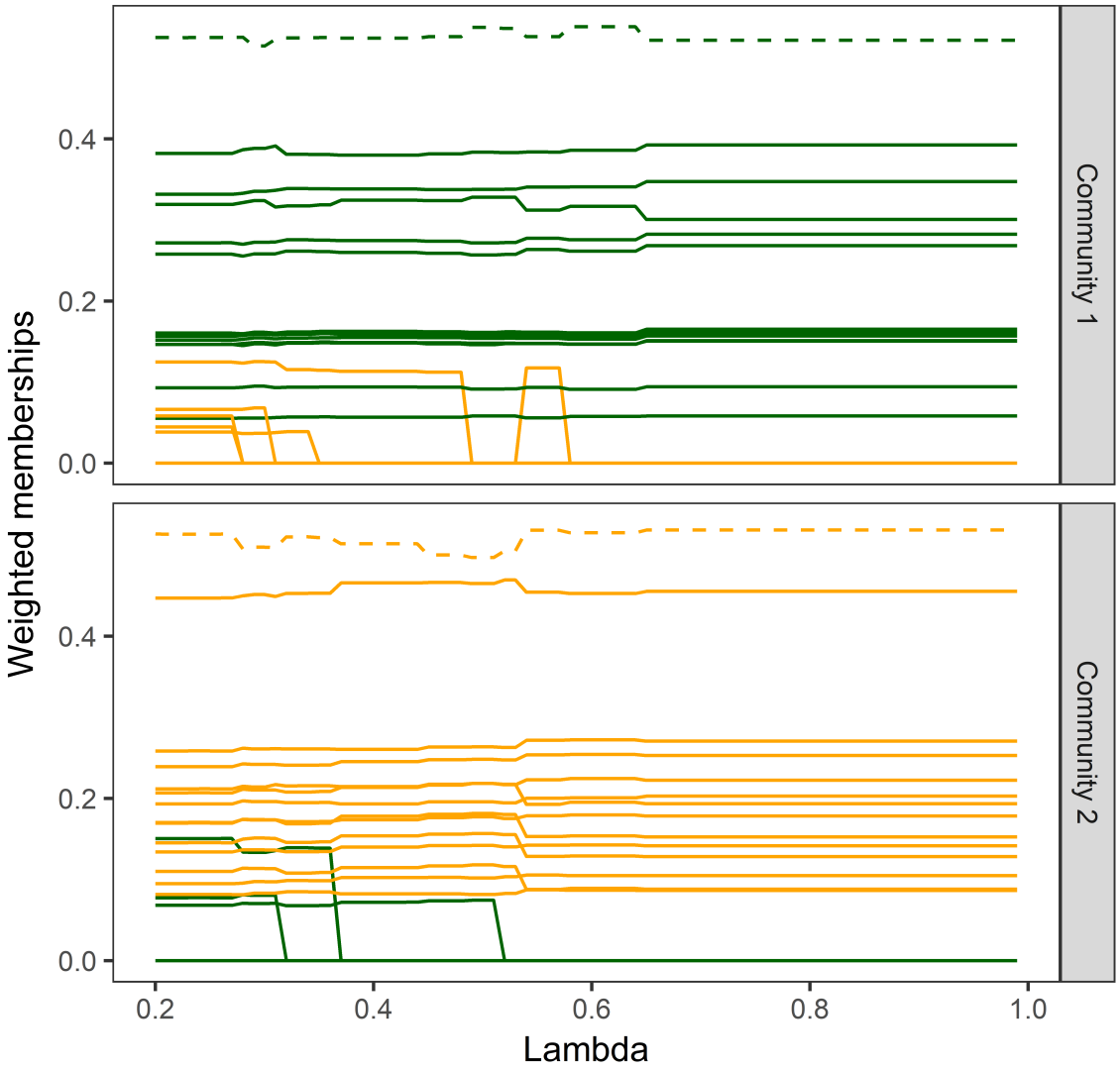
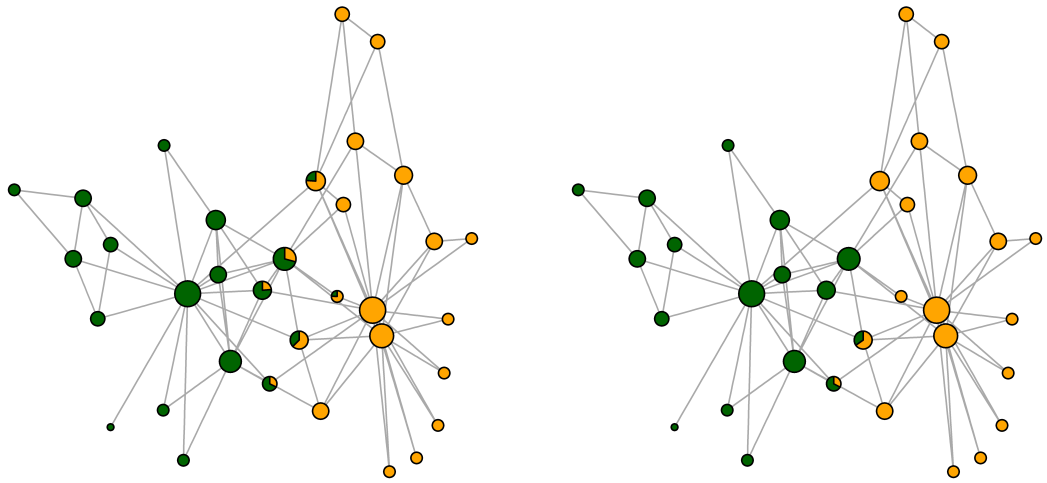


Figure 4.7: Node membership paths to each community (left and right) as a function of the thresholding parameter λ.



$\lambda = 0.3$ $\lambda = 0.4$

Figure 4.8: Solutions of SPCA-eig for different values of λ in the Zachary's karate club network data.

2015). Here, we evaluate the performance of our method for community detection on this network by using the labels of the blogs as ground truth.Note that, in this case, the assumed truth does not contain overlapping nodes, so in order to compare the community assignments, an overlapping membership Z can be converted to a binary membership $\tilde{Z}$ by taking $\tilde{Z}_{ij} = 1$ when $j = \arg\max_{k=1,\ldots,K} |Z_{ik}|$. In order to perform community detection, we first select the largest connected component of the network, which contains 1222 nodes, and we treat the edges as undirected, so $A_{ij} = 1$ if either blog i has an hyperlink to blog j or viceversa.

Figure 4.11 shows the plot of the political blog network colored by the blog labels (left side) and the colors obtained with overlapping communities using Algorithm 4.2 (Algorithm 4.1 obtains a similar solution). Using the tuning parameter selected by BIC, the algorithm assigns only 5 nodes to the overlap of the communities, and the binarized memberships miscluster 56 nodes in the wrong community, which is a similar performance reported in other methods that are able to operate in networks with heterogeneous (Jin, 2015). On the other hand, other overlapping community methods usually assign too many nodes to the overlap. In particular, the solution of OCCAM assigns 299 to both communities, while mixed-SCORE assigns 195.

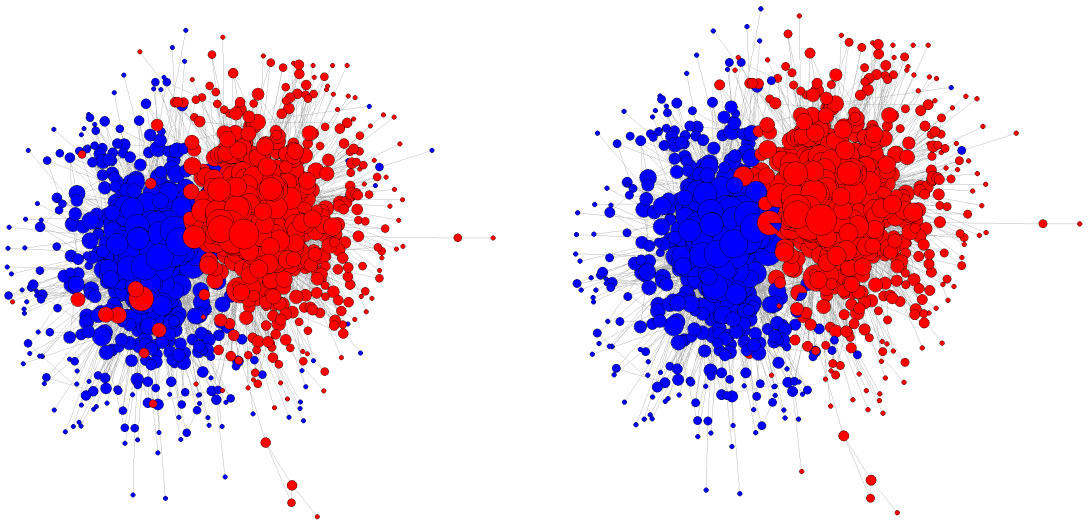


Figure 4.9: Nodes labeled by political view (blue = liberal, red = conservative).

Figure 4.10: Nodes labeled by fitted communities, with λ selected by BIC.

Figure 4.11: Political blog network (Adamic and Glance, 2005).

4.4.3 Evaluation of the methods on SNAP social networks

Social media platforms provide a rich source of data to study social interactions. (McAuley and Leskovec, 2012) presented a large collection of ego-networks from Facebook, Google Plus and Twitter. An ego-network represents the virtual friendships or following-follower relationships between a group of people that are connected to a central user. Those platforms allow the users to manually label or classify their friends into groups or social circles, and this information can be used as a ground truth to compare the performance of methods for detecting communities. In Zhang et al. (2014), several state-of-the-art overlapping community detection methods were compared on these data, showing a competitive performance of OCCAM. Here, we evaluate the performance of our methods with respect to OCCAM and the recently introduced mixed-SCORE method. We obtained a preprocessed version of the data directly from the first author of Zhang et al. (2014), which performed standard preprocessing steps to clean the networks (for the specific details of those steps, see Section 6 of Zhang et al. (2014)).

Table 4.1 shows the average performance measured by NVI of the different community detection methods we considered. For our methods, we fit the solutions with different values of λ and choose the best solution according to BIC, as we did in simulations. For OCCAM and mixed-SCORE, we threshold the continuous memberships

Dataset (sample size)	SPCA-Eig	SPCA-CD	OCCAM	M-SCORE
Facebook (6)	0.573 (0.090)	0.588 (0.088)	0.548 (0.118)	0.493 (0.137)
Google Plus (39)	0.408 (0.047)	0.427 (0.048)	0.501 (0.039)	0.475 (0.039)
Twitter (168)	0.435 (0.021)	0.477 (0.021)	0.450 (0.021)	0.391 (0.020)

Table 4.1: Average performance (and standard errors) of different methods for overlapping community detection in SNAP ego-networks.

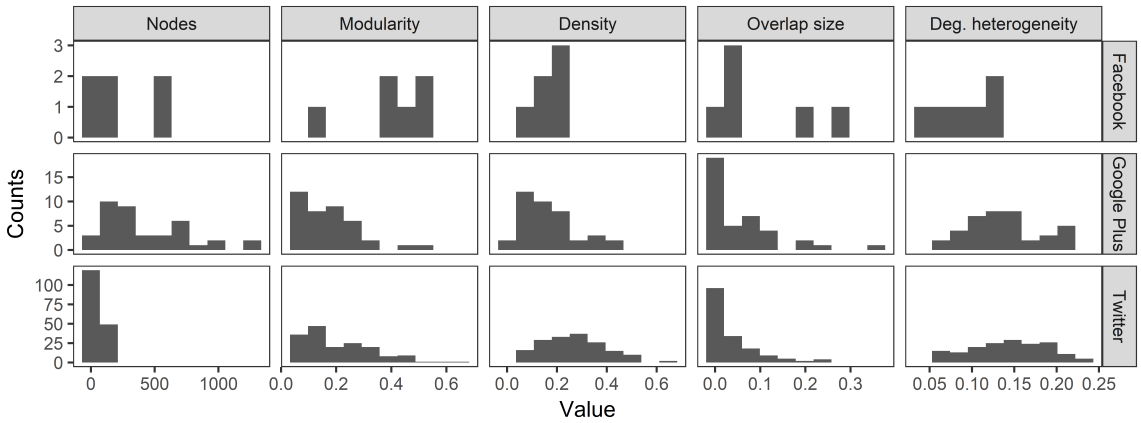


Figure 4.12: Histograms of summary statistics for SNAP ego-networks. (McAuley and Leskovec, 2012). The histograms show the number of nodes (n), Newman-Girvan modularity, density (number of edges divided by n^2), overlap size (percentage of nodes with overlapping memberships) and degree heterogeneity (standard deviation of the node degrees divided by n).

by $1/K$. Our methods (SPCA-eig and SPCA-CD) show a slightly better performance than the rest of the methods in the Facebook networks. SPCA-CD performs better than other methods on the Twitter networks, but SPCA-eig does not perform better than OCCAM. For Google Plus networks, OCCAM and mixed-SCORE have a clear advantadge. Figure 4.12 shows a histogram of different network summaries for each dataset. The histograms suggest that Google Plus networks might be harder for our methods, as these networks are generally larger but have a larger number of overlapping nodes. On the other hand, Facebook networks have higher modularity and lower overlap size, so these networks should be easier to cluster. In general, the results show the ability of all methods to recover reasonable overlapping communities, although our methods might achieve better performance when the memberships are actually sparse.

4.5 Discussion

In this chapter, we presented a new approach for overlapping community detection based on the estimation of a sparse basis of the principal space. The results of our methods in simulated data are encouraging, as the methods show good accuracy in estimating the overlapping memberships and are computationally very efficient, making them scalable to large networks. In future work, we plan to extend the analysis of the theoretical properties of our methods.

CHAPTER 5

Future work

Big data has opened many challenges and opportunities in statistics. In particular, this thesis explored different problems in the analysis of network data, and we focused in developing computationally efficient methods that can uncover parsimonious underlying structures of the data. Here, we discuss some open problems of interest .

Statistical inference in prediction with network covariates

Part of the work in this thesis focused on developing network-aware statistical methods for prediction with network-valued covariates. Combining the local edge information with the node or community structure can have benefits in estimation accuracy and interpretability. An important question for future work is how to effectively incorporate the network structure of the data in order to asses the significance of the predictors. Statistical inference in high-dimensional settings is an active research area, but in most cases the solutions are based on sparsity assumptions which are only able to report the significance of one edge at a time. As we explored in Chapter 3, brain network data is usually characterized by groups of correlated variables with similar predictive power, and finding those groups can improve prediction accuracy. Assessing the significance of these groups or other structures of interest for the analysis of network data is an interesting problem for future work.

Analysis of distributed network data

Distributed datasets are nowadays commonly present in different domains. Usually, communication costs are the bottleneck when dealing with distributed data. Due to

privacy or memory constraints, having access to all the data on a single computer is sometimes not possible, and new statistical methodologies require new efficient solutions with limited use of communication and bandwidth. In Arroyo and Hou (2016), we studied this problem in the context of graphical model estimation when samples are distributed across different machines. In many applications, networks are not directly observed, but inferred from a sample of observations, which is commonly done by estimating the underlying graphical model. In brain connectomics, for instance, a brain connectivity network is inferred based on a time series of brain activity measurements. Using debiasing and thresholding steps, we proposed an efficient algorithm that needs only one round of communication and limited bandwidth. Theoretically, the error of our solution is comparable to the solution when non-distributed data is available.

The analysis of distributed network data is particularly challenging. In some situations, computers in the distributed system might not have access to information about all the nodes in the network. For example, in sensor networks, a computer can only access information about its neighbors. Developing efficient methods for classic network problems, such as community detection, is a problem of interest in the context of distributed data.

Dimensionality reduction in multiple networks analysis

Many approaches for the statistical analysis of a single network have focused in fitting lower dimensional structures, such as communities (Holland et al., 1983) or latent positions (Hoff et al., 2002; Young and Scheinerman, 2007), that can help in understanding the relationship between the nodes. When analyzing a sample of networks with labeled nodes, the interest is not only in extracting the variability of the nodes, but also the subjects' variability, for which dimensionality reduction methods that can effectively deal with these two aspects are required. As in prediction problems, classic statistical tools for dimensionality reduction, such as PCA or ICA, can be directly applied to the vectorized adjacency matrix, but this approach ignores the network structure of the data. Incorporating these elements together is an interesting problem that we leave as a future work.

APPENDIX A

Network classification

A.1 Theoretical results of node group penalty

Here we prove the bounds on Frobenius norm error and probability of support selection in Proposition 2.1, following the framework of Lee et al. (2015) based on geometrical decomposability. A penalty Ω is geometrically decomposable if it can be written as

$$\Omega(B) = h_A(B) + h_I(B) + h_{E^\perp}(B)$$

for all B, with A, I closed convex sets, E a subspace, and $h_C(B) = \sup\{\langle Y, B\rangle \,|Y \in C\}$ the support function on C.

The proof proceeds in the following steps. Lemma 1 shows that an equivalent form of our penalty (2.2) is geometrically decomposable, allowing us to use the framework of Lee et al. (2015). Lemma 3 shows the Assumption 2.2 together with a lower bound on ρ imply that the irrepresentability assumption of Lee et al. (2015) holds. Assumption 2.2 is directly on the entries of the loss Hessian, which simplifies the very general form of the assumption in Lee et al. (2015). Lemma 3 gives a bound on the entries of the loss gradient under the sub-Gaussianity assumption 2.3. Lemma 4 gives explicit bounds for the compatibility constants that appear on Theorem 1 of Lee et al. (2015). Finally, we combine these results to prove Proposition 2.1.

Without loss of generality, to simplify notation we assume that $\mathcal{G} = \{1, \ldots, G\}$, that is, the active subgraph is in the first G rows of the matrix.

Lemma 1. *The penalty* (2.2) *can be written as geometrically decomposable.*

Proof of Lemma 1. We use an equivalent formulation of the penalty in which every coefficient is penalized only once. Let $B^{(1)}, B^{(2)} \in \mathbb{R}^{N\times N}$ be matrices such that the

upper triangular part of $B^{(2)}$ and the diagonals of $B^{(1)}$ and $B^{(2)}$ are zero. Define

$$\tilde{\Omega}\left(B^{(1)}, B^{(2)}\right) = \sum_{i=1}^{N} \|B^{(1)}_{(i)}\|_2 + \rho\|B^{(1)}\|_1,$$

$$E = \{(B^{(1)}, B^{(2)}) \in \mathbb{R}^{N\times 2N} \ : \ B^{(1)} = B^{(1)^T}, B^{(2)}_{ij} = B^{(1)}_{ij}, \text{ for } i < j \text{ and } B^{(2)}_{ij} = 0 \text{ for } i \geq j\}.$$

Denote by R the transformation from $\mathbb{R}^{N\times N}$ to $\mathbb{R}^{N\times 2N}$ that replicates entires appropriately,

$$(RB)_{ij} = \begin{cases} B_{ij} & \text{if } 1 \leq j \leq N \\ B_{i(j-N)} & \text{if } j > N. \end{cases} \tag{A.1}$$

Therefore, for any $B \in \mathbb{R}^{N\times N}$, we can uniquely define $RB = (B^{(1)}, B^{(2)})$ such that $\Omega(B) = \tilde{\Omega}(B^{(1)}, B^{(2)})$. We then show that $\tilde{\Omega}$ is geometrically decomposable. Moreover, for any $(B^{(1)}, B^{(2)}) \in E$ we can define R^{-1}, so the penalties Ω and $\tilde{\Omega}$ on E are equivalent. Define the sets $A, I \subset \mathbb{R}^{N\times 2N}$ such that

$$\begin{aligned} A &= \Big\{(B^{(1)}, B^{(2)}) : \max_{i\in\mathcal{G}} \|B^{(1)}_{(i)}\|_2 \leq 1, \max_{i\in\mathcal{G}^C} \|B^{(1)}_{(i)}\|_2 = 0, \\ &\qquad \max|B^{(2)}_{ij}| \leq \rho, B^{(2)}_{ij} = 0, (i,j) \in (\mathcal{G}\times\mathcal{G})^C\Big\}, \\ I &= \Big\{(B^{(1)}, B^{(2)}) : \max_{i\in\mathcal{G}^C} \|B^{(1)}_{(i)}\|_2 \leq 1, \max_{i\in\mathcal{G}} \|B^{(1)}_{(i)}\|_2 = 0, \\ &\qquad \max|B^{(2)}_{ij}| \leq \rho, B^{(2)}_{ij} = 0, (i,j) \in \mathcal{G}\times\mathcal{G}\Big\}. \end{aligned}$$

If $\left\langle Y, (B^{(1)}, B^{(2)})\right\rangle = \mathrm{Tr}(Y^{(1)}B^{(1)^T}) + \mathrm{Tr}(Y^{(2)}B^{(2)^T})$, combining the arguments of Lee et al. (2015) for lasso and group lasso penalties,

$$\begin{aligned} h_A\left(B^{(1)}, B^{(2)}\right) &= \sum_{i\in\mathcal{G}} \|B^{(1)}_{(i)}\|_2 + \rho \sum_{(i,j)\in\mathcal{G}\times\mathcal{G}} |B^{(2)}_{ij}|, \\ h_I\left(B^{(1)}, B^{(2)}\right) &= \sum_{i\in\mathcal{G}^C} \|B^{(1)}_{(i)}\|_2 + \rho \sum_{(i,j)\in(\mathcal{G}\times\mathcal{G})^C} |B^{(2)}_{ij}|, \\ h_E\left(B^{(1)}, B^{(2)}\right) &= \begin{cases} 0 & \text{if } \left(B^{(1)}, B^{(2)}\right) \in E \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

Hence, Ω can be written as a geometrically decomposable penalty

$$\Omega(B) = \tilde{\Omega}\left(B^{(1)}, B^{(2)}\right) = \lambda\left(h_A\left(B^{(1)}, B^{(2)}\right) + h_I\left(B^{(1)}, B^{(2)}\right) + h_E\left(B^{(1)}, B^{(2)}\right)\right).$$

□

We introduce some notation in order to state the irrepresentability condition of

Lee et al. (2015). For a set $F \subset \mathbb{R}^{N\times 2N}$ and $Y \in \mathbb{R}^{N\times 2N}$, denote by $\gamma_F(Y) = \inf\{\lambda > 0 : Y \in F\}$ the gauge function on C. Thus,

$$\gamma_I\left(B^{(1)}, B^{(2)}\right) = \max\left\{\max_{i\in\mathcal{G}^C} \|B^{(1)}_{(i)}\|_2,\ \frac{1}{\rho}\max_{(i,j)\in(\mathcal{G}\times\mathcal{G})^C} |B^{(2)}_{ij}|\right\} + \mathbf{1}_I\left(B^{(1)}, B^{(2)}\right),$$

where $\mathbf{1}_I(B) = 0$ if $B \in I$ and ∞ otherwise. Define

$$V(Z) = \inf\{\gamma_I(Y) : \quad Z - Y \in E^\perp, Y \in \mathbb{R}^{N\times 2N}\}$$

for $Z \in \mathbb{R}^{N\times 2N}$. Let $\tilde{\mathcal{M}} = E \cap \operatorname{span}(I)^\perp$ be the set of matrices with correct support in the extended space $\mathbb{R}^{N\times 2N}$, similarly to $\mathcal{M}$ in (2.13). Denote by P_M and $P_{M^\perp}$ the projections onto $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}^\perp$. Define the function $\mathcal{H}(Z) : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ as

$$\mathcal{H}(Z)_{ij} = \begin{cases} \operatorname{Tr}\left(H_{(i,j),\mathcal{G}}(P_M Z)_{\mathcal{G},\mathcal{G}}\right) & \text{if } j \in \mathcal{G}, \\ 0 & \text{otherwise.} \end{cases}$$

where $H_{(i,j),\mathcal{G}}$ is the matrix defined in (2.14). The Irrepresentability Assumption 3.2 of Lee et al. (2015) requires the existence of $0 < \tilde{\tau} < 1$ such that

$$\sup_{Z\in A} V\left(P_{M^\perp}\left(R\mathcal{H}(Z) - Z\right)\right) < 1 - \tilde{\tau}. \tag{A.2}$$

For a support function h, denote by $\partial h(M) = \bigcup_{Y\in M} \partial h(Y)$ the set of subdifferentials of h in M. Note that $\partial h_A(M) = A$, since $0 \in M$ and $\partial h_A(0) = A$.

Lemma 2. *If Assumption 2.2 holds and $\rho > \frac{1}{\tau} - \frac{1}{\sqrt{G}}$, then there exists $0 < \tilde{\tau} < 1$ such that* (A.2) *holds.*

Proof of Lemma 2. Since V is sublinear (Lemma 3.3 of Lee et al. (2015)),

$$\sup_{Z\in A} V\left(P_{M^\perp}\left(R\mathcal{H}(Z) - Z\right)\right) \le \sup_{Z\in A} V\left(P_{M^\perp}\left(R\mathcal{H}(Z)\right)\right) + \sup_{Z\in A} V\left(P_{M^\perp} Z\right). \tag{A.3}$$

To bound the first term, note that $E^\perp = \{(Z^{(1)}, Z^{(2)}) | Z^{(1)}_{ij} + Z^{(1)}_{ji} + Z^{(2)}_{ij} = 0, 1 \le j <$

$i \le N\}$.

$$\begin{aligned} V(Y^{(1)},Y^{(2)}) &= \inf\left\{\gamma\left(U^{(1)},U^{(2)}\right):\ (U_{ij}^{(1)}-Y_{ij}^{(1)})+(U_{ji}^{(1)}-Y_{ji}^{(1)})+(U_{ij}^{(2)}-Y_{ij}^{(2)})=0,\right. \\ &\quad \left. 1\le j<i\le N\right\} \\ &\le \inf\left\{\gamma(U^{(1)},U^{(2)}):\ \ U_{(i)}^{(1)}=Y_{(i)}^{(1)}, i\in\mathcal{G}^C; U_{\mathcal{G}^C,\mathcal{G}^C}^{(2)}=Y_{\mathcal{G}^C,\mathcal{G}^C}^{(2)};\right. \\ &\quad \left.(U^{(1)},U^{(2)})-(Y^{(1)},Y^{(2)})\in E^{\perp}\right\} \\ &\le \max\left\{\max_{i\in\mathcal{G}^C}\|Y_{(i)}^{(1)}\|_2,\ \frac{1}{\rho}\left\|Y_{\mathcal{G}^C,\mathcal{G}^C}^{(2)}\right\|_{\infty}\right\}. \end{aligned}$$

Therefore,

$$\begin{aligned} V\left(P_{M^{\perp}}\left(R\mathcal{H}(Z)\right)\right) &\le \max\left\{\max_{i\in\mathcal{G}^C}\left\|\left(P_{M^{\perp}}\left(R\mathcal{H}(Z)\right)\right)_{(i)}^{(1)}\right\|_2,\ \frac{1}{\rho}\left\|\left(P_{M^{\perp}}\left(R\mathcal{H}(Z)\right)\right)_{\mathcal{G}^C,\mathcal{G}^C}^{(2)}\right\|_{\infty}\right\} \\ &= \max_{i\in\mathcal{G}^C}\left\|\mathcal{H}(Z)_{(i)}\right\|_2, \end{aligned}$$

which implies that

$$\begin{aligned} \sup_{Z\in A} V\left(P_{M^{\perp}}\left(R\mathcal{H}(Z)\right)\right) &\le \sup_{Z\in A}\left\{\max_{i\in\mathcal{G}^C}\left\|\mathcal{H}(Z)_{(i)}\right\|_2\right\} \\ &\le \sup_{B\in\mathbb{R}^{G\times G},\|B_{(i)}\|_2\le 1}\left\{\max_{i\in\mathcal{G}^C}\left\|\left(\mathrm{Tr}\left(H_{(i,j),\mathcal{G}}B\right)\right)_{j=1}^{N}\right\|_2\right\} \\ &\le \max_{i\in\mathcal{G}^C}\left\|\left(\sum_{k=1}^{G}\|(H_{(i,j),\mathcal{G}})_{k\cdot}\|_2\right)_{j=1}^{N}\right\|_2 = 1-\tau. \end{aligned} \tag{A.4}$$

Let $Z=(Z^{(1)},Z^{(2)})\in A$. Without loss of generality, assume that $Z_{\mathcal{G},\mathcal{G}}^{(1)}=Z_{\mathcal{G},\mathcal{G}}^{(2)}=0$ (note that these entries do not change $V(P_{M^{\perp}}Z)$). Therefore, $P_{M^{\perp}}Z=Z$. Hence,

$$\begin{aligned} V(Z) &= \inf\left\{\gamma\left(U^{(1)},U^{(2)}\right):\ \ \left(U^{(1)},U^{(2)}\right)\in I,\ \left(U^{(1)},U^{(2)}\right)-\left(Z^{(1)},Z^{(2)}\right)\in E\right\} \\ &= \inf\left\{\gamma\left(U^{(1)},U^{(2)}\right):\ \ U_{ij}^{(1)}+U_{ij}^{(2)}=Z_{ji}^{(1)}, 1\le j\le G, G<i\le N\right\} \\ &= \inf\left\{\max\left\{\max_{i\in\mathcal{G}^C}\|U_{(i)}^{(1)}\|_2, \frac{1}{\rho}\max_{\substack{1\le j\le G\\ G<i\le N}}|U_{ij}^{(2)}|\right\}:\ U_{ij}^{(1)}+U_{ij}^{(2)}=Z_{ji}^{(1)}, 1\le j\le G, G<i\le N\right\} \\ &\le \inf\left\{\max\left\{\max_{i\in\mathcal{G}^C}\|U_{(i)}^{(1)}\|_2, \frac{1}{\rho}\max_{\substack{1\le j\le G\\ G<i\le N}}|U_{ij}^{(2)}|\right\}:\ \ U_{ij}^{(1)}+U_{ij}^{(2)}=1, 1\le j\le G, G<i\le N\right\} \end{aligned}$$

The last bound from $|Z_{ji}^{(1)}|\le 1$ and no longer depends on Z. It is easy to see that

the minimum is attained when, for each $i > G$,

$$\left\|U^{(1)}_{(i)}\right\|_2 = \frac{1}{\rho}\left|U^{(2)}_{ij}\right|, \quad 1 \leq j \leq G,$$

and therefore

$$V(Z) \leq \frac{\sqrt{G}}{1+\rho\sqrt{G}}. \tag{A.5}$$

Moreover, if $Z^* \in A$ is defined such that $(Z^*)^{(1)}_{G+1,i} = 1$ for $i = 1, \ldots, G$ and 0 elsewhere, then $V(Z^*)$ achieves this bound, which shows that $\rho > 1 - \frac{1}{\sqrt{G}}$ is a necessary condition for the irrepresentability to hold, even in the case where the entries of the Hessian that denote the information between active and inactive edges is zero. Therefore, plugging the bounds (A.4) and (A.5) into equation (A.3), we obtain (A.2) holds as long as $1 - \tau + \frac{\sqrt{G}}{1+\rho\sqrt{G}} < 1$, which implies that $\rho > \frac{1}{\tau} - \frac{1}{\sqrt{G}}$. □

The next lemma establishes a bound on the dual norm of Ω of the loss gradient function under a sub-Gaussian assumption. Let Ω^* denote the dual norm of Ω, so $\Omega^*(B) = \sup\{\langle Y, B\rangle \mid Y \in C, \Omega(Y) \leq 1\}$.

Lemma 3. *Under Assumption 2.3,*

$$\mathbb{P}\left(\Omega^*(\nabla\ell(B^\star)) > t\right) \leq 2N^2 \min\left\{\exp\left(-\frac{n(1+\rho)^2t^2}{N(\sigma^2)}\right), \exp\left(-\frac{n\rho^2t^2}{\sigma^2}\right)\right\}. \tag{A.6}$$

Proof of Lemma 3. By Hoeffding's inequality for sub-Gaussian variables, for all j, k and $t > 0$,

$$\mathbb{P}\left(\left|\nabla_{jk}\ell\left(B^\star\right)\right| > t\right) \leq \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}\nabla_{jk}\ell_i\left(B^\star\right)\right| > t\right) \leq 2\exp\left(-n\frac{t^2}{\sigma^2}\right).$$

Note that $(1+\rho)\sum_{i=1}^{N}\|B_{(i)}\|_2 \leq \Omega(B)$. Let $\Phi(B) = \frac{1}{1+\rho}\max_{i=1,\ldots N}\|B_{(i)}\|_2$. Thus,

$$\Omega^*(B) \leq \sup_{Y\in\mathbb{R}^{N\times N}}\left\{\mathrm{Tr}(YB) : \Omega_{\rho=0}(Y) \leq \frac{1}{1+\rho}\right\} = \Phi(B). \tag{A.7}$$

In a similar manner, $\rho\|B\|_1 \leq \Omega(B)$. Setting $\Xi(B) = \frac{1}{\rho}\|B\|_\infty$, we have

$$\Omega^*(B) \leq \Xi(B). \tag{A.8}$$

Using (A.7) and setting $\Lambda = \nabla \ell(B^\star)$,

$$\begin{aligned}
\mathbb{P}\left(\Omega^*(\Lambda) > t\right) &\leq \mathbb{P}\left(\Phi(\Lambda) > t\right) \\
&= \mathbb{P}\left(\max_{1\leq i\leq N} \|\Lambda_{(i)}\|_2 > (1+\rho)t\right) \\
&\leq \mathbb{P}\left(\max_{1\leq i\leq N} \max_{j\neq i} |\Lambda_{ij}| > (1+\rho)\frac{t}{\sqrt{N}}\right) \\
&\leq 2N(N-1)\exp\left(-\frac{n(1+\rho)^2 t^2}{2\sigma^2(N-1)}\right),
\end{aligned}$$

the last inequality obtained by arguments similar to Lemma 4.3 of Lee et al. (2015). In the same way, we can also bound the previous quantity using (A.8) by

$$\begin{aligned}
\mathbb{P}\left(\Omega^*(\Lambda) > t\right) &\leq \mathbb{P}\left(\Xi(\Lambda) > t\right) \\
&= \mathbb{P}\left(\|\Lambda_{(i)}\|_\infty > \rho t\right) \\
&\leq N(N-1)\exp\left(-\frac{n\rho^2 t^2}{2\sigma^2}\right).
\end{aligned}$$

Combining (A.9) and (A.9), we can obtain equation (A.6). □

For a semi-norm $\Psi : \mathbb{R}^{N\times N} \to \mathbb{R}$, let κ_Ω be the compatibility constant between Ψ and the Frobenius norm, defined as

$$\kappa_\Psi = \sup\left\{\Psi(B) : \|B\|_2 \leq 1, B \in M\right\},$$

and let κ_{IC} be the compatibility constant between the irrepresentable term and the dual norm Ω^* given by

$$\kappa_{\mathrm{IC}} = \sup\left\{V(P_{M^\perp}(R\mathcal{H}Z - Z)) : \Omega^*(Z) \leq 1\right\} .$$

Lemma 4. *The following bounds on the compatibility constants hold:*

$$\begin{aligned}
\kappa_\Omega &= \sqrt{G} + \rho\sqrt{G(G-1)}, \\
\kappa_{\Omega^*} &\leq \frac{1}{1+\rho}, \\
\kappa_{IC} &\leq 3 - \tau.
\end{aligned}$$

Proof of Lemma 4. Note that $\Omega(Y)$ is maximized on $\{Y : \|Y\|_2 \leq 1\}$ when all entries

of Y have magnitude equal to $\frac{1}{\sqrt{G(G-1)}}$. Therefore

$$\kappa_\Omega = G\sqrt{\frac{G-1}{G(G-1)}} + \rho\frac{G(G-1)}{\sqrt{G(G-1)}} = \sqrt{G} + \rho\sqrt{G(G-1)}. \tag{A.9}$$

Similarly, (A.7) implies

$$\kappa_{\Omega^*} \leq \sup\left\{\frac{1}{1+\rho}\max_{i\in\mathcal{G}}\|B_{(i)}\|_2 : \|B\|_2 \leq 1\right\} \leq \frac{1}{1+\rho}. \tag{A.10}$$

Finally,

$$\begin{aligned}
\kappa_{IC} &= \sup\left\{V(P_{M^\perp}(R\mathcal{H}Z - Z)) : \Omega^*(Z) \leq 1\right\} \\
&\leq \sup\left\{V(P_{M^\perp}(R\mathcal{H}Z)) : \Omega^*(Z) \leq 1\right\} + \sup\left\{V(P_{M^\perp}(Z)) : \Omega^*(Z) \leq 1\right\} \\
&\leq (1-\tau) + 2 = 3 - \tau.
\end{aligned}$$

□

Proof of Proposition 2.1. Part (a). Since $\hat{B}$ minimizes the problem (2.4),

$$\ell(\hat{B}) + \lambda\Omega(\hat{B}) \leq \ell(B^\star) + \lambda\Omega(B^\star).$$

Rearranging the terms, using Assumption 2.1, by the triangle inequality and the generalized Cauchy-Schwarz inequality,

$$\begin{aligned}
0 &\geq \ell(\hat{B}) - \ell(B^\star) + \Omega(\hat{B}) - \Omega(B^\star) \\
&\geq \left\langle \nabla\ell(B^\star)^T, \hat{B} - B^\star \right\rangle + \frac{m}{2}\|\hat{B} - B^\star\|_2^2 - \Omega\left(\hat{B} - B^\star\right) \\
&\geq -\Omega\left(\hat{B} - B^\star\right)\Omega^*\left(\nabla\ell(B^\star)\right) - \Omega\left(\hat{B} - B^\star\right) + \frac{m}{2}\|\hat{B} - B^\star\|_2^2.
\end{aligned} \tag{A.11}$$

Using the argument for computing κ_Ω in (A.9), $\Omega(Y) \leq \left(\sqrt{N} + \rho\sqrt{N(N-1)}\right)\|Y\|_2$. Rearranging the terms in (A.11),

$$\|\hat{B} - B^\star\|_2 \leq \frac{2}{m}\left(\sqrt{N} + \rho\sqrt{N(N-1)}\right)\left(\lambda + \Omega^*\left(\hat{B} - B^\star\right)\right).$$

For any ρ, setting $\lambda = 2\sqrt{\frac{\sigma^2\log N}{n}}\min\left(\frac{\sqrt{N}}{1+\rho}, \frac{1}{\rho}\right)$, by Lemma 3, with probability at least

$1 - 2/N$,

$$\begin{aligned}
\|\hat{B} - B^\star\|_2 &\leq \frac{4}{m}\left(\sqrt{N} + \rho\sqrt{N(N-1)}\right)\lambda \\
&\leq \frac{4}{m}\sqrt{\frac{\sigma^2 \log N}{n}}\left(\sqrt{N} + \rho N\right)\min\left(\frac{\sqrt{N}}{1+\rho}, \frac{1}{\rho}\right) \\
&\leq \frac{4}{m}\sqrt{\frac{\sigma^2 \log N}{n}} N \min\left(1 + \rho\sqrt{N}, 1 + \frac{1}{\rho\sqrt{N}}\right) \\
&\leq \frac{4}{m}\sqrt{N^2\frac{\sigma^2 \log N}{n}}. \qquad \text{(A.12)}
\end{aligned}$$

Part (b). Lemma 1 gives a geometric decomposition of the penalty. Therefore, we can directly use Theorem 3.1 of Lee et al. (2015), since Lemma 2 also ensures that their irrepresentability condition holds. Thus,

$$\|\hat{B} - B^\star\|_2 \leq \frac{2}{m}\kappa_\Omega\left(1 + \frac{\tau}{4\kappa_{\text{IC}}}\right)\lambda,$$

and $\hat{\mathcal{G}} \subseteq \mathcal{G}$ as long as

$$\frac{4\kappa_{\text{IC}}}{\tau}\Omega^*\left(\nabla\ell(B^\star)\right) < \lambda < \frac{m^2\tau}{2L\kappa_\Omega^2\kappa_{\Omega^*}\kappa_{\text{IC}}}\left(1 + \frac{\tau}{4\kappa_{\text{IC}}}\right)^{-2}. \qquad \text{(A.13)}$$

Setting

$$\lambda = \frac{8\kappa_{\text{IC}}}{\tau}\sqrt{\frac{\sigma^2 \log N}{n}}\min\left(\frac{\sqrt{N}}{1+\rho}, \frac{1}{\rho}\right),$$

using a similar argument than (A.12), the left hand size of (A.13) holds with probability at least $1 - 2/N$. The right hand side of (A.13) holds as long as the sample size satisfies

$$n > C(L, m, \tau, \kappa_\Omega, \kappa_{\Omega^*}, \kappa_{IC})\left(\sqrt{G} + G\right)^2\sigma^2 \log N,$$

with $C(L, m, \tau, \kappa_\Omega, \kappa_{\Omega^*}, \kappa_{IC}) > 0$ a positive constant. Therefore, claims (2.17) and (2.18) follow.

□

APPENDIX B

Proofs of overlapping community detection via sparse principal component analysis

B.1 Proof ot Theorem 4.1

Proof. Since Z and $\tilde{Z}$ are basis of the column space of W, then $\text{rank}(Z) = \text{rank}(\tilde{Z}) = K$, and there exists a full rank matrix $V \in \mathbb{R}^{K\times K}$ such that $Z = \tilde{Z}V$. We will show that $V = PD$.

Let $\mathcal{S}_1 = (i_1, \ldots, i_K)$ be the indexes that satisfy $Z_{i_j j} = 1$ and $Z_{i_j j'} = 0$ for $j' \neq j$, and $j = 1, \ldots, j$ (these indexes exist by the assumptions of the theorem). In the same way, define $\mathcal{S}_2 = (i'_1, \ldots, i'_K)$ such that $\tilde{Z}_{i'_j j} = 1$ and $\tilde{Z}_{i_j j'} = 0$ for $j' \neq j$. $j = 1, \ldots, j$. Denote by $Z_{\mathcal{S}}$ to the $K \times K$ matrix that is formed with the rows indexed by $\mathcal{S}$. Therefore $Z_{\mathcal{S}_1} = I$ and $\tilde{Z}_{\mathcal{S}_2} = I$. Thus, $Z_{\mathcal{S}_2} = V$, and since V is full rank, $\tilde{Z}_{\mathcal{S}_1} = V^{-1}$. Thus, both V and V^{-1} are nonnegative matrices, which in turn implies that V is a positive generalized permutation matrix, so $V = PD$ for some permutation matrix P and a diagonal D with $\text{diag}(D) > 0$. □

B.2 Proof of Proposition 4.3

Proof. Set $\tilde{Q} = [q_1 \cdots q_K]$ and $\tilde{\Lambda}$ a diagonal matrix with the corresponding eigenvalues of $\tilde{Q}$, so $A\tilde{Q} = \tilde{Q}\tilde{\Lambda}$. Note that by equation (4.11), there exists a full-rank matrix $M \in \mathbb{R}^{K\times K}$ such $Z = \tilde{Q}M$. Therefore, Γ can be expressed as

$$\begin{aligned}
\Gamma &= (Z^T Z)^{-1} Z^T A Z (Z^T Z)^{-1} \\
&= (M^T M)^{-1} M^T \tilde{Q}^T \tilde{\Lambda} \tilde{Q} M (M^T M)^{-1} \\
&= M^{-1} \tilde{\Lambda} (M^T)^{-1}.
\end{aligned}$$

Hence,

$$\begin{aligned} AZ(Z^TZ)^{-1} &= A(\tilde{Q}M)(M^TM)^{-1} \\ &= \tilde{Q}\tilde{\Lambda}M^{-1}\Gamma^{-1}\Gamma \\ &= \tilde{Q}M\Gamma \\ &= Z\Gamma, \end{aligned}$$

which completes the proof. □

B.2.1 Proof of Theorem 1

Proof. The proof consists of a one-step analysis of the algorithm 4.2. We will show that if $Z^{(t)} = Z$, then $Z^{(t+1)} = Z$ with high probability. Let $T = T^{(t+1)} = AZ$ be value after the multiplication step. Define $C \in \mathbb{R}^{K\times K}$ as a diagonal matrix containing the sizes of the communities on the diagonal, with $C_{kk} = C_k = \|Z_{\cdot k}\|_1$. Thus, $U = U^{(t+1)} = TC^{-1}$. In order for the threshold to appropriately set to zero the correct set of entries, a sufficient condition is that on each row i the largest element of $U_{i\cdot}$ corresponds to the correct community. Define $\mathcal{C}_k \subset \{1,\dots,N\}$ as the subset of indexes of the nodes corresponding to community k. Then,

$$U_{ik} = \frac{1}{C_k}A_{i\cdot}Z_{\cdot k} = \frac{1}{C_k}\sum_{j\in\mathcal{C}_k} A_{ij}.$$

Then, U_{ik} is an averaged sum of independent and identically distributed Bernoulli random variables. Moreover, for each k_1 and k_2, U_{ik_1} and U_{ik_2} are independent of each other. For some $\lambda \in (0,1)$, let $E_i(\lambda) = \{\lambda U_{ik_i} > U_{ik_j}, i \in \mathcal{C}_{k_i}, \forall k_j \neq k_i\}$ be the event in which the largest index of $U_{i\cdot}$ is k_i, the community of node i, and all the other indexes in that row are smaller than λU_{ik_i}. Under the event $E(\lambda) = \bigcap_{i=1}^N E_i(\lambda)$, note that for $V = V^{(t+1)}$, we have that $\|V_{i\cdot}\|_\infty = U_{ik_i}$, and hence

$$V_{ik} = \begin{cases} U_{ik_i} & \text{if } k = k_i, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, under the event $E(\lambda)$, the thresholding step recovers again the correct support, so $Z^{(t+1)} = Z$. Now, we verify that under the conditions of Theorem 4.13,

the event $E(\lambda)$ has high probability. By using a union bound,

$$\begin{aligned}\mathbb{P}(E(\lambda)) &\geq 1 - \sum_{i=1}^{N} \mathbb{P}(E_i(\lambda)^C) \\ &\geq 1 - \sum_{i=1}^{N} \sum_{j \neq k_i} \mathbb{P}(U_{ij} > \lambda U_{ik_i}). \end{aligned} \tag{B.1}$$

Note that for $j \neq k_i$, $U_{ij} - \lambda U_{ik_i}$ is a sum of independent random variables by the arguments stated before, with expectation

$$\begin{aligned}\mathbb{E}\left[U_{ij} - \lambda U_{ik_i}\right] &= \frac{1}{C_j} \sum_{j \in \mathcal{C}_j} \mathbb{E}A_{ij} - \lambda \frac{1}{C_{k_i}} \sum_{j \in \mathcal{C}_{k_i}} \mathbb{E}A_{ij} \\ &= q - \lambda \frac{C_{k_i} - 1}{C_{k_i}} p. \end{aligned}$$

Hence, by Hoeffding's inequality, we have that for any $\tau \in \mathbb{R}$,

$$\begin{aligned}\mathbb{P}\left(U_{ij} - \lambda U_{ik_i} \geq \tau + \mathbb{E}\left[U_{ij} - \lambda U_{ik_i}\right]\right) &\leq 2\exp\left(\frac{-2\tau^2}{\frac{1}{C_j} + \frac{\lambda^2}{C_{k_i}}}\right) \\ &\leq 2\exp\left(-C_{\min}\tau^2 \frac{2}{1+\lambda^2}\right). \end{aligned}$$

Setting $\tau = -\mathbb{E}\left[U_{ij} - \lambda U_{ik_i}\right]$, and using equation (4.13) we obtain that

$$\mathbb{P}\left(U_{ij} > \lambda U_{ik_i}\right) \leq 2\exp\left(-\frac{2}{1+\lambda^2}\log(KN)\right) = \frac{2}{(KN)^{2/(1+\lambda^2)}}.$$

Plugging in the previous bound on equation (B.1), we obtain that

$$\mathbb{P}(E(\lambda)) \geq 1 - \frac{2(K-1)N}{(KN)^{2/(1+\lambda^2)}} \geq 1 - \frac{2}{(KN)^{2/(1+\lambda^2)-1}}$$

□

BIBLIOGRAPHY

Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2009). Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems*, pages 33–40.

Amini, A. A. and Levina, E. (2018). On semidefinite relaxations for the block model. *The Annals of Statistics*, 46(1):149–179.

Amini, A. A. and Wainwright, M. J. (2008). High-dimensional analysis of semidefinite relaxations for sparse principal components. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 2454–2458. IEEE.

Arroyo, J. and Hou, E. (2016). Efficient distributed estimation of inverse covariance matrices. In *Statistical Signal Processing Workshop (SSP), 2016 IEEE*, pages 1–5. IEEE.

Arroyo, J., Kessler, D., Levina, E., and Taylor, S. (2017). Network classification with applications to brain connectomics. *arXiv preprint arXiv:1701.08140*.

Arslan, S., Ktena, S. I., Makropoulos, A., Robinson, E. C., Rueckert, D., and Parisot, S. (2017). Human brain mapping: a systematic comparison of parcellation methods for the human cerebral cortex. *NeuroImage*.

Association, A. P., Association, A. P., et al. (1994). Diagnostic and statistical manual of mental disorders (DSM). *Washington, DC: American Psychiatric Association*, pages 143–7.

Athreya, A., Fishkind, D. E., Levin, K., Lyzinski, V., Park, Y., Qin, Y., Sussman, D. L., Tang, M., Vogelstein, J. T., and Priebe, C. E. (2017). Statistical inference on random dot product graphs: a survey. *arXiv preprint arXiv:1709.05454*.

Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al. (2012). Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468.

Bach, F. R. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9(Jun):1179–1225.

Ball, B., Karrer, B., and Newman, M. (2011). Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Becker, N., Werft, W., Toedt, G., Lichter, P., and Benner, A. (2009). penalizedSVM: a R-package for feature selection SVM classification. *Bioinformatics*, 25(13):1711–1712.

Behzadi, Y., Restom, K., Liau, J., and Liu, T. T. (2007). A component based noise correction method (compcor) for bold and perfusion based fMRI. *Neuroimage*, 37(1):90–101.

Bengio, Y. and Monperrus, M. (2005). Non-local manifold tangent learning. In *Advances in Neural Information Processing Systems*, pages 129–136.

Bhattacharyya, S. and Chatterjee, S. (2017). Spectral clustering for dynamic stochastic block model. *Working paper*.

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

Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function,'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, pages 989–1010.

Bondell, H. D. and Reich, B. J. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

Broyd, S. J., Demanuele, C., Debener, S., Helps, S. K., James, C. J., and Sonuga-Barke, E. J. (2009). Default-mode brain dysfunction in mental disorders: a systematic review. *Neuroscience & Biobehavioral Reviews*, 33(3):279–296.

Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198.

Bullmore, E. T. and Bassett, D. S. (2011). Brain graphs: graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7:113–140.

Bunney, W. E. and Bunney, B. G. (2000). Evidence for a compromised dorsolateral prefrontal cortical parallel circuit in schizophrenia. *Brain Research Reviews*, 31(2):138–146.

Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., and Munafò, M. R. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14(5):365–376.

Calhoun, V. D., Sui, J., Kiehl, K., Turner, J., Allen, E., and Pearlson, G. (2011). Exploring the psychosis functional connectome: aberrant intrinsic networks in schizophrenia and bipolar disorder. *Magnetic resonance imaging of disturbed brain connectivity in psychiatric illness*, 35.

Campanella, M. (2015). Syd Barrett: Was he suffering from schizophrenia or Asperger's syndrome? *Clinical Neuropsychiatry*, (3).

Chai, X. J., Castañón, A. N., Öngür, D., and Whitfield-Gabrieli, S. (2012). Anticorrelations in resting state networks without global signal regression. *Neuroimage*, 59(2):1420–1428.

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

Chen, X., Lin, Q., Kim, S., Carbonell, J. G., and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752.

Chen, Z. J., He, Y., Rosa-Neto, P., Germann, J., and Evans, A. C. (2008). Revealing modular architecture of human brain structural networks by using cortical thickness from MRI. *Cerebral cortex*, 18(10):2374–2381.

Conover, M., Ratkiewicz, J., Francisco, M. R., Gonçalves, B., Menczer, F., and Flammini, A. (2011). Political polarization on twitter. *ICWSM*, 133:89–96.

Cortes, C. and Vapnik, V. (1995). Support vector machine. *Machine learning*, 20(3):273–297.

Craddock, R. C., Holtzheimer, P. E., Hu, X. P., and Mayberg, H. S. (2009). Disease state prediction from resting state functional connectivity. *Magnetic Resonance in Medicine*, 62(6):1619–1628.

Daubechies, I., DeVore, R., Fornasier, M., and Güntürk, C. S. (2010). Iteratively reweighted least squares minimization for sparse recovery. *Communications on pure and applied mathematics*, 63(1):1–38.

Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011). Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106.

Deshpande, M., Kuramochi, M., Wale, N., and Karypis, G. (2005). Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036–1050.

Diamond, S., Takapoui, R., and Boyd, S. (2016). A general system for heuristic solution of convex problems over nonconvex sets. *arXiv preprint arXiv:1601.07277*.

Dong, D., Wang, Y., Chang, X., Luo, C., and Yao, D. (2017). Dysfunction of large-scale brain networks in schizophrenia: a meta-analysis of resting-state functional connectivity. *Schizophrenia bulletin*, 44(1):168–181.

Duchi, J. and Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(Dec):2899–2934.

Durante, D., Dunson, D. B., et al. (2017). Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis*.

Fair, D. A., Dosenbach, N. U., Church, J. A., Cohen, A. L., Brahmbhatt, S., Miezin, F. M., Barch, D. M., Raichle, M. E., Petersen, S. E., and Schlaggar, B. L. (2007). Development of distinct control networks through segregation and integration. *Proceedings of the National Academy of Sciences*, 104(33):13507–13512.

Fei, H. and Huan, J. (2010). Boosting with structure information in the functional space: an application to graph classification. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 643–652. ACM.

Finn, E. S., Shen, X., Scheinost, D., Rosenberg, M. D., Huang, J., Chun, M. M., Papademetris, X., and Constable, R. T. (2015). Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nature neuroscience*, 18(11):1664.

First, M. B., Spitzer, R. L., Gibbon, M., Williams, J. B., et al. (1995). Structured clinical interview for DSM-IV axis I disorders. *New York: New York State Psychiatric Institute*.

Fornito, A., Zalesky, A., Pantelis, C., and Bullmore, E. T. (2012). Schizophrenia, neuroimaging and connectomics. *NeuroImage*, 62(4):2296–2314.

Fox, M. D., Snyder, A. Z., Vincent, J. L., Corbetta, M., Van Essen, D. C., and Raichle, M. E. (2005). The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9673–9678.

Friedman, J., Hastie, T., and Tibshirani, R. (2009). glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. (2015). Achieving optimal misclassification proportion in stochastic block model. *arXiv preprint arXiv:1505.03772*.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer.

Ginestet, C. E., Li, J., Balachandran, P., Rosenberg, S., Kolaczyk, E. D., et al. (2017). Hypothesis testing for network data in functional neuroimaging. *The Annals of Applied Statistics*, 11(2):725–750.

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

Gonzalez, J., Holder, L. B., and Cook, D. J. (2000). Graph based concept learning. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. AAAI Press*, page 1072. The MIT Press.

Gotts, S., Saad, Z., Jo, H. J., Wallace, G., Cox, R., and Martin, A. (2013). The perils of global signal regression for group comparisons: a case study of Autism Spectrum Disorders. *Frontiers in Human Neuroscience*, 7:356.

Grosenick, L., Klingenberg, B., Katovich, K., Knutson, B., and Taylor, J. E. (2013). Interpretable whole-brain prediction analysis with graphnet. *NeuroImage*, 72:304–321.

Hanlon, F. M., Houck, J. M., Pyeatt, C. J., Lundy, S. L., Euler, M. J., Weisend, M. P., Thoma, R. J., Bustillo, J. R., Miller, G. A., and Tesche, C. D. (2011). Bilateral hippocampal dysfunction in schizophrenia. *Neuroimage*, 58(4):1158–1168.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Helma, C., King, R. D., Kramer, S., and Srinivasan, A. (2001). The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108.

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

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

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Hu, Y. and Allen, G. I. (2015). Local-aggregate modeling for big data via distributed optimization: Applications to neuroimaging. *Biometrics*, 71(4):905–917.

Inokuchi, A., Washio, T., and Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. In *Principles of Data Mining and Knowledge Discovery*, pages 13–23. Springer.

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440. ACM.

Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *Neuroimage*, 17(2):825–841.

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

Jin, J. (2015). Fast community detection by SCORE. *Ann. Statist.*, 43(1):57–89.

Jin, J., Ke, Z. T., and Luo, S. (2017). Estimating network memberships by simplex vertex hunting. *arXiv preprint arXiv:1708.07852*.

Johnstone, I. M. and Lu, A. Y. (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693.

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, 12(3):531–547.

Kashima, H., Tsuda, K., and Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *International Conference of Machine Learning*, volume 3, pages 321–328.

Kessler, D., Angstadt, M., and Sripada, C. (2016). Growth charting of brain connectivity networks and the identification of attention impairment in youth. *JAMA psychiatry*, 73(5):481–489.

Ketkar, N. S., Holder, L. B., and Cook, D. J. (2009). Empirical comparison of graph classification algorithms. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 259–266. IEEE.

Kudo, T., Maeda, E., and Matsumoto, Y. (2004). An application of boosting to graph classification. In *Advances in Neural Information Processing Systems*, pages 729–736.

Lancichinetti, A., Fortunato, S., and Kertész, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015.

Latouche, P., Birmelé, E., and Ambroise, C. (2011). Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, pages 309–336.

Le, C. M. and Levina, E. (2015). Estimating the number of communities in networks by spectral methods. *arXiv preprint arXiv:1507.00827*.

Le, C. M., Levina, E., and Vershynin, R. (2015). Sparse random graphs: regularization and concentration of the Laplacian. *arXiv preprint arXiv:1502.03049*.

Lee, J. D., Sun, D. L., Sun, Y., Taylor, J. E., et al. (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927.

Lee, J. D., Sun, Y., Taylor, J. E., et al. (2015). On model selection consistency of regularized M-estimators. *Electronic Journal of Statistics*, 9:608–642.

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

Li, T., Levina, E., and Zhu, J. (2016). Network cross-validation by edge sampling. *arXiv preprint arXiv:1612.04717*.

Lindquist, M. A. et al. (2008). The statistical analysis of fMRI data. *Statistical science*, 23(4):439–464.

Liu, Y., Liang, M., Zhou, Y., He, Y., Hao, Y., Song, M., Yu, C., Liu, H., Liu, Z., and Jiang, T. (2008). Disrupted small-world networks in schizophrenia. *Brain*, 131(4):945–961.

Lockhart, R., Taylor, J., Tibshirani, R. J., and Tibshirani, R. (2014). A significance test for the lasso. *The Annals of Statistics*, 42(2):413.

Lyzinski, V., Sussman, D. L., Tang, M., Athreya, A., Priebe, C. E., et al. (2014). Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics*, 8(2):2905–2922.

Ma, Z. (2013). Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801.

Mayer, A. R., Ruhl, D., Merideth, F., Ling, J., Hanlon, F. M., Bustillo, J., and Cañive, J. (2013). Functional imaging of the hemodynamic sensory gating response in schizophrenia. *Human brain mapping*, 34(9):2302–2312.

McAuley, J. J. and Leskovec, J. (2012). Learning to discover social circles in ego networks. In *NIPS*, volume 2012, pages 548–56.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Menon, V. (2011). Large-scale brain networks and psychopathology: a unifying triple network model. *Trends in cognitive sciences*, 15(10):483–506.

Meunier, D., Achard, S., Morcom, A., and Bullmore, E. (2009). Age-related changes in modular organization of human brain functional networks. *Neuroimage*, 44(3):715–723.

Mossel, E., Neeman, J., and Sly, A. (2015). Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461.

Narayan, M., Allen, G. I., and Tomson, S. (2015). Two sample inference for populations of graphical models with applications to functional connectivity. *arXiv preprint arXiv:1502.03853*.

Newman, M. (2010). *Networks: an introduction*. Oxford university press.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104.

Olhede, S. C. and Wolfe, P. J. (2014). Network histograms and universality of blockmodel approximation. *Proceedings of the National Academy of Sciences*, 111(41):14722–14727.

Öngür, D., Lundy, M., Greenhouse, I., Shinn, A. K., Menon, V., Cohen, B. M., and Renshaw, P. F. (2010). Default mode network abnormalities in bipolar disorder and schizophrenia. *Psychiatry Research: Neuroimaging*, 183(1):59–68.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231.

Peeters, S. C., van de Ven, V., Gronenschild, E. H. M., Patel, A. X., Habets, P., Goebel, R., van Os, J., Marcelis, M., et al. (2015). Default mode network connectivity as a function of familial and environmental risk for psychotic disorder. *PloS one*, 10(3):e0120030.

Power, J. D., Barnes, K. A., Snyder, A. Z., Schlaggar, B. L., and Petersen, S. E. (2012). Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. *Neuroimage*, 59(3):2142–2154.

Power, J. D., Cohen, A. L., Nelson, S. M., Wig, G. S., Barnes, K. A., Church, J. A., Vogel, A. C., Laumann, T. O., Miezin, F. M., Schlaggar, B. L., et al. (2011). Functional network organization of the human brain. *Neuron*, 72(4):665–678.

Prasad, G., Joshi, S. H., Nir, T. M., Toga, A. W., Thompson, P. M., (ADNI, A. D. N. I., et al. (2015). Brain connectivity and novel network measures for Alzheimer's disease classification. *Neurobiology of aging*, 36:S121–S131.

Psorakis, I., Roberts, S., Ebden, M., and Sheldon, B. (2011). Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114.

Richiardi, J., Eryilmaz, H., Schwartz, S., Vuilleumier, P., and Van De Ville, D. (2011). Decoding brain states from fMRI connectivity graphs. *NeuroImage*, 56(2):616–626.

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

Scheinberg, K., Goldfarb, D., and Bai, X. (2014). Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14(3):389–417.

Schlitt, T. and Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8(Suppl 6):S9.

Schwarz, A. J., Gozzi, A., and Bifone, A. (2008). Community structure and modularity in networks of correlated brain activity. *Magnetic resonance imaging*, 26(7):914–920.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Scott, J. G., Kelly, R. C., Smith, M. A., Zhou, P., and Kass, R. E. (2015). False discovery rate regression: an application to neural synchrony detection in primary visual cortex. *Journal of the American Statistical Association*, 110(510):459–471.

Shah, R. D. and Samworth, R. J. (2013). Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(1):55–80.

Smith, S. M., Miller, K. L., Moeller, S., Xu, J., Auerbach, E. J., Woolrich, M. W., Beckmann, C. F., Jenkinson, M., Andersson, J., Glasser, M. F., et al. (2012). Temporally-independent functional modes of spontaneous brain activity. *Proceedings of the National Academy of Sciences*, 109(8):3131–3136.

Smith, S. M., Vidaurre, D., Beckmann, C. F., Glasser, M. F., Jenkinson, M., Miller, K. L., Nichols, T. E., Robinson, E. C., Salimi-Khorshidi, G., Woolrich, M. W., et al. (2013). Functional connectomics from resting-state fMRI. *Trends in Cognitive Sciences*, 17(12):666–682.

Srinivasan, A., Muggleton, S. H., Sternberg, M. J., and King, R. D. (1996). Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85(1):277–299.

Sripada, C., Angstadt, M., Kessler, D., Phan, K. L., Liberzon, I., Evans, G. W., Welsh, R. C., Kim, P., and Swain, J. E. (2014a). Volitional regulation of emotions produces distributed alterations in connectivity between visual, attention control, and default networks. *Neuroimage*, 89:110–121.

Sripada, C., Kessler, D., Fang, Y., Welsh, R. C., Prem Kumar, K., and Angstadt, M. (2014b). Disrupted network architecture of the resting brain in attention-deficit/hyperactivity disorder. *Human brain mapping*, 35(9):4693–4705.

Sripada, C. S., Kessler, D., and Angstadt, M. (2014c). Lag in maturation of the brain's intrinsic functional architecture in attention-deficit/hyperactivity disorder. *Proceedings of the National Academy of Sciences*, 111(39):14259–14264.

Stephen, J. M., Coffman, B. A., Jung, R. E., Bustillo, J. R., Aine, C., and Calhoun, V. D. (2013). Using joint ICA to link function and structure using MEG and DTI in schizophrenia. *Neuroimage*, 83:418–430.

Supekar, K., Menon, V., Rubin, D., Musen, M., and Greicius, M. D. (2008). Network analysis of intrinsic functional brain connectivity in alzheimer's disease. *PLoS Comput Biol*, 4(6):e1000100.

Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V., Park, Y., and Priebe, C. E. (2017a). A semiparametric two-sample hypothesis testing problem for random graphs. *Journal of Computational and Graphical Statistics*, 26(2):344–354.

Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V., Priebe, C. E., et al. (2017b). A nonparametric two-sample hypothesis testing problem for random graphs. *Bernoulli*, 23(3):1599–1630.

Thirion, B., Varoquaux, G., Dohmatob, E., and Poline, J.-B. (2014). Which fMRI clustering gives good brain parcellations? *Frontiers in neuroscience*, 8:167.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society; Series B*, 73(3):267–288.

Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R., et al. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202.

Varoquaux, G. and Craddock, R. C. (2013). Learning and comparing functional connectomes across subjects. *NeuroImage*, 80:405–415.

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. (2010). Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242.

Vogelstein, J. T., Roncal, W. G., Vogelstein, R. J., and Priebe, C. E. (2013). Graph classification using signal-subgraphs: Applications in statistical connectomics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1539–1551.

Vu, V. Q., Lei, J., et al. (2013). Minimax sparse principal subspace estimation in high dimensions. *The Annals of Statistics*, 41(6):2905–2947.

Wang, Y. and Bickel, P. J. (2015). Likelihood-based model selection for stochastic block models. *arXiv preprint arXiv:1502.02069*.

Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*, volume 8. Cambridge university press.

Watanabe, T., , D., Scott, C., Angstadt, M., and Sripada, C. (2014). Disease prediction based on functional connectomes using a scalable and spatially-informed support vector machine. *NeuroImage*, 96:183–202.

Whitfield-Gabrieli, S., Thermenos, H. W., Milanovic, S., Tsuang, M. T., Faraone, S. V., McCarley, R. W., Shenton, M. E., Green, A. I., Nieto-Castanon, A., LaViolette, P., et al. (2009). Hyperactivity and hyperconnectivity of the default network in schizophrenia and in first-degree relatives of persons with schizophrenia. *Proceedings of the National Academy of Sciences*, 106(4):1279–1284.

Xin, B., Kawahara, Y., Wang, Y., and Gao, W. (2014). Efficient generalized fused lasso and its application to the diagnosis of Alzheimer's disease. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2163–2169.

Yan, C.-G., Craddock, R. C., Zuo, X.-N., Zang, Y.-F., and Milham, M. P. (2013). Standardizing the intrinsic brain: towards robust measurement of inter-individual variation in 1000 functional connectomes. *Neuroimage*, 80:246–262.

Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer.

Yuan, L., Liu, J., and Ye, J. (2011). Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems*, pages 352–360.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society; Series B*, 68(1):49–67.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473.

Zalesky, A., Fornito, A., and Bullmore, E. T. (2010). Network-based statistic: identifying differences in brain networks. *NeuroImage*, 53(4):1197–1207.

Zhang, A. Y., Zhou, H. H., et al. (2016a). Minimax rates of community detection in stochastic block models. *The Annals of Statistics*, 44(5):2252–2280.

Zhang, J., Cheng, W., Wang, Z., Zhang, Z., Lu, W., Lu, G., and Feng, J. (2012). Pattern classification of large-scale functional brain networks: identification of informative neuroimaging markers for epilepsy. *PLoS One*, 7(5):e36733.

Zhang, L., Guindani, M., Versace, F., Engelmann, J. M., Vannucci, M., et al. (2016b). A spatiotemporal nonparametric bayesian model of multi-subject fMRI data. *The Annals of Applied Statistics*, 10(2):638–666.

Zhang, Y., Levina, E., and Zhu, J. (2014). Detecting overlapping communities in networks with spectral methods. *arXiv preprint arXiv:1412.3432*.

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

Zhou, H. and Li, L. (2014). Regularized matrix regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):463–483.

Zhou, H., Li, L., and Zhu, H. (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552.

Zhu, J., Rosset, S., Tibshirani, R., and Hastie, T. J. (2004). 1-norm support vector machines. In *Advances in Neural Information Processing Systems*, pages 49–56.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286.