Penalized likelihood methods for covariance selection in the context of non-stationary data

A thesis presented for the degree of

Doctor of Philosophy of Imperial College London

and the

Diploma of Imperial College

by

Ricardo Pio Monti

Department of Mathematics

Imperial College

180 Queen's Gate, London SW7 2AZ

APRIL 21, 2017

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Signed:

Copyright

Dedicado al *Padre*, a la *Madre* y a Simon que siempre me corrigió los textos.

Abstract

Graphical models have established themselves as fundamental tools through which to understand complex relationships in high-dimensional datasets. Applications abound, a pertinent example being neuroimaging where Gaussian graphical models are employed to model statistical dependencies across spatially remote brain regions. Often such models are estimated under regularization penalties which help to enforce properties such as sparsity. Much of the current methodology is rooted on the assumption that the same covariance structure characterizes all observations and may be summarized using a single graphical model. However, such an assumption is untenable in the context of many applications. In order to address this issue, we propose a host of algorithms through which to accurately estimate Gaussian graphical models in the context of data with heterogeneous covariance structure.

Formally, this thesis is focused in studying graphical models in two distinct manifestations of heterogeneity in covariance structure. The first relates to the task of estimating time-varying graphical models and two algorithms are proposed to this end. A related challenge is associated with the choice of regularization parameter: in the presence of variable covariance structure such a parameter is both difficult to estimate and potentially time-varying itself. In order to address these challenges, a novel framework is proposed through which to iteratively tune this parameter. The second manifestation is related to the presence of heterogeneity in covariance structure across multiple related graphical models. In such a setting scientific objectives consist in inferring the covariance structure shared across all graphs together with the idiosyncrasies of each specific graph. A related objective which is often overlooked consists of quantifying variability across all graphs. A novel algorithm is proposed through which to simultaneously address the aforementioned objectives.

Acknowledgements

First and foremost I must thank my advisors, Christoforos Anagnostopoulos and Giovanni Montana, for all their expertise, patience, support and guidance throughout the past four years. Without their help none of this would have been possible.

Throughout my PhD I have had the pleasure of collaborating with many talented researchers. As a result of these collaborations I have been able to learn a great deal about many diverse topics. Thank you to Romy Lorenz, Robert Leech, Peter Hellyer, Rodrigo Braga, Emanuele Pesce and Ai Wern Chung.

To my friends with whom I have shared office 526: thank you for the many lunches, discussions and for tolerating the small mess in my corner of the office. Thank you also to everyone at the Imperial College Tennis Club.

Finally, I must thank my parents, my brother, my sister and Annabel for their support and encouragement.

Ricardo Pio Monti

Contents

Chapter 1

Introduction

Undirected graphical models have established themselves as fundamental tools through which to understand and describe complex statistical relationships in high-dimensional data [100]. In such models, each node represents a random variable and edges encode statistical dependencies. The popularity of graphical models arises due to their ability to reduce complex statistical structures to simple, modular graphs [181]. Graphical models may therefore leverage the wide literature relating to graph theory and network analysis. There are numerous applications for such models. In finance they are employed to describe systematic dependencies across stocks and the estimated graphical models may then be used to optimize portfolios [101]. In the context of neuroscience, a primary focus of this thesis, they are employed to quantify dependencies across spatially remote brain regions [159]. This allows for the estimation of brain *networks* which provide novel insights into the architecture and function of the human brain [30]. It follows that such networks provide a platform through which to further understand the effects of many neurological and psychiatric conditions [72]. Other applications include the study of genome data [104], cyber-security [79] and sensor networks [4].

Arguably the most widely employed class of graphical models are Gaussian graphical models (GGMs) [100]. In this setting, data is assumed to follow a multivariate Gaussian distribution and the edge structure encodes the conditional dependence structure across random variables. More concretely, the edges capture the partial correlations, defined as the correlation once the effects of all other variables has been removed [95]. Due to the unique

interpretation of edges in GGMs, it is often desirable to learn sparse edge structures for reasons of interpretability and data-efficiency as well as domain specific motivations [77]. Such models are often referred to as covariance selection models and can be estimated by learning the non-zero entries of the inverse covariance (precision) matrix from data [46].

The problem of estimating a sparse inverse covariance matrix has been widely studied in the context of statistics and machine learning. Traditional methods have involved the use of multiple hypothesis testing. Such methods iteratively add or remove edges based on hypothesis tests [49, 50], but cannot easily be extended to high-dimensional settings for two fundamental reasons: first, the need to correct for multiple comparisons results in increasingly conservative tests as the number of nodes in the graph increases. Second, in the case of high-dimensional data the sample covariance matrix may be poorly conditioned and potentially singular [101]. In such a setting, it is not possible to obtain estimates of the inverse covariance thereby impeding the use of the aforementioned approaches.

In order to estimate GGMs in the context of high-dimensional data, a host of *penalized* methods have been proposed. Such methods introduce regularization penalties whose purpose is to enforce the estimated inverse covariance matrix to display certain properties by constraining the set of candidate solutions. A popular form of regularization involves the use of an ℓ_1 penalty [77], which is able to yield sparse estimates of the inverse covariance matrix by effectively constraining the magnitude of elements in the estimated inverse covariance. An important advantage of the ℓ_1 penalty is that it is able to retain the convexity of the associated optimization problem. This allows for the optimization problem to be solved using first order (i.e., gradient) methods and has resulted in a wide range of algorithms being proposed through which to estimate ℓ_1-penalized precision matrices [10, 68, 85] as well as computationally efficient approximations [117].

The aforementioned covariance selection methods are rooted on the assumption that the data is such that the associated graphical model remains fixed, indicating that the conditional dependence structure may be adequately summarized by a single graph. However, there are a wide range of applications where such an assumption is tenuous. A pertinent example corresponds to the estimation of brain connectivity networks where there is wide-spread evidence to suggest the data displays variation in covariance structure over time [32, 91]. This has led to large-scale interest in quantifying the dynamic properties of

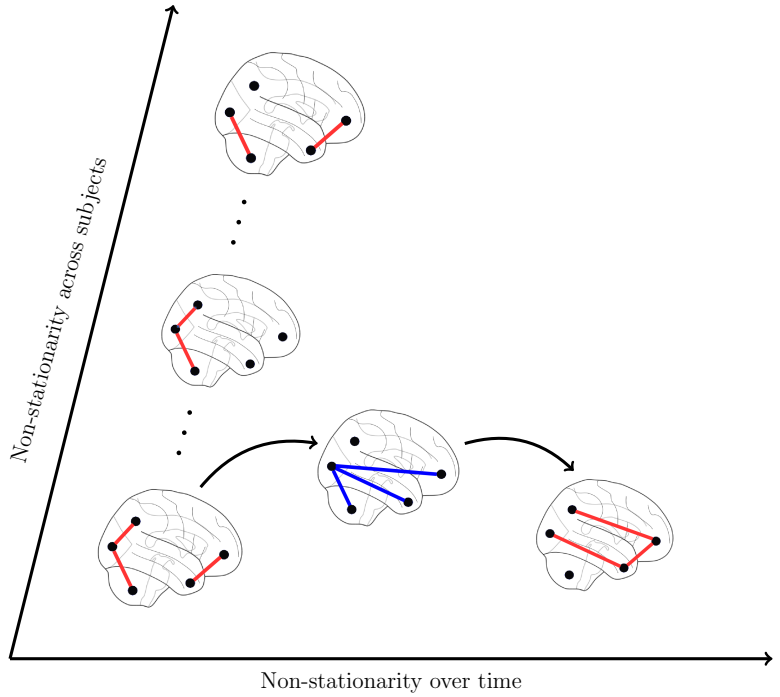


Figure 1.1: The two axes along which we study variability in covariance structure, which for the sake of notational convenience we refer to as non-stationarity, are depicted pictorially. The first axis corresponds to understanding non-stationary over time. This involves accurately quantifying changes in edge structure over time. The second axis corresponds to understanding non-stationarity over multiple related datasets.

brain networks and understanding how these may be modulated by distinct cognitive tasks [31, 90].

It is important to note that variability in the covariance matrix (and hence also in its inverse) may manifest itself in a host of different ways. In this thesis we are focused on studying variation in the edge structure of GGMs along some covariate, for example time. In particular, the objectives of this thesis are to quantify variability across two distinct scenarios depicted pictorially in Figure [1.1]. The first scenario corresponds to quantifying temporal diversity in edge structure with a view to potentially relating such diversity to external circumstances. Returning to the example of brain networks, a fundamental question of modern neuroscience relates to understanding the changes in functional connectivity that occur during distinct cognitive tasks. To date, a wide range of methods have been employed in order to estimate time-varying GGMs. A popular approach involves the use

of sliding windows in order to obtain a *local* estimate of the sample covariance matrix, which may be subsequently employed to estimate the associated graphical models [187]. Alternative approaches involve the use of change-point detection [38] and hidden Markov models [163].

The second scenario corresponds to studying heterogeneity in edge structure across multiple related graphical models. In such a setting, the data available typically corresponds to observations across several different classes. Two relevant example applications include the study of brain connectivity across a cohort of subjects and gene-gene interaction networks across multiple different tissue types. In the case of the former we may consider each subject as a distinct class. Such an approach allows for the quantification of inter-subject variability which is known to exist. In the case of the latter example each class may correspond to distinct tissue types (e.g., cancer tissue compared to normal tissue). In such a setting, typical scientific objectives include both estimation of a GGM for each class as well as understanding the variability in edge structure across classes. From a methodological perspective, a potential solution involves estimating each GGM independently. However, such an approach fails to exploit any shared edge structure across classes. This may jeopardize the accuracy of estimated networks, especially in a high-dimensional setting when only a reduced number of observations are available for each class. Further complications are introduced by the fact that given multiple estimated GGMs, obtaining a single representative graph is non-trivial and yet hugely important. For example, in the context of neuroscience it is imperative to distinguish between the edge structure shared across a population of subjects and subject-specific idiosyncrasies. Recently, these issues have been partially addressed through the use of novel regularization schemes [42, 172]. Such methods propose to jointly estimate networks across classes while introducing constraints over the edge structure. In this manner, the edge structure of each class is informed by the estimated structure of all remaining classes.

We conclude this section by noting that the two aforementioned examples of variability in covariance structure correspond to two very specific manifestations of non-stationarity. However, for the sake of notational brevity we will henceforth refer to data displaying such properties as non-stationary data.

Description of the thesis

The aim of this thesis is to extend traditional covariance selection methods to the domain of non-stationary data. As a result, the remainder of this thesis describes a collection of novel algorithms through which to estimate GGMs when faced with non-stationarity data. The proposed methodology, which is focused on the two manifestations of non-stationarity depicted in Figure [1.1], is rooted on the use of penalized likelihood methods. Background on these methods is provided in Chapter 2, together with the associated convex optimization methods. Moreover, the methodological contributions of this thesis are motivated by the study of fMRI data and the estimation of functional connectivity networks. As a result, Chapter 2 also provides a brief discussion of the properties of fMRI data and the role of graphical models in modeling connectivity networks.

In Chapter 3 we consider the problem of estimating time-varying GGMs in the context of functional MRI data. The methodology presented in this chapter was published in [122] and extends traditional sliding window methods by incorporating an additional regularization penalty which enforces sparse differences in estimated edge structure over time. This penalty serves to ensure that changes in edge structure are only reported when substantiated by evidence in the data, a property we refer to as temporal homogeneity. An algorithm is proposed through which to efficiently estimate time-varying GGMs which are both sparse and temporally homogeneous. Throughout a series of simulations the performance of the proposed algorithm is empirically validated and benchmarked against alternative algorithms from the literature. Finally, the proposed algorithm is applied to fMRI data collected while subjects were asked to perform a Choice Reaction Task (CRT). During the CRT task, subjects were required to alternate between performing a cognitively demanding task and resting. As a result, network structure was expected to alternate depending on the underlying task.

An exciting avenue for modern neuroscientific research involves the study of fMRI data in real-time. Potential applications include neurofeedback, where participants learn to modulate brain activity within a specified brain region [178], and brain decoding [99], where multivariate classification techniques are employed to predict brain states in real-time. However, the majority of real-time fMRI studies to date have only studied mea-

surements of individual brain regions. This fails to take into consideration the notion that the brain is a functionally connected network [162]. As a result, Chapter 4 extends the methodology presented in Chapter 3 to the domain of streaming data, allowing for connectivity networks to be estimated in real-time. This work, published in [123], presents important practical challenges as edge structure must be efficiently estimated using only a reduced subset of relevant observations. In order to address this challenge, we first introduce the use of adaptive filtering techniques [78]. Throughout a series of simulations, such methods are shown to empirically out-perform traditional sliding windows. The methodology presented in Chapter 3 is then extended by presenting a computationally efficient approximation which can be solved in real-time. The chapter concludes with an application to data taken from the Human Connectome Project (HCP). While this data was not acquired and analyzed in real-time, it may be treated as such by only considering a single observation at a time. This serves to demonstrate that the proposed algorithm is able to accurately estimate task-related changes in network structure in real-time.

The primary focus of Chapters 3 and 4 has been associated with the accurate estimation of edge structure via the introduction of regularization penalties. However, one fundamental aspect which these chapters overlook corresponds to the choice and tuning of the associated regularization parameters, which dictate the degree of sparsity in estimated networks, together with the implicit assumption that these parameters should remain fixed. While the choice of such regularization parameters has been extensively studied in the context of stationary data, such methods cannot trivially be extended to non-stationary or streaming data scenarios. As a result, in Chapter 5 we present a framework through which to learn a time-varying regularization parameter in the context of streaming data. The proposed framework effectively recasts the selection of a sparsity parameter in the context of adaptive filtering, thereby relegating the choice of such a parameter to the data. This reformulation also allows for the tracking of a time-varying regularization parameter as well as the derivation of convergence guarantees in a non-stochastic setting. The proposed framework is developed for streaming lasso models and then extended to GGMs via neighborhood selection techniques. A series of simulation studies are employed to empirically validate the proposed framework. Finally, the chapter concludes with an application to task based fMRI data taken from the HCP. This work has lead to the following paper [121] which is currently

under review.

The preceding three chapters have addressed the challenge of accurately estimating time-varying GGMs in the context of non-stationary data. By estimating a GGM at each observation, such methods provide unprecedented insights relating to the dynamic restructuring and temporal evolution of functional connectivity networks. However, the increased temporal resolution also makes it difficult to obtain robust and easily interpretable insights. This is particularly true in the context of high-dimensional GGMs. The objective of methodology presented in Chapter 6 is therefore to address the challenges associated with the interpretation of time-varying, high-dimensional networks via the use of linear graph embedding methods. This serves to facilitate tasks such as visualization and classification by translating the problem from the graph domain into a Euclidean space, where traditional classification and visualization techniques can be readily applied. While a wide range of graph embedding techniques may be employed, we focus on the use of graph embeddings which are based on linear projections over the edge structure of estimated graphs. This allows us to obtain a clear interpretation of the embedding in the context of functional connectivity. Two distinct graph embedding algorithms are presented; the first based on principal component analysis and the second on regularized linear discriminant analysis. The capabilities of the proposed embeddings are quantified via an extensive simulation study. This chapter concludes with an application of the proposed embeddings to data taken from the HCP. The work presented in this chapter resulted in the following publications [124, 125].

In Chapter 7 we consider a distinct manifestation of non-stationarity in the form of heterogeneity in covariance structure across multiple related GGMs. The focus of this chapter therefore revolves around the estimation of multiple related GGMs. The study of neuroimaging data serves as an appropriate application given that one of the hallmarks of fMRI data is its reproducible nature. Observed patterns in functional connectivity have been shown to demonstrate reproducible properties across subjects [41, 191]. This motivates the need for novel methodologies with two overriding objectives. First, it is important to exploit the presence of shared connectivity structure in order to yield more accurate network estimates for each subject. Second, there is also a critical need to understand and quantify inter-subject variability in the context of functional connectivity. To date, the

aforementioned challenges have not been simultaneously addressed. Instead previous work has considered one of two main avenues which involve either learning a separate GGM for each subject or a single GGM that is representative of the entire population. The objective of the work presented in Chapter 7 is to reconcile the two popular approaches presented above, thus allowing for accurate network estimation at subject-specific and population levels while also quantifying variability present across a cohort. This is achieved by extending neighborhood selection techniques to incorporate an additional random effects component. This corresponds to learning a novel model for covariance structure across a cohort of subjects which is able to distinguish between shared covariance structure and subject-specific idiosyncrasies. The capabilities of the proposed method are demonstrated throughout an extensive series of simulation studies. Moreover, an application to resting-state data taken from the ABIDE consortium is presented. The methodology presented in this chapter has lead to the following paper [120] which is currently under review.

Chapter 8 concludes the thesis with a summary of the main results as well as a discussion of future work.

Notation and terminology

The notation and terminology are employed throughout this thesis is as follows:

- We write Σ, Θ and S to denote the covariance, inverse covariance and sample covariance matrices respectively.
- The subscript notation is employed to denote time dependence or components of a vector or matrix but never both. Moreover, i or t are reserved the denote time while u, v are reserved to denote elements (or subsets) of the vertex set, V, which indexes our set of variables. We therefore write u or v to indicate components of a vector or columns of a matrix. By way of example, we write S_t to denote an estimate of the covariance matrix at time t (or at the tth observation). Conversely, we write S_v to denote the entries of S along the column v.
- In the context of iterative algorithms, we employ the super-script notation to denote an estimated quantity at the kth iteration. As such, Θ^k denotes the estimated precision

matrix at the kth iteration.

- Unless stated otherwise, all vectors are assumed to be column vectors. As such, we write $\beta \in \mathbb{R}^p$ to denote $\beta \in \mathbb{R}^{p \times 1}$. Furthermore, the superscript T notation is reserved to denote a vector or matrix transpose.

- The bracketed superscript notation is employed when estimating graphical models across multiple classes or subjects. As such, $\Theta^{(s)}$ denotes the estimated precision matrix for subject s.

- This entire thesis is focused exclusively on the use of GGMs. As a result, whenever the term graphical model is employed this will refer to a GGM. Similarly, the terms graph and network are used interchangeably and refer to a GGM.

- Furthermore, we use the term covariance structure to refer to the edge structure of a GGM. Such terminology is widely used in the literature due to the mapping between the non-zero entries of the inverse covariance matrix and the edges of a GGM.

- As mentioned previously, this thesis is also focused on the study of non-stationary data. In an abuse of terminology, the term "non-stationary" is employed to denote a scenario where the edge structure of a GGM varies as a function of some covariate (for example as a function of time). It is also important to note that such non-stationarity may occur both as a change in the presence of an edge, for example an edge between two nodes may alternate between being present and absent, or in the nature of the edge, for example the conditional dependence between two nodes may alternative from positive to negative or may vary in magnitude.

List of publications

The work presented in Chapters 3 to 7 is based on the following preprints or publications:

- Chapter 3 is based on the following publication:

  R. P. Monti *et al.,* (2014). "Estimating time-varying brain connectivity networks from functional MRI time series", *NeuroImage* (103):427-443

- Chapter 4 is based on the following publication:

  R. P. Monti *et al.,* (2017). "Real-time estimation of dynamic functional connectivity networks", *Human Brain Mapping*, (38):202-220

- Chapter 5 is based on the following preprint:

  R. P. Monti *et al.,* (2016). "A framework for adaptive regularization in streaming lasso models", Under review at *Statistical Analysis and Data Mining*. Available online at: *arXiv:1610.09127*

- Chapter 6 is based on the following publications:

  R. P. Monti *et al.,* (2015). "Graph embeddings of dynamic functional connectivity reveal discriminative patterns of task engagement in HCP data", *International workshop on Pattern Recognition in Neuroimaging*

  R. P. Monti *et al.,* (2016). "Decoding time-varying functional connectivity networks via linear graph embedding methods", *Frontiers in Computational Neuroscience*, (11):1-14

- Chapter 7 is based on the following preprint:

  R. P. Monti *et al.,* (2015). "Learning population and subject-specific brain connectivity networks via Mixed Neighborhood Selection", Under review at *The Annals of Applied Statistics* (received minor corrections). Available online at: *arXiv:1512.01947*

Code

Freely available code is provided for all the methodology presented in this thesis. Below, we detail some of the implementations provided.

- Chapter 3: A python implementation of the SINGLE algorithm is available at `www.github.com/piomonti/pySINGLE`. The `multiprocessing` package is employed to provide a multi-core implementation [114]. Furthermore, Cython [14]

is employed to optimize a subset of computationally expensive routines within the SINGLE algorithm.

- Chapter 4: A python implementation of the rt-SINGLE algorithm is available at `www.github.com/piomonti/RTN`.
- Chapter 5: An R implementation of the RAP framework is available in the `rRAP` package which can be downloaded from Comprehensive R Archive Network (CRAN) [143].
- Chapter 6: A python implementation of the linear graph embedding methods is available at `www.github.com/piomonti/pyLGE`.
- Chapter 7: An R implementation of the MNS algorithm is available in the `MNS` package which can be downloaded from CRAN [143]. This implementation includes multi-core capabilities via the use of the `doParallel` package [179]. This package also includes the network simulation algorithm discussed in Appendix C.2.

Chapter 2

Prerequisites

The methodology presented in this thesis relies largely upon a collection of mathematical techniques which we introduce in this chapter. The content in this chapter thereby serves to introduce the building blocks for the remainder of this thesis. This chapter is not intended to serve as a thorough review of the topics discussed but rather to provide the necessary background to derive and motivate the work discussed in the remaining chapters.

We begin the chapter by introducing and discussing penalized likelihood methods in Section 2.1, where we review maximum likelihood based parametric estimation methods under the presence of regularization penalties. These methods will form the backbone of the methodology discussed in this thesis and we therefore provide a justification for the use of such regularization penalties. In Section 2.2 we introduce and discuss Gaussian graphical models (GGMs), which are the focus of much of this thesis. After detailing some of the properties of GGMs, this section discusses the challenges associated with their estimation in the context of high-dimensional data and details how regularization penalties may be introduced in order to yield graphs with sparse edge structure. This provides a segue into Section 2.3, where several optimization techniques are introduced and discussed. Finally, in Section 2.4, the aforementioned methods are tied into the study of functional MRI (fMRI) data. This section serves to provide a concise background on the methodologies applied in the context of fMRI data as well as to further motivate the introduction of regularization penalties.

2.1 Penalized likelihood methods

The objective of this thesis is to employ parametric methods to accurately model high-dimensional data. The work presented in this section will focus on two associated challenges: accurately estimating the model parameters and performing model selection. We discuss each of these challenges in turn.

Arguably the most popular and widely used method through which to estimate parameters, $\theta \in \mathbb{R}^p$, in a parametric model is maximum likelihood estimation [175]. Such methods typically assume observations are independent and identically (IID) sampled from a distribution with a probability density function parameterized by unknown θ. The associated likelihood function corresponds to the joint density of the data as a function of θ and can be subsequently maximized as a function of parameters to obtain maximum likelihood estimates. As a result, given random variables $X_1, \ldots, X_n$ sampled IID from a probability density, $f(x; \theta)$, the associated likelihood is defined as:

$$\mathcal{L}_n(\theta) = \prod_{i=1}^{n} f(X_i; \theta). \tag{2.1}$$

The maximum likelihood estimator, $\hat{\theta}$, is subsequently defined as:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \ \{\log \mathcal{L}_n(\theta)\} \tag{2.2}$$

The widespread use of maximum likelihood estimation is justified by the wealth of theory available regarding the properties associated with such estimators, such as asymptotic normality and efficiency [175].

However, in this thesis we are interested in the special case where the number of relevant observations, n, in small relative to the number of parameters which must be estimated, p. This is generally referred to as the large p, small n scenario where traditional maximum likelihood methods are often not applicable or desirable. Formally, if $p \approx n$ or $p > n$ maximum likelihood estimates will be poor estimates of the *true* parameter θ. Furthermore, in the latter case the estimates will not be unique as the problem is over-parameterized. This raises concerns both from the perspective of model interpretation as well as model

performance, where overfitting is likely to occur [77]. This problem is particularly relevant in the context of fMRI data as the number of parameters, p, may sometimes be in the thousands while the number of samples remains relatively low due to the low temporal resolution (see Section 2.4 for further discussion and examples). Furthermore, in the context of non-stationary data, the number of relevant observations is further reduced as the statistical properties vary.

The aforementioned challenges have traditionally been handled via the use of model selection techniques. The objective of such methods is to automatically select a subset of all candidate variables [82], thereby reducing the number of parameters to be estimated. Formally, such methods are motivated by the following concerns:

(a) **interpretability**: reducing the number of parameters helps to improve model interpretability. This is particularly relevant when there is reason to suspect the underlying model is sparse, as is often the case in neuroscientific and many biological applications [77, 149].

(b) **performance**: reducing model complexity reduces the risk of overfitting and may therefore lead to better performance on unseen data.

(c) **computational considerations**: more parsimonious models will incur a lower computational cost during prediction or classification. Moreover, many of methods discussed also provide computational advantages during model estimation. Finally, it is important to note that parsimonious models may result in significant reductions to the memory burden.

However, while model selection yields several important advantages, it also presents a significant methodological challenge. Formally, searching over the parameter space to learn the optimal subset of parameters is non-trivial. Intuitive methods based on greedy algorithms are known to perform poorly and may incur a large computational cost as models are re-fitted at each iteration [53].

In this thesis, we study algorithms which perform model selection by minimizing a penalized negative log-likelihood objective*. Such methods effectively constrain the pa-

*note that this is equivalent to maximizing the the log-likelihood function, as detailed in equation (2.2).

rameter space of candidate solutions, thereby enforcing constraints on the estimated models. From an optimization perspective, the class of penalized likelihood methods solve the following problem:

$$\underset{\theta}{\text{minimize}} \{-\log \mathcal{L}_n(\theta)\} \quad \text{subject to} \quad \mathcal{P}(\theta) \leq t, \tag{2.3}$$

where $\mathcal{P}(\theta)$ denotes a measure of model complexity which we wish to constrain. While a vast array of penalties may be considered, in this work we focus primarily on the use of ℓ_1 penalties where $\mathcal{P}(\theta) = ||\theta||_1$. The use of ℓ_1 regularization is widely employed as it retains the convexity of the original objective function whilst also enforcing sparsity in ways we will demonstrate in the next section. Convexity is a highly desirable property as it greatly simplifies the corresponding optimization procedures and allows for the derivation of scalable and efficient algorithms [27].

It is often convenient to re-write equation (2.3) in Lagrangian form [77]:

$$\underset{\theta}{\text{minimize}} \{-\log \mathcal{L}_n(\theta) + \lambda \, \mathcal{P}(\theta)\} \tag{2.4}$$

where $\lambda \in \mathbb{R}_+$ is a regularization parameter which dictates the severity of the regularization. It is important to note that there is a one-to-one relationship exists between λ in equation (2.4) and the budget constraint, t, in equation (2.3). The objective in equation (2.4) is subsequently minimized as a function of θ using an array of convex optimization techniques, some of which are discussed in Section 2.3.

2.1.1 Penalized linear regression

In order to provide a flavor for the penalized likelihood methods studied in this thesis we consider the special case of penalized linear regression. This corresponds to the Least Absolute Selection and Shrinkage Operator (lasso) model originally proposed by [166].

Given a univariate response variable, $Y \in \mathbb{R}$, and associated p-dimensional covariate, $X^T \in \mathbb{R}^p$, the simplest parametric model to consider is linear regression. Such a model

assumes the response is a linear combination of the covariates[†]:

$$Y = X\beta + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{2.5}$$

where $\beta \in \mathbb{R}^p$ is a vector of regression coefficients. Such a model is parameterized by $\theta = (\beta, \sigma)$ where σ is typically treated as a nuisance parameter. Given a dataset consisting of multiple response-covariate pairs, $\{(y_i, X_i) : i = 1, \ldots, n\}$, it is possible to estimate the regression coefficients as follows:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\text{minimize}} \left\{ \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \, ||\beta||_1 \right\}. \tag{2.6}$$

The regularization parameter, $\lambda \in \mathbb{R}_+$, dictates the severity of the regularization penalty. For $\lambda = 0$, the objective in equation (2.6) corresponds to traditional least squares and $\hat{\beta}$ will correspond to the maximum likelihood estimate. As a result, no regularization is enforced and all regression coefficients are included in the model. As λ increases, the ℓ_1 penalty increasingly dominates the objective function leading to estimated of regression coefficients with increasing levels of sparsity.

From a geometric perspective, the introduction of an ℓ_1 penalty forces the solution to lie on a scaled ℓ_1-simplex, shown in the middle panel of Figure [2.1]. The remaining two panels of Figure [2.1] visualize the constraint regions for ℓ_2 and $\ell_{0.5}$ norms respectively and serves to visualize the advantages of the ℓ_1 norm. Firstly, the ℓ_1 norm retains the convexity of the optimization problem. Moreover, the irregularities which occur at the corners of the ℓ_1-simplex serve to directly encourage sparse solutions. The reason for this is that in the vicinity of such a corner, the objective function specified in equation (2.6) is dominated by the ℓ_1 loss, as opposed to squared residual error. As a result, the lasso is able to perform model selection by setting some components of $\hat{\beta}$ to be exactly zero [29].

[†]We ignore the presence of an intercept as this may be absorbed into the covariate X

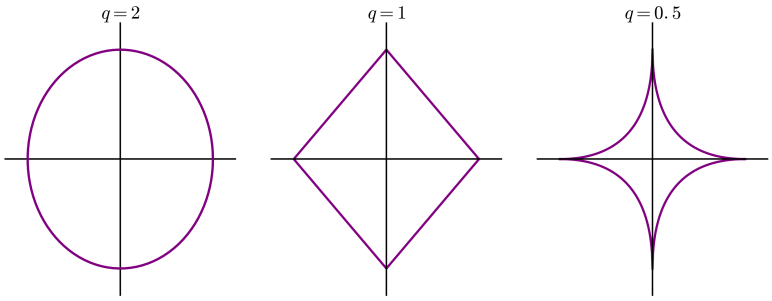


Figure 2.1: Constraint regions specified by three distinct ℓ_q regularization penalties for $q = 0.5, 1$ and 2. We note that as soon as $q < 1$ the constraint region is no longer convex.

2.2 Gaussian graphical models

The majority of this thesis is dedicated to the study of data assumed to follow a multivariate Gaussian distribution. From a methodological perspective, this choice is motivated by the fact that such an assumption yields graphical models over covariates which are easily interpretable and for which efficient and tractable estimation algorithms can be derived [100]. In this section we provide a brief review of the methodology and algorithms associated with Gaussian graphical models (GGMs). We provide justifications for such models in the context of fMRI data in Section 2.4.

2.2.1 Properties of multivariate Gaussian data

The multivariate Gaussian distribution is an extension of the univariate Gaussian distribution to higher dimensions. We write $X \sim \mathcal{N}(\mu, \Sigma)$ to denote a random vector following a multivariate Gaussian distribution with mean μ and covariance Σ. The associated probability density function is defined as:

$$f(X; \mu, \Sigma) = (2\pi)^{-\frac{p}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right\}, \tag{2.7}$$

where parameters $\mu \in \mathbb{R}^p$ and $\Sigma \in \mathbb{R}^{p\times p}$ denote the mean and covariance respectively. No assumptions are made with respect to these parameters expect to enforce that Σ should be symmetric and positive-definite.

A p-dimensional random vector $X \sim \mathcal{N}(\mu, \Sigma)$ may be partitioned in to $X_v \in \mathbb{R}^q$, $X_u \in \mathbb{R}^r$ with $q + r = p$. The mean and covariance matrix are correspondingly partitioned into:

$$\mu = \begin{pmatrix} \mu_v \\ \\ \mu_u \end{pmatrix} \qquad \text{and} \qquad \Sigma = \begin{pmatrix} \Sigma_{vv} & \Sigma_{vu} \\ \\ \Sigma_{uv} & \Sigma_{uu} \end{pmatrix}$$

Proposition 1 *The marginal distribution of X_v is a multivariate Gaussian with mean μ_v and covariance Σ_{vv} (similarly for X_u).*

Proposition 2 *The conditional distribution of X_v given $X_u = x_u$ follows a multivariate Gaussian distribution with mean:*

$$\mu_{v|u} = \mu_v - \Sigma_{vu}\Sigma_{uu}^{-1}(x_u - \mu_u), \tag{2.8}$$

and covariance:

$$\Sigma_{v|u} = \Sigma_{vv} - \Sigma_{vu}\Sigma_{uu}^{-1}\Sigma_{uv}. \tag{2.9}$$

Proposition 3 *Vectors X_v and X_u are independent if and only if $\Sigma_{vu} = \mathbf{0}$.*

Proof The proofs for Propositions 1 and 2 are standard and can be found in [100] while the proof for Proposition 3 follows by considering the conditional mean and covariance of X_v given X_u described in equations (2.8) and (2.9).

Proposition 4 *Given $X \sim \mathcal{N}(\mu, \Sigma)$, then two distinct univariate components X_v and X_u will be conditionally independent given all remaining variables if and only if the corresponding entries in the precision matrix are zero. Moreover, we write:*

$$X_v \perp\!\!\!\perp X_u | X_{\setminus\{v,u\}} \iff \Theta_{vu} = 0 \tag{2.10}$$

to denote the conditional independence of v and u given all remaining variables. Further, $\Theta = \Sigma^{-1}$ denotes the inverse covariance (i.e., precision) matrix and Θ_{vu} denotes the corresponding entry in the precision matrix

Proof We note that the precision matrix for the conditional distribution of $\{v, u\}$ given $X_{\setminus\{v,u\}}$ is specified as:

$$\Theta_{\{v,u\}} = \begin{pmatrix} \Theta_{vv} & \Theta_{vu} \\ \Theta_{uv} & \Theta_{uu} \end{pmatrix} \in \mathbb{R}^{2\times 2}. \tag{2.11}$$

As a result, the corresponding covariance matrix is defined as:

$$\frac{1}{\det \Theta_{\{v,u\}}} \begin{pmatrix} \Theta_{uu} & -\Theta_{vu} \\ -\Theta_{uv} & \Theta_{vv} \end{pmatrix}. \tag{2.12}$$

It therefore follows from Proposition 3 that Θ_{vu} must be zero for the associated components to be conditionally independent.

Proposition 4 is significant as it implies that we may infer the conditional dependence structure of a multivariate Gaussian distribution by recovering the non-zero entries of the precision matrix.

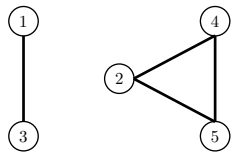
2.2.2 Covariance selection

Graphical models, by leveraging concepts in both probability theory and graph theory, have established themselves as important tools in applied statistics. Such methods represent random vectors, $X \in \mathbb{R}^p$, via a set of nodes $V = \{1, \ldots, p\}$. The corresponding edge structure, E, across nodes is then used to encodes the statistical dependencies across components of X. In this manner, we may describe high-dimensional distributions in a concise manner.

The methods described in this thesis are focused exclusively on GGMs, where random vectors are assumed to follow a multivariate Gaussian distribution. Following Proposition 4, estimating the edge structure for GGMs requires inferring the sparse support for the associated precision matrix, Θ. Formally, in GGMs an edge exists between random variables if and only if the corresponding entry in the precision matrix is non-zero. Due to the symmetric nature of the precision matrix, the edge structure in GGMs is undirected. GGMs thereby provide a medium through which to visualize and study the covariance structure in an intuitive manner as demonstrated in Figure [2.2].

$$\Theta = \begin{pmatrix} 1 & 0 & 0.3 & 0 & 0 \\ 0 & 1 & 0 & 0.3 & 0.2 \\ 0.3 & 0 & 1 & 0 & 0 \\ 0 & 0.3 & 0 & 1 & 0.2 \\ 0 & 0.2 & 0 & 0.2 & 1 \end{pmatrix}$$

a)



b)

Figure 2.2: An example of a sparse precision, Θ, is shown in panel **a)**. The graph of the GGM associated with such a precision matrix is visualized in panel **b)**. We note that edges are only present if the corresponding entry in Θ is non-zero.

The challenge of estimating the sparse support of a precision matrix is widely referred to as *covariance selection* [46]. Covariance selection is a difficult statistical problem. In a high-dimensional setting, this is primarily a result of the sample covariance matrix being poorly conditioned [86, 101]. As a result, it is often difficult to distinguish true conditional dependence relationships from spurious correlations induced by noise. In order to address this concern, a wide range of techniques have been proposed through which to estimate a sparse precision matrix. For example, [49] propose a simultaneous hypothesis testing procedure which is capable of controlling the overall error rate for incorrect edge inclusion. Alternative approaches look to recover a sparse precision via the optimization of a penalized likelihood objective [10, 68, 172], as discussed in Section 2.1. In this work, we focus on the latter approach. This is motivated by the fact that such methods correspond to solving convex optimization problems, for which there is a wide range of highly efficient and scalable algorithms. Moreover, such methods can be easily adapted to handle non-stationary data and provide the additional advantage of allowing for the introduction of further regularization penalties, for example Chapters 3 and 4 consider the introduction of an additional Fused lasso penalty [167].

As a result, the methodology detailed in the remainder of this thesis is primarily concerned with the estimation of sparse precision matrices via the introduction of regularization constraints, typically in the form of ℓ_1 penalization. As demonstrated in Figure [2.2], the estimated precision matrix can be represented as an undirected graph or network.

In the remainder of this section we detail two penalized likelihood methods for estimating sparse edge structure in a GGM. The first method corresponds to the Graphical lasso,

proposed by [68]. We then introduce neighbourhood selection, proposed by [117].

Graphical lasso

The Graphical lasso seeks to recover a sparse GGM by imposing an ℓ_1 regularization penalty on the entries of the precision matrix. Following from Section 2.2.1, the introduction of an ℓ_1 penalty is motivated by the desire to recover a parsimonious GGM which accurately describes the conditional dependence structure.

We assume we have n IID samples $X_1, \ldots, X_n$ from a multivariate Gaussian distribution. Following from equation (2.7), we note that the (scaled) log-likelihood takes the following form:

$$\log \mathcal{L}_n(\Theta) = \log \det \Theta - \text{trace}\left(S\Theta\right), \tag{2.13}$$

where S denotes the sample covariance matrix. The maximum likelihood estimate of the precision matrix corresponds to the inverse covariance matrix, however, this is may be poorly conditioned in the context of high-dimensional data [101]. As a result, we look to estimate the precision matrix by solving the following optimization problem:

$$\Theta = \underset{\Theta \succ 0}{\text{argmin}} \left\{ - \log \mathcal{L}_n(\Theta) + \lambda \, ||\Theta||_1 \right\}, \tag{2.14}$$

where we write $\Theta \succ 0$ to denote that the estimated precision matrix must be positive definite and $||\Theta||_1$ denotes the sum of absolute values of the precision matrix. The introduction of an ℓ_1 penalty serves to constrain the elements of the estimated precision matrix, potentially setting entries to be exactly zero.

The choice of the regularization parameter, λ, is an important consideration as it has a significant effect on the properties of the estimated GGM. A wide variety of methods have been proposed through which to tune λ. These include traditional cross-validation methods [68], minimizing information theoretic measures such as BIC [100] or employing stability based methods [107]. The latter methods, inspired by [118], involve randomly sub-sampling the data and either selecting edges which are consistently present under randomized penalization (thus avoiding the explicit choice of the regularization parameter) or selecting λ in order to minimize variability across sub-sampled graphs [107].

We further note that the objective function detailed in equation (2.14) is convex, implying a wide range of convex optimization techniques may be employed [170]. These are discussed in further detail in Section 2.3.

Neighbourhood selection

In this section we consider neighbourhood selection [117]. This can be considered as an approximation to the exact optimization problem described in equation (2.14) which enjoys the benefits of being computationally efficient and highly scalable. The underlying intuition behind neighbourhood selection stems from the fact that we may derive the overall edge structure for a GGM by iteratively inferring the conditional dependence structure for each node. The latter is referred to the neighbourhood of a node [100]. We write $ne(v)$ to denote the neighbourhood for a given node $v \in V$.

In order to derive neighbourhood selection, we return to Proposition 2 and consider the conditional distribution of a node, v, given all remaining nodes, $X_{V\setminus\{v\}} = x_{V\setminus\{v\}}$. This follows a univariate Gaussian distribution whose mean is defined as [100]:

$$\mu_{v|V\setminus\{v\}} = \mu_v - \Sigma_{v,V\setminus\{v\}} \left(\Sigma_{V\setminus\{v\},V\setminus\{v\}}\right)^{-1} \left(x_{V\setminus\{v\}} - \mu_{V\setminus\{v\}}\right) \tag{2.15}$$

$$= \mu_v + \sum_{u\in V\setminus\{v\}} \beta_u^v \left(x_u - \mu_u\right). \tag{2.16}$$

Equations (2.15) and (2.16) demonstrate that the observations for a given node, v, can be decomposed as a linear prediction based on the data at the remaining nodes where the conditional dependence structure is entirely captured by the regression coefficients, β^v.

As a result it follows that we are able to learn the conditional dependence structure for each node, termed the neighbourhood of a node, by considering the optimal prediction given the observations of the remaining nodes [77]. From equation (2.16), this reduces to a linear regression of X_v on $X_{\setminus\{v\}}$:

$$X_v = X_{\setminus\{v\}}\beta^v + \epsilon^v, \tag{2.17}$$

where ϵ^v is a univariate Gaussian variable with mean zero. In such a regression model,

it follows that nodes which are not in the the neighbourhood of v will be omitted from the set of optimal predictors. Neighbourhood selection can therefore be reformulated as variable selection in a linear regression model, which can be achieved via the introduction of an ℓ_1 penalty as discussed in Section 2.1.1. Due to the parsimony properties of the lasso, certain elements of β^v will be shrunk to zero. An estimate for the neighbourhood of v is subsequently defined as:

$$\hat{ne}(v) = \left\{ u \in V \backslash \{v\} : \hat{\beta}_u^v \neq 0 \right\}. \tag{2.18}$$

The estimated neighbourhood of a node is therefore defined as the set of all nodes included in the lasso solution. Given the estimated neighborhood of each node in a graph it is possible to infer the edge structure for the entire GGM. In practice, estimation error may introduce discrepancies into the estimated neighborhoods across nodes. For example, it may be the case that $v \in \hat{ne}(u)$ while $u \notin \hat{ne}(v)$, which is incompatible with the undirected nature of edges in GGMs. As a result, the following heuristic rules are proposed [117]:

$$E_{OR} = \{(v,u) : u \in \hat{ne}(v) \textbf{ or } v \in \hat{ne}(u)\} \text{ or } E_{AND} = \{(v,u) : u \in \hat{ne}(v) \textbf{ and } v \in \hat{ne}(u)\}. \tag{2.19}$$

The use of neighbourhood selection leads to many important advantages. Primarily, it is easily amenable to parallelization, allowing such methods to be employed in high-dimensional settings. This is in contrast to the graphical lasso, which simultaneously estimates the entire GGM. However, one fundamental shortcoming of neighbourhood selection methods is that we are only able to recover the edge structure as opposed to the associated precision matrix.

2.3 Convex optimization methods

In the preceding sections we have discussed penalized likelihood methods and their application to covariance selection. One of the fundamental advantages of such methods is their convexity, allowing a wide array of convex optimization methods to be employed.

Convex optimization techniques are prevalent within statistics, the prototypical example being least squares regression. Formulating problems in the context of convex optimization yields important theoretical and practical advantages. From a theoretical perspective,

we are able to obtain a host of necessary and sufficient conditions through which to check the optimality of a solution [27].

In the remainder of this section we consider two methods through which to solve convex optimization problems and relate them back to the original problems described in Section 2.1 and 2.2. We begin by discussing traditional gradient and proximal gradient methods in Section 2.3.1 before discussing Alternative Direction Methods of Multipliers (ADMM) algorithms in Section 2.3.2.

2.3.1 Gradient methods

Gradient descent methods correspond to a class of iterative algorithms for solving optimization problems. In this section we focus exclusively on first order descent methods, implying only information relating to the gradient is employed. Such methods are attractive as they avoid the computational burden associated with calculating higher order derivatives, making them suitable for large scale problems.

Throughout this section we consider the optimization of an objective function,

$$f : \mathbb{R}^p \to \mathbb{R}.$$

In many cases, we are able to exploit specific properties of the objective function in order to derive efficient optimization algorithms. Two properties which will prove fundamental to the algorithms presented in this thesis are convexity and separability, defined below.

Definition 2.3.1 *A function, $f : \mathbb{R}^p \to \mathbb{R}$, is convex if it satisfies:*

$$f\left(\alpha x_1 + (1-\alpha)x_2\right) \leq \alpha f(x_1) + (1-\alpha) f(x_2) \tag{2.20}$$

for any x_1, x_2 in the domain of f and any $\alpha \in [0, 1]$.

As such, a convex function is any function where the tangent between any two points on the curve always lies above the curve. From a geometric perspective, this property implies that the gradient always points in the direction of the global minimum.

Definition 2.3.2 *A function, $f : \mathbb{R}^p \to \mathbb{R}$, is separable with respect to partition or splitting the variable θ if we may write:*

$$f(\theta) = \sum_{i=1}^{N} f_i(\theta_i) \tag{2.21}$$

where $\theta = (\theta_1, \ldots, \theta_N)$ and the variables $\theta_i \in \mathbb{R}^{n_i}$ are subvectors of θ.

Throughout this section we focus on optimization problems were the objective function is convex and separable but not necessarily differentiable. For example, this may correspond to the lasso objective presented in equation (2.6) or the Graphical lasso objective detailed in equation (2.14). Under the assumption that the objective function is differentiable and convex, a necessary and sufficient condition for a solution, θ^*, to be optimal is [134]:

$$\nabla f(\theta)|_{\theta=\theta^*} = 0. \tag{2.22}$$

In the remainder of this thesis we abuse notation and directly write $\nabla f(\theta^*)$ to denote the derivative of f evaluated at θ^*. In some cases, such as linear regression, it is possible to solve equation (2.22) directly. However, for general complex problems a solution must be obtained by iterating through a minimizing sequence of candidate solutions, $\theta^1, \theta^2, \ldots$, each of which progressively approximates the solution [15].

Gradient descent algorithms iteratively produce such a sequence as follows:

$$\theta^{k+1} = \theta^k - \eta^k \nabla f(\theta^k), \tag{2.23}$$

where η^k denotes the stepsize parameters which can be selected in various different ways in order to guarantee convergence [26, 27]. The update described in equation (2.23) has a natural geometric interpretation. At each step the gradient is calculated, which indicates the direction of steepest descent along the objective function. Each iteration therefore takes a step in this direction before the gradient is calculated once more. The stopping criteria for such algorithms is typically determined by measuring the magnitude of each gradient, $\nabla f(\theta^k)$, and declaring convergence once this falls below a threshold.

One of the advantages of employing gradient methods to optimize convex objective functions is that regardless of the initial choice of the parameter of interest, θ^0, the proposed

algorithm is still guaranteed to converge to the global minimum [27]. In other words, iteratively applying the recursive update defined in equation (2.23) guarantees convergence regardless of the initial θ^0. This is desirable property as it implies that θ^0 may be specified arbitrarily. However, it follows that specifying θ^0 close to the global minimum will greatly reduce the computational burden associated with iterative gradient descent methods. A frequently employed approach is that of *warm starts*, where information from a highly related problem is employed to guide the choice of θ^0 [92]. While there are many distinct *warm start* strategies available, throughout this thesis we only consider specifying θ^0 to be the solution (i.e., θ^*) for a related problem. Many of the problems considered in this thesis will require solving a sequence of closely related optimization problems and are therefore well suited to benefit from *warm starts*.

Proximal gradient algorithms

Optimization problems which involve a non-differentiable objective, such as problems which enforce ℓ_1 regularization, preclude the use of traditional gradient methods [155]. As a result, we consider proximal algorithms, which are a general class of optimization algorithms which can handle non-differentiable objectives [136].

Definition 2.3.3 *For a convex function $g : \mathbb{R}^n \to \mathbb{R}$ the associated proximal operator, denoted $\mathbf{prox}_g : \mathbb{R}^n \to \mathbb{R}^n$, is defined as follows:*

$$\mathbf{prox}_g(\omega) = \underset{\theta}{\operatorname{argmin}} \left(g(\theta) + \frac{1}{2}||\theta - \omega||_2^2 \right) \tag{2.24}$$

The operator derives it name from the fact that $\mathbf{prox}_f(\omega)$ balances a compromise between minimizing f and remaining in the vicinity of (i.e., proximal to) ω.

One of the crucial properties of such operators is that the fixed point of a proximal operator are precisely the minimizers of g [136]:

$$\mathbf{prox}_g(\theta^*) = \theta^* \iff \theta^* \text{ is a minimizer of } g.$$

Proximal algorithms iteratively evaluate the proximal operator of an objective. As such, they can be interpreted as solving optimization problems by finding the fixed points of the appropriate operator.

Recall that in Section 2.1 we considered penalized likelihood functions of the form:

$$f(\theta) + g(\theta) \tag{2.25}$$

where f typically denotes the negative log-likelihood and g denotes the regularization penalty. Furthermore, in the context of penalized likelihoods it is often the case that while both f and g are convex, only f is differentiable. Proximal gradient algorithms are particularly well-suited to the minimization of such objectives as they exploit both the separable nature of the problem as well as the differentiability of f.

Formally, proximal gradient algorithms produce a sequence of candidate solutions as follows:

$$\begin{aligned} \omega^{k+1} &= \theta^k - \eta^k \nabla f(\theta^k) \\ \theta^{k+1} &= \mathbf{prox}_g\left(\omega^{k+1}\right) \end{aligned} \tag{2.26}$$

Setting $g = 0$ removes regularization constraints and recovers gradient descent described in equation (2.23).

The appeal of proximal gradient methods is derived from the fact that while evaluating the proximal operator involves solving a convex optimization problem, in many cases the solution is available in closed form. In particular, when $g(\theta) = \lambda||\theta||_1$ the proximal operator is the element-wise soft-thresholding operator [77]:

$$(\mathbf{prox}_g(\omega))_i = \mathcal{S}_\lambda(\omega_i) = \begin{cases} \omega_i - \lambda, & \text{if } \omega_i \geq \lambda \\ 0, & \text{if } |\omega_i| \leq \lambda \\ \omega_i + \lambda, & \text{if } \omega_i \leq \lambda \end{cases} \tag{2.27}$$

In order to provide a flavor for the proposed methods we consider the lasso. The objective, originally defined in equation (2.6), is separable with:

$$f(\beta) = \sum_{i=1}^{n} (Y_i - X_i\beta)^2 \quad \text{and} \quad g(\beta) = \lambda||\beta||_1.$$

As a result, the associated proximal gradient algorithm employs the following update step:

$$\beta^{k+1} = \mathcal{S}_{\eta^k\lambda}\left(\beta^k - 2\eta^k \sum_{i=1}^{n} X_i^T(Y_i - X_i\beta^k)\right), \tag{2.28}$$

where the soft-thresholding function is applied element-wise. Equation (2.28) is then iteratively applied until convergence. This is closely related to the shooting algorithm for the lasso, proposed in [67] and [71] with the only difference that all regression coefficients are updated simultaneously. In practice, the use of a proximal algorithm of the shooting algorithm has been shown to yield similar performance [77].

2.3.2 ADMM algorithms

Alternating directions method of multipliers (ADMM) algorithms are a special case of proximal algorithms which are typically employed to further exploit the separability of the objective function or when the evaluation of the proximal operator is non-trivial [26]. Formally, the ADMM algorithm proceeds to solve the following problem:

$$\begin{aligned} &\text{minimize} \quad f(\theta) + g(z) \\ &\text{subject to} \quad \theta - z = 0. \end{aligned} \tag{2.29}$$

This is often referred to as the *consensus form* of the problem [136]. We note that the parameter of interest has been split into two parts denoted by θ and z such that the objective function is separable across this split. An additional *consensus constraint* has also been added to ensure the variables agree.

The augmented Lagrangian [15] associated with the optimization problem detailed in

equation (2.29) is:

$$L(\theta, z, u) = f(\theta) + g(z) + u^T(\theta - z) + \frac{1}{2}||\theta - z||_2^2 \tag{2.30}$$

$$= f(\theta) + g(z) + \frac{1}{2}||\theta - z + u||_2^2 - \frac{1}{2}||u||_2^2, \tag{2.31}$$

where u is the Lagrangian dual variable associated with the constraint that $\theta = z$. Equation (2.30) corresponds to the usual Lagrangian with an additional quadratic penalty. The introduction of this term is desirable as it often facilitates the minimization of the Lagrangian [26]. The ADMM algorithm then iterates:

$$\theta^{k+1} = \mathbf{prox}_f(z^k - u^k) = \underset{\theta}{\operatorname{argmin}} \left\{ L(\theta, z^k, u^k) \right\} \tag{2.32}$$

$$z^{k+1} = \mathbf{prox}_g(\theta^{k+1} + u^k) = \underset{z}{\operatorname{argmin}} \left\{ L(\theta^{k+1}, z, u^k) \right\} \tag{2.33}$$

$$u^{k+1} = u^k + \theta^{k+1} - z^{k+1}. \tag{2.34}$$

This decoupling of the original objective allows for ADMM algorithms to exploit the structure associated with each of the functions f and g separately. At each iteration, minimization is performed first with respect to θ variables and subsequently with respect to z variables. Finally, a dual variable update is then employed in order to ensure variables θ and z converge towards each other [15].

An ADMM algorithm for the Graphical lasso

In order to highlight the simplicity of ADMM algorithms we consider the graphical lasso problem, defined in equation (2.14). This corresponds to a widely studied optimization problem whose difficulty also lead to novel approximation algorithms such as neighbourhood selection presented in Section 2.2.2. The associated ADMM algorithm looks to solve the following problem:

$$\begin{aligned} &\text{minimize} \quad -\log\det\Theta + \text{trace}\,(S\Theta) + \lambda||Z||_1 \\ &\text{subject to} \quad \Theta - Z = 0, \end{aligned} \tag{2.35}$$

for which each of the steps outlined in equations (2.32) - (2.34) are as follows:

$$\Theta^{k+1} = \underset{\Theta}{\operatorname{argmin}} \left\{ -\log\det\Theta + \text{trace}\,(S\Theta) + \frac{1}{2}||\Theta - Z^k + U^k||_2^2 \right\} \tag{2.36}$$

$$Z^{k+1} = \underset{Z}{\operatorname{argmin}} \left\{ \lambda||Z||_1 + \frac{1}{2}||\Theta^{k+1} - Z + U^k||_2^2 \right\} \tag{2.37}$$

$$U^{k+1} = U^k + \Theta^{k+1} - Z^{k+1}. \tag{2.38}$$

From equation (2.27) we note that the update of the Z variable is nothing more than an element-wise application of soft-thresholding. With regards to updating Θ, we note that the objective specified in equation (2.36) is differentiable with respect to Θ. As a result, differentiating the right hand side of equation (2.36) and setting the derivative to zero yields:

$$\Theta^{-1} - \Theta = S - \left(Z^k - U^k\right). \tag{2.39}$$

Equation (2.39) yields an expression quadratic in Θ which can be solved analytically, as detailed in the following proposition.

Proposition 2.3.4 *If symmetric matrices $X, Y \in \mathbb{R}^{p\times p}$ satisfy $X^{-1} - \alpha X = Y$ for some constant α then it follows that X and Y have the same eigenvectors. Furthermore it is also the case that the ith eigenvalues of X and Y, denoted by λ_{X_i} and λ_{Y_i} respectively, will satisfy $\lambda_{X_i}^{-1} - \alpha\lambda_{X_i} = \lambda_{Y_i}$ for $i \in \{1, \ldots, p\}$*

Proof In order to prove claim 2 we begin taking the eigendecompositions of X and Y as $\Omega_X\Lambda_X\Omega_X^T$ and $\Omega_Y\Lambda_Y\Omega_Y^T$ respectively. Substituting these into $X^{-1} - \alpha X = Y$ we obtain:

$$(\Omega_X\Lambda_X\Omega_X^T)^{-1} - \alpha(\Omega_X\Lambda_X\Omega_X^T) = \Omega_Y\Lambda_Y\Omega_Y^T$$

Expanding the left hand side yields:

$$\begin{aligned}\Omega_X\Lambda_X^{-1}\Omega_X^T - \alpha(\Omega_X\Lambda_X\Omega_X^T) &= \Omega_Y\Lambda_Y\Omega_Y^T \\ &= \Omega_X(\Lambda_X^{-1} - \alpha\Lambda_X)\Omega_X^T\end{aligned}$$

where we have made use of the fact that Ω_X is an orthonormal matrix. Thus it follows that

$\Omega_X = \Omega_Y$ and since both Λ_X and Λ_Y are diagonal matrices we also have that $\lambda_{X_i}^{-1} - \alpha\lambda_{X_i} = \lambda_{Y_i}$ for $i \in \{1, \ldots, p\}$ ∎

From Proposition 2.3.4, it follows that the solution to equation (2.36) can be obtained in closed form where the update for Θ is obtained by computing the eigendecomposition of $S - Z^k + U^k$. As a result, the ADMM algorithm specified by equations (2.36)-(2.38) alternates between updating variables Θ, Z and U; all of which involve solving optimization problems with closed-form solutions.

2.4 Analysis of functional MRI data

The preceding sections have set the foundations from a mathematical perspective. The objective of this section is to provide the necessary background on the study of functional MRI (fMRI) data as well as the closely related topic of functional connectivity. Both these topics raise important statistical challenges, which we hope to address using the aforementioned techniques.

While there are a wide range of imaging modalities available, the applications presented in this thesis are focused on the use of fMRI data. Functional MRI is an imaging technique which is widely employed as a result of its safe, non-invasive nature together with the high spatial resolution it is able to provide [88]. The underlying idea behind fMRI relies on quantifying changes in blood oxygenation which occur during neuronal activity. As a result, the signal measured by fMRI is referred to as the blood oxygenation level dependent (BOLD) signal [140].

As fMRI is fundamentally based on blood flow, there are several important properties to consider. Formally, the increase in blood flow which follows a period of neuronal activity is determined by the hemodynamic response. One of the important properties of the hemodynamic response is that it occurs over a time scale of seconds. As a result, the temporal resolution of fMRI data is far lower than alternative modalities which are not based on blood flow [105]. From a practical perspective, the low resolution of fMRI data results in a reduced number of observations. This may cause challenges, as the high spatial resolution yields high-dimensional data, resulting in the class of large p, small n problems discussed

in Section 2.1. As a result, regularization penalties are often introduced when studying fMRI data.

There are several common objectives typically associated with the study of fMRI data. These include understanding which brain regions are modulated by specific tasks as well as investigating the networks of multiple regions associated with specific brain functions. Throughout this thesis the focus is on the latter objective by estimating functional connectivity networks, introduced below.

2.4.1 Functional connectivity

The primary objective of early neuroscientific research was to establish and explore the functional segregation of the human brain. This refers to the view that specific brain regions support specific tasks, ofter termed functional localization or segregation [69, 161]. While the functional segregation of brain is firmly established, there has since been a significant shift towards the study of functional dependence or connectivity across regions, termed functional integration. One potential avenue through which to study the functional integration of the brain is via functional connectivity. Formally, the objective of functional connectivity is to quantify the statistical dependencies which exist across spatially remote brain regions. Accurately estimating functional connectivity networks therefore corresponds to a statistical challenge.

It follows that studying the brain as network of functionally related regions can provide new insights relating to the architecture and large-scale structure of the human connectome, such as the small-world structure [13, 162] and the presence of hubs [54]. Moreover, functional connectivity provides a platform through which to examine how changes in organization and structure in such networks may relate to a various neurological and psychiatric conditions [168].

Throughout this thesis we focus on the study of brain connectivity from the perspective of functional connectivity. However, it is important to note that brain connectivity is a widely studied topic which can be addressed by many alternative frameworks. Two popular alternatives include the study of anatomical and effective connectivity. The former is concerned with the study of physical connections between brain regions and is typi-

cally quantified by studying white matter tracts between regions [73]. It follows that the anatomical connectivity between regions will directly affect their capability to share functional dependencies [40]. This has motivated the introduction of anatomical connectivity networks as additional constraints in the estimation of functional connectivity [81, 132]. Conversely, the study of effective connectivity is focused on quantifying directional effects of one region over another and is therefore far more challenging to quantify [70].

A cornerstone in the study of functional connectivity, as well as brain connectivity in general, is the notion that connectivity can be represented as a graph or network composed of a set of nodes inter-connected by a set of edges. This allows for connectivity to be studied using a rich set of graph theoretic tools [63, 131] and has resulted in widespread use of graph theoretic techniques in neuroscience [1, 58].

The first step when looking to study brain connectivity is to define a set of nodes. This can be achieved through various distinct methods depending on the underlying objectives and ambitions of the data analysis. It is important to note that the choice of such regions is paramount to both the estimation and subsequent interpretation of networks [171]. Of a large number of strategies, two popular examples include anatomical parcellation methods and independent component analysis. In each case, multiple brain regions are obtained which serve as the nodes in the corresponding graph. Each node is associated with its own BOLD time series, which is subsequently studied to determine statistical dependencies and infer functional connectivity structure.

Traditionally, functional connectivity networks have been estimated by measuring pairwise dependencies across regions quantified via Pearson's correlation coefficient [63, 90]. This corresponds to estimating the correlation matrix where each entry corresponds to the correlation between a distinct pair of nodes. Partial correlations, summarized in the precision or inverse covariance matrix [181] have also been employed extensively [81, 87, 112]. In this case, the correlations between nodes are inferred once the effects of all other units have been removed. Partial correlations are typically preferred to Pearson's correlation coefficient for a host of reasons. Firstly, the use of partial correlations provides a clear interpretation for the edges in the network based on conditional dependence. Moreover, by considering correlations across nodes after regressing out the effects of other nodes, the risk of confounding is reduced [171]. As a result, partial correlations have been shown to

be better suited to detecting changes in connectivity structure [112, 157]. The focus on partial correlations, summarized by the precision matrix, makes GGMs a natural candidate in the modeling of functional connectivity networks.

2.4.2 Functional connectivity via sparse GGMs

Throughout this work we look to model functional connectivity networks using GGMs with a sparse edge structure, as introduced in Section 2.2. This yields several important benefits. Firstly, the use of GGMs allows for the derivation of scalable and efficient estimation algorithms. In particular, GGMs are easily amenable to the introduction of regularization. Second, the edge structure associated with GGMs encodes the conditional dependencies across nodes, which may be interpreted as functional relationships. This is in contrast to alternative methods, such as those based on the correlation matrix where the edges encodes marginal dependence structure [171].

Throughout this thesis the edge structure of the associated GGMs is inferred via the use of regularization methods. In particular, this thesis is focused on the use ℓ_1 regularization penalties such as those discussed earlier in this chapter which yield graphs with sparse edge structure. Intrinsically related to the use of such regularization methods is the issue of estimating the *true* sparsity of the network in question. There are many studies reporting brain networks display varying degrees of sparsity. For example, [30] suggest that connectivity networks have evolved to achieve high efficiency of information transfer at a low connection cost, resulting in sparse networks. Conversely, [111] propose a high-density model where efficiency is achieved via the presence of highly heterogeneous edge strengths between nodes. Throughout this thesis we consider the degree of sparsity as a question to be answered by the data. However, we note that regularization must be introduced to some extent in order to ensure the problem is well-posed from an optimization perspective. Table 2.1 provides a description of the datasets employed in this thesis, which are representative of the type of datasets typically employed in the study of functional connectivity. For the vast majority of the datasets employed the size of the edge set exceeds the number of observations available, indicating that regularization of some form must be introduced.

Dataset	p	n	$\lvert E \rvert$	Chapter
CRT task	18	126	153	Chapter 3
HCP Motor task	11	404	55	Chapter 4
HCP Emotion task	20	404	190	Chapter 5
HCP Working Memory task	84	404	3468	Chapter 6
ABIDE resting state	92	230	4168	Chapter 7

Table 2.1: A summary of the various datasets employed throughout this thesis. For each dataset we report the number of nodes, p, the number of observations, n, the size of the edge set, $|E|$, and the chapter in which the results are provided. It is important to note that for the vast majority of the datasets, the size of the edge set larger than the number of observations available. This issue is further exasperated when we consider the non-stationary nature of the data, which implies that only local observations may be employed.

While alternative regularization schemes, such as ℓ_2 penalization, may also be employed we focus on the use of ℓ_1 regularization for reasons of interpretability and insight. Recall that in the context of GGMs, edges denote conditional dependencies and may interpreted as functional relationships between spatially remote brain regions [157]. By pruning the edge set we are able to recover the minimal set of conditional dependencies across nodes which adequately describes the data. This serves to provide easily interpretable networks from which insights may be easily inferred.

2.5 Conclusion

In this chapter we have reviewed several topics which will form the foundation of the methods derived in subsequent chapters. We have discussed penalized likelihood methods and their relationship to covariance selection in GGMs. This in turn motivated the discussion of the convex optimization methods required to solve such problems. We concluded with a brief review fMRI and the study of functional connectivity networks together with a description of how GGMs may be employed to model functional connectivity. Finally, the introduction of sparsity constraints was motivated from the perspectives of interpretability and feasibility.

Chapter 3

Time-varying covariance selection

The focus of this chapter revolves around the estimation of time-varying Gaussian graphical models (GGMs) in the context of non-stationary data. While we consider the analysis of fMRI data as our motivation, the methods described in this chapter are applicable in many other contexts.

As described in Section 2.4, GGMs are frequently employed to model statistical dependencies across spatially remote brain regions, known as functional connectivity [70]. Traditionally, functional connectivity networks had been assumed to remain fixed over time, implying that a single GGM was sufficient to summarize the covariance structure within an fMRI dataset. However, there is growing evidence to suggest that fMRI data is non-stationary over time [91]; this is particularly true in the context of task-based fMRI studies [32], as noted in Section 2.4. In particular, functional relationships are hypothesized to be modulated by certain cognitive tasks [24, 62] implying that certain edges may be present across brain regions during a particular cognitive tasks but absent during others. As a result there is a need to quantify dynamic changes in network structure over time. From a statistical perspective, this translates to estimating time-indexed GGMs which capture the statistical dependencies at a specific point in time. Specifically, there is a need to estimate a graph at each observation in order to accurately quantify temporal diversity induced by cognitive tasks.

Formally, the objective of the methodology presented in this chapter is to infer dynamic functional connectivity networks from fMRI data which display two overriding proper-

ties: sparsity and temporal homogeneity. The former property implies estimated networks should consist only of a reduced set of all possible edges. Such constraints are introduced in order to yield easily interpretable networks as well as ensure the estimation problem is well-posed and feasible, as discussed in Section 2.4.2. Meanwhile, the property of temporal homogeneity implies estimated networks should display sparse changes in edge structure over time, thereby encouraging constant covariance structure across temporally adjacent networks. From a biological perspective, this property is motivated by reports that functional connectivity is modulated by distinct cognitive tasks [62] indicating the network structure should remain approximately constant within a neighbourhood of any observation and vary only over a larger time horizon. Moreover, the introduction of a temporal homogeneity constraint also serves as an additional mechanism, analogous to shrinkage in the static case, through which to differentiate variation in edge structure which is driven by statistical noise from true variation in the underlying covariance structure [122].

To date, the most common approach to study dynamic functional connectivity involves the use of sliding windows [90]. Such methods allow for the estimation of time-varying networks by considering only a reduced subset of temporally adjacent observations. While sliding window methods are easily amenable to the introduction of sparsity [3], they lack a mechanism through which to enforce temporal homogeneity. To this end, we present the Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm, which directly extends sliding window algorithms by enforcing additional constraints across temporally adjacent network estimates. The additional constraints, in the form of ℓ_1 penalties across temporally adjacent graphs, serve to encourage sparse differences across consecutive network estimates. The aforementioned constraints are enforced via the introduction of convex regularization penalties. As a result, the SINGLE algorithm involves the minimization of an objective function that is convex but not differentiable, thereby motivating the use of ADMM algorithms, introduced in Section 2.3.2.

The remainder of this chapter is structured as follows: in Section 3.1 we introduce and describe the SINGLE framework as well as the corresponding optimization algorithm in detail. In Section 3.2 we present the results of our simulation study which aims to study the empirical properties of the SINGLE algorithm. Throughout these simulations, synthetic data is generated in order to recreate many of the statistical properties observed in fMRI

data. We benchmark the performance of the SINGLE algorithm against sliding window based algorithms as well as recently proposed algorithms from the literature.

3.1 Time-varying GGMs

We assume we have observed fMRI time series data denoted by $X_1, \ldots, X_n$, where each vector $X_i \in \mathbb{R}^p$ contains the BOLD measurements of p nodes at the ith observation. Throughout the remainder of this section we assume that each X_i follows a multivariate Gaussian distribution, $X_i \sim \mathcal{N}(\mu_i, \Sigma_i)$. Here the mean and covariance are dependent on the observation index in order to accommodate the non-stationary nature of fMRI data.

We aim to infer functional connectivity networks over time by estimating the corresponding precision (inverse covariance) matrices $\{\Theta_i\} = \{\Theta_1, \ldots, \Theta_n\}$. Here, Θ_i encodes the partial correlation structure at the ith observation [181]. As described in Section 2.2, we may encode Θ_i as a graph or network G_i where the presence of an edge implies a non-zero entry in the corresponding precision matrix and can be interpreted as a functional relationship between the two nodes in question. Thus our objective is equivalent to estimating a sequence of time indexed graphs $\{G_i\} = \{G_1, \ldots, G_n\}$ where each G_i summarizes the functional connectivity structure at the ith observation.

The objective of this work is to estimate a sequence of graphs, $\{G_i\}$, which display the following two properties:

1. **Sparsity**: The introduction of sparsity is motivation by two reasons; first, the number of parameters to estimate often exceeds the number of observations. In this case the introduction of regularization is required in order to formulate a well-posed problem. Second, due to the presence of noise, all entries in the estimated precision matrices will be non-zero. This results in dense, unparsimonious networks with limited interpretability.

2. **Temporal homogeneity**: From a biological perspective, it has been reported that functional connectivity networks exhibit changes due to task based demands [57, 62, 64, 165]. As a result, we expect the network structure to remain constant within a neighbourhood of any observation but to vary over a larger time horizon. This

is particularly true for task-based fMRI studies where stimulus presentation often occurs in alternating blocks. In light of this, we wish to encourage estimated graphs with sparse changes in edge structure over time.

We split the problem of estimating $\{\Theta_i\}$ into two independent tasks. First we look to obtain local estimates of time-varying covariance matrices, which we denote by $\{S_i\} = \{S_1, \ldots, S_n\}$. Here each S_i is a function of the data which serves as an estimate of the time-varying covariance matrix, Σ_i. By some abuse of terminology, we refer to each S_i as the sample covariance at the ith observation in the sense that the estimate S_i serves the same purpose locally as the sample covariance would serve if the data were IID. In this work kernel functions are employed to obtain sample covariance matrices, as detailed in Section 3.1.1. Given such a sequence we wish to estimate the corresponding precision matrices $\{\Theta_i\}$ with the aforementioned properties while ensuring that each Θ_i adequately describes the corresponding S_i. The latter is quantified by considering the negative log-likelihood:

$$f\left(\{\Theta_i\}, \{S_i\}\right) = \sum_{i=1}^{n} -\text{log det } \Theta_i + \text{trace } (S_i\Theta_i). \tag{3.1}$$

While it would be possible to estimate $\{\Theta_i\}$ by directly minimizing f, this would not guarantee either of the properties discussed previously. In order to enforce sparsity and temporal homogeneity we introduce the following regularization penalty:

$$g_{\lambda_1,\lambda_2}(\{\Theta_i\}) = \lambda_1 \sum_{i=1}^{n} ||\Theta_i||_1 + \lambda_2 \sum_{i=2}^{n} ||\Theta_i - \Theta_{i-1}||_1. \tag{3.2}$$

Sparsity is enforced by the first penalty term which assigns a large cost to matrices with large absolute values, thus effectively shrinking elements towards zero. This can be seen as a convex approximation to the combinatorial problem of selecting the number of edges. The second penalty term, parameterized by λ_2, encourages temporal homogeneity by penalizing the difference between consecutive networks. This can be seen as an extension of the Fused lasso penalty [167], typically applied in the context of linear regression.

The proposed method therefore minimizes the following loss function:

$$l(\{\Theta_i\}, \{S_i\}) = f(\{\Theta_i\}, \{S_i\}) + g_{\lambda_1,\lambda_2}(\{\Theta_i\}). \tag{3.3}$$

This allows for the estimation of time-indexed precision matrices which display the properties of sparsity and temporal homogeneity while providing an accurate representation of the data, as detailed in Section 3.1.2. The choice of regularization parameters λ_1 and λ_2 allow us to balance this trade-off and can be learnt in a data driven manner as described in Section 3.1.3.

3.1.1 Estimation of time-varying covariance matrices

The loss function, summarized in equation (3.3), requires the input of sample covariance matrices $\{S_i\}$. Estimating time-varying covariance matrices is itself a non-trivial and widely studied problem. Under the assumption of IID data, the sample covariance matrix serves as a suitable estimator of the true covariance and may be readily calculated as $S = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{x})^T(X_i - \bar{x})$ where $\bar{x}$ is the sample mean. However, when faced with time-varying covariance matrices, such an approach is untenable.

A potential approach involves the use of change-point detection to segment the data into piece-wise stationary segments, as is the case in the DCR algorithm [38]. Alternatively, a sliding window may be used to obtain an estimate of the covariance matrix at each observation. Due to the sequential nature of the observations, sliding windows allow us to obtain adaptive estimates by considering only temporally adjacent observations.

A natural extension of sliding windows is to obtain adaptive estimates by downweighing past observations, thereby yielding weighted sample covariance matrices. This can be achieved using kernel functions. Formally, kernel functions have the form $K_h(i, j)$ where $K_h(\cdot, \cdot)$ is a symmetric, non-negative function, h is a specified fixed width and i and j are time indices. By considering the uniform kernel $K_h(i, j) = \mathbb{I}\{|i - j| < h\}$, where $\mathbb{I}(x)$ denotes the indicator function, we can see that sliding windows are a special case of kernel functions. This allows us to contrast the behavior of sliding windows against alternative kernels, such as the Gaussian kernel:

$$K_h(i, j) = \exp\left\{\frac{-(i - j)^2}{h}\right\}. \tag{3.4}$$

Finally, given a kernel function, we are able to obtain the weighted sample mean and sample

covariance matrix at the ith observation as follows:

$$\bar{x}_i = \frac{\sum_{j=1}^{n} K_h(i,j) X_j}{\sum_{j=1}^{n} K_h(i,j)}, \tag{3.5}$$

$$S_i = \frac{\sum_{j=1}^{n} K_h(i,j)(X_j - \bar{x}_j)^T (X_j - \bar{x}_j)}{\sum_{j=1}^{n} K_h(i,j)}. \tag{3.6}$$

It follows that for both the Gaussian kernel as well as the sliding window the choice of h plays a fundamental role. It is typically advised to set h to be large enough to ensure robust estimation of covariance matrices without making h too large [150]. However, data-driven approaches are rarely proposed [90]. This is partly because the choice of h will depend on many factors, such as the rate of change of the underlying networks, which are rarely known apriori. Here we propose to estimate h using cross-validation. This is discussed in detail in Section 3.1.3.

3.1.2 Proposed algorithm

Having obtained weighted sample covariance matrices, as discussed in Section 3.1.1, we turn to the problem of minimizing the loss function (3.3). Whilst this loss is convex, it is not continuously differentiable due to the presence of the penalty terms. In particular, the presence of the Fused lasso penalty poses a real restriction as it introduces dependencies across chronologically adjacent network estimates. Additional difficulty is introduced by the structured nature of the problem: we require that each Θ_i be symmetric and positive definite.

The approach taken here is to exploit the separable nature of equation (3.3). As discussed previously, the loss function is composed of two components; the first of which is proportional to the sum of likelihood terms and the second containing the sum of the penalty components. This separability allows us to take advantage of the structure of each component.

There has been a rapid increase in interest in the study of such separable loss functions in the statistics, machine learning and optimization literature. Here we capitalize on the separability of the objective function by employing ADMM algorithms, introduced in Section 2.3. We note that the use of an ADMM algorithm is able to guarantee estimated

precision matrices, $\{\Theta_i\}$, are symmetric and positive definite as we outline below.

Formally, the separability of the loss function (3.3) is exploited via the introduction of a set of auxiliary variables denoted $\{Z_i\} = \{Z_1, \ldots, Z_n\}$ where each $Z_i \in \mathbb{R}^{p \times p}$ corresponds to each Θ_i. This allows us to minimize the loss with respect to each set of variables, $\{\Theta_i\}$ and $\{Z_i\}$ in iterative fashion while enforcing an equality constraint on each Θ_i and Z_i respectively. Consequently, equation (3.3) can be reformulated as the following constrained minimization problem:

$$\underset{\{\Theta_i\},\{Z_i\}}{\text{minimize}} \quad \sum_{i=1}^{n} (-\log\det\Theta_i + \text{ trace }(S_i\Theta_i)) + \lambda_1 \sum_{i=1}^{n} ||Z_i||_1 + \lambda_2 \sum_{i=2}^{n} ||Z_i - Z_{i-1}||_1 \tag{3.7}$$

$$\text{subject to} \quad \Theta_i = Z_i \quad i = 1, \ldots, n \tag{3.8}$$

where we have replaced Θ_i with Z_i in both of the penalty terms. As a result, $\{\Theta_i\}$ terms are involved only in the likelihood component of equation (3.7) while $\{Z_i\}$ terms are involved in the penalty components. This decoupling allows for the individual structure associated with the functions f and g_{λ_1,λ_2} to be exploited.

The use of an ADMM algorithm requires the formulation of the augmented Lagrangian corresponding to equations (3.7) and (3.8), defined as:

$$\begin{aligned}\mathcal{L}_\gamma(\{\Theta_i\},\{Z_i\},\{U_i\}) = &-\sum_{i=1}^{n} (\log\det\Theta_i - \text{ trace }(S_i\Theta_i)) + \lambda_1 \sum_{i=1}^{n} ||Z_i||_1 \\ &+ \lambda_2 \sum_{i=2}^{n} ||Z_i - Z_{i-1}||_1 + \gamma/2 \sum_{i=1}^{n} \left(||\Theta_i - Z_i + U_i||_2^2 - ||U_i||_2^2\right),\end{aligned} \tag{3.9}$$

where $\{U_i\} = \{U_1, \ldots, U_n\}$ are scaled Lagrange multipliers such that $U_i \in \mathbb{R}^{p \times p}$. Equation (3.9) corresponds to the Lagrangian for equations (3.7) and (3.8) together with an additional quadratic penalty term. The latter is multiplied by a constant stepsize parameter γ which can typically be set to one. The introduction of this term is desirable as it often facilitates the minimization of the Lagrangian; specifically in our case it will make our problem substantially easier as we outline below.

We write $\{\Theta_i^j\} = \{\Theta_1^j, \ldots, \Theta_n^j\}$ where Θ_i^j denotes the estimate of Θ_i in the jth iteration. The same notation is used for $\{Z_i\}$ and $\{U_i\}$. The algorithm is initialized with $\Theta_i^0 = I_p$, $Z_i^0 = U_i^0 = \mathbf{0} \in \mathbb{R}^{p \times p}$ for $i = 1, \ldots, n$. At the jth iteration of the proposed

algorithm three steps are performed as outlined below.

Step 1: Update $\{\Theta_i^j\}$

At the jth iteration, each Θ_i is updated independently by minimizing equation (3.9). At this step we treat all $\{Z_i^j\}$, $\{U_i^j\}$ and Θ_k^j, for $k \neq i$ as constants. As a result, minimizing equation (3.9) with respect to Θ_i corresponds to setting:

$$\Theta_i^j = \underset{\Theta_i}{\text{argmin}} \left\{ -\text{log det } \Theta_i + \text{trace}(S_i\Theta_i) + \gamma/_2||\Theta_i - Z_i^{j-1} + U_i^{j-1}||_2^2 \right\}. \tag{3.10}$$

From equation (3.10) we can further understand the process occurring at this step. If γ is set to zero only the negative log-likelihood terms will be left in equation (3.10) resulting in $\Theta_i^j = S_i^{-1}$. However, this will not enforce either sparsity or temporal homogeneity and requires the assumption that S_i is invertible. Setting γ to be a positive constant implies that Θ_i will be a compromise between minimizing the negative log-likelihood and remaining in the proximity of Z_i^{j-1}. The extent to which the latter is enforced will be determined by both γ and the Lagrange multiplier, U_i^{j-1}. As we will see in step 2, it is the $\{Z_i\}$ terms which encode the sparsity and temporal homogeneity constraints. Differentiating the right hand side of equation (3.10) with respect to Θ_i and setting the derivative equal to zero yields:

$$\Theta_i^{-1} - \gamma\Theta_i = S_i - \gamma\left(Z_i^{j-1} - U_i^{j-1}\right) \tag{3.11}$$

which is a matrix quadratic in Θ_i (after multiplying through by Θ_i). Following from Proposition 2.3.4, we note that the quadratic defined in equation (3.11) has a closed form solution. Formally, we have that both Θ_i and $S_i - \gamma\left(Z_i^{j-1} - U_i^{j-1}\right)$ share the same eigenvectors. This allows us to solve equation (3.10) using an eigendecomposition as outlined below. Letting θ_r and s_r denote the rth eigenvalues of Θ_i and $S_i - \gamma\left(Z_i^{j-1} - U_i^{j-1}\right)$ respectively we have that:

$$\theta_r^{-1} - \gamma\theta_r = s_r. \tag{3.12}$$

Solving the quadratic in equation (3.12) yields

$$\theta_r = \frac{1}{2\gamma}\left(-s_r + \sqrt{s_r^2 + 4\gamma}\right), \tag{3.13}$$

for $r = 1, \ldots, p$. We note that the quadratic equation specified by equation (3.12) also contains a negative solution, however, this is ignored as each θ_i corresponds to an eigenvalue. By only considering the positive solution to the aforementioned quadratic equation we are able to ensure all eigenvalues are positive, thereby guaranteeing that the associated matrix will be positive semi-definite [27]. Due to the nature of equation (3.13) it follows that all eigenvalues, θ_i will be greater than zero. Thus Step 1 involves an eigendecomposition and update

$$\Theta_i = V_i \tilde{D}_i V_i^T \tag{3.14}$$

for each $i = 1, \ldots, n$. Here V_i is a matrix containing the eigenvectors of $S_i - \gamma\left(Z_i^{j-1} - U_i^{j-1}\right)$ and $\tilde{D}_i$ is a diagonal matrix containing entries $\theta_1, \ldots, \theta_p$. As discussed, all of the entries in $\tilde{D}_i$ will be strictly positive, ensuring that each Θ_i will be positive definite. Moreover, we also note from equation (3.14) that each Θ_i will also be symmetric.

Step 2: Update $\{Z_i^j\}$

As in step 1, all variables $\{\Theta_i^j\}$ and $\{U_i^j\}$ are treated as constants when updating $\{Z_i\}$. Due to the presence of the Fused lasso penalty in equation (3.9) we cannot update each Z_i^j separately as was the case with each Θ_i^j in step 1. Instead, at the jth iteration the $\{Z_i^j\}$ variables are updated by solving:

$$\{Z_i^j\} = \underset{\{Z_i\}}{\text{argmin}} \quad \left\{ \gamma/2 \sum_{i=1}^{T} ||\Theta_i^j - Z_i + U_i^{j-1}||_2^2 + \lambda_1 \sum_{i=1}^{T} ||Z_i||_1 + \lambda_2 \sum_{i=2}^{T} ||Z_i - Z_{i-1}||_1 \right\}, \tag{3.15}$$

where we note that only element-wise operations are applied. As a result it is possible to break down equation (3.15) into element-wise optimizations of the following form:

$$\underset{\{Z_i\}_{k,l}}{\text{argmin}} \quad \left\{ \gamma/2 \sum_{i=1}^{T} ||(\Theta_i^j - Z_i + U_i^{j-1})_{k,l}||_2^2 + \lambda_1 \sum_{i=1}^{T} ||(Z_i)_{k,l}||_1 + \lambda_2 \sum_{i=2}^{T} ||(Z_i - Z_{i-1})_{k,l}||_1 \right\} \tag{3.16}$$

where we write $(M)_{k,l}$ to denote the (k, l) entry for any square matrix M. Moreover, we write $\{Z_i\}_{k,l}$ to denote the (k, l) entires for all matrices in $\{Z_i\}$. This corresponds to a

Fused lasso signal approximator (FLSA) problem [83]. Moreover, due to the symmetric nature of matrices $\{\Theta_i\}$, $\{Z_i\}$ and $\{U_i\}$ we require $\frac{p(p+1)}{2}$ optimizations of the form shown in equation (3.16). Thus by introducing auxiliary variables $\{Z_i\}$ and formulating the augmented Lagrangian we are able to enforce both the sparsity and temporal homogeneity penalties by solving a series of one-dimensional Fused lasso optimizations.

Step 3: Update $\{U_i^j\}$

Step 3 corresponds to an update of Lagrange multipliers $\{U_i^j\}$ as follows:

$$U_i^j = U_i^{j-1} + \Theta_i^j - Z_i^j \text{ for } i = 1, \ldots, T \tag{3.17}$$

Convergence Criteria

The proposed algorithm is an iterative procedure consisting of Steps 1-3 described above until convergence is reached. In order to guarantee convergence we require both primal and dual feasibility: primal feasibility refers to satisfying the constraint $\Theta_i = Z_i$ while dual feasibility refers to minimization of the Augmented Lagrangian. For dual feasibility to be satisfied we require both that $\nabla_\Theta \mathcal{L}(\Theta, Z^j, U^j) = 0$ and $\nabla_Z \mathcal{L}(\Theta^{j+1}, Z, U^j) = 0$. We can check for primal feasibility by considering $||\Theta_i^j - Z_i^j||_2^2$ at each iteration. In order to ensure dual feasibility we employ the following proposition from [26].

Proposition 3.1.1 *The update in Step 3 guarantees dual feasibility in the $\{Z_i\}$ variables and dual feasibility in the $\{\Theta_i\}$ variables can be checked by considering $||Z^{j+1} - Z^j||_2^2$.*

Step 3 thereby ensures that $\{Z_i\}$ are always dual feasible [26] and it suffices to consider $||Z^j - Z^{j-1}||_2^2$ to verify dual feasibility in $\{\Theta_i\}$ variables. Thus the SINGLE algorithm is said to converge when $||\Theta_i^j - Z_i^j||_2^2 < \epsilon_1$ and $||Z_i^j - Z_i^{j-1}||_2^2 < \epsilon_2$ for $i = 1, \ldots, T$ where ϵ_1 and ϵ_2 are user specified convergence thresholds. The complete procedure is given in Algorithm 1.

Algorithm 1: Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm

Input: Multivariate fMRI time series $X_1, ..., X_n$, Gaussian kernel width h and regularization parameters λ_1, λ_2 and convergence tolerance ϵ_1, ϵ_2

Set $\Theta_i^0 = I_p$, $Z_i^0 = U_i^0 = \mathbf{0}$ for $i \in \{1, ..., n\}$ and $j = 1$
Compute weighted sample covariance matrices
for *i in* $\{1, \ldots, n\}$ **do**
    $\mu_i = \frac{\sum_{j=1}^{n} K_h(i,j) \cdot X_j}{\sum_{j=1}^{n} K_h(i,j)}$
for *i in* $\{1, \ldots, n\}$ **do**
    $S_i = \frac{\sum_{j=1}^{n} K_h(i,j) \cdot (X_j - \mu_j)^T (X_j - \mu_j)}{\sum_{j=1}^{n} K_h(i,j)}$
Estimate sparse precision matrices
while *Convergence == False* **do**
    ## $\{\Theta\}$ Update
    for *i in* $\{1, \ldots, n\}$ **do**
        $V, D = \text{eigen}\left(S_i - \gamma\left(Z_i^{j-1} - U_i^{j-1}\right)\right)$
        $\tilde{D} = \text{diag}\left(\frac{1}{2\gamma}\left(-D + \sqrt{D^2 + 4\gamma}\right)\right)$
        $\Theta_i^j = V \tilde{D} V'$
    ## $\{Z\}$ Update
    for *l in* $\{1, \ldots, p\}$ **do**
        for *k in* $\{l, \ldots, p\}$ **do**
            $x = \text{concat}\left(\left(\Theta_1^j - U_1^{j-1}\right)_{k,l}, \ldots, \left(\Theta_T^j - U_T^{j-1}\right)_{k,l}\right)$
            $\left(Z_1^j, \ldots, Z_T^j\right)_{k,l} = \text{FLSA}(x, \lambda_1, \lambda_2)$
    ## $\{U\}$ Update
    for *i in* $\{1, \ldots, n\}$ **do**
        $U_i^j = U_i^{j-1} + \Theta_i^j - Z_i^j$
    if $||\Theta_i^j - Z_i^j||_2^2 < \epsilon_1$ *and* $||Z_i^j - Z_i^{j-1}||_2^2 < \epsilon_2, \ \forall i$ **then**
        Convergence=True
    else
        j = j + 1
return $\{\Theta\}$

Computational complexity

As discussed previously the optimization of the SINGLE objective function involves the iteration of three steps. In step 1 we perform n eigen-decompositions, each of complexity $\mathcal{O}(p^3)$ where p is the number of nodes (i.e., the dimensionality of the data). Thus step 1 has

a computational complexity of $\mathcal{O}(np^3)$. We note that step 2 requires $\frac{p(p+1)}{2}$ iterations of the Fused lasso* where each iteration is $\mathcal{O}\left(n\log(n)\right)$ [83]. Thus the computational complexity of step 2 is $\mathcal{O}\left(p^2 n\log(n)\right)$. Finally step 3 only involves matrix addition implying that the final computational complexity of the SINGLE algorithm is $\mathcal{O}\left(p^2 n\log(n) + np^3\right)$. This is dominated by the number of nodes, p, not the number of observations. As a result the limiting factor is likely to be the number of nodes in a study.

3.1.3 Tuning parameters

The SINGLE algorithm requires the input of three parameters which can be tuned using the available data: λ_1, λ_2 and h. Each of these parameters has a direct interpretation. Parameter h is the width of the Gaussian kernel. Following from our discussions in Section 3.1.1, similar considerations should be made when tuning h as when tuning the width of a sliding window. Parameters λ_1 and λ_2 affect the sparsity and temporal homogeneity respectively. In particular, increasing λ_1 will result in network estimates with a higher degree of sparsity whereas increasing the value of λ_2 will encourage the fusion of temporally adjacent estimates. We discuss each of these three parameters in turn.

The choice of parameter h describes certain assumptions relating to the nature of the available data which are often not formally discussed. The use of a kernel (be it in the form of a sliding window or otherwise) also reflects an assumption of local, as opposed to global, stationarity. This assumption is that it is possible to obtain time dependent parameter estimates that accurately reflect the correlation structure within a neighbourhood of any observation but possibly not over an arbitrarily long time horizon. The choice of h can therefore be seen as an assumption relating to the extent of non-stationarity of the available data (for an attempted definition of the degree of non-stationarity see [78]).

On the one hand, the choice of a large value of h is indicative of an assumption that the data is close to stationary. If this is the case, a large choice of h allows for the accurate estimation of sample covariance matrices by incorporating information across a wide range of observations. However, if this assumption is incorrect, the choice of a large h can result in overly smoothed estimates where short term variation is overlooked. On the other hand,

* $\frac{p(p-1)}{2}$ edges and p more along the diagonal

the choice of a small h implies an assumption of a higher degree of non-stationarity. Here the choice of a small h can allow for the accurate estimation of sample covariance matrices by correctly discarding irrelevant information. However reducing the value of h will result in an increase in the variance of the estimators as it implies that a smaller sample size is used to estimate parameters. This effect is more dramatic for large values of p as a greater number of parameters must be estimated. Overall, the best performing value of h in any given setting will depend on the difficulty of the estimation task, in particular the dimensionality of p, as well as the rate of change of the underlying networks. The latter is not known apriori in many fMRI applications.

To avoid making specific assumptions about the nature of the temporal variability we rely on an entirely data-driven technique when choosing h that best describes the observations. The approach taken here is to use cross-validation [156]. As before, goodness-of-fit is employed to quantify how well estimated sample covariance matrices describe the observed time series. We define the leave-one-out (LOO) log-likelihood for the ith observation and some fixed choice of h as follows:

$$\mathcal{L}_{-i}(h) = -\frac{1}{2}\log\det\left(S_{-i}^{(h)}\right) - \frac{1}{2}\left(X_i - \mu_{-i}^{(h)}\right)^T\left(S_{-i}^{(h)}\right)^{-1}\left(X_i - \mu_{-i}^{(h)}\right), \tag{3.18}$$

where both $\mu_{-i}^{(h)}$ and $S_{-i}^{(h)}$ are estimated with the ith observation removed for a given h. Thus $\mathcal{L}_{-i}(h)$ allows us to estimate the goodness-of-fit at X_i for any fixed h. We subsequently choose h in order to maximize the following score function:

$$CV(h) = \sum_{i=1}^{n}\mathcal{L}_{-i}(h). \tag{3.19}$$

Parameters λ_1 and λ_2 determine the sparsity and temporal homogeneity of the estimated networks respectively. Therefore λ_1 and λ_2 directly affect the degrees of freedom of the estimated networks. In this case we can employ a more sophisticated parameter tuning technique based on the Akaike Information Criterion (AIC). The use of AIC allows us to estimate the in-sample prediction error for each choice of parameters λ_1 and λ_2, allowing for a clear comparison across different values of each parameter [76]. For any pair λ_1, λ_2

we define the AIC as:

$$AIC(\lambda_1, \lambda_2) = 2 \sum_{i=1}^{T} \left(-\text{log det } (\Theta_i) + \text{ trace } (S_i \Theta_i)\right) + 2K \tag{3.20}$$

where K is the estimated degrees of freedom. For a given range of λ_1 and λ_2 values an extensive grid-search is performed with the resulting choices of λ_1 and λ_2 being the pair that minimises AIC.

Following [167] we define K to be the number of non-zero coefficient blocks in $\{(\Theta_i)_{r,s}\}$ for $1 \leq r \neq s \leq p$. That is, we count a sequence of one or more consecutive non-zero and equal estimates of partial correlations as one degree of freedom. This can be formally written as:

$$K = \sum_{r,s} \sum_{i=2}^{n} \mathbb{1} \left((\Theta_i)_{r,s} \neq (\Theta_{i-1})_{r,s}\right). \tag{3.21}$$

Equation (3.21) therefore corresponds to counting the number of consecutive changes in estimated edge structure.

3.1.4 Related work

There are currently limited methodologies available for estimating dynamic functional connectivity networks. A novel approach has recently been proposed in the form of the DCR algorithm [38]. The DCR is able to estimate functional connectivity networks by first partitioning time series into piece-wise stationary segments. This allows the DCR to exploit the vast literature relating to stationary network estimation. Formally, the DCR algorithm detects statistically significant change-points by applying a block bootstrap permutation test. The use of a block bootstrap allows the DCR algorithm to account for autocorrelation present in fMRI data. The DCR will be employed extensively throughout the remainder of this chapter as a benchmark for the SINGLE algorithm. As a result, we provide a brief overview of the DCR algorithm in Appendix B.1.

Other widely employed approaches involve the use of a sliding windows [90]. This involves recursively estimating covariance matrices by re-weighting observations according to a sliding window or kernel. Subsequently, analysis can be performed directly on the

sample covariance, S_i, to infer the network structure at the ith observation. This approach is studied in detail by [187]. However, sliding window approaches face the potential issue of variability between temporally adjacent networks. This arises as a direct consequence of the fact that each network is estimated independently without any mechanism present to encourage temporal homogeneity. This additional variability can jeopardize the accuracy of the estimation procedure and can result in networks which do not reflect the true network structure over time. The SINGLE algorithm addresses precisely this problem by introducing an additional Fused lasso penalty. In this way, changes in the connectivity structure are only reported when strongly validated by the data. The beneficial effects of the additional Fused lasso penalty are studied extensively in the simulation study provided in Section 3.2.

Finally, the SINGLE algorithm is formally related to the Joint Graphical lasso (JGL) [42]. The JGL was designed with the motivation of improving network inference by leveraging information across related observations and data sets. However, while the JGL focuses on stationary network estimation the SINGLE algorithm is designed to estimate dynamic networks. This manifests itself in two main differences to the overall objective functions of each of the algorithms. Firstly, the SINGLE algorithm only employs the Fused lasso penalty as the Group lasso penalty proposed in [42] cannot be used in the context of temporal homogeneity. This is due to the fact that the Group lasso penalty encourages all coefficients to either be zero or non-zero in unison and therefore ignores temporal behaviour. Secondly, while both algorithms contain a Fused lasso penalty the nature of these penalties are vastly different. In the case of the JGL there is no natural ordering to observations and therefore *fusions* are present between all networks (i.e., the penalty is of the form $\sum_{i \neq j} ||\Theta_i - \Theta_j||_1$). This is not the case in the SINGLE algorithm where there is a chronological ordering. This results in a penalty of the form $\sum_{i=2}^{T} ||\Theta_i - \Theta_{i-1}||_1$. From an algorithmic perspective, this greatly reduces the computational burden as each estimated network only depends on the previous network [83].

3.2 Simulation study

In this section we evaluate the performance of the SINGLE algorithm through a series of simulation studies. In each simulation we produce simulated time series data giving

rise to a number of connectivity patterns which reflect those reported in real fMRI data. The objective is then to measure whether the proposed algorithm is able to recover the underlying patterns. That is, we are interested primarily in the correct estimation of the presence or absence of edges.

3.2.1 Simulation settings

There are two main properties of fMRI data which we wish to recreate in this simulation study. The first is the high autocorrelation which is typically present in fMRI data [140]. The second and main property we wish to recreate is the structure of the connectivity networks themselves. It is widely reported that brain networks have a small-world topology as well as highly connected hub nodes [30] and we therefore look to enforce these properties in our simulations.

Vector Autoregressive (VAR) processes are well suited to the task of producing autocorrelated multivariate time series as they are capable of encoding autocorrelations within components as well as cross-correlations across components [38]. Moreover, when simulating connectivity structures we study the performance of the proposed algorithm using three types of random graphs; Erdős-Rényi random graphs [56], scale-free random graphs obtained by using the preferential attachment model [11] and small-world random graphs obtained using the Watts-Strogatz model [177]. Erdős-Rényi random graphs are included as they correspond to the simplest and most widely studied type of random network while the use of scale-free and small-world networks is motivated by the fact that they are each known to each resemble different aspects of fMRI networks. A detailed description of each of the three aforementioned algorithms is provided in Appendix C.1.

When simulating Erdős-Rényi random networks we maintain the edge strength of the connectivity between nodes fixed at 0.6. We note that the edge strength refers to the correlation coefficient between simulated random variables. However, when simulating scale-free and small-world networks we randomly sample the edge strengths uniformly from $[-1/2, -1/4] \cup [1/4, 1/2]$. The motivation behind randomly sampling the edge strengths was due to the fact that such an approach would add additional variability in the edge strength together. Moreover, the expected magnitude of edges was also reduced further increasing

the difficulty of the estimation task.

The first three simulations considered are aimed at studying the performance of the SINGLE algorithm in three different scenarios. We begin by considering the overall performance of the SINGLE algorithm by generating connectivity structures according to Erdős-Rényi, scale-free and small-world networks in simulations 1a, 1b and 1c respectively. In many task-based experiments it is the case that the task is repeated several times, thus we expect there to be cyclic behavior within the true functional connectivity structure (i.e., connectivity alternates between two structures) and we study this scenario in simulations 2a, 2b and 2c. In simulation 3 we study the performance of the algorithm as the ratio of observations, n, to nodes, p, decreases. This simulation is critical as it is often the case that there is a low ratio of observations to nodes, especially when considering subject specific fMRI data. We further note that only scale-free and small-world networks are considered in this simulation. This is motivated by the fact that Erdős-Rényi networks are too simplistic and scale-free and small-world methods provide more realistic and challenging synthetic networks.

Throughout each of these simulations we benchmark the performance of the SINGLE algorithm against both the DCR algorithm and two sliding window based algorithms. In the case of the latter, a sliding window is employed to obtain time-dependent estimates of the sample covariance matrices and the Graphical lasso is subsequently used to estimate a sparse connectivity structure. In order to ensure a fair comparison, the sliding window approach is employed using both a uniform kernel as well as a Gaussian kernel.

3.2.2 Performance measures

When evaluating the performance of the SINGLE algorithm we are primarily interested in the estimation of the functional connectivity graphs at every time point. In our setting this corresponds to correctly identifying the non-zero entries in estimated precision matrices, Θ_i, at each $i = 1, \ldots, n$. An edge is assumed to be present between the jth and kth nodes if $(\Theta_i)_{j,k} \neq 0$. At the ith observation we define the set of all reported edges as $D_i = \{(j,k) : (\Theta_i)_{j,k} \neq 0\}$. We define the corresponding set of true edges as $T_i = \{(j,k) : (K_i)_{j,k} \neq 0\}$ where we write K_i to denote the true precision matrix at the ith observation. Given D_i and

T_i we consider a number of performance measures at each observation i.

First we measure the precision, P_i. This measures the percentage of reported edges which are actually present (i.e., true edges). Formally, the precision is given by:

$$P_i = \frac{|D_i \cap T_i|}{|D_i|}.$$

Second we also calculate the recall, R_i, formally defined as:

$$R_i = \frac{|D_i \cap T_i|}{|T_i|}.$$

This measures the percentage of true edges which were reported by each algorithm. Ideally we would like to have both precision and recall as close to one as possible. Finally, the F_i *score*, defined as

$$F_i = 2\frac{P_i R_i}{P_i + R_i}, \tag{3.22}$$

summarizes both the precision and recall by taking their harmonic mean [169].

3.2.3 Results

The objective of the simulation study presented is to provide an overview of the performance of the SINGLE algorithm with respect to alternative methods. The data is simulated as follows: each data set consists of 3 segments each of length 100 (i.e., overall duration of 300). Thus each data set consists of 2 change-points at times $t = 100$ and 200 respectively resulting in a network structure that is piece-wise constant over time. The correlation structure within each segment is randomly generate in three distinct methods which capture different properties known to arise in functional connectivity networks.

Throughout these simulations, the parameters for the SINGLE algorithm were set as per Section 3.1.3. This involved specifying the value of h via maximizing the leave-one-out log-likelihood, defined in equation (3.19). Furthermore, we note that in the case of the SINGLE algorithm, a Gaussian kernel was employed throughout all simulations. Values of λ_1 and λ_2 were estimated by minimizing AIC. For the DCR algorithm, the block size for the block bootstrap permutation tests was set to be 15 and one thousand permutations

where used for each permutation test. We note that alternative values for the block size and number of permutation tests were explored but did not result in significant differences in performance of the DCR algorithm. In the case of the sliding window and Gaussian kernel algorithms the kernel width was estimated using leave-one-out log-likelihood and regularization parameter was estimated by minimizing AIC.

Simulation 1a - Erdős-Rényi random networks

In this simulation, networks are simulated according to the Erdős-Rényi random graph model [56]. This corresponds to the simplest possible manner of generating random networks and therefore does not reproduce properties of fMRI data. As a result, we also study the performance of the SINGLE algorithm when networks are simulated using a preferential attachment model [11] and the Watts-Strogatz model [177] in Simulations 1b and 1c respectively.

The Erdős-Rény model treats all edges as independent random variables which are present with some fixed probability $\alpha \in [0, 1]$. This allows for random networks to be easily simulated. In the case of this simulation, random graphs were generated with 10 nodes and the probability of an edge between two nodes was fixed at $\alpha = 0.1$.

The top-left panel of Figure [3.1] shows the average F_t scores for each of the four algorithms over 500 simulations. We can see that the SINGLE algorithm performs competitively relative to the other algorithms. Specifically we note that the performance of the SINGLE algorithm mimics that of the Gaussian kernel algorithm. We also note that all four algorithms experience a dramatic drop in performance in the vicinity of change-points. This effect is most pronounced for the sliding window algorithm.

Simulation 1b - Scale-free networks

It has been reported that brain networks follow a scale-free distribution. A hallmark of scale-free networks is the degree centrality across all nodes follows a power law. From a biological perspective this implies that there are a small but finite number of hub regions which have access to most other regions [54]. While Erdős-Rényi random graphs offer a simple and powerful model from which to simulate random networks they fail to generate

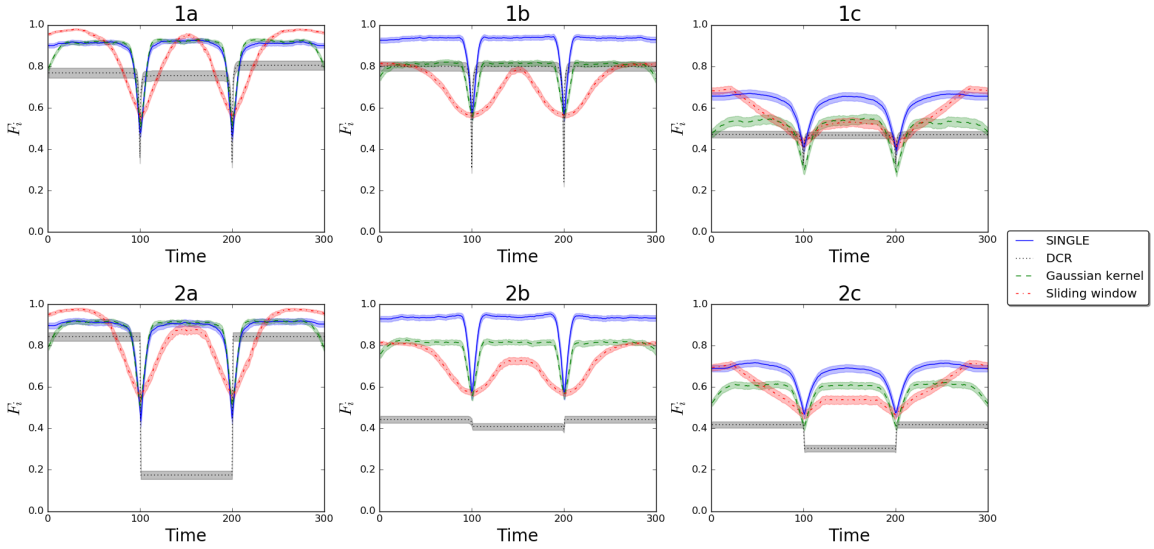


Figure 3.1: Mean F scores are shown across all four algorithms for simulations 1 and 2. Shaded regions correspond to 95% confidence intervals. Underlying network structure was simulated using three distinct algorithms. In the case of simulation 2, the covariance structure was cyclic in nature.

networks where the degree distribution follows a power law. In this simulation we analyze the performance of the SINGLE algorithm by simulating random networks according to the preferential attachment model proposed by [11]. Further details are provided in Appendix C.1. Here the power of preferential attachment was set to one. Additionally, edge strength was also simulated according to a uniform distribution on $[-1/2, -1/4] \cup [1/4, 1/2]$, introducing further variability in the estimated networks.

The top-middle panel of Figure [3.1] shows the average F_t scores for each of the four algorithms over 500 simulations. We note that the performance of the SINGLE and DCR algorithms is largely unaffected by the increased complexity of the simulation. This is not true in the case of the sliding window and Gaussian kernel algorithms, both of which see their performance drop. We hypothesize this drop in performance may be caused by the increased complexity of the network structure. Similar results confirming that networks with skewed degree distributions (e.g., power law distributions) are typically harder to estimate have also been described in [137].

Simulation Ic - Small-world networks

In addition to scale-free properties, brain networks have also been reported to display a small-world topology [13, 162, 164]. In this simulation, random networks demonstrating such properties were simulated by employing the Watts-Strogatz model [177]. Such a model is parameterized by $\beta \in [0, 1]$ which quantifies the probability of randomly rewiring an edge. A detailed description of this model is provided in C.1. Throughout this simulation we set $\beta = 3/4$ and edge strength was simulated according to a Uniform distribution on $[-1/2, -1/4] \cup [1/4, 1/2]$.

The top-right panel of Figure [3.1] shows the average F_t scores for each of the four algorithms over 500 simulations. There is a significant drop in the performance of all the algorithms relative to their performance in simulations 1a and 1b, however, the SINGLE algorithm continues to out-perform both the DCR and sliding window alternatives. An interesting related research question, which lies beyond the scope of this work, is to understand the effect of network structure on the accuracy of the estimated networks. In the case of this simulation, we believe that the drop in performance may be related to the increased complexity of small-world networks compared to alternative network models. In particular, due to the high local clustering present in small-world networks, the path length between any two nodes will remain relatively short. As a result, we expect there to be a large number of correlated variables that are not directly connected. It has been reported that the lasso (and therefore by extension the Graphical lasso) cannot guarantee consistent variable selection in the presence of highly correlated predictors [189, 190]. Since all four algorithms are related to the Graphical lasso, this may be the cause of the overall drop in performance.

Simulation 2a - Alternating Erdős-Rényi networks

In task related experiments subjects are typically asked to alternate between performing a cognitive task and resting. As a result, we expect the functional connectivity structure to alternate between two states: a task related state and the resting state. In order to recreate this scenario, network structures are simulated in a recurring fashion such that the first and third correlation structures are identical.

The results are shown in the bottom-left panel of Figure [3.1]. The performance of the SINGLE, sliding window and Gaussian kernel algorithms is largely unaffected. This is to be expected as such methods consider only an adequately re-weighted subset of nearby observations when computing the sufficient statistics. However, the DCR algorithm suffers a clear drop in performance relative to simulation 1a. The drop in performance of the DCR algorithm is partly due to the presence of the recurring correlation structure. More specifically, the problem is related to the use of block bootstrapping permutation test to determine the significance of change-points in the DCR. This test assumes that local data points are identically distributed but expects data points that are far away not to be. Typically this assumption holds, however, in the context of an alternating correlation structure, points which are far away may also follow the same underlying distribution. As a result the power of the permutation test is heavily reduced and many change-points are missed.

Simulation 2b - Alternating Scale-free networks

In this simulation multivariate time series are generated where the underlying correlation structure is alternating and follows a scale-free distribution. The results are summarized in the bottom-middle panel of Figure [3.1]. As in simulation 1b, there is no noticeable difference in the performance of the SINGLE algorithm. However, there is a drop in the performance of the sliding window, Gaussian kernel and DCR algorithms. This is particularly evident in the case of the DCR algorithm. As mentioned previously the drop in performance of the sliding window and Gaussian kernel algorithms is due to the increased complexity of the network structure as well as the fall in the signal to noise ratio. In the case of the DCR the drop in performance can be partly explained by the fact the assumptions behind the use of the block bootstrap no longer hold (see simulation 2a for a discussion) and the increased complexity of the network structure. These two factors combine to greatly affect the performance of the DCR algorithm.

Simulation 2c - Alternating Small-world networks

In this simulation the performance of the SINGLE algorithm is assessed in a scenario that is representative of experimental data typically obtained from fMRI studies. As such, the

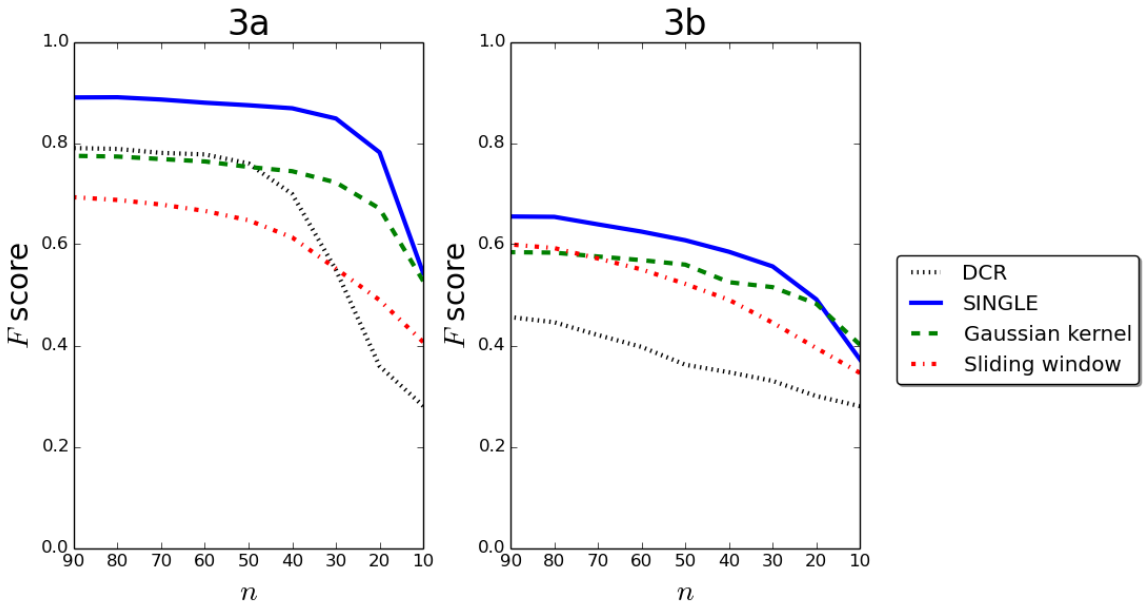


Figure 3.2: Mean F scores for simulation 3. Underlying networks followed scale-free (left) and small-world (right) distributions. The performance of each of the four algorithms is studied as the number of observations, n, decreases for a fixed number of nodes, p.

network structure in this simulation was composed of alternating networks where each network structure is simulated according to a small-world network as in Simulation 1c. This simulation gives us a clear insight into the performance of the SINGLE algorithm in a scenario that is very similar to that proposed in the experimental data.

The results are summarized in the bottom-right panel of Figure [3.1]. As in Simulation 1c, there is drop in the performance of all four algorithms relative to their performance in simulations 2a and 2b. This is due to the increased complexity of the underlying networks structures, specifically the high levels of clustering we experience in small-world networks which are not seen in Erdős-Rényi or scale-free random networks.

Simulation 3a - Scale-free networks with decreasing n/p ratio

The objective of this simulation is to study the behavior of the proposed method as the ratio of observations, n, to the number of nodes, p, decreases. This is a particularly relevant problem in the case of fMRI data as it is often the case that the number of nodes in the study (typically the number of ROIs) will be much larger than the number of observations.

In this simulation we fix $p = 10$ and allow the value of n to decrease. As such, data is simulated with three segments each of length n where the connectivity structure within each segment is randomly simulated according to a scale-free network. As the value of n decreases we are able to quantify the performance of the SINGLE algorithm in the presence of rapid changes in network structure.

In the case of the SINGLE, sliding window and Gaussian kernel algorithms all parameters are estimated as discussed previously. In the case of the DCR algorithm, the value of block sizes for the block bootstrap test was reduced as a function of n.

Results for Simulation 3a are given in the left panel of Figure [3.2]. Error bars have been omitted in the interest of clarity however detailed results are available in Table 3.1. As expected, the performance of all four algorithms diminishes when n is small relative to p. However, from Figure [3.2] we note that the performance of both the SINGLE and DCR algorithms quickly improves as n increases.

n	μ	σ
10	0.28	0.09
20	0.36	0.15
30	0.55	0.21
40	0.70	0.15
50	0.76	0.08
60	0.78	0.07
70	0.78	0.06
80	0.79	0.03
90	0.79	0.02

(a) DCR

n	μ	σ
10	**0.54**	**0.13**
20	**0.78**	**0.08**
30	**0.85**	**0.06**
40	**0.87**	**0.05**
50	**0.87**	**0.05**
60	**0.88**	**0.05**
70	**0.89**	**0.04**
80	**0.89**	**0.04**
90	**0.89**	**0.04**

(b) SINGLE

n	μ	σ
10	**0.53**	**0.09**
20	0.67	0.07
30	0.72	0.05
40	0.74	0.05
50	0.75	0.04
60	0.76	0.04
70	0.77	0.03
80	0.77	0.03
90	0.77	0.03

(c) Gaussian Kernel

n	μ	σ
10	**0.41**	**0.09**
20	0.49	0.10
30	0.55	0.10
40	0.61	0.09
50	0.65	0.07
60	0.67	0.07
70	0.68	0.06
80	0.69	0.05
90	0.69	0.05

(d) Sliding window

Table 3.1: Detailed results from Simulation 3a. For each algorithm the mean F score, μ, is reported together with the sample standard deviation, σ. The algorithm with the best performance is highlighted in bold for each value of n.

Simulation 3b - Small-world networks with decreasing n/p ratio

As with Simulation 3a, the purpose of this simulation is to evaluate the performance of the proposed algorithm as the ratio of observations, n, relative to the dimensionality of the data, p, decreases. However, here the underlying network structure are simulated according

to small-world networks. This simulation therefore provides an insight into how accurately proposed algorithm is able to estimate networks in the presence of rapid changes.

Results for Simulation 3b are shown in the right panel of Figure [3.2] and detailed results are provided in Table [3.2]. As with the previous simulations we note that the performance of all four algorithms is affected by the presence of small-world networks (see simulation 1c for a discussion). Furthermore, as in simulation 3a, the performance of all four algorithms also deteriorates as the ratio n/p decreases. Moreover, as in Simulation 3a, the performance of the SINGLE algorithm improves as n/p increases.

3.3 Application

In this section the SINGLE algorithm is employed to estimate connectivity networks associated with fMRI data evoked during a simple cognitive task, the Choice Reaction Time (CRT) task. The CRT is a forced choice visuo-motor decision task that reliably activates visual, motor and many cognitive control regions. The task was blocked into alternating task and rest periods. As a result we expect the task onset to evoke an abrupt change in the correlation structure that is cyclical in nature.

This corresponds to a challenging dataset for several reasons. Firstly, it corresponds to the scenario where $n/p = 126/18$ is small. Secondly, a change in the covariance structure is expected roughly every 15 seconds, suggesting that an even smaller number of relevant observations are available through which to estimate networks. Finally, given the nature of the CRT task there is a recurring correlation structure with subjects alternating between two cognitive states: resting and performing the CRT task.

3.3.1 Choice Reaction task data

The data was collected from 24 healthy subjects performing a simple but attentionally demanding cognitive task. Subjects were presented with an initial fixation cross for 350ms. This was followed by a response cue in the form of an arrow in the direction of the required response and lasting 1400ms. The inter-stimulus interval was 1750ms. Finger-press responses were made with the index finger of each hand. Subjects were instructed to respond

n	μ	σ
10	**0.31**	**0.06**
20	0.32	0.07
30	0.33	0.08
40	0.35	0.10
50	0.36	0.11
60	0.40	0.12
70	0.42	0.11
80	0.45	0.10
90	0.46	0.10

(a) DCR

n	μ	σ
10	**0.37**	**0.08**
20	**0.49**	**0.07**
30	**0.56**	**0.07**
40	**0.59**	**0.07**
50	**0.61**	**0.07**
60	**0.62**	**0.07**
70	**0.64**	**0.07**
80	**0.65**	**0.06**
90	**0.66**	**0.06**

(b) SINGLE

n	μ	σ
10	**0.40**	**0.06**
20	**0.48**	**0.05**
30	**0.52**	**0.05**
40	**0.54**	**0.05**
50	**0.56**	**0.05**
60	**0.57**	**0.05**
70	0.58	0.05
80	0.58	0.05
90	0.58	0.05

(c) Gaussian Kernel

n	μ	σ
10	**0.35**	**0.07**
20	**0.39**	**0.08**
30	**0.44**	**0.08**
40	**0.49**	**0.09**
50	**0.52**	**0.08**
60	**0.55**	**0.07**
70	0.57	0.07
80	0.59	0.06
90	0.60	0.06

(d) Sliding window

Table 3.2: Detailed results from Simulation 3b. For each algorithm the mean F score, μ, is reported together with the sample standard deviation, σ. The algorithm with the best performance is highlighted in bold for each value of n.

as quickly and as accurately as possible. To maximise design efficiency, stimulus presentation was blocked, with five repeated blocks of 14 response trials interlaced with five blocks of 14 rest trials, and four response trials at the start of the experiment. This resulted in a total of 74 response trials per subject.

Image pre-processing involved realignment of EPI images to remove the effects of motion between scans, spatial smoothing using a 6mm full-width half-maximum Gaussian kernel, pre-whitening using FILM and temporal high-pass filtering using a cut-off frequency of $1/50$ Hz to correct for baseline drifts in the signal. FMRIB's Linear Image Registration Tool (FLIRT) [158] was used to register EPI functional data sets into standard MNI space using the participant's individual high-resolution anatomical images.

The nodes were eighteen cortical spherical regions based on [135]. Briefly, these nodes were defined based on peak regions from a spatial group independent components analysis of resting state fMRI. The regions were chosen for the nodes to encompass a wide range of cortical regions including regions within two well recognized functional connectivity networks, the fronto-parietal cognitive control network (FPCN) and default mode network (DMN) regions, as well as motor, visual and auditory cortical regions. For each subject and node the mean time-course from within a 10mm diameter sphere centered on each of the 18 peaks was calculated. Six motion parameters, estimated during realignment, were filtered out of each time-course using linear regression. The resulting 18 time-courses were

subsequently used.

3.3.2 Results

The SINGLE algorithm was employed to estimate time-varying functional connectivity networks for each subject. This required the specification three parameters: the Gaussian kernel width, h, was fixed across all subjects and selected via cross-validation as described in Section 3.1.3. The remaining regularization parameters, λ_1 and λ_2 were selected on a subject-by-subject basis by minimizing AIC. The choice of kernel width was estimated to be $\hat{h} = 10$ which is roughly in line with the block length for the CRT task. As mentioned, the regularization parameters were estimated separately for each subject with mean values of $\bar{\lambda}_1 = 0.15$ and $\bar{\lambda}_2 = 0.05$. However, the interpretation of such regularization parameters is challenging [77].

In order to study the roles of the various ROIs during the CRT task we consider the changes in betweenness centrality of each node over time. The betweenness centrality of a node is the sum of how many shortest paths between all other nodes pass through it [135]. Nodes with high betweenness centralities are considered to be of important, hub nodes in the network [73]. As described previously the CRT task involves subjects alternating between performing a visual stimulus task (on task) and resting state (off task). Figure [3.3] shows the average estimated functional connectivity networks for a patient on and off task respectively. Here the size of each node is proportional to the sum of the betweenness centralities of the corresponding ROI and the edge thickness is proportional to the partial correlation between nodes.

We note that there are changes in the betweenness centralities of several nodes between tasks. In order to determine the significance of any changes betweenness centrality as a result of the changing cognitive state of the subjects we study the estimated graphs for each of the 24 subjects both on and off task. To determine the statistical significance of reported changes a Wilcoxon rank sum test was employed. The resulting p-values where adjusted according to the Bonferroni-Holm method in order to account for multiple tests. The results indicated that at the $\alpha = 5\%$ level there was a statistically significant increase in betweenness centrality for the Right Inferior Frontal Gyrus and Right Inferior Parietal

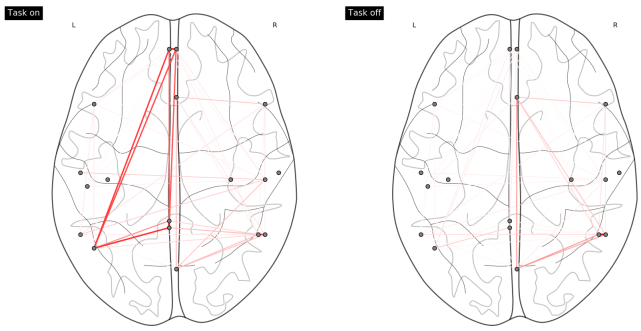


Figure 3.3: Mean estimated graphs on and off task for a given subject. Here node size is proportional to betweenness centrality and edge width is proportional to the magnitude of their partial correlations.

regions. This indicates that during this simple, cognitive task the Right Inferior Frontal Gyrus and the Right Inferior Parietal increase their connectivity across remaining nodes, potentially indicating an increase in their importance in the network.

These findings suggest that the Right Inferior Frontal Gyrus and Right Inferior Parietal play a key role in cognitive control and executive functions as demonstrated by their dynamically changing betweenness centrality throughout the task. This result agrees with the proposed functional roles for the Right Inferior Frontal Gyrus (and adjacent right anterior insula), which is assumed to play a fundamental role in attention and executive function during cognitively demanding tasks and may have an important role in regulating the balance between other brain regions [7, 74, 20]. The findings also agree with the proposed function of the Right Inferior Parietal lobe, which has been reported to play a role in high-level cognition [113] and sustaining attention [37, 89].

We may further study the behavior of the dynamic networks estimated by the SINGLE algorithm by considering the time courses of individual edges. Figure [3.4] provides a visualization of three edges which were found to be significantly correlated with task onset. Specifically, Figure [3.4] shows the average edge weights over time for all subjects. We note that in the case of all three edges show, there is an increase in connectivity during the CRT task and a corresponding drop during rest.

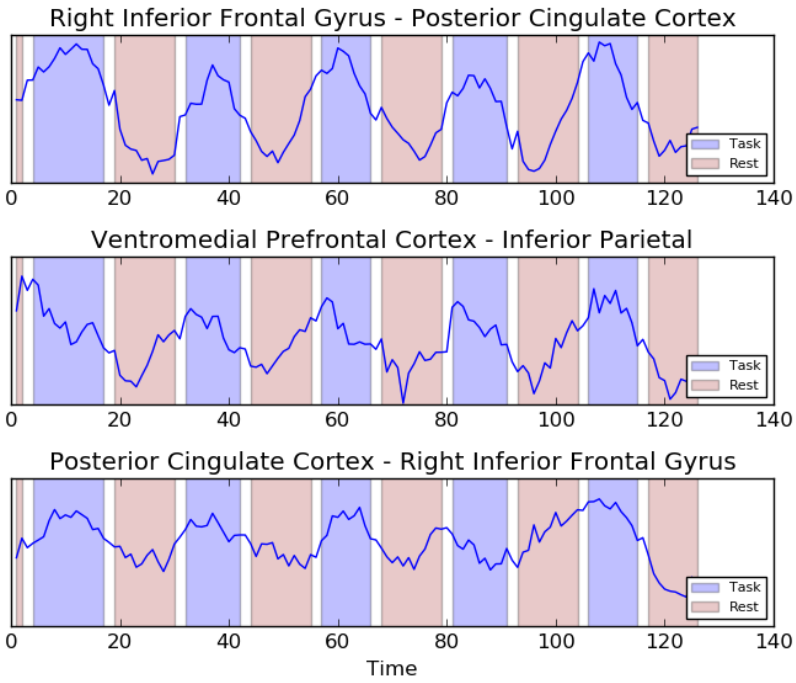


Figure 3.4: Time courses are visualized for three edges which were found to be highly correlated with task onset. Each time course corresponds to the average edge connectivity across all subjects. Background shading is indicative of the underlying task performed by subjects; blue indicates the CRT task whilst red indicates rest.

3.4 Conclusion

In this chapter we have studied the problem of learning time-varying GGMs. We have introduced the Smooth Incremental Graphical lasso Estimation (SINGLE) algorithm as a new methodology for estimating sparse dynamic functional connectivity networks from non-stationary fMRI data. The proposed algorithm provides two main advantages. First, it is able to accurately estimate functional connectivity networks at each observation. This allows for the quantification the dynamic behavior of brain networks at a high temporal granularity. The second advantage lies in the SINGLE algorithm's ability to quantify network variability over time. In SINGLE, networks are estimated simultaneously in a unified framework which encourages temporal homogeneity. This results in networks with sparse innovations in edge structure over time and implies that changes in connectivity structure

are only reported when substantiated by evidence in the data.

The SINGLE algorithm is closely related to sliding window based algorithms. We note that [187] have extensively studied the combined use of kernel methods and constrained optimization to estimate dynamic networks and provide a theoretical guarantee that accurate estimates of time varying network structure can be obtained in such a manner under mild assumptions [187]. The approach taken there is to estimate sample covariance matrices at each $i \in \{1, \ldots, T\}$ using kernel methods with the Graphical lasso being used subsequently to estimate the corresponding precision matrices. However, given T time points this approach corresponds directly to T independent iterations of the Graphical lasso. As a result, while smooth estimates of the sample covariance matrix are obtained via the use of kernels, there is no mechanism in place to enforce temporal homogeneity in the corresponding precision matrices. Consequently the estimated partial correlations may not accurately represent the functional connectivity over time. The SINGLE algorithm addresses precisely this problem by directly enforcing temporal homogeneity. This is achieved via the introduction an additional constraint inspired by the Fused lasso. As shown in our simulation study, this additional constraint results in higher accuracy of estimated networks in a vast array of scenarios.

The SINGLE algorithm requires the input of 3 parameters, λ_1, λ_2 and h, each of which has a natural interpretation for the user. Penalty parameters λ_1 and λ_2 enforce sparsity and temporal homogeneity respectively. They can be tuned by minimizing AIC over a given range of values. The choice of h can be interpreted as the window length and we provide an data-driven method for tuning parameter h using the leave-one-out log-likelihood. We note that the choice of h is a delicate matter as well as an active area of research in its own right. The choice of h can be seen as a trade-off between stability and temporal adaptivity. Setting h to be too large will result in network estimates that resemble the global mean and omit valuable short-term fluctuations in connectivity structure. Conversely, setting h to be too small will lead to networks that are dominated by noise. Given this reasoning, it is often desirable to have a kernel width which is dependent on the location within the time series. This allows the kernel width to decrease in the proximity of a change-point (allowing for rapid temporal adaptivity) and increase when data is piece-wise stationary (in order to fully exploit all relevant data). The idea of adaptive values of h has been studied in literature

[78, 142], and is discussed in further detail in Chapter 4.

Our simulation results indicate that the SINGLE algorithm can accurately estimate the true underlying functional connectivity structure when provided with non-stationary multivariate time series data. We identify three relevant scenarios where the proposed method performs competitively. The first, demonstrated by Simulation 1, quantifies our claim that the SINGLE algorithm is able to accurately estimate dynamic functional connectivity networks. In task based experiments it is often the case that tasks are repetitively performed followed by a period of rest, resulting in the presence of a cyclic functional connectivity structure. This scenario is studied in Simulation 2 which serves as an indication that the SINGLE algorithm is not adversely affected in such cases. Furthermore, we have shown that the SINGLE algorithm is relatively robust when the ratio of observations to nodes falls, meaning that the SINGLE algorithm can be applied on a subject-by-subject basis. This is a great advantage as it avoids the issue of subject-to-subject variability and allows for the estimation of functional connectivity networks for each subject. This potentially allows for estimated dynamic connectivity to be used to differentiate between subjects. A summary of all the simulation results is provided in Table 3.3.

	SINGLE	DCR	Glasso methods
Temporal adaptivity	✓	✓	✓
Temporal homogeneity	✓	✓	X
Cyclic correlation structure	✓	X	✓
Parameters	h, λ_1, λ_2	Δ, λ_1	h, λ_1
Computational Complexity	$\mathcal{O}(np^3 + p^2 n \log(n))$	$\mathcal{O}((n+b)p^3)$	$\mathcal{O}(np^3)$

Table 3.3: Comparative summary of each algorithm. A derivation of the computational cost of the DCR algorithm is provided in Appendix B.1 where b refers to the number of bootstrap permutation tests performed at each iteration.

In conclusion, the SINGLE algorithm provides an alternative and novel method for estimating the underlying network structure associated with dynamic fMRI data. It is ideally suited to analyzing data where a change in the correlation structure is expected but little more is known. An exciting avenue of neuroscientific research corresponds to studying fMRI data in real-time. Here the objective is to study the data as a stream of observations. To date, the majority of real-time fMRI applications have not incorporated information

relating to functional connectivity networks. This is primarily due to the complexity of estimating such networks in realt-time. In Chapter 4 we present an extension of the proposed methodology through which to accurately estimate connectivity networks in the context of streaming data.

Moreover, an important theoretical consideration which has not been addressed relates to the choice of the regularization parameters, λ_1 and λ_2. In particular, throughout this chapter it has been assumed that the aforementioned parameters remain fixed. However, since the data is assumed to be non-stationary, such an assumption cannot easily be justified. This aspect this considered in further detail in Chapter 5, were a framework is proposed through which to learn a time-varying regularization parameter.

A further problem raised by the proposed algorithm is related to the interpretation and understanding of results. This is especially the case when networks are estimated across a large number of subjects. In the application presented here, graph metrics such as betweenness centrality have been employed. However, it follows that such metrics may not necessarily capture relevant changes in connectivity structure in a concise manner. In order to address this issue, we consider two distinct graph embedding methods in Chapter 6.

Chapter 4

Streaming covariance selection

In Chapter 3 we proposed novel methodology through which to estimate time-varying Gaussian graphical models (GGMs). This work was motivated by the desire to quantify the dynamic properties of functional connectivity networks, which are often modeled as GGMs. In this chapter we focus on extending the methodology proposed in Chapter 3 to the context of streaming data. Formally, a data stream is defined as a potentially unending sequence of ordered observations where each observation may be read or studied only once [17]. Streaming applications may arise in settings where observations are continually arriving or when the data itself is too large to store in memory for analysis. Examples include the study of financial data and cyber-security [18, 19].

The work presented in this chapter is motivated by the study of fMRI data in real-time, a rapidly expanding avenue of neuroscience research [178]. The dominant applications of real-time fMRI are centered around *neurofeedback* [43], where the objective is to train participants to modulate BOLD activity within a specified brain region, and *brain decoding* [99], which seeks to predict brain states "on the fly" based on BOLD measurements obtained in real-time. However, both of the aforementioned applications are typically based on the study of individual brain regions. In the context of neurofeedback, the use such of region of interest based approaches fails to take into consideration the notion of the brain as a functionally connected network [162]. Furthermore, in addition to stimulating a particular brain region it may also be of scientific interest to stimulate entire networks [148]. Conversely, in the case of brain decoding, it is reasonable to suggest that the predictive power

of such methods could be further boosted by providing them with additional information relating to functional connectivity.

Accurately estimating functional connectivity networks in real-time is therefore an important methodological challenge in the context of real-time fMRI. To date, only a limited number of studies, primarily focused on neurofeedback, have considered the estimation of connectivity networks in real-time [96, 148, 188]. A limitation of the above mentioned studies is that only a reduced number of regions have been employed.

The objectives of this chapter is therefore to extend the methods presented in Chapter 3 to facilitate the estimation of functional connectivity networks in real-time. This presents significant practical as well as methodological challenges. From a practical perspective, it is imperative to derive closed-form, recursive updates for sufficient statistics as well as computationally efficient optimization algorithms. An additional challenge is introduced by the potentially non-stationary nature of the data. It follows that rapid changes may occur in functional connectivity structure, indicating that proposed methods should be highly adaptive to change. In order to address these issues we leverage a host of previous research on streaming data analysis. In particular, we advocate the use of exponentially weighted moving average (EWMA) modes [106] as well as adaptive filtering techniques [78]. Such methods effectively discard past observations, allowing for *local* estimates of sufficient statistics. Through an extensive simulation study, we provide evidence indicating that such methods should be preferred to traditional sliding window approaches. The SINGLE algorithm, presented in the previous chapter, is then extended to the real-time scenario, allowing for functional connectivity to be estimated on the basis of conditional dependencies while encouraging the properties of sparsity and temporal homogeneity. These methods are discussed in Section 4.1.

The remainder of this chapter is organized as follows: In Section 4.1, we detail the extension of the SINGLE algorithm to handle streaming data. In order to achieve this, we introduce computationally efficient methods through which to update sufficient statistics as well as solve the associated optimization problem. An extensive simulation study is presented in Section 4.2. Finally, in Section 4.3 we present an application of the proposed methods to data from the Human Connectome Project (HCP).

4.1 Streaming GGMs

We assume we have access to a stream of multivariate fMRI measurements across p nodes where each node represents a spatially remote brain region. We write $X_t \in \mathbb{R}^p$ to denote the BOLD measurements at the tth observation; thus $X_{t,j}$ corresponds to the BOLD measurement at the jth node at time t. The objective of this work is to sequentially use all observations up to and including X_t in order to recursively estimate the underlying connectivity networks. As new observations, X_{t+1}, arrive they are employed to update the estimated networks accordingly. Throughout this chapter it is assumed each X_t follows a multivariate Gaussian distribution, $X_t \sim \mathcal{N}(\mu_t, \Sigma_t)$, where both the mean and covariance are assumed to vary over time.

Following from the previous chapter, our objective is to estimate functional connectivity networks based on conditional dependencies across nodes. As noted in Section 2.2, this corresponds to estimating the support of the precision matrix. The objective of this work is therefore to estimate an increasing sequence of connectivity networks, $\{\Theta_1, \ldots, \Theta_t, \ldots\}$, where each Θ_t captures the conditional dependence structure at the tth observation.

The task of estimating functional connectivity networks in real-time is divided into two independent steps. First, an update of the sample covariance, S_t, is obtained. To this end, we consider two related methods, EWMA models and adaptive filtering methods, which are formally outlined in Section 4.1.1. The second step corresponds to estimation of a sparse precision matrix. This is achieved by extending the SINGLE algorithm, detailed in Chapter 3. This step is discussed in Section 4.1.2.

4.1.1 Recursive covariance estimation

As noted in Section 2.2, the sample covariance is a sufficient statistic when estimating connectivity networks based on the precision matrix, Θ_t. In this section we focus on the challenge of obtaining adaptive estimates of the sample covariance in a recursive fashion. The recursive nature of the proposed methods is fundamental in order to adequately handle streaming datasets.

Within the neuroimaging community, arguably the dominant approach used to obtain

adaptive estimates of the sample covariance involves sliding windows [90]. This also holds true in the case of real-time fMRI analysis [148, 188]. Such methods are able to obtain adaptive estimates of S_t in real-time by only considering a fixed number of past observations, defined as the window. Using only the observations within the predefined window, an adaptive estimate of functional connectivity is obtained at time t as follows:

$$S_t = \frac{1}{h}\sum_{i=0}^{h-1}(X_{t-i} - \bar{x}_t)^T(X_{t-i} - \bar{x}_t). \tag{4.1}$$

Here $\bar{x}_t$ denotes the mean of all observations within the window and the parameter h denotes the length of the window.

A natural extension of sliding windows is the use of EWMA models, first introduced by [145]. Such methods re-weight observations according to their chronological proximity. The rate at which past information is discarded is determined by a fixed forgetting factor, $r \in (0, 1]$. In this manner, EWMA models are able to give greater weight to more recent observations. Furthermore, as detailed by [106], these methods enjoy superior statistical properties when compared to traditional sliding window methods. EWMA models thereby provide a conceptually simple and robust method with which to handle a wide range of non-stationary processes. For a given forgetting factor, r, the estimated mean at time t can be recursively computed as:

$$\bar{x}_t = \left(1 - \frac{1}{\omega_t}\right)\bar{x}_{t-1} + \frac{1}{\omega_t}X_t, \tag{4.2}$$

where ω_t is a normalizing constant defined as:

$$\omega_t = \sum_{i=1}^{t} r^{t-i} = r\omega_{t-1} + 1. \tag{4.3}$$

The sample covariance can be computed as follows:

$$\Pi_t = \left(1 - \frac{1}{\omega_t}\right)\Pi_{t-1} + \frac{1}{\omega_t}X_t^T X_t \tag{4.4}$$

$$S_t = \Pi_t - \bar{x}_t^T \bar{x}_t \tag{4.5}$$

We note that equations (4.4) and (4.5) are equivalent to iteratively estimating the sample covariance as:

$$S_t = \left(1 - \frac{1}{\omega_t}\right) S_{t-1} + \frac{1}{\omega_t}(X_t - \bar{x}_t)^T(X_t - \bar{x}_t). \tag{4.6}$$

While equation (4.6) is arguably a more intuitive formulation, the parameterization presented in equations (4.4) and (4.5) will serve to simplify future discussion as it is more amenable to streaming data and incremental updates.

From equations (4.2) and (4.4) it is clear that past observations gradually receive less importance whenever $r < 1$. This is in contrast to sliding windows, where all observations within the window receive equal weighting. It follows that the choice of parameter r dictates the rate at which past information is discarded. This parameter therefore directly relates to the adaptivity of the proposed method, as noted by studying the extreme case of $r = 1$. This implies that $\omega_t = t$ and consequently that $\bar{x}_t$ and S_t correspond to the sample mean and covariance in an offline setting (using all observations up to time t). As a result, equal importance is given to all past observations, resulting in reduced adaptivity to changes. As the value of r is reduced, greater importance is given to more recent observations. This leads to increasingly adaptive estimates of the sample covariance. However, decreasing the value of r also increases the susceptibility of the proposed methods to outliers and noise. The choice of r therefore constitutes a trade-off between adaptivity and stability.

Adaptive filtering methods

In the context of non-stationary data, it is important to note that the optimal choice of forgetting factor, r, may itself by time-varying. By this, we mean that in the proximity of a change-point it is clearly desirable to employ a small choice of r, thereby reducing the importance of past observations which are no longer relevant. Conversely, within a locally stationary region we wish to employ a large value for r as this will allow us to learn from a wide range of pertinent observations. This concept is visualized in Figure [4.1]. In the case of real-time fMRI, we inherently expect the statistical properties of a subject's data to vary depending on a wide range of factors (e.g., varying cognitive tasks). A fixed choice of forgetting factor may therefore be suboptimal.

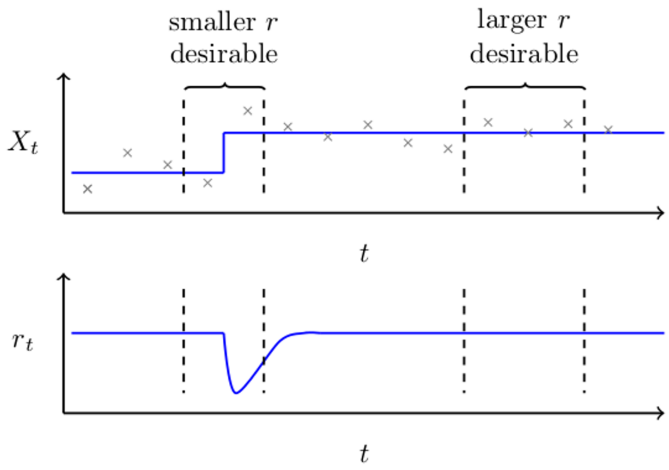


Figure 4.1: Measurements of a non-stationary univariate random variable, X_t, are shown in grey together with the true mean in blue. This figure serves to highlight how the optimal choice of a forgetting factor or window length may vary over time. It follows that in the neighborhood of the change-point we wish r to be small in order for it to adapt to change quickly. However, during locally stationary regimes, we wish for r to be large in order to be able to fully exploit all relevant data. Bottom: An illustration of how an ideal adaptive forgetting factor would behave; decreasing directly after a change occurs and quickly recovering thereafter.

To address this issue we propose the use of adaptive filtering methodology [78]. In such methods, the magnitude of the forgetting factor is iteratively adjusted as new observations arrive. This is achieved by approximating the derivative of the likelihood for new observations with respect to the current forgetting factor. In this manner, the forgetting factor may be updated in a stochastic gradient descent framework [22]. As such, the value of the forgetting factor will have a direct dependence on the time index, t. We write r_t to make this dependence explicit. The bottom panel of Figure [4.1] provides an example illustration of desirable behavior of an adaptive forgetting factor. In the context of a change-point, the adaptive forgetting factor should drop. This allows for the discarding of past observations which are no longer relevant while giving increased importance to new observations. Conversely, in the presence of piece-wise stationary data the value of r_t should increase. This allows for a wider range of relevant observations to be leveraged, resulting in more stable and accurate estimates.

Furthermore, the use of adaptive filtering methods provides an additional monitoring

mechanism. By considering the value of r_t at any given point in time we are able to obtain a rudimentary understanding as to the current degree of non-stationarity in the data [5]. This follows from the fact that the estimated forgetting factor quantifies the influence of recent observations on the sample mean and covariance. Therefore, large values of r_t are indicative of the presence of a stationary regime whereas small values of r_t provide evidence of non-stationarity.

Adaptive filtering methods are able to tune the forgetting factor, r_t, by iteratively quantifying the performance of the current parameter on new observations, X_{t+1}. Throughout this work we denote such a measure by $C(X_{t+1})$. Assuming that the derivative of $C(X_{t+1})$ with respect to the forgetting factor can be efficiently computed or approximated, we are able to update the parameter of interest in a stochastic gradient descent framework:

$$r_{t+1} = r_t - \epsilon \left. \frac{\partial C(X_{t+1})}{\partial r} \right|_{r=r_t} \tag{4.7}$$

Here ϵ is a small step-size parameter which may be viewed as a learning rate [17].

The approach we advocate here is to iteratively tune the adaptive forgetting factor to maximize the likelihood of unseen observations. In this case, $C(X_{t+1})$ is defined to be the likelihood of X_{t+1} given the current estimates of the mean and covariance matrix. Under the assumption of Gaussianity, we therefore have that:

$$C(X_{t+1}) = C(X_{t+1}; \bar{x}_t, S_t) = -\frac{1}{2}\text{log det } S_t - \frac{1}{2}(X_{t+1} - \bar{x}_t)^T S_t^{-1}(X_{t+1} - \bar{x}_t). \tag{4.8}$$

Unfortunately, due to the recursive definition of ω_t, provided in equation (4.3), the derivative of $C(X_{t+1})$ is not available analytically and must be approximated. In this work we employ the approximation detailed in [5].

From equations (4.2), (4.4) and (4.5) we can see the direct dependence of estimates $\bar{x}_t$ and S_t on a fixed forgetting factor r. This suggests that the likelihood is itself a function of the forgetting factor, allowing us to calculate its derivative with respect to r as follows:

$$\frac{\partial C(X_{t+1})}{\partial r} = \frac{1}{2}(X_{t+1} - \bar{x}_t)^T \left(2S_t^{-1}\bar{x}_t' - S_t^{-1}S_t'S_t^{-1}(X_{t+1} - \bar{x}_t)\right) - \frac{1}{2}\text{trace } (S_t^{-1}S_t'). \tag{4.9}$$

Full details relating to the derivation of equation (4.9) are provided in Appendix A. Given the derivative, the adaptive forgetting factor is updated as described in equation (4.7). Once

r_{t+1} has been calculated, the estimates of the mean and sample covariance are subsequently updated as described in equations (4.2-4.5) with the minor adjustment that the effective sample size is updated as:

$$\omega_{t+1} = r_t \omega_t + 1. \tag{4.10}$$

4.1.2 Recursive network estimation

In this section we detail how the SINGLE algorithm, introduced in Chapter 3, can be extended to recursively estimate sparse and temporally homogeneous precision matrices in the context of streaming observations.

We recall that given a sequence of sample covariance matrices $\{S_t\} = \{S_1, \ldots, S_T\}$, the SINGLE algorithm is able to estimate corresponding precision matrices by solving the following convex optimization problem:

$$\{\Theta_t\} = \underset{\{\Theta_t\}}{\text{argmin}} \left\{ \sum_{i=1}^{T} -\text{log det } \Theta_i + \text{trace } (S_i \Theta_i) + \lambda_1 \sum_{i=1}^{T} ||\Theta_i||_1 + \lambda_2 \sum_{i=2}^{T} ||\Theta_i - \Theta_{i-1}||_1 \right\}. \tag{4.11}$$

The first sum in equation (4.11) corresponds to a likelihood term while the remaining terms, parameterized by λ_1 and λ_2 respectively, enforce sparsity and temporal homogeneity constraints.

However, in the real-time setting, a new S_t is constantly obtained implying that the dimension of the solution to equation (4.11) grows over time as we look to estimate a network at each observation. It follows that iteratively re-solving equation (4.11) is both wasteful and computationally expensive; in particular, valuable computational resources will be spent estimating past networks which are no longer of interest. In order to address this issue the following objective function is proposed to estimate the functional connectivity network at time t:

$$f(\Theta) = -\text{log det } \Theta + \text{trace } (S_t \Theta) + \lambda_1 ||\Theta||_1 + \lambda_2 ||\Theta - \Theta_{t-1}||_1, \tag{4.12}$$

where Θ_{t-1} corresponds to the estimate of the precision matrix at time $t-1$ and is assumed to be fixed. The proposed real-time SINGLE (rt-SINGLE) algorithm therefore estimates Θ_t by minimizing equation (4.12). In doing so the proposed method must find a balance between goodness-of-fit and satisfying the regularization constraints. The former is captured

by the likelihood term:

$$l(\Theta, S_t) = -\text{log det } \Theta + \text{trace } (S_t\Theta), \tag{4.13}$$

and provides a measure of how precisely Θ describes the current estimate of the sample covariance, S_t. The latter two terms of the objective correspond to regularization penalty terms:

$$g_{\lambda_1,\lambda_2}(\Theta) = \lambda_1||\Theta||_1 + \lambda_2||\Theta - \Theta_{t-1}||_1 \tag{4.14}$$

The first of these, parameterized by λ_1, encourages sparsity while the second, parameterized by λ_2, determines the extent of temporal homogeneity. By penalizing changes in functional connectivity networks, the second penalty encourages sparse changes in edge structure over time. As a result, network changes are only reported when substantiated by evidence in the data.

Optimization algorithm

Equations (4.12)-(4.14) expose the separable nature of the objective function. As a result we follow the methods described in Chapter 3 and employ an ADMM algorithm introduced in Section 2.3.

As in the SINGLE algorithm, we proceed by introducing an auxiliary variable $Z \in \mathbb{R}^{p\times p}$. Here Z corresponds directly to Θ and we require $Z = \Theta$ for convergence. Minimizing equation (4.12) can subsequently be cast as the following constrained optimization problem, where only a single precision matrix is estimated:

$$\underset{\Theta,Z}{\text{minimize}} \quad \{-\text{log det } \Theta + \text{ trace } (S_t\Theta) + \lambda_1||Z||_1 + \lambda_2||Z - \Theta_{t-1}||_1\} \tag{4.15}$$

$$\text{subject to} \quad \Theta = Z. \tag{4.16}$$

We note that Θ is now only involved in the likelihood component while Z is involved exclusively in the penalty components. Thus, by introducing Z we have decoupled the initial objective function — allowing us to take advantage of the individual structure associated with each term. We formulate the augmented Lagrangian corresponding to equations (4.15)

and (4.16), which is defined as:

$$\begin{aligned}\mathcal{L}_\gamma\left(\Theta, Z, U\right) = &- \log\det\Theta + \text{ trace}\left(S_t\Theta\right) + \lambda_1||Z||_1 \\ &+ \lambda_2||Z - \Theta_{t-1}||_1 + 1/2\left(||\Theta - Z + U||_2^2 - ||U||_2^2\right),\end{aligned} \tag{4.17}$$

where $U \in \mathbb{R}^{p\times p}$ is the associated Lagrange multiplier.

The proposed estimation algorithm works by iteratively minimizing equation (4.17) with respect to Θ and Z while maintaining all other variables fixed. In this way, we are able to decouple the augmented Lagrangian and exploit the individual structure corresponding to each of these variables. Due to the iterative nature of the algorithm, in what follows we write Θ^i to denote the estimate of Θ at the ith iteration. The same notation is used for both Z and U. The algorithm is initialized with $\Theta^0 = I_p$, $Z^0 = U^0 = \mathbf{0} \in \mathbb{R}^{p\times p}$. We note that the Θ and U update steps remain unchanged from the original offline algorithm. However, in the case of the Z update an adjustment is required due to the fact that past networks, Θ_{t-1}, are treated as constants. Subsequently, Z is updated by solving:

$$Z^i = \underset{Z}{\text{argmin}} \quad \left\{1/2||\Theta^i - Z + U^{i-1}||_2^2 + \lambda_1||Z||_1 + \lambda_2||Z - \Theta_{t-1}||_1\right\}, \tag{4.18}$$

where Θ^i, U^i and Θ_{t-1} are treated as constants. We note that equation (4.18) involves a series of one-dimensional problems as only element-wise operations are applied. This implies that we may solve an independent problem of the following form for each entry in Z^i:

$$(Z^i)_{k,l} = \underset{(Z)_{k,l}\in\mathbb{R}}{\text{argmin}} \quad \left\{1/2||(\Theta^i - Z + U^{i-1})_{k,l}||_2^2 + \lambda_1||(Z)_{k,l}||_1 + \lambda_2||(Z - \Theta_{t-1})_{k,l}||_1\right\} \tag{4.19}$$

where we write $(M)_{k,l}$ to denote the (k,l) entry for any square matrix M. Thus each element of Z^i can be updated by solving a one-dimensional convex problem. While there is no closed form solution, we may employ efficient line search algorithms [134]. Due to the symmetric nature of Z it follows that only $\frac{p(p+1)}{2}$ of such problems must be solved.

Burn-in period

It is common for streaming algorithms to incorporate a brief burn-in phase when they are initialized. This involves collecting the first N_{BurnIn} observations to initialize parameter estimates. Many times such an approach is motivated by the need to ensure sample statistics are well-defined, however, due to the presence of regularization the proposed method does not require a burn-in *per se*. That said the use of a burn-in phase can improve initial network estimates and may thereby result in improved network estimation initially. As a result, the first N_{BurnIn} observations are collected and used to estimate the corresponding precision matrices by directly applying the offline SINGLE algorithm. This involves solving equation (4.11). From then onward, new estimates of the precision matrix are obtained as described previously.

4.1.3 Tuning parameters

Parameter estimation is challenging in the context of streaming data. Approaches such as cross-validation, which are inherently difficult to implement due to the non-stationarity of the data, are further hampered by the limited computational resources. As an alternative, information theoretic approaches such as minimizing the AIC or BIC may be employed but these too may incur a high computational burden.

In this chapter we advocate the use of adaptive filtering as such methods provide a flexible framework through which to handle temporal variation in the data which cannot easily be modeled. In this context of this work, the use of adaptive filtering methods designates the choice of forgetting factor, r_t, to the data. As a result, only a stepsize parameter ϵ is required. This is desirable as the choice of fixed forgetting factor (or sliding window) requires knowledge regarding the *degree* of non-stationarity of the data which is both difficult to justify as well as problem specific. In contrast, we can interpret the choice of ϵ as a stepsize parameter in a stochastic gradient descent scheme. As a result, there are clear guidelines which can be followed when selecting ϵ [21, 22].

Parameters λ_1 and λ_2 enforce sparsity and temporal homogeneity respectively. The choice of these parameters affects the degrees of freedom of estimated networks, suggesting the use of information theoretic approaches such as AIC. However, in a real-time setting,

choosing λ_1 and λ_2 in such a manner presents a computational burden. As a result, we propose two heuristics for choosing appropriate values of λ_1 and λ_2 respectively. One potential approach involves studying a previous scan of the subject in question. If this is available then the regularization parameters may be chosen by minimizing AIC over this scan. Alternatively, the burn-in phase may be used to choose adequate parameters. Such an approach would involve choosing λ_1 and λ_2 which minimized AIC over the burn-in period.

Finally, we note that the implicit assumption that regularization parameters should remain constant is difficult to justify in the context of non-stationary data. Ideally, such parameters should themselves to be time-varying. To this end, a framework through which to iteratively update the sparsity parameter, λ_1, is presented in Chapter 5.

4.2 Simulation study

In this section we evaluate the performance of the rt-SINGLE algorithm throughout a series of simulation studies. In each simulation we produce simulated time series data giving rise to a number of connectivity patterns and properties which reflect those reported in fMRI data. The objective is then to measure whether our proposed algorithm is able recover the underlying patterns in real-time. We are primarily interested in studying the performance of the proposed methods in two ways; first we wish to study the quality of the estimated covariance matrices over time. That is to say, we study how accurately our sample covariances represent the true underlying covariance structure. Second, we are also interested in the correct estimation of the presence or absence of edges.

In Simulation 1 we study how reliably we are able to track changes in the correlation structure using forgetting factors and adaptive filtering techniques. In Simulations 2 and 3 we consider the overall performance of the proposed method by generating connectivity structures according to scale-free and small-world networks respectively. Finally, in Simulation 4 we look to quantify the computational cost of the proposed method as the number of nodes, p, increases; a crucial aspect to study given the objectives of this work.

Throughout this section we compare results for the rt-SINGLE algorithm where the sample covariance matrix is iteratively updating in three ways: a sliding window, a fixed forgetting factor (corresponding to an EWMA model) and an adaptive forgetting factor.

Further, we also consider the performance of the offline SINGLE algorithm as a benchmark. Naturally, we expect the rt-SINGLE algorithms to generally perform below its offline counterpart.

Throughout each of these simulations, the parameters for the offline SINGLE algorithm where determined as described Chapter 3. As such, the choice of kernel width was obtained by maximizing leave-one-out log-likelihood while the regularization parameters where chosen by minimizing AIC. In the case of the real-time algorithms the parameters where chosen as follows: sliding windows of with a window length of 20 observations were employed. The fixed forgetting factor was chosen to be $r = 0.95$ as this corresponded approximately to an effective sample size of twenty observations. While in the case of adaptive forgetting, $\eta = 0.005$ was employed. In the case of the rt-SINGLE algorithm, regularization parameters were selected by minimizing AIC over a burn-in of 15 observations.

4.2.1 Performance measures

As discussed previously, we wish to evaluate the performance of the proposed method in two distinct ways. First, we wish to study the reliability with which we can track changes in correlation structure using either a fixed forgetting factor or an adaptive forgetting factor. In order to quantify the difference between the true correlation structure and our estimated covariance matrix, S, we consider the distance defined by the trace inner product:

$$d(\Sigma, S) = \text{Trace } (\Sigma^{-1}S). \tag{4.20}$$

We note that equation (4.20) is proportional to a Gaussian log-likelihood without the log-determinant term, which may be interpreted as a penalty on the complexity of the sample covariance [27]. It follows that if the sample covariance, S, is a good estimate of the true covariance, Σ, we will have that $d(\Sigma, S) \approx p$. However, if S is a poor estimate, the distance d will be large. Moreover, since both Σ and S are positive definite we have that $d(\Sigma, S)$ will always be positive.

Second, we wish to consider the estimated functional connectivity networks at each point in time. As in Chapter 3, we are interested in correctly identifying the non-zero entries in estimated precision matrices, Θ_i, at each $i = 1, \ldots, T$. An edge is assumed to be

present between the jth and kth nodes if $(\Theta_i)_{j,k} \neq 0$. At the ith observation we define the set of all reported edges as $D_i = \{(j,k) : (\Theta_i)_{j,k} \neq 0\}$. We define the corresponding set of true edges as $T_i = \{(j,k) : (K_i)_{j,k} \neq 0\}$ where we write K_i to denote the true precision matrix at the ith observation. Given D_i and T_i we consider a number of performance measures at each observation. As detailed in Section 3.2.2 we consider the precision, recall and F-score.

4.2.2 Results

The objective of the simulation study presented below is to empirically quantify the performance of the proposed rt-SINGLE algorithm. In Simulation 1, we consider the challenges of tracking correlation structure while in Simulations 2 and 3 we consider recovering the sparse support of the covariance structure. Finally, the computational demands of the proposed method are studied in Simulation 4.

Simulation 1 - Correlation tracking

In this simulation we look to assess how accurately we are able to track changes in correlation structure via the use of sliding windows as well as fixed (i.e., EWMA models) and adaptive forgetting factors.

Datasets were simulated as follows: each dataset consisted of five segments each of length 100 (i.e., overall duration of 500). The network structure within each segment was simulated according to either the preferential attachment model of Barabási and Albert [11] or using the Watts-Strogatz models [177]. A detailed description of each of these network generation models is provided in Appendix C.

Figure [4.2] shows results when scale-free (top) and small-world (bottom) network structures are simulated. We note that the performance of the sample covariance drops in the proximity of a change-point for all algorithms. In the case of the offline SINGLE algorithm this drop is symmetric due to the symmetric nature of the Gaussian kernel employed. However, in the case of the real-time algorithms the drop is highly asymmetric and occurs directly after the change-point, as is to be expected. Due to the sudden change in correlation structure, the performance of streaming methods drops immediately after a

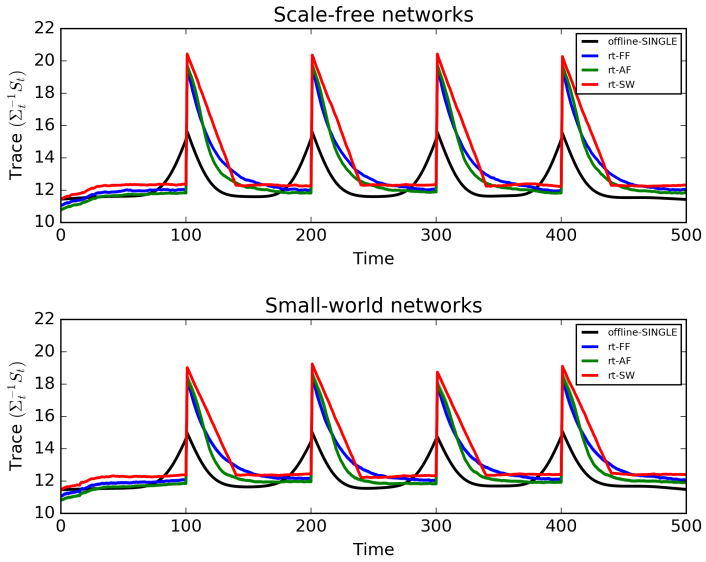


Figure 4.2: Mean trace scores, defined in equation (4.20), are shown when the sample covariance matrix is estimated using a symmetric Gaussian kernel (as in the offline SINGLE algorithm) and three real-time methods: sliding windows, fixed and adaptive forgetting factors. Results are shown when the underling correlation structure displays scale-free (top) or small-world properties.

change occurs. Moreover, from Figure [4.2] we note that the correlation tracking capabilities of the proposed methods are not adversely affected by the choice of underlying network structure.

Simulation 2 - scale-free networks

In this simulation we look to obtain a general comparison between the rt-SINGLE algorithm and its offline counterpart. Datasets were simulated as described in Simulation 1 using the Barabási and Albert preferential attachment model [11]. This generated scale-free networks were the the degree distribution of nodes followed a power law. This implies the presence of a reduced number of *hub* nodes which have access to many other regions, while the remaining majority of nodes have a small number of edges [54]. The entire dataset was simulated *apriori*. In the case of the rt-SINGLE algorithms, one observation was provided at time, thereby treating the dataset as if it was a stream arriving in real-time. The offline SINGLE algorithm was provided with the entire dataset and this was treated as

an offline task.

The left panel of Figure [4.3] shows the average F_t scores for each of the real-time algorithms as well as the offline algorithm over 500 simulations. We note that all three algorithms experience a drop in F-score in the proximity of change-points. The offline SINGLE algorithm is based on a symmetric Gaussian kernel and as a result it suffers a a symmetric drop in performance in the vicinity of a change-point before quickly recovering. Alternatively, the drop in performance of the rt-SINGLE algorithms is asymmetric. This is due to the real-time nature of these algorithms. Moreover, we note that while the rt-SINGLE algorithm performs worse than its offline counterpart directly after change-points, it is able to quickly recover to the level of the offline SINGLE algorithm. Specifically, in the case where adaptive forgetting is used, the real-time algorithm is able to outperform its offline counterpart in sections where the data remains piece-wise stationary for long periods of time. This is because it is able to increase the value of the adaptive forgetting factor accordingly. This allows the algorithm to exploit a larger pool of relevant information compared to its offline counterpart. This is demonstrated on the right panel of Figure [4.3] where the mean value of the adaptive forgetting factor is plotted. We see there is a drop directly after changes occur; this allows the algorithm to quickly forget past information which is no longer relevant. We also note that the estimated value of the forgetting factor increases quickly after changes occur.

Simulation 3 - small-world networks

While Simulation 2 studied scale-free networks, it has been reported that brain networks follow a small-world topology [13]. Such networks are characterized by their high clustering coefficients which has been reported in both anatomical as well as functional brain networks [162]. Datasets were simulated as described in Simulations 1 and 2, with the exception that individual networks were generated according to the Watts-Strogatz preferential attachment model [177].

Average F_t scores for each of the algorithms over $N = 500$ simulations are shown on the left panel of Figure [4.4]. As in Chapter 3, we note that the performance drops compared to scale-free networks considered in Simulation 2. We further note that the rate at which

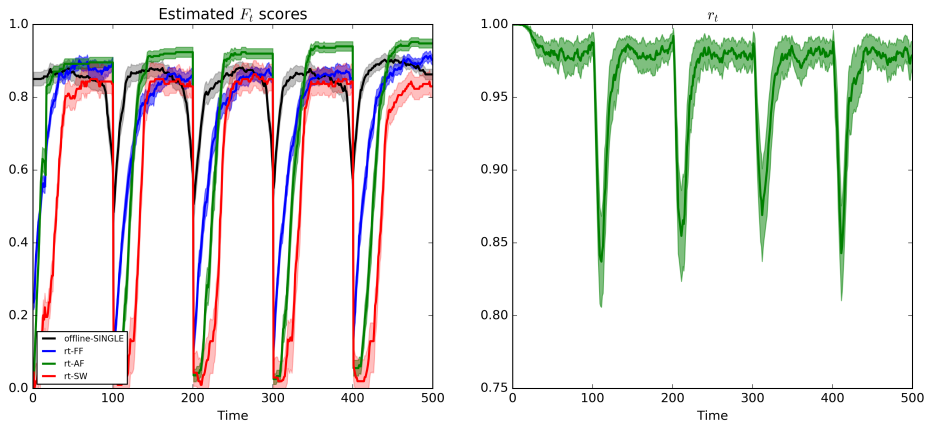


Figure 4.3: Results are shown for Simulation 2, where the underlying covariance structure followed a scale-free distribution. The left panel plots the mean F_t score across four distinct algorithms: the offline-SINGLE algorithm and the rt-SINGLE algorithm where the underlying covariance is estimated using sliding windows and fixed and adaptive forgetting factors. The right panel visualizes the mean adaptive forgetting factor over all simulations. We note there is a sudden drop immediately after each change-point before quickly recovering.

the real-time networks recover after a change-point is reduced. As with Simulation 2, we note that both of the real-time algorithms are able to reach the same level of performance as their offline counterpart if given sufficient time. Moreover, in the case where adaptive forgetting is employed we once again find that the performance of the real-time algorithm exceeds that of the offline algorithm when the data is remains piece-wise stationary for a sufficiently long period of time. In the right panel of Figure [4.4] we see the estimated adaptive forgetting factor over each of the 500 simulations. Again, we see the drop in the value of the forgetting factor directly after change-points, allowing past information to be discarded.

Simulation 4 — Computational cost

A fundamental aspect of real-time algorithms is that they must be computationally efficient in order to be able to update parameter estimates in the limited time provided. The main computational cost of the rt-SINGLE algorithm is related to the eigendecomposition of the Θ update, which has a complexity of $\mathcal{O}(p^3)$, as discussed in Section 3.1.2.

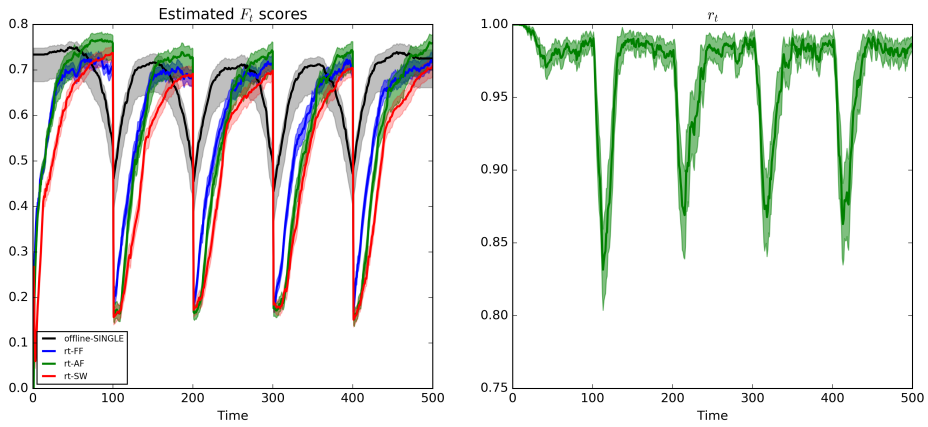


Figure 4.4: Results are shown for Simulation 3, where the underlying covariance structure followed a small-world distribution. The left panel plots the mean F_t score across four distinct algorithms: the offline-SINGLE algorithm and the rt-SINGLE algorithm where the underlying covariance is estimated using sliding windows and fixed and adaptive forgetting factors. The right panel visualizes the mean adaptive forgetting factor over all simulations. We note there is a sudden drop immediately after each change-point before quickly recovering.

In this simulation we look to empirically study the computational cost. In this manner, we are able to provide a rough guide as to the number of ROIs which can be employed in a real-time neurofeedback study while still reporting network estimates at every point in time. This was achieved by measuring the mean running time of each update iteration of the rt-SINGLE algorithm for various numbers of ROIs, p.

Here each dataset was simulated as in Simulation 2; that is the underlying correlation structure was randomly generated according to a small-world network. However, here we choose to only simulate three segments, each of length 50, resulting in a dataset consisting of 150 observations. For increasing values of p, the time taken to estimate a new precision matrix was calculated. Figure [4.5] shows the mean running time for the rt-SINGLE algorithm where either sliding window, a fixed forgetting factor or adaptive forgetting was used. We note that the difference in computational cost between each of the algorithms is virtually indistinguishable. More importantly, we note that the running times for each of the rt-SINGLE algorithms was significantly lower than the offline SINGLE algorithm; on average, the computational time associated with the offline SINGLE algorithm was 5-10

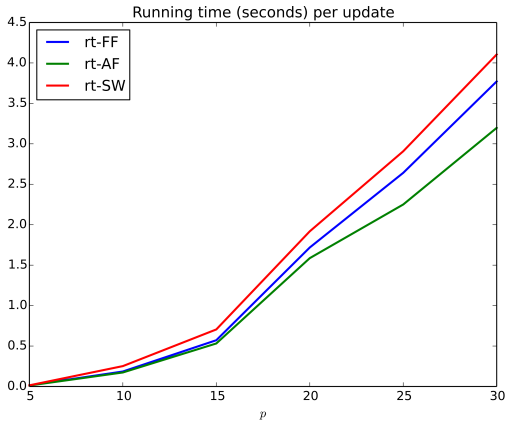


Figure 4.5: Mean running time (seconds) per update iteration of the rt-SINGLE algorithm when either a sliding window (rt-SW), fixed forgetting factor (rt-FF) or adaptive forgetting (rt-AF) was employed.

times that of the computational time associated with the real-time algorithms. For example, in the case of $p = 20$ nodes, rt-SINGLE algorithms required approximately 1.75 seconds whilst the offline SINGLE algorithm required 12 seconds.

Finally we note that when the number of nodes is below 20 it is possible to estimate functional connectivity networks in under two seconds, making the proposed method practically feasible in real-time studies. This simulation was run on a computer with an INTEL CORE I5 CPU at 2.8 GHz.

4.3 Application

In this section we present an applications of the rt-SINGLE algorithm to Motor task data taken from the Human Connectome Project (HCP) [55]. Subjects were required to perform a range of motor tasks such as tapping their fingers or squeezing their toes. While this data was not acquired and analyzed in real-time, it may be treated as such by only single observation at a time. In this manner, we are able to validate the performance of the rt-SINGLE algorithm on fMRI data.

4.3.1 HCP Motor task data

Twenty of the 500 available task-based fMRI datasets provided by the HCP were selected at random and data corresponding to a motor task was studied. The task, adapted from [186], involved the presentation of visual queues to subjects who had to perform one of five motor tasks. Each movement type was blocked, lasting 12 seconds and preceded by a three second visual cue. Each motor task was performed twice together with three additional fixation blocks of 15 seconds. This resulted in a total of 13 blocks.

While this data is not intrinsically real-time (as the preprocessing was conducted after data acquisition) it is included as a proof-of-concept study. The data was pre-processed offline as the focus lies on the comparison between the real-time and offline network estimation approaches rather than different preprocessing pipelines. Preprocessing involved regression of Fristons 24 motion parameters and high-pass filtering using a cut-off frequency of $\frac{1}{150}$Hz.

Eleven bilateral cortical ROIs were defined based on the Desikan-Killiany atlas [47] covering the occipital, parietal and temporal lobe. These regions were selected based on the hypothesis that changes would occur in the sensory-motor and higher-level visual areas. The extracted time courses from these regions were subsequently used for the analysis. By treating the extracted time course data as if it was arriving in real-time (i.e., considering one observation at a time), we can compare the results of the proposed real-time method to offline algorithms while using the same underlying pre-processed data.

4.3.2 Results

Both the SINGLE as well as the rt-SINGLE algorithms where applied to the motor-task fMRI dataset. Our primary interest here is to report task-driven changes in functional connectivity. In this way, we are able to examine if the rt-SINGLE algorithm is capable of reporting the changes functional connectivity induced by the motor task. The functional relationships that were modulated by the motor task were studied. This corresponds to studying the edges in the estimated networks which are significantly correlated with task onset. This was achieved by first estimating time-varying functional connectivity networks using both the offline SINGLE algorithm as well as the proposed real-time algorithm. In

the case of the SINGLE algorithm, parameters where chosen as described in Chapter 3. This involved estimating the width of the Gaussian kernel via leave-one-out cross validation and estimating regularization parameters via minimizing AIC. In the case of the real-time algorithm, an adaptive forgetting factor was employed with $\eta = 0.005$. The sparsity and temporal homogeneity parameters where set to the same values as those employed in the offline SINGLE algorithm as the focus here was to study the differences induced by estimating networks in real-time as opposed to differences resulting from distinct regularization penalties.

To determine which edges where modulated by the motor task a non-parametric statistical test was performed on an edge by edge basis. Formally, the Spearman rank correlation coefficient was estimated between the time-varying estimated partial correlation values for each edge and the task-evoked HRF function. It follows that edges which are modulated by the task will display strong correlations with the task HRF, thus allowing us to obtain a network of edges which are modulated by the motor task. Each estimated correlation coefficient was subsequently tested to determine if the correlation was statistically significant. The resulting p-values (one for each edge) were then corrected for multiple comparisons via the Holm-Bonferroni method [84]. In this manner, an activation network was obtained. This network summarized the set of edges which were statistically activated by the motor task for each algorithm.

Figure [4.6] shows task activation networks for both the SINGLE and rt-SINGLE algorithms. Edges are only present if they were reported as being significantly correlated with task-evoked HRF function. Red edges indicated the strength of the edge increase during task while blue edges indicate the strength of the edge decrease during task (i.e., a negative correlation). Furthermore, edge thickness is indicative of the magnitude of the correlation. Figure [4.6] shows clear similarities across each of the algorithms, with 84% of edges reported by both the rt-SINGLE and SINGLE algorithms. This would suggest that the rt-SINGLE algorithm is accurately detecting task-modulated changes in functional connectivity. In particular, we observe increased functional coupling between the motor-sensory and visual regions in the occipital cortex as well as inferior and middle temporal heteromodal regions. These results are plausible with regard to the task that involved high-level visual and heteromodal processing of the preceding visual cues and the execution of

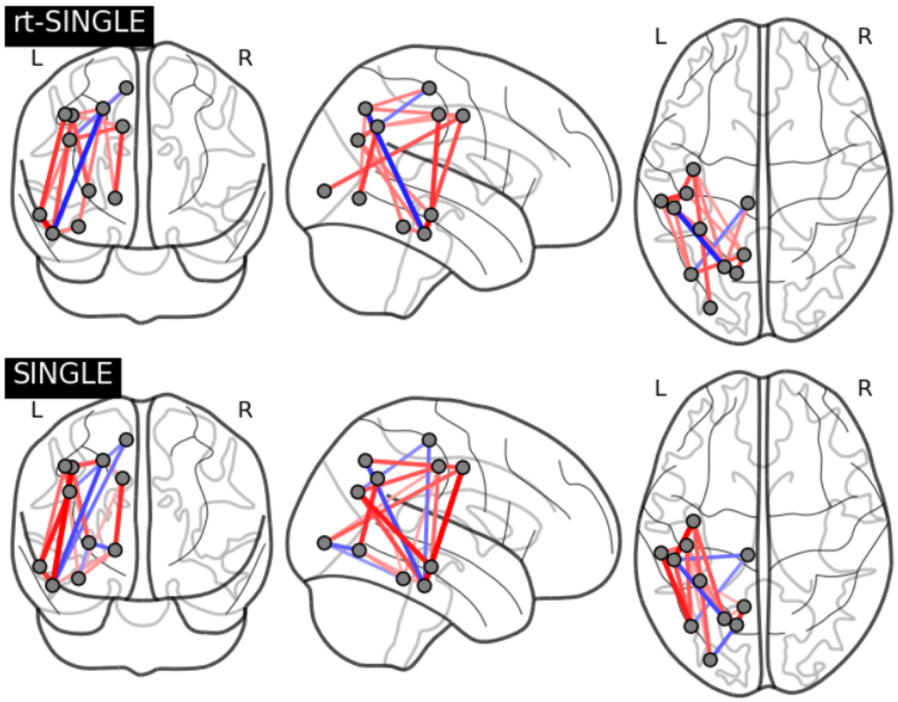


Figure 4.6: Task activation networks for rt-SINGLE (top) and SINGLE (bottom) algorithms, respectively. Present edges had statistically significant correlations with task HRF after correction for multiple comparisons. Red edges indicate edge strength increased during task while blue edges indicate edge strength decreased during task. In order to facilitate interpretation of the plot, only the right-hemispheric coordinates are shown here. We note there is consistent activation pattern across both algorithms, particularly across nodes nodes corresponding to the motorsensory areas.

the actual movement and have been previously reported [80, 188].

While Figure [4.6] serves to visually demonstrate that the rt-SINGLE algorithm is accurately detecting task modulated changes in connectivity, we also studied graph theoretic properties to quantify if there are significant differences in the graph structure of networks estimated using offline SINGLE and rt-SINGLE algorithms. While there are many candidate graph statistics which can be studied, in this work we look to study the three key properties; the mean degree centrality across nodes, the mean betweenness centrality over edges in the network and the transitivity of the network. Furthermore, the changes in network statistics where studied in the context of task positive and task negative modulation, thereby allowing us to study in detail if significant differences occurred in the estimated

Graph statistics	**Offline SINGLE**		**Real-time SINGLE**	
	Task positive	Task negative	Task positive	Task negative
Degree centrality	0.29 (0.10)	0.09 (0.05)	0.27 (0.11)	0.07 (0.04)
Betweenness centrality	0.07 (0.02)	0.01 (0.01)	0.01 (0.03)	0.02 (0.01)
Transitivity	0.24	0.06	0.22	0.06

Table 4.1: Summary graph statistics for networks estimated using the offline and real-time SINGLE algorithms respectively. Graph statistics are provided for task positive and task negative networks (correspond to red and blue edges in Figure [4.6] respectively) in order to allow for a detailed study of graph properties across both algorithms.

network structure. Graph statistics were calculated for the network of positively and negatively task-modulated edges respectively (that is the networks corresponding to the red and blue edges in Figure [4.6] respectively). The results are provided in Table 4.1. We note that no significant differences are reported for each of the graph statistics considered. These results serve as evidence that the proposed method can perform comparably with offline methods despite facing the additional challenge of estimating networks on-the-fly.

Furthermore, in real-time fMRI studies it is crucial to be able to accurately estimate functional connectivity networks on a subject-by-subject basis. While the true underlying functional connectivity networks are unknown (and may vary for each subject), we are able to quantify how closely the networks estimated in real-time recreate the results of an offline analysis. As a result, the correlation was studied between the estimated edges using both the rt-SINGLE and the offline SINGLE algorithms. This was performed on a subject-by-subject basis. For each edge, the correlation between the estimated edge values using each of the two algorithms was quantified using Spearmans rank correlation coefficient and the corresponding p-values were corrected for multiple comparisons. Figure [4.7] shows the subject-specific networks containing only edges that were significantly correlated across both algorithms. As before, red edges indicate a positive correlation with task while blue edges are indicative of negative correlations and the thickness of the edges is proportional to the strength of the correlation. We note the resulting networks are dense across all subjects and the vast majority of edges indicate positive correlations. In particular, an average of 74% of edges were positively correlated across all subjects.

As noted previously, it is also important to study graph theoretic properties of the esti-

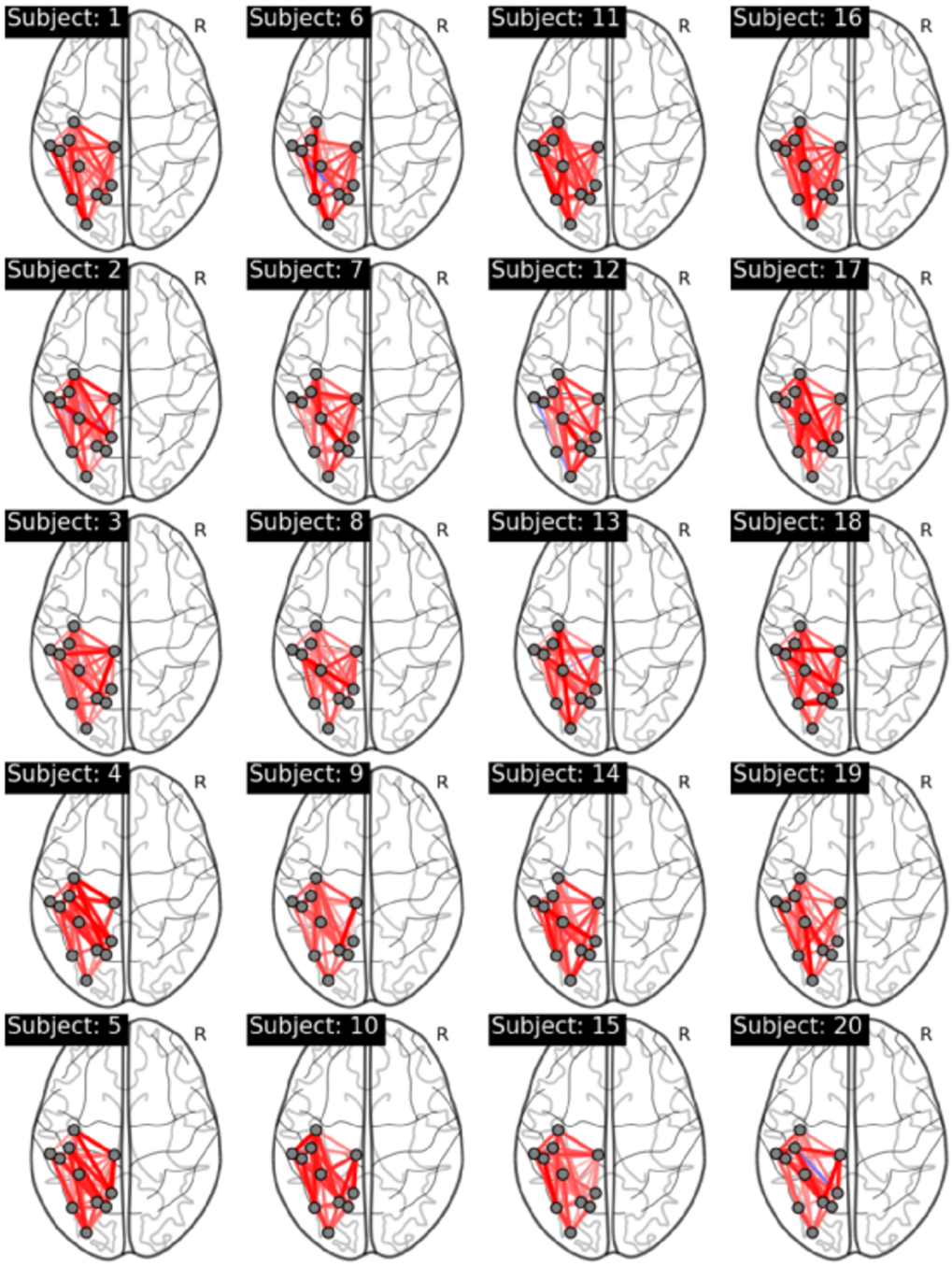


Figure 4.7: Subject specific networks visualizing edges that were significantly correlated across both the rt-SINGLE algorithm and its offline counterpart. Red edges indicate positive correlations while blue edges indicate negative correlations. We note that networks are dense across all subjects, indicating that the rt-SINGLE algorithm is able to accurately recover network structures similar to an offline study. Associated summary graph statistics of the task positive and task negative networks across all subjects are provided in Table 4.2.

mated networks to quantitatively study whether there are significant differences in the network structure across subjects. As a result, we computed the three aforementioned graph statistics over the subject-specific estimated networks shown in Figure [4.7]. The results are provided in Table 4.2 which indicate significant differences in the edge statistics from task positive to task negative states. We note that in the case of each of the three graph statistics considered, the difference in the mean values is over two standard deviations away. These results serve to indicate that the estimated networks are stable and consistent across both algorithms as well as across the cohort of subjects.

Finally, we note that the use of the real-time SINGLE algorithm also resulted in significant computational improvements. More specifically, inline with the improvements reported in Simulation 4, we observed a 6-fold decrease in the running time of the real-time SINGLE algorithm when compared to its offline counterpart.

4.4 Conclusion

In this chapter we have extended the methods proposed in Chapter 3 to perform covariance selection in the context of streaming data. This has required both the introduction of efficient methods through which to iteratively obtain estimates of sufficient statistics as well as adapting the original optimization algorithm so that sparsity and temporal homogeneity may be enforced in a real-time estimation framework.

In order to incrementally obtain estimates of sufficient statistics such as the sample covariance matrix, the rt-SINGLE algorithm extends widely used sliding window methods by considering fixed and adaptive forgetting methods. Such methods are preferred in this work due to their superior theoretic properties [106] as well as improved empirical results obtained through simulations. In particular, adaptive forgetting methods provide several important advantages. Firstly, such methods effectively designate that choice of the forgetting factor to the data, making them highly adaptive and allowing them to handle non-stationary data without requiring an explicit model for such behavior.

The proposed methods directly extend the SINGLE algorithm to the context of the streaming data by deriving a computationally tractable approximation to the SINGLE objective function. The proposed approximation involves iteratively estimating a precision

Graph statistics	Estimated value across subjects	
	Task positive	Task negative
Degree centrality	0.31 (0.05)	0.11 (0.06)
Betweenness centrality	0.06 (0.01)	0.01 (0.01)
Transitivity	0.27 (0.06)	0.08 (0.04)

Table 4.2: Summary graph statistics for networks estimated using the real-time SINGLE algorithm across the cohort of subjects. Graph statistics are provided for task positive and task negative networks (correspond to red and blue edges in Figure [4.7] respectively) in order to allow for a detailed study of the robust nature of graph statistics across all subjects.

matrix for each incoming observation, as opposed to jointly estimating multiple matrices. As in the SINGLE algorithm, sparsity and temporal homogeneity constraints are introduced. As we demonstrate through a series of simulation studies, the rt-SINGLE algorithm is able to both obtain accurate estimates of functional connectivity networks at each point in time as well as accurately describe the evolution of networks over time.

The proposed method requires the input of three parameters. The first of these parameters, stepsize η, governs the rate at which an adaptive forgetting factor, r_t, varies and can be interpreted as the stepsize in a stochastic gradient descent scheme [22]. The final two parameters enforce sparsity and temporal homogeneity respectively. These parameters remain fixed throughout in a similar manner to the fixed forgetting factor and two heuristic approaches are proposed to tune these parameters. However, the assumption that such parameters remain fixed is difficult to justify. This is particularly the case in the context of non-stationary data. This issue is addressed in Chapter 5 where an adaptive update for regularization parameters is proposed for streaming regression models.

An application of the proposed algorithm to task-based fMRI data is presented. The results demonstrate that the rt-SINGLE algorithm was able to accurately detect functional networks which are modulated by motor task. While the data is not intrinsically real-time, observations were treated as such and therefore serves as a proof-of-concept. Moreover, the results indicate that functional connectivity networks may be reliably estimated both at a group level as well as on a subject-by-subject basis.

In conclusion, the rt-SINGLE algorithm provides a novel method for estimating functional connectivity networks in real-time. In future, the proposed method could be incor-

porated into rt-fMRI studies, potentially providing neurofeedback based on functional connectivity. One exciting avenue would be to integrate this work with the recently proposed Automatic Neuroscientist framework of [109]. Such a framework combines real-time fMRI with machine learning techniques to optimize experimental conditions to maximize a given target brain state [110, 108]. While the target brain state in the original proof-of-principle study presented in [109] was simply based on BOLD differences, the proposed method can be utilized to extend the Automatic Neuroscientist to target entire functional connectivity networks.

Chapter 5

Adaptive penalization in streaming regression models

Motivated by the non-stationary properties of fMRI data, novel methodologies for quantifying the dynamic properties of covariance structure were proposed in Chapters 3 and 4. The proposed methods relied heavily on the use regularization penalties. However, the introduction of such penalties in the context of non-stationary data raises important methodological questions relating to the choice of the associated regularization parameters as well as the implicit assumption that such parameters remain fixed. In this chapter, we look to address some of these questions by presenting a framework through which to learn a time-varying sparsity parameter in the context of streaming data. The proposed framework effectively recasts the selection of a sparsity parameter in the context of adaptive filtering, thereby relegating the choice of such a parameter to the data. This reformulation also allows for the derivation of convergence guarantees in a non-stochastic setting. Such a framework is developed for streaming lasso models and then extended to GGMs via neighborhood selection techniques described in Section 2.2.

As a result, this chapter is focused on in learning ℓ_1 regularized linear regression models in the context of streaming, non-stationary data. While there has been significant research relating to the estimation of such models in a streaming data context [23, 52], a fundamental aspect which has been overlooked is the selection of the regularization parameter. The choice of this parameter dictates the severity of the regularization penalty. While the un-

derlying optimization problem remains convex, distinct choices of such a parameter yield models with vastly different characteristics. This poses significant concerns from the perspective of model performance and interpretation. It therefore follows that selecting such a parameter is an important problem that must be addressed in a data-driven manner.

Many solutions have been proposed through which to select the regularization parameter in a non-streaming context. For example, stability based approaches have been proposed in the context of linear regression [118]. Other popular alternatives include cross-validation [65] and information theoretic techniques. However, in a streaming setting such approaches are infeasible due to the limited computational resources available. Moreover, the statistical properties of the data may vary over time; a common manifestation being concept drift [2, 182]. This complicates the use of sub-sampling methods as the data can no longer be assumed to follow a stationary distribution. Furthermore, as we argue in this work, it is conceivable that the optimal choice of regularization parameter may itself vary over time. It is also important to note that traditional approaches such change-point detection cannot be employed as there is no readily available pivotal quantity. It therefore follows that novel methodologies are required in order to tune regularization parameters in an online setting.

The remainder of this Chapter is organized as follows: the proposed framework is detailed in Section 5.1. This involves a discussion of the computational demands of the framework as well as the associated convergence guarantees. We present an extensive simulation study in Section 5.2 and an application to data from the Human Connectome Project (HCP) is presented in Section 5.3.

5.1 Real-time adaptive penalization framework

In this work we are interested in streaming linear regression problems. Here it is assumed that pairs (X_t, y_t) arrive sequentially over time, where $X_t \in \mathbb{R}^{p-1}$ corresponds to a $(p-1)$-dimensional vector of predictor variables and y_t is a univariate response. The objective of this work is to learn time-varying linear regression models from which to accurately predict future responses, y_{t+1}, from predictors, X_{t+1}. An ℓ_1 penalty, parameterized by $\lambda \in \mathbb{R}_+$, is introduced in order to encourage sparse solutions as well as to ensure the problem is well-posed from an optimization perspective. This corresponds to the lasso model intro-

duced by [166]. For a given choice of regularization parameter, λ, time-varying regression coefficients can be estimated by minimizing the following convex objective function:

$$L_t(\beta, \lambda) = \sum_{i=1}^{t} w_i \left(y_i - X_i^T \beta\right)^2 + \lambda ||\beta||_1, \tag{5.1}$$

where $w_i > 0$ are weights indicating the importance given to past observations. Typically, w_i decay monotonically as a function of the chronological proximity of the ith observation. For example, weights w_i may be tuned using a fixed forgetting factor or a sliding window as discussed in Chapter 4.

In a non-stationary context, the optimal estimates of regression parameters, $\hat{\beta}_t$, may vary over time. The same argument can be posed in terms of the selected regularization parameter, λ. For example, this may arise due to changes in the underlying sparsity or changes in the signal-to-noise ratio of the data. While there exists a wide range of methodologies through which to update regression coefficients in a streaming fashion, the choice of regularization parameter has been largely overlooked in the literature. As such, the primary objective of this work is to propose a framework through which to learn time-varying regularization parameter in real-time. The proposed framework is based on adaptive filtering theory, described in Chapter 4.

As noted previously, the choice of parameter λ dictates the severity of the regularization penalty. Different choices of λ result in vastly different estimated models. While several data-driven approaches are available for selecting λ in an offline setting, such methods are typically not feasible for streaming data for two reasons. First, limited computational resources pose a practical restriction. Second, data streams are often non-stationary and rarely satisfy IID assumptions required for methods based on the bootstrap [2]. Moreover, it is important to note that traditional methods such as change point detection cannot be employed due to the absence of a readily available pivotal quantity for λ.

In this section we detail the proposed framework for real-time adaptive penalization (RAP) in the context of streaming lasso models. We begin by outlining the RAP framework and deriving the necessary machinery in Section 5.1.1. Section 5.1.2 outlines the resulting algorithm. Computational considerations are discussed in Section 5.1.3 and an efficient approximation is presented. In Section 5.1.4 we study some of the theoretical properties of

the proposed framework and provide some convergence results in a non-stationary setting.

5.1.1 Proposed framework

We propose to learn a time-varying sparsity parameter in an adaptive filtering framework. This allows the proposed method to relegate the choice of sparsity parameter to the data. Moreover, by allowing λ_t to vary over time the proposed method is able to naturally accommodate datasets where the underlying sparsity may vary over time.

We define the empirical objective to be the look-ahead residual error, defined as:

$$C_{t+1} = C(X_{t+1}, y_{t+1}) = ||y_{t+1} - X_{t+1}\hat{\beta}_t(\lambda_t)||_2^2, \tag{5.2}$$

where we write $\hat{\beta}_t(\lambda_t)$ to emphasize the dependence of the estimated regression coefficients on the current value of the regularization parameter, λ_t. In an adaptive filtering framework, the regularization parameter can be iteratively updated as follows:

$$\lambda_{t+1} = G(\lambda_t) = \lambda_t - \epsilon\frac{\partial C_{t+1}}{\partial \lambda_t}. \tag{5.3}$$

We note that for convenience we write $\frac{\partial C_{t+1}}{\partial \lambda_t}$ to denote the derivative of C_{t+1} with respect to λ evaluated at $\lambda = \lambda_t$ (i.e, $\frac{\partial C_{t+1}}{\partial \lambda}|_{\lambda=\lambda_t}$). We note that λ_t is bounded below by zero, in which case no regularization is applied, and above by $\lambda_t^{max} = \max_j \left\{|\sum_{i=1}^t w_i y_i X_{i,j}|\right\}$, in which case all regression coefficients are set to zero [66].

The proposed framework requires only the specification of an initial sparsity parameter, λ_0, together with a stepsize parameter, ϵ. In this manner the proposed framework effectively replaces a fixed sparsity parameter with a stepsize parameter, ϵ. This is desirable as the choice of a fixed sparsity parameter is difficult to justify in the context of streaming, non-stationary data. Moreover, any choice of λ is bound to be problem specific. In comparison, we are able to interpret ϵ as a stepsize parameter in a stochastic gradient descent scheme. As a result, there are clear guidelines which can be followed when selecting ϵ [21].

Once the regularization parameter has been updated, estimates for the corresponding regression coefficients can be obtained by minimizing $L_{t+1}(\beta, \lambda_{t+1})$, for which there is a wide literature available [21, 23, 52]. The challenge in this work therefore corresponds to

efficiently calculating the derivative in equation (5.3). Through the chain rule, this can be decomposed as:

$$\frac{\partial C_{t+1}}{\partial \lambda_t} = \frac{\partial C_{t+1}}{\partial \hat{\beta}_t} \cdot \frac{\partial \hat{\beta}_t}{\partial \lambda_t}. \tag{5.4}$$

The first term in equation (5.4) can be obtained by direct differentiation. In the case of the second term, we leverage the results of [53] and [146] who demonstrate that the lasso solution path is piecewise linear as a function of λ. By implication, $\frac{\partial \hat{\beta}_t}{\partial \lambda_t}$ must be piecewise constant. Furthermore, there is a simple, closed-form solution for $\frac{\partial \hat{\beta}_t}{\partial \lambda_t}$.

Proposition 5 *[Adapted from [146]] In the context of ℓ_1 penalized linear regression models, the derivative $\frac{\partial \beta_t}{\partial \lambda_t}$ is piecewise constant and can be obtained in closed form.*

Proof Since $\hat{\beta}_t$ minimizes the objective function specified in equation (5.1), it satisfies:

$$\nabla_\beta \left(L_t(\beta, \lambda) \right) |_{\beta = \hat{\beta}_t} \ni 0 \tag{5.5}$$

This follows from Section 2.3.1, which states that a convex objective function is minimized when the gradient is zero. Recall from equation (5.1) that $L_t(\beta, \lambda)$ is composed of the sum of a squared error term and an ℓ_1 penalty term. As such, the we have that the derivative of $L_t(\beta, \lambda)$ with respect to β is as follows:

$$\nabla_\beta L_t(\beta, \lambda) = -X_{1:t}^T W_{1:t} (Y_{1:t} - X_{1:t}^T \beta) + \lambda \, \text{sign}(\beta) \tag{5.6}$$

where $W_{1:t}$ is a diagonal matrix with entries $w_1 \ldots, w_t$. We may therefore take the derivative of equation (5.5) with respect to λ and obtain the following:

$$\begin{aligned}
\frac{\partial}{\partial \lambda} \left(\nabla_\beta L_t(\beta, \lambda)|_{\beta = \hat{\beta}_t} \right) &= 0 && (5.7) \\
&= \frac{\partial \hat{\beta}_t}{\partial \lambda} \nabla \left(-X_{1:t}^T W_{1:t} (Y_{1:t} - X_{1:t}^T \hat{\beta}_t) \right) + \text{sign}(\hat{\beta}_t) && (5.8) \\
&= \frac{\partial \hat{\beta}_t}{\partial \lambda} \left(X_{1:t}^T W_{1:t} X_{1:t} \right) + \text{sign}(\hat{\beta}_t) && (5.9)
\end{aligned}$$

where equation (5.8) is obtained by applying the chain rule. Rearranging equation (5.9)

yields:

$$\frac{\partial \hat{\beta}_t}{\partial \lambda} = -\left(X_{1:t}^T W X_{1:t}\right)^{-1} \text{ sign}(\hat{\beta}_t), \tag{5.10}$$

$$= -(S_t)^{-1} \text{ sign}(\hat{\beta}_t). \tag{5.11}$$

From Proposition 1 we have that the derivative, $\frac{\partial C_{t+1}}{\partial \lambda_t}$, can be computed in closed form. Moreover, we note that the derivative in equation (5.10) is only defined over the active set of regression coefficients, denoted by $\mathcal{A}_t = \{i : (\hat{\beta}_t(\lambda_t))_i \neq 0\}$, and zero elsewhere. In practice we must therefore consider two scenarios:

- the active set is non-empty (i.e., $\mathcal{A}_t \neq \emptyset$). In this case equation (5.10) is well-defined.
- the active set is empty. In this case we proceed to take a step in the direction of the most correlated predictor: $\hat{j} = \underset{j}{\text{argmax}} \left\{ |\sum_{i=1}^t w_i y_i X_{i,j}| \right\}$. Thus we have that:

$$\left(\frac{\partial \hat{\beta}_t}{\partial \lambda}\right)_i = -\delta_{i,\hat{j}} \text{ sign}\left(\sum_{i=1}^t w_i y_i X_{i,\hat{j}}\right), \tag{5.12}$$

  where $\delta_{i,j}$ is the dirac-delta function.

5.1.2 Streaming lasso regression

At each iteration, a new pair (X_{t+1}, y_{t+1}) is received and employed to update both the time-varying regularization parameter, λ_t, as well as the corresponding estimate of regression coefficients, $\hat{\beta}_t(\lambda_t)$. The former involves computing the derivative $\frac{\partial C_{t+1}}{\partial \lambda_1}$ as outlined in Section 5.1.1. As noted in equation (5.11), a current estimate of the sample covariance matrix is sufficient. This may be recursively estimated in a variety of ways, for example using a fixed forgetting factor as detailed in Chapter 4. The latter involves solving a convex optimization problem which can be addressed in a variety of ways. In this work we look to iteratively estimate regression coefficients using coordinate descent method, discussed in Section 2.3. Such methods are easily amenable to streaming data and allow us to exploit previous estimates as warm starts. In our experience, the use of warm starts leads to convergence within a handful of iterations. Psuedo-code detailing the proposed RAP framework

is given in Algorithm 2.

Algorithm 2: Real-time Adaptive Penalization

Input: Stepsize $\epsilon \in \mathbb{R}_+$, fixed forgetting factor $r \in [0, 1)$ used to compute S_t

begin
  for $t \leftarrow 1, \ldots, t \ldots$ **do**
    receive new (X_{t+1}, Y_{t+1});
    if $\mathcal{A}_t \neq \emptyset$ **then**
      set $\frac{\partial \hat{\beta}_t}{\partial \lambda_t}$ using equation (5.10);
    else
      set $\frac{\partial \hat{\beta}_t}{\partial \lambda_t}$ using equation (5.12) ;
    set $\frac{\partial C_{t+1}}{\partial \lambda_t} = \frac{\partial C_{t+1}}{\partial \beta_t} \frac{\partial \hat{\beta}_t}{\partial \lambda_t}$;
    update $\lambda_{t+1} = \lambda_t - \epsilon \frac{\partial C_{t+1}}{\partial \lambda_t}$;
    $\hat{\beta}_{t+1}(\lambda_{t+1}) = \underset{\beta}{\operatorname{argmin}} \{L_{t+1}(\beta, \lambda_{t+1})\}$

5.1.3 Computational considerations

With respect to the computational and memory demands, we note that the major expense incurred when calculating $\frac{\partial \beta_t}{\partial \lambda_t}$ involves inverting the sample covariance matrix. While only the dimensions corresponding to active variables need to be considered, this still corresponds to inverting a $|\mathcal{A}_t| \times |\mathcal{A}_t|$ matrix. It is possible to alleviate the computational burden by efficiently updating $(S_t)_{\mathcal{A}_t, \mathcal{A}_t}$ using the Sherman - Morrison formula. In this case, care must be taken to ensure that the support of $\mathcal{A}_t$ has not changed from iteration $t - 1$ to t. If this is not the case (i.e., a regression coefficient has either added/removed from $\mathcal{A}_t$) then the inverse must be calculated directly from $(S_t)_{\mathcal{A}_t, \mathcal{A}_t}$.

However, computational and memory efficiency is paramount to streaming methods. The need to compute and store the inverse of the sample covariance is undesirable in the context of high-dimensional data. As a result, the following approximation is proposed:

$$\frac{\partial \hat{\beta}_t}{\partial \lambda_t} \approx - \left(\text{diag}\left(S_t\right)\right)^{-1} \text{ sign } \left(\hat{\beta}_t\right). \tag{5.13}$$

Here a diagonal approximation to the sample covariance is employed, implying that only

the diagonal elements of the sample covariance must be stored and inverted. Such approximations are frequently employed in streaming or large data applications [52]. This serves to reduce the computational burden of updating the sparsity parameter in the proposed manner. The approximate update therefore has a time and memory complexity that is proportional to the cardinality of the active set, $\mathcal{A}_t$.

5.1.4 Fixed point convergence

In this section we study some of the properties of the proposed framework from a theoretical perspective. We note that a formal theoretical treatment would require analysis of convergence using the tools of stochastic approximation theory [21, 23]. However, in this section we derive some preliminary properties of the proposed framework and provide a sketch of what future theoretical results may resemble. This is achieved by studying the behavior of the RAP framework in a non-stochastic setting. As such, our objective is to demonstrate convergence to a fixed point when the gradient updates are iteratively applied to the data. We begin by noting that this update rule is piecewise non-expansive over the support of regularization parameter. We then show that iteratively applying equation (5.3) leads to convergence to a fixed point.

Recall that $G(\lambda_t) = \lambda_t - \epsilon \frac{\partial C_{t+1}}{\partial \lambda_t}$ is a self-mapping defined on the support $\Lambda = [0, \lambda_t^{max}]$. We study the behavior of iteratively applying the update rule $G(\lambda_t)$ for fixed new data pair (X_{t+1}, y_{t+1}). This corresponds to iteratively performing the gradient descent update to minimize residual error, C_{t+1}, for some fixed unseen pair, (X_{t+1}, y_{t+1}). While the proposed algorithm is stochastic in the sense that distinct random samples, (X_{t+1}, y_{t+1}), are employed at each update step, the results presented below provide reassuring guarantees in a non-stochastic setting. We note that such non-stochastic results are often presented when studying online algorithms.

Throughout the remainder of this section we abuse notation and write $\lambda_{t+1} = G(\lambda_t)$ to denote the result of applying the gradient update for t iterations. Our goal is to show that the limit of $\delta_t = |\lambda_{t+1} - \lambda_t|$ converges to zero as the number of iterations, t, increases.

First, we demonstrate that the support of the regularization parameter, Λ, can be partitioned into $p - 1$ subsets where G is a contraction mapping. Then the mappings across

subsets are studied to show that proposed algorithm does not exhibit periodic, expansive behavior. These two results are combined to obtain convergence to a fixed point.

Assumption 1 *The coefficient profiles for each regression coefficient,* $\{\beta(\lambda) : \lambda \in \Lambda\}$*, are monotone.*

Assumption 1 implies that $|\beta(\lambda_1)| < |\beta(\lambda_2)|$ for all $\lambda_1 > \lambda_2$. For any λ we define the corresponding active set as $\mathcal{A}(\lambda)$. Assumption 1 thus implies $\mathcal{A}(\lambda_1) \subset \mathcal{A}(\lambda_2)$ for all $\lambda_1 > \lambda_2$. We note that the assumption of monotonicity has been previously employed to study the properties of lasso estimators [75]. Furthermore, this assumption can be verified in practice by checking that the inverse covariance is diagonally dominant [75].

Lemma 1 *Under Assumption 1,* Λ *can be divided into* $p-1$ *open subsets,* $\{\mathcal{S}_i\}_{i=1}^{p-1}$*, where* G *is a contraction mapping for suitably selected stepsize parameter,* ϵ*.*

Proof We assume without loss of generality that $\lambda_1 > \lambda_2$. We consider:

$$|G(\lambda_1) - G(\lambda_2)| = \left|\lambda_1 - \lambda_2 - \epsilon\left(\frac{\partial C_{t+1}}{\partial \lambda_1} - \frac{\partial C_{t+1}}{\partial \lambda_2}\right)\right|. \tag{5.14}$$

Our objective is to show that $\frac{\partial C_{t+1}}{\partial \lambda_1} - \frac{\partial C_{t+1}}{\partial \lambda_2} > 0$, thereby showing that G is a contraction for suitably chosen ϵ. Recall the gradient with respect to regularization parameter λ is defined as:

$$\frac{\partial C_{t+1}}{\partial \lambda} = \left(y_{t+1} - X_{t+1}\beta_t(\lambda)\right)^T X_{t+1}^T \left(S_t\right)^{-1} \text{sign}\left(\beta_t(\lambda)\right)$$

Due to Assumption 1, we have that:

$$\frac{\partial C_{t+1}}{\partial \lambda_1} - \frac{\partial C_{t+1}}{\partial \lambda_2} = \underbrace{\sum_{i \in \mathcal{A}(\lambda_1) \cap \mathcal{A}(\lambda_2)} \left[\left(\beta_t(\lambda_2) - \beta_t(\lambda_1)\right)^T \left(X_{t+1}^T X_{t+1}\right) \left(S_t\right)^{-1} \text{sign}(\beta_t(\lambda_1))\right]_i}_{A_1}$$
$$\underbrace{- \sum_{i \in \mathcal{A}(\lambda_2) \setminus \mathcal{A}(\lambda_1)} \left[\left(y_{t+1} - X_{t+1}\beta_t(\lambda_2)\right)^T X_{t+1}^T \left(S_t\right)^{-1} \text{sign}(\beta_t(\lambda_2))\right]_i}_{A_2}$$

We note that A_2 will be zero whenever $\mathcal{A}(\lambda_1) \setminus \mathcal{A}(\lambda_2) = \emptyset$. Moreover, the term A_1 will always be greater than or equal to zero. This follows from the fact that $A_1 = g(\lambda_1) - g(\lambda_2)$

where

$$\begin{aligned} g(\lambda) &= -\left(\beta_t(\lambda)^T \left(X_{t+1}^T X_{t+1}\right) (S_t)^{-1} \operatorname{sign}(\beta_t(\lambda))\right) \\ &= \beta_t(\lambda)^T \left(X_{t+1}^T X_{t+1}\right) \frac{\partial \beta_t(\lambda)}{\partial \lambda}. \end{aligned}$$

Recall from Proposition 5 that the derivative, $\frac{\partial \beta_t(\lambda)}{\partial \lambda}$, is piecewise constant as a function of λ. As a result, it follows that $\frac{\partial^2 \beta_t(\lambda)}{\partial \lambda^2} = 0$. Therefore, we have that:

$$\frac{\partial g(\lambda)}{\partial \lambda} = \left(\frac{\partial \beta_t(\lambda)}{\partial \lambda}^T \left(X_{t+1}^T X_{t+1}\right) \frac{\partial \beta_t(\lambda)}{\partial \lambda} \right) \geq 0, \tag{5.15}$$

due to the positive semi-definite nature of $X_{t+1}^T X_{t+1}$. This indicates that $g(\lambda)$ is a monotone, non-decreasing function in λ. As a result, we are able to divide Λ into $p-1$ open subsets where the update rule G is a contraction for suitably selected ϵ. These subsets correspond to the regions where the support of the lasso solution is constant, thus implying that A_2 is zero. Finally, we note that there are many different rules which are frequently employed when setting stepsize parameters such as ϵ [27]. In the context of this work, it sufficies to check that:

$$|\lambda_1 - \lambda_2| \leq \epsilon \left| \frac{\partial C_{t+1}}{\partial \lambda_1} - \frac{\partial C_{t+1}}{\partial \lambda_2} \right|, \tag{5.16}$$

which can easily be checked for all λ_1, λ_2 as $\frac{\partial C_{t+1}}{\partial \lambda}$ can be evaluated in closed form.

By Lemma 1, we have that $|G(\lambda_1) - G(\lambda_2)| < |\lambda_1 - \lambda_2|$ for all $\lambda_1, \lambda_2 \in \mathcal{S}_i$. It remains to study the (possibly expansive) behavior across the subsets $\{\mathcal{S}_i\}_{i=1}^{p-1}$; in particular there remains a need to mitigate against potential periodic, expansive behavior.

Lemma 2 *If periodic behavior occurs across subsets, then this must be a contraction.*

Proof We consider periodic behavior of the form:

$$G(\lambda_t) \in \begin{cases} \mathcal{S}_j & \text{if } t \text{ is even} \\ \mathcal{S}_{j-1} & \text{if } t \text{ is odd} \end{cases} \tag{5.17}$$

We consider two subsets which we label $\mathcal{S}_1$ and $\mathcal{S}_2$. Without loss of generality we assume that $\mathcal{S}_1 > \mathcal{S}_2$ in the sense that $\lambda_1 > \lambda_2$ for all $\lambda_1 \in \mathcal{S}_1$ and $\lambda_2 \in \mathcal{S}_2$. We consider the periodic behavior described in equation (5.17).

Therefore, at an odd iteration the gradient update maps from $\mathcal{S}_2$ into $\mathcal{S}_1$. Thus we have $\lambda_t = G(\lambda_{t-1}) > \lambda_{t-1}$ by construction. This implies that the gradient, $\frac{\partial C_t}{\partial \lambda_{t-1}}$, is negative here. This can be seen by noting that

$$\lambda_t = \lambda_{t-1} - \epsilon \frac{\partial C_t}{\partial \lambda_{t-1}} > \lambda_{t-1}.$$

Conversely, at every even iteration the gradient update maps from $\mathcal{S}_1$ into $\mathcal{S}_2$, implying that $\lambda_t = G(\lambda_{t-1}) < \lambda_{t-1}$. This in turn implies that $\frac{\partial C_t}{\partial \lambda_{t-1}}$ must be positive here. As a result, we have that for any $\lambda_1 \in \mathcal{S}_1$ and $\lambda_2 \in \mathcal{S}_2$:

$$\frac{\partial C_t}{\partial \lambda_1} - \frac{\partial C_t}{\partial \lambda_2} > 0$$

indicating that cyclic mapping must be contractions.

By Lemma 2 we have that any periodic behavior across subsets must be a contraction. Due to the compact nature of Λ, it follows that at most $p - 1$ (possibly expansive) non-periodic mappings across subsets occur, after which only contraction mappings occur.

Proposition 6 *Iteratively applying the gradient descent mapping G over a fixed training example, (X_{t+1}, y_{t+1}), leads to convergence to a fixed point.*

Proof We consider the sequence $\{\delta_t\}_t$ where $\delta_t = |\lambda_{t+1} - \lambda_t|$. The terms in $\{\delta_t\}_t$ can be split exactly into two subsequences; one containing all mappings within the same subset S_i for some i and another containing all mappings across subsets. We denote these subsequences by $\{\delta_t\}_{t_{(1)}}$ and $\{\delta_t\}_{t_{(2)}}$ respectively.

By Lemma 1, the first subsequence consists of purely contraction mappings and therefore converges to zero. Similarly, Lemma 2 states that all periodic behavior across subsets must be a contraction, thereby implying that the second subsequence also converges to zero. As both subsequences contain all elements of $\{\delta_t\}_t$ and converge to zero, it follows that $\{\delta_t\}_t$ also converges to zero.

We note that the aforementioned results also hold when either the exact or approximate gradient as well as when multiple unseen samples $\{(X_i, y_i) : i = 1, \ldots, N\}$ are employed (in the case of mini-batch updates).

5.1.5 Related work

Regularized methods have established themselves as popular and effective tools through which to handle high-dimensional data [77]. Such methods employ regularization penalties as a mechanism through which to constraint the set of candidate solutions, often with the goal of enforcing specific properties such as parsimony. In particular, ℓ_1 regularization is widely employed as a convex approximation to the combinatorial problem of model selection. As a result of the convex nature of ℓ_1 penalties, efficient and highly scalable estimation algorithms can be derived [9].

However, the introduction of an ℓ_1 penalty requires the specification of the associated regularization parameter. The task of tuning such a parameter has primarily been studied in the context of non-streaming, stationary data. Stability selection procedures, introduced by [118], effectively look to by-pass the selection of a specific regularization parameter by instead fitting multiple models across sub-sampled data. Variables are subsequently selected according to the proportion of all models in which they are present. In this manner, stability selection is able to provide important theoretical guarantees while incurring an additional computational burden. Other popular approaches involve the use of cross-validation [67] or information theoretic techniques [166]. However, such methods have been reported to perform poorly in high-dimensional settings [102, 107, 176] and cannot easily be adapted to handle streaming data.

Online learning with the ℓ_1 constraints has also been studied extensively and many computationally efficient algorithms are available. A stochastic gradient descent algorithm is proposed by [23]. More generally, online learning of regularized objective functions has been studied extensively by [52] who propose a general class of computationally efficient methods based on proximal gradient descent. The aforementioned methods all constitute important advances in the study of sparse online learning algorithms. However, a fundamental issue that has been overlooked corresponds to the selection of the regularization parameters. As such, current methodologies are rooted on the assumption that the regularization parameter remains fixed. It follows that the regularization parameter may itself vary over time, yet selecting such a parameter in a principled manner is non-trivial. The focus of this work is to present and validate a framework through which to automatically select and

update the regularization parameter in real-time. The framework presented in this work is therefore complementary and can be employed in conjunction with many of the preceding techniques.

More generally, the automatic selection of hyper-parameters has recently become an active topic in machine learning [154]. Interest in this topic has been catalyzed by the success of deep learning algorithms, which typically involve many such hyper-parameters. Sequential model based optimization (SMBO) methods such as Bayesian optimization employ a probabilistic surrogate to model the generalization performance of learning algorithms as samples from a Gaussian process [154], leading to expert level performance in many cases. It follows that such methods may be employed to tune regularization parameters in the context of penalized linear regression models. However, there are several important differences between the SMBO framework and the proposed framework. The most significant difference relates to the fact that the proposed framework employs gradient information in order to tune the regularization parameter while SMBO methods such as Bayesian optimization are rooted in the use of a probabilistic surrogate model. This allows the SMBO framework to be applied in a wide range of settings while the proposed framework focuses exclusively on lasso regression models. However, as we describe in this work, the use of gradient information makes the RAP framework ideally suited in the context of non-stationary, streaming data. This is in contrast to SMBO techniques, which typically assume the data is stationary.

5.2 Empirical results

In this section we look to empirically demonstrate the capabilities of the proposed framework by studying a variety of real and simulated datasets. In order to provide a flavor for the capabilities of the RAP algorithm, we being by studying the diabetes dataset in Section 5.2.1. This corresponds to a publicly available dataset which has been widely studied in the context of lasso models [53]. We complement these results with a more extensive simulation study presented in Section 6.2.

5.2.1 Diabetes dataset

We begin by considering the diabetes dataset presented in [53]. Here the response is a quantitative measure of disease progression. The covariates associated with each observation correspond to four measurements over baseline variables such as average blood pressure and body mass index as well as six blood serum measurements.

The objective of this example is to provide empirical evidence that the RAP algorithm is able to reliably track the regularization parameter. As such, cross-validation with $K = 10$ folds was employed in order to estimate the regularization parameter. The regularization parameter was subsequently estimated using the RAP framework in a streaming fashion. The RAP algorithm was applied over $N = 500$ iterations. At each iteration, the dataset rows were randomly permuted such that the order in which observations arrived varied. Due to the stationary nature of the data, dynamically tracking regression coefficients was not of interest here. As such, the sample covariance was recursively estimated using a fixed forgetting factor of $r = 1$. This corresponds to an online analysis where information from past observations is not discarded. For each iteration, the proposed method was initialized with $\beta = \mathbf{0} \in \mathbb{R}^p$ Both the exact and approximate gradient updates were considered. Finally, a SMBO approach, in the form of Bayesian optimization, was also considered. This involved modeling the generalization performance of the penalized regression model as a Gaussian process with a square exponential covariance function. The expected improvement acquisition function was employed to search the parameter space for λ [154].

The results are shown in Figure [5.1], where the ℓ_1 norm* is plotted against the number of iterations of the RAP algorithm. The horizontal red and blue lines indicates the regularization parameter as selected by 10-fold cross-validation and SMBO respectively. The dashed lines represent the mean ℓ_1 norm selected over $N = 500$ permutations when the exact (black) or approximate (brown) gradients where employed. We note that in both cases the estimated ℓ_1 norm quickly increases away from zero and converges to the cross-validated norm.

*The ℓ_1 norm was considered as opposed to the estimated sparsity parameter in order to avoid potential confusion arising due to scaling of regularization parameters and other idiosyncrasies. There is a one-to-one relationship between the sparsity parameter, λ, and the ℓ_1 norm.

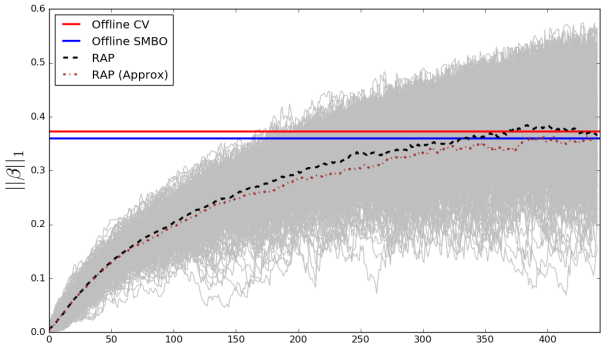


Figure 5.1: Mean ℓ_1 norm over all permutations for exact and approximate RAP algorithms is plotted in the dashed black and brown lines respectively. Each grey line corresponds to the ℓ_1 norm over each of the $N = 500$ iterations when applying RAP algorithm with exact gradient updates. The solid, horizontal lines correspond to the regularization parameters selected via two offline methods: cross-validation and Bayesian optimization.

5.2.2 Simulation study

In this section we look to compliment the results presented in Section 5.2.1 with a more extensive set of simulations. We begin by considering the performance of the RAP algorithm in the context of stationary data. This simulation serves to demonstrate that the proposed method is capable of accurately tracking the regularization parameter. We then study the performance of RAP algorithm in the context of non-stationary data. Throughout this simulation study the RAP algorithm is benchmarked against two offline methodologies: cross-validation and SMBO. In the context of SMBO methods, we study the performance against Bayesian optimization methods. Here a Gaussian process with a square exponential kernel was employed as a surrogate model together with the expected improvement acquisition function.

Simulation settings

In order to thoroughly test the performance of the RAP algorithm, we look to generate synthetic data were we are able to control both the underlying structure as well as the dimensionality of the data. In this work, data was generated according to a multivariate Gaussian distribution with a block covariance structure. This introduced significant correlations across covariates, thereby increasing the difficultly of the regression task [190]. Formally,

the data simulation process followed that described by [116]. This involved sampling each covariate as follows:

$$X_t \sim \mathcal{N}(0, \Sigma),$$

where $\Sigma \in \mathbb{R}^{p \times p}$ is a block diagonal matrix consisting of five equally sized blocks. Within each block, the off-diagonal entries were fixed at 0.8, while the diagonal entries were fixed to be one. Having generated covariates, X_t, a sparse vector of regression coefficients, β, was simulated. This involved randomly selecting a proportion, ρ, of coefficients and randomly generating their values according to a standard Gaussian distribution. All remaining coefficients were set to zero. Finally, univariate responses were obtained as follows:

$$y_t \sim \mathcal{N}(X_t\beta, 1).$$

In this manner, it is possible to generate piece-wise stationary data, $\{(y_t, X_t) : t = 1, \ldots, T\}$. When studying the performance of the RAP algorithm in the context of stationary data, it sufficed to simulate one such dataset. In order to quantify performance in the context of non-stationary data, we concatenate multiple piece-wise stationary datasets. This results in datasets with abrupt changes. We note that in the non-stationary setting the block structure was randomly permuted at each iteration in order to avoid covariates constantly sharing the same set of highly correlated variables.

Performance metrics

In order to asses the performance of the RAP algorithm we consider various performance metrics. In the context of stationary data, our primary objective is to demonstrate that the proposed method is capable of tracking the regularization parameter when benchmarked against traditional methods such as cross-validation. As a result, we consider the difference in ℓ_1 norms of the regression model estimated by each algorithm. This is defined as:

$$\Delta = ||\beta(\lambda^{CV})||_1 - ||\beta(\lambda^{RAP})||_1, \tag{5.18}$$

where we write λ^{CV} and λ^{RAP} to denote the regularization parameters selected by cross-validation and RAP algorithms respectively. We choose to employ the ℓ_1 norm (as opposed to directly considering the sparsity parameter, λ) as there is a one-to-one relationship between λ and the ℓ_1 norm with the added benefit that the ℓ_1 norm is not affected by arbitrary changes to the data (e.g., scaling observations).

In the context of non-stationary data we are interested in two additional metrics. The first corresponds to the residual error over unseen observations, C_{t+1}, initially defined in equation (5.2). Secondly, we also consider the correct recovery of the sparse support of β_t. In this context, we treat the recovery of the support of β_t as a binary classification problem and quantify the performance using the F score; defined as the harmonic mean between the precision and recall of a classification algorithm [169].

Stationary data

While the results presented in Section 5.2.1 provide reassuring empirical evidence, we consider a more extensive simulation study here. In particular, we study the performance of the RAP algorithm as the dimensionality of regression coefficients, p, increases. The goal of this simulation therefore is to demonstrate that proposed algorithm is able to accurately track the regularization parameter.

Data was generated as described previously and the dimensionality of the covariates, X_t, was varied from $p = 10$ through to $p = 100$. For each value of p, $N = 500$ datasets were randomly generated. The regularization parameter was first estimated using $K = 10$ fold cross-validation. The RAP algorithm was subsequently employed and the difference in ℓ_1, defined in equation (5.18), was then computed.

This procedure was repeated to produce data with dimensionality $p = 10$ through to $p = 100$. For each simulated dataset, the regularization parameter was selected via $K = 10$ fold cross-validation in an offline manner (i.e., by considering the entire dataset). The RAP algorithm was subsequently employed to estimate the regularization parameter. This involved iterating through observations in a streaming fashion.

The difference in selected regularization parameters over $N = 500$ simulations is visualized in Figure [5.2]. It is reassuring to note that the differences are both small in

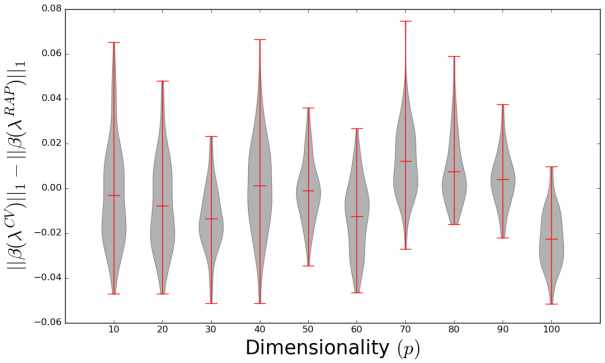


Figure 5.2: Violin plots visualizing the difference in ℓ_1 norms of estimated regression coefficients as a function of the dimensionality, p. It is reassuring to note that the difference is both small in magnitude and centered around the origin, indicating the absence of a large systematic bias.

magnitude as well as centered around the origin. The latter serves to indicate the absence of a large systematic bias. The figure also does not show evidence of any systematic change in the bias as the dimensionality increases.

Non-stationary data

While the previous simulation provided empirical evidence demonstrating that the RAP framework can be effectively employed to track regularization parameters in a stationary setting, we are ultimately interested in streaming, non-stationary datasets. As a result, in this simulation we study the performance of the proposed framework in the context of non-stationary data.

While there are a multitude of methods through which to simulate non-stationary data, in this simulation study we chose to generate data with piece-wise stationary covariance structure. As a result, the underlying covariance alternated between two regimes: a sparse regime where the response was driven by a reduced subset of covariates and a dense regime where the converse was true. Thus, pairs (y_t, X_t) of response and predictors were simulated in a piece-wise stationary regimes. The dimensionality of the covariates was fixed at $p = 20$, implying that $X_t \in \mathbb{R}^{20}$. Changes occurred abruptly every 100 observations and two change-points were considered, resulting in 300 observations in total.

Covariates, X_t, were simulated with two alternating regimes; dense and sparse. The

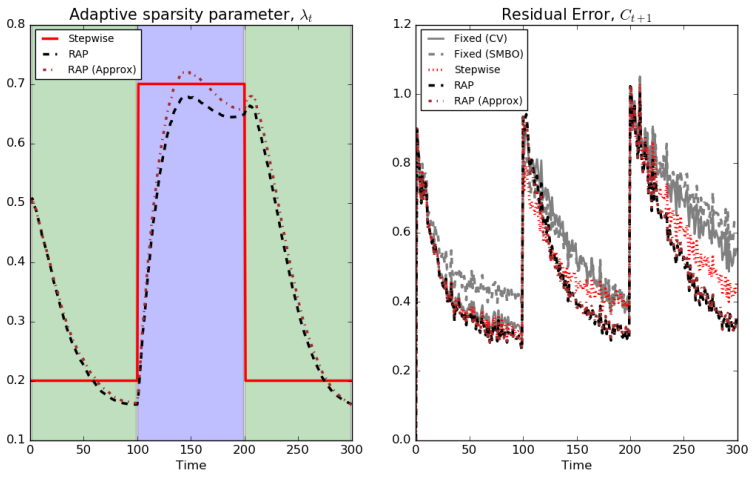


Figure 5.3: *Left*: Mean estimates of the regularization parameter are shown for the RAP algorithm as well as a the optimal piece-wise constant value selected by cross-validation. The background color indicates the nature of the underlying regime (green indicating dense, blue sparse). *Right*: Residual error, C_{t+1}, is plotted as a function of time. We note that the RAP algorithms outperforms the offline approaches employed.

block-covariance structure remained fix within each regime (i.e., for 100 observations). Within the dense regime, a proportion $\rho_1 = 0.8$ of regression coefficients were randomly selected and their values sampled from a standard Gaussian distribution. All remaining coefficients were set to zero. Similarly, in the case of the sparse regime, $\rho_2 = 0.2$ regression coefficients were randomly selected with remaining coefficients set to zero. The regression coefficients remained fixed within each regime.

In order to benchmark the performance of the proposed RAP framework, streaming penalized lasso models were also estimated using a fixed and piece-wise constant sparsity parameters. As a result, the RAP algorithm was benchmarked against three distinct offline methods for selecting the regularization parameter. In the case of a fixed sparsity parameter, $K = 10$ fold cross-validation as well as Bayesian optimization were employed. Finally, cross-validation was also employed to learn a piece-wise constant regularization parameter. This was achieved by performing cross-validation for the data within each regime. For each of these methods, their offline nature dictated that the entire dataset should be analyzed simultaneously (as opposed to in a streaming fashion by the RAP algorithm). As such, they serve to provide a benchmark but would infeasible in the context of streaming data.

Algorithm	$\bar{C}_t$	$\bar{F}_t$
Fixed (CV)	0.58 (0.05)	0.49 (0.05)
Fixed (SMBO)	0.63 (0.05)	0.50 (0.07)
Piecewise	0.51 (0.04)	0.56 (0.04)
RAP	**0.47** (0.04)	**0.64** (0.06)
RAP (Approx)	**0.48** (0.05)	**0.63** (0.07)

Table 5.1: Detailed results consisting of the mean negative log-likelihood, $\bar{C}_t$, as well as the mean F-score, $\bar{F}_t$. Standard errors are provided in brackets.

Results are shown in Figure [5.3], where the left panel shows the estimated regularization parameter as a function of time. These results provide further evidence to indicate that the RAP algorithm is able to reliably track the piece-wise constant parameters selected by cross-validation. As expected, there is some lag directly after each change occurs, however, the estimated regression parameters quickly adapt. The right panel of Figure [5.3] shows the mean residual error, C_{t+1}, for unseen data. We note there are abrupt spikes every 100 observations, corresponding to the abrupt changes in the underlying dependence structure. Detailed results are provided in Table 5.1. We note that the proposed framework is able to outperform the alternative offline approaches. In the case of the offline cross-validation and SMBO, this is to be expected as a fixed choice of regularization parameter is misspecified.

5.3 Application

In this section we present an application of the RAP algorithm to task-based fMRI data. This data corresponds to time-series measurements of blood oxygenation, a proxy for neuronal activity, taken across a set of spatially remote brain regions [105]. Our objective in this work is to quantify pairwise statistical dependencies across brain regions, typically referred to as functional connectivity within the neuroimaging literature [157].

While traditional analysis of functional connectivity was rooted on the assumption of stationarity, there is growing evidence to suggest this is not the case [90]. This particularly true in the context of task-based fMRI studies. Several methodologies have been proposed to address the non-stationary nature of fMRI data [3, 122], many of which are premised on the use of penalized regression models such as those studied in this work. While such

methods have made important progress in the study of non-stationary connectivity networks, they have typically employed fixed regularization parameters. This is difficult to justify in the context of non-stationary data and plausible biological justifications are not readily available. The RAP algorithm is therefore ideally suited to both accurately estimating non-stationary connectivity structure as well as providing insight regarding whether the assumption of a fixed sparsity parameter is reasonable.

5.3.1 HCP Emotion task Data

Emotion task data from the Human Connectome Project (HCP) was studied with 20 subjects selected. During the task participants were presented with blocks of trials that either required them to decide which of two faces presented on the bottom of the screen match the face at the top of the screen, or which of two shapes presented at the bottom of the screen match the shape at the top of the screen. The faces had either an angry or fearful expression while the shapes represented the emotionally neutral condition. Preprocessing involved regression of Fristons 24 motion parameters from the fMRI data. Sixty-eight cortical and 16 subcortical ROIs were derived from the Desikan-Killiany atlas and the ASEG atlas, respectively. Mean BOLD time series for each of these 84 ROIs were extracted and further cleaned by regressing out time series sampled from white matter and cerebrospinal fluid. Finally, the extracted time courses were high-pass filtering using a cut-off frequency of $\frac{1}{130}$ Hz. Neurosynth, a platform for large-scale automated synthesis of neuroimaging data,was employed to reduce the number of regions studied [185]. This provided an automatically generated forward inference map based on 790 studies quantifying the activation all regions in emotion studies. Twenty regions identified as core emotion hubs were selected. Data for each subject therefore consisted of $n = 175$ observations across $p = 20$ nodes.

5.3.2 Results

Data for each subject was analyzed independently and the time-varying estimates of the conditional dependence structure where obtained for each node. A fixed forgetting factor of $r = .95$ was employed throughout with a stepsize parameter $\eta = .025$. The exact gradient was employed when updating the sparsity parameter at each iteration. In order to

avoid unreliable initial performance of the algorithms a burn-in of twenty observations was employed. Finally a time-varying estimate of the functional connectivity was employed by applying neighborhood selection, introduced in Section 2.2.2.

The mean sparsity parameter over all subjects is shown in the top panel Figure [5.4]. We observe decreased sparsity parameters for blocks in which subjects were presented with emotional (i.e., angry or fearful) faces (top panel, purple shaded areas) as compared to blocks in which subjects were shown neutral shapes (top panel, green shaded areas). The oscillation in sparsity parameter is highly correlated with task onset. When inspecting the networks estimated using the time varying sparsity parameter (bottom panel), we find strong coupling amongst many of the regions during the emotion processing blocks (A and C) compared to a clearly sparser network representation for blocks that require no emotion processing (i.e., neutral shapes, block B). This is to be expected as the selected regions are core hubs involved with emotion processing; therefore explaining the higher network activity during the emotion task when compared to the neutral task

5.4 Conclusion

In this work we have presented a novel framework through which to learn a time-varying sparsity parameters in the context of streaming lasso models. An approximate algorithm is also provided to address issues concerning computational efficiency; a factor of paramount importance in the context of high-dimensional data. We provide theoretical results regarding the convergence in a non-stochastic scenario. These results hold for both the exact and approximate gradient algorithms as well as in the context of mini-batch updates. Finally, empirical evidence is provided to validate the proposed algorithm.

We present two simulation studies which demonstrate the capabilities of the proposed method. These simulations demonstrate that the proposed RAP framework is capable of tracking the regularization parameter both in a stationary as well as non-stationary context. Finally, we present an application to fMRI data, which is widely accepted to be non-stationary.

Future work will involve extending the RAP framework to consider alternative regularization schemes. In particular, the popular ridge or ℓ_2 penalty could be incorporated as the

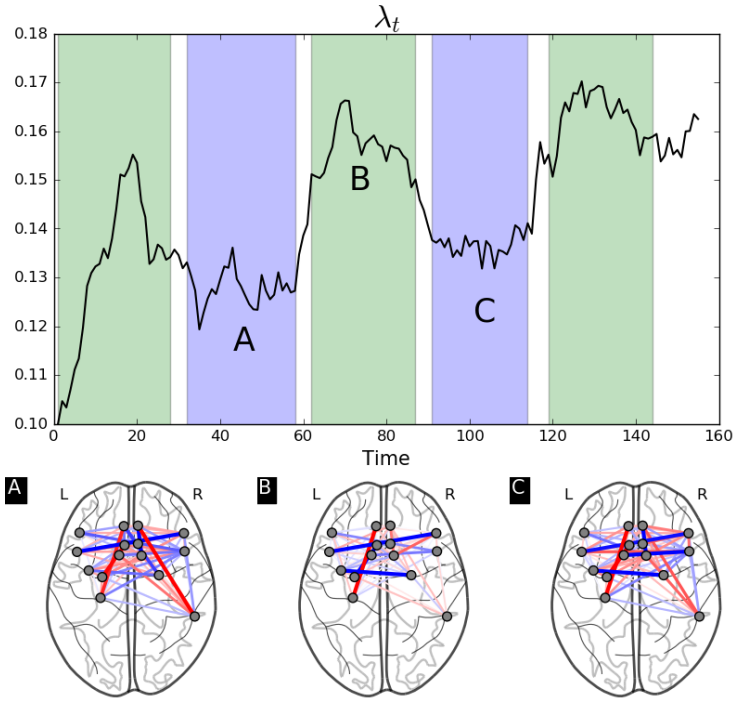


Figure 5.4: *Top*: the mean sparsity parameter is shown as a function of time. The background color indicates the nature of the task at hand (green indicates neutral task while blue indicates the emotion task). *Bottom*: estimated networks visualizing the estimated connectivity structure at three distinct points in time. Edge colors indicate the nature of the dependence (blue indicates a positive dependence, red a negative dependence).

derivative, $\frac{\partial \beta}{\partial \lambda}$, is also available in closed form. Furthermore, it would also be possible to extend the framework to consider a wider range of models. In this setting, the results of [146] could be leveraged to consider time-varying sparsity for generalized linear models.

Chapter 6

Linear graph embedding methods for dynamic networks

The focus of previous chapters has been to propose novel algorithms through which to accurately infer sparse covariance structure in time-varying GGMs. Such methods are relevant in a wide range of applications. In particular, within the context of neuroscience they may be used to study the dynamic properties of functional connectivity networks. This corresponds to an exciting avenue of modern neuroscientific research [31]. As a result, the methods presented in Chapters 3 to 5 form part of a larger ecosystem of methods dedicated to the study of time-varying connectivity networks [90]. These methods have provided unprecedented insights relating to the dynamic restructuring and temporal evolution of the human connectome and may potentially provide important insights relating to various neurological and psychiatric conditions [39, 44, 160].

However, obtaining robust and easily interpretable insights from the results of such algorithms raises novel statistical challenges. Difficulties arise both due to the need to summarize high-dimensional graphs intuitively as well as the need to compare multiple estimated graphs. Such difficulties are further exacerbated by the fact that often a distinct network is estimated at each observation and potentially across many subjects, resulting in a large number of estimated networks. One potential solution involves testing for statistical correlations across the estimated edge structure and underlying changes in cognitive task, thereby recovering the set of edges which are functionally modulated by a given task. Such

an approach was taken in Chapter 4 and is widely advocated [90, 123, 184]. However, it fails to account for the structured nature of networks. Crucially, by studying edges on an individual basis such methods fundamentally ignore the notion that the brain is a functionally connected network [28, 162]. A related approach involves the use of clustering methods: for example [3] employ k-means clustering on estimates of time-varying covariance matrices. Such methods are able to identify *state* networks which may capture the current connectivity structure at specific points in time. However, clustering based methods require the definition of a distance metric which is difficult to define in the context of graphs [144]. Finally, time-varying graph metrics may also be employed, where metrics such as the degree or betweenness centrality are tracked over time. This was the approach taken in Chapter 3. However, it if often difficult to know *a priori* which metrics to consider and there is no guarantee that predefined metrics will necessarily capture all the relevant changes in connectivity structure.

In this chapter, we look to address the challenges associated with interpreting time-varying, high-dimensional networks via the use of linear graph embedding methods. Generally speaking, the objective of graph embedding techniques is to map estimated graphs into a (potentially low-dimensional) vector space [183]. This facilitates tasks such as visualization and classification by translating the problem from the graph domain into a Euclidean space, where traditional classification and visualization techniques can be readily applied.

While a wide range of graph embedding techniques may be employed, in this work we limit ourselves to consider only methods based on linear projections over the edge structure of an estimated graph. This allows us to obtain a clear interpretation of the embedding in the context of functional connectivity. Moreover, the proposed embeddings are based on the Laplacian of the estimated graphs. This serves to normalize the estimated edge weights. As a result, we consider two distinct graph embedding algorithms. The first embedding considered is based on Principal Component Analysis (PCA). This embedding, which is closely related to the work of [103], can be interpreted as mapping graphs into a low-dimensional vector space that captures the maximal variability. Due to the unsupervised nature of this embedding, it is ideally suited for the study of both resting-state as well as task-based fMRI data. The second approach is based on regularized Linear Discrim-

inant Analysis (LDA). This method serves to recover a low-dimensional embedding that maximizes the discriminatory power across various tasks or states. The supervised nature of such an embedding is particularly suitable for task-based experiments, where changes in cognitive task are known and the objective is to recover the associated changes in the connectivity structure.

The remainder of this chapter is organized as follows: linear graph embedding techniques based on principal component and linear discriminant analysis are presented in Section 6.1. A simulation study is presented in Section 6.2. Finally, in Section 6.3 we present an application of the proposed graph embedding methods to working memory task data taken from the Human Connectome Project (HCP).

6.1 Linear embedding methods

Throughout this section it is assumed that estimates of time-varying functional connectivity networks have been obtained across a cohort of S subjects. Recall that we write $\Theta_i^{(s)} \in \mathbb{R}^{p\times p}$ to denote the estimated precision for the sth subject at the ith observation. Each $\Theta_i^{(s)}$ therefore captures the statistical dependencies across p regions of interest (ROIs) at the ith observations and may be subsequently interpreted as a encoding the functional dependencies across such ROIs. The dynamic properties of functional connectivity networks can be quantified in many ways. One popular method for estimating such networks involves the use of sliding windows, discussed in Chapter 4. Alternative methods, based on approaches such as change-point detection [38] and forgetting factors have also been proposed [123].

In this work our objective is to understand dynamic functional connectivity networks using linear graph embedding methods. Such methods allow for the representation of graphs or networks in real-valued vector spaces, resulting in two advantages. First, by embedding graphs in a Euclidean vector spaces we are able to employ traditional visualization and classification techniques. Second, by focusing on linear projections over the set of all edges we are able to directly interpret the embeddings in the context of functional connectivity networks. The linear embedding methods considered in this work are based on PCA and regularized LDA. Such methods correspond to unsupervised and supervised

learning algorithms respectively, indicating the they may be used in conjunction to further understand dynamic connectivity networks.

The remainder of this section is organized as follows: we introduce and discuss graph Laplacians in Section 6.1.1. In Sections 6.1.2 and 6.1.3 we introduce two distinct graph embedding methods.

6.1.1 Graph Laplacians

The graph embedding techniques described in this work are based on the Laplacian of each estimated functional connectivity network, formally defined as:

$$L_i^{(s)} = \left(D_i^{(s)}\right)^{-\frac{1}{2}} \left(D_i^{(s)} - \Theta_i^{(s)}\right) \left(D_i^{(s)}\right)^{-\frac{1}{2}}, \tag{6.1}$$

where $D_i^{(s)}$ is a diagonal matrix containing the diagonal elements of $\Theta_i^{(s)}$. The use of a Laplacian is desirable as it serves to normalize the estimated edge weights (corresponding to the off-diagonal entries) of estimated networks. It therefore follows that each Laplacian matrix, $L_i^{(s)}$, is fully characterized by its upper-triangular entries. We define the set of Laplacian matrices for a given subject to be $L^{(s)} = \{L_i^{(s)} : i = 1 \ldots, n\}$. In the remaining sections, we employ $L^{(s)}$ directly as input to the proposed graph embedding algorithms. We define

$$\text{vec}(L^{(s)}) = \mathbb{R}^{n \times \binom{p}{2}} \tag{6.2}$$

as a matrix where the ith row corresponds to the vectorized upper-triangular entries of the Laplacian at the ith observation. The matrix, L, consisting of all vectorized Laplacians across all subjects can subsequently be defined as:

$$L = \left[\text{vec}(L^{(1)})^T, \ldots, \text{vec}(L^{(S)})^T\right]^T \in \mathbb{R}^{Sn \times \binom{p}{2}}. \tag{6.3}$$

This process is described in Figure [6.1]. It follows that each column of L corresponds directly to one of the $\binom{p}{2}$ possible edges. As both embeddings studied here consist of linear projections of L onto lower-dimensional subspaces, they can each be understood as a a linear combination of edges and interpreted as functional connectivity networks.

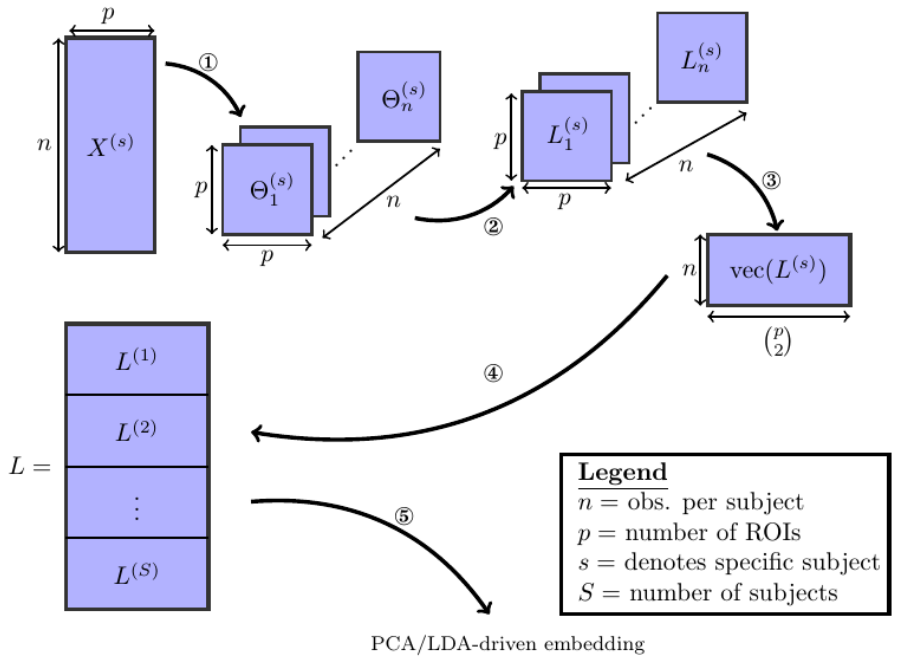


Figure 6.1: The various steps involved in the proposed embedding method are visualized: 1) the SINGLE algorithm is used to obtain estimates of time-varying precision matrices. 2) The precision matrices are transformed to Laplacian matrices. 3) The Laplacian matrices are vectorized by taking their upper-triangular components. 4) The vectorized Laplacians of all subjects are stacked vertically. 5) Finally the PCA/LDA-driven embeddings are estimated.

6.1.2 Unsupervised PCA-driven embedding

Here we look to obtain a low-dimensional embedding that maximizes the amount of explained variance. Following from the method described in [103], we look to achieve this by applying PCA to L. Recall that L is a matrix consisting of vectorized graph Laplacians across all S subjects. In particular, each row of L corresponds to a specific edge. This will yield the linear combination of edges that best summarize the variability in functional connectivity networks over time.

Formally, PCA is an unsupervised dimensionality reduction technique which produces a new set of uncorrelated variables from the original data. This is achieved by considering the k leading eigenvectors of the covariance matrix $L^T L$, defined as the principal components $P_k \in \mathbb{R}^{k \times \binom{p}{2}}$. The principal components, P_k, can be studied in two ways. First, by

considering the entries of each principal component we are able to quantify the contribution of the corresponding edges. Edges which vary highly within a dataset can therefore be expected to provide a large contribution to the leading principal components. The set of these edges can subsequently be interpreted as pertaining to a network which is strongly modulated by underlying cognitive task. Second, the embedding produced by P_k is obtained as:

$$P_k \cdot \text{vec}(L^{(s)}) \in \mathbb{R}^{k \times n}. \tag{6.4}$$

This yields a k-dimensional graph embedding for each subject at each of the n observations. This serves as a low-dimensional representation of the time-varying networks which can be employed in tasks such as classification or visualization.

6.1.3 Supervised LDA-driven embedding

While the PCA-driven embedding was motivated by understanding the components of functional connectivity which demonstrated the greatest variability, we may also be interested in understanding which components of the functional networks are most discriminative across multiple tasks. To this end, a supervised learning approach is taken here.

We propose the use of LDA to learn the functional connectivity networks which are most discriminative across tasks. LDA is a simple and robust classification algorithm which can also be interpreted as a linear projection. As a result, LDA reports the linear combination of edges which are most discriminative between tasks. These can subsequently be interpreted as a discriminative embedding which reports changes in functional connectivity induced by a given task.

In high-dimensional supervised learning problems, such as the one considered in this work, it is of paramount importance to avoid overfitting. Two popular methods to guard against overfitting involve the introduction of regularization, thereby penalizing overly complex models which are more susceptible to overfitting, and cross-validation. Here a combination of both approaches is employed. First a variable screening procedure is applied, reducing the number of candidate variables (in our case edges) to $p' << \binom{p}{2}$. This serves to greatly reduce the risk of overfitting as well as yield a sparse embedding which is easily interpretable. The remaining p' selected edges are subsequently used to train an

LDA classifier. Such a classifier will learn the linear projection of selected edges which is most discriminative across tasks. This projection will serve as our LDA-driven embedding.

The screening method employed in this work selected the most reproducible edges across all S subjects. This was achieved by fitting an independent LDA classifier for the data of each subject. Due to the limited observations per subject, regularization was introduced in the form of an l_1 penalty. As a result, an l_1 penalized LDA model was estimated for each subject. Such models can be estimated efficiently as described in [35] and provide the additional benefit of performing variable selection. As discussed on Chapter 5, the choice of regularization parameter will play a fundamental role in the variable selection procedure and must therefore be carefully tuned. The gradient-based methods for selecting the regularization parameter discussed in Chapter 5 are not easily extended to the context of LDA. As a result, a distinct regularization parameter was selected for each subject via cross-validation. A regularized LDA model was then estimated for each subject and the active variables were noted. In this manner, variables which were consistently active across all subjects where retained while all others were discarded.

Such a screening approach is analogous to performing stability selection, as described in [118], where the sub-sampling is performed by studying each subject independently. This serves to discard a large number of noisy and non-informative variables, yielding a Laplacian matrix, $L' \in \mathbb{R}^{S \cdot n \times p'}$, consisting of only selected variables which have demonstrated reproducible discriminative power across all subjects. In practice, a threshold $\rho \in [0, 1]$ is proposed and all edges which are active in at least $\rho\%$ of subjects are included in the final model. The screening procedure is summarized in Algorithm 3.

6.2 Simulation study

In this section we provide empirical evidence to demonstrate the capabilities of the two graph embeddings methods introduced in Section 6.1. Throughout these simulations, we produce simulated time series data giving rise to a number of connectivity patterns which reflect those reported in real fMRI data. The data is generated such that the underlying connectivity varies over time and the SINGLE algorithm, introduced in Chapter 3, was subsequently employed to obtain estimates of time-varying connectivity networks. While

Algorithm 3: Screening procedure for sparse LDA-driven embedding

Input: Threshold $\rho \in [0, 1]$

begin

  for *each subject* $s \in \{1, \ldots, S\}$ **do**

    - Perform 10-fold cross-validation to select regularization parameter ;

    - Estimate a regularized LDA model using subject specific data (i.e., vec$(L^{(s)})$);

    - Selected variables (i.e., edges) stored;

  - All edges present in at least $\rho\%$ of subjects are retained to obtain the screened Laplacian matrix L';

return L'

the SINGLE algorithm was employed in this work, it follows that any alternative algorithm could also have been used. The objective of this simulation is therefore to quantify how reliably the proposed graph embedding algorithms are able to capture changes in connectivity structure.

6.2.1 Simulation settings

In order to thoroughly test the capabilities of the proposed graph embedding algorithms, we follow the simulation study described in previous chapters. This involved the use of vector autoregressive (VAR) processes to generate autocorrelated, multivariate time-series. The use of VAR models allowed for the encoding of both autocorrelations within components as well as cross-correlations across nodes. Furthermore, we validate the performance of each graph embedding method using three distinct random graph algorithms: Erdős-Rényi random graphs [56], scale-free random graphs obtained by using the preferential attachment model of Barabási and Albert [11] and small-world random graphs obtained using the Watts-Strogatz model [177]. Each of these random graph algorithms is discussed in detail in Appendix C.

Following from the simulations detailed in Chapter 3, we fix the edge strength between nodes to be 0.6 in the case of Erdős-Rényi random networks. In the case of the scale-free and small-world networks we randomly sample the edge strengths uniformly from $[-1/2, -1/4] \cup [1/4, 1/2]$. This serves to introduce additional variability and further increase the difficulty of the task at hand.

In many task based studies, subjects are required to alternate between performing a cognitive task and resting in a cyclic fashion. As such, the simulations presented in the work consist of a cyclic connectivity structure, where the underlying connectivity varies between two simulated networks. As a result, multivariate, simulated data was generated where the underlying covariance structure alternated in a cyclic fashion. Three distinct network structures were considered: Erdős-Rényi, scale-free and small-world networks. Furthermore, networks were simulated with $p = 10, 25$ and 50 nodes respectively while the number of observations within each segment remained fixed at $n = 100$. This allows for the study of the behavior the proposed of graph embedding techniques as the ratio n/p decreases. Throughout this simulation, $S = 20$ datasets where independently simulated as described above.

6.2.2 Performance metrics

In order to evaluate the empirical performance of the graph embedding methods we consider the discriminatory power of the estimated embeddings when predicting the underlying covariance structure. As the underlying covariance structure is simulated to alternate between two network structures, this corresponds to binary classification task and traditional classification scores, such as the area under the ROC curve (AUC), can be employed [97]. The nature of the AUC score implies that the embedding scores obtained from the proposed methods can be directly employed. In the case of the PCA-driven embedding, the leading principal component score was considered while in the case of the LDA-driven embedding the discriminant scores where employed.

6.2.3 Results

Data was simulated as described in Section 6.2.1. The SINGLE algorithm, introduced in Chapter 3, was subsequently applied in order to estimate time-varying functional connectivity networks for each subject. The application of the SINGLE algorithm required the specification of three hyper-parameters which were selected as follows: the kernel width parameter was estimated once across all subjects using cross-validation. The remaining regularization parameters were selected by minimizing AIC on a subject-by-subject basis.

Given the estimated networks, the two graph embedding methods introduced in Section 6.1 were applied. Half of the $S = 20$ subjects were selected as a training sample, and the networks for the remaining subjects were kept as a validation set.

PCA-driven embeddings

We begin by studying the performance of the PCA-driven embeddings. Recall that the objective of this method is to obtain a low-dimensional embedding which maximizes the amount of explained variance. Figure [6.2] provides an initial flavor for the capabilities of the proposed method where the embedding based on the leading principal component has been visualized over unseen subjects. Recall that the underlying covariance structure has been simulated in a cyclic fashion such that the first and third segments share the same connectivity structure. Change points are denoted by dashed, vertical lines. The results presented in Figure [6.2] therefore indicate that the proposed embedding has accurately captured the cyclic variations which occur in the underlying connectivity structure. Moreover, as these results corresponds to the mean embedding over unseen subjects, this serves as an indication that the estimated PCA-driven embedding is robust and reproducible on unseen data.

In order to obtain a more comprehensive understanding regarding the performance of the embedding, we consider the predictive power of the embeddings when trying to uncover the underlying covariance structure. In this setting, the underlying covariance structure was treated as a binary variable with two classes: each of which serves to indicate one of the two underlying connectivity regimes. The embedding corresponding to the leading principal component was then employed to discriminate across the class. The AUC score was then employed to obtain a measure of the discriminative capabilities of the embedding [97]. Detailed results are provided in Table [6.1a] where the mean AUC score across all unseen simulated subjects is reported together with the standard deviation. As expected, there is a clear decline in the discriminative capabilities of the embedding as the dimensionality of the network increases. As reported in previous chapters, we note there is a drop in the performance as the underlying network structure changes from Erdős-Rényi to scale-free and finally to small-world.

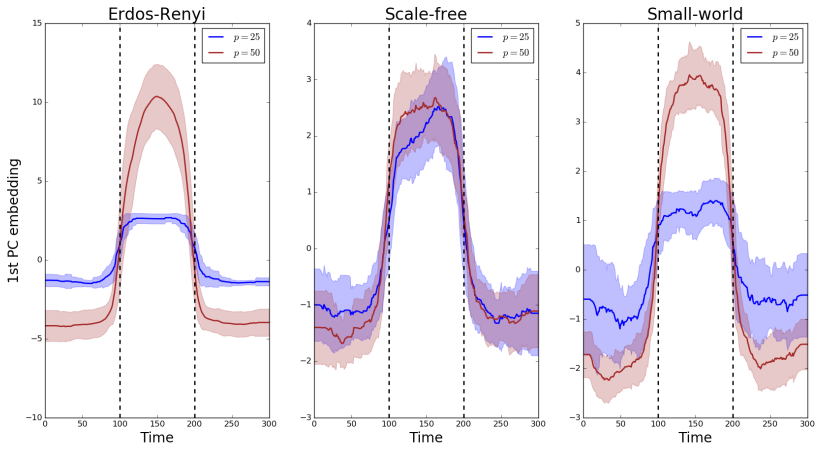


Figure 6.2: An example visualization of the results for the PCA-driven embedding for simulations with $p = 25$ and $p = 50$ nodes (results for $p = 10$ are omitted in the interest of clarity). Each panel shows the mean PCA-driven embedding over 10 unseen, simulated datasets (i.e., 10 unseen subjects). This corresponds to the loading over the leading principal component. Standard deviations are indicated by the shaded regions. Results are shown when the underlying connectivity structure was simulated using three distinct graph algorithms: Erdős-Rényi, scale-free and small-world random graphs. Vertical dashed lines indicate a change in covariance structure.

LDA-driven embeddings

While the PCA-driven embeddings are motivated by the need to understand components of estimated networks which demonstrate the greatest variability, it is also important to consider embeddings which are discriminative across multiple cognitive tasks. The LDA-driven embeddings introduced in Section 6.1.3 are one potential method through which to achieve this. Briefly, the objective of such an embedding is to learn a linear combination of edges which are maximally discriminative across across tasks.

The fundamental difference between the PCA and LDA-driven embeddings is that the latter is a supervised embedding. As a result, it is crucial to avoid any potential overfitting. As described in Section 6.1.3, the proposed method employs a variable screening procedure based on ℓ_1 regularized models. This use of regularization also serves to penalize complex models which are naturally more prone to overfit.

We note that the underlying covariance structure was simulated in a cyclic fashion

p	Erdős-Rényi	Scale-free	Small-world
10	0.94 (0.02)	0.94 (0.04)	0.92 (0.05)
25	0.95 (0.03)	0.88 (0.07)	0.80 (0.08)
50	0.91 (0.03)	0.84 (0.06)	0.79 (0.07)

(a) PCA-driven

p	Erdős-Rényi	Scale-free	Small-world
10	0.97 (0.01)	0.96 (0.05)	0.97 (0.06)
25	0.95 (0.03)	0.93 (0.06)	0.83 (0.07)
50	0.90 (0.04)	0.89 (0.06)	0.78 (0.07)

(b) LDA-driven

Table 6.1: Mean AUC scores for each of the proposed graph embeddings are shown when the underlying covariance structure is simulated using three distinct methods. Results are presented for networks with varying numbers of nodes, p. Standard deviation are provided in brackets.

which alternated between two distinct regimes. As a result, the objective of the proposed embedding is to differentiate between two distinct classes. Due to the properties of LDA, this results in a 1-dimensional embedding [76]. This embedding is visualized in Figure [6.3], which provides an initial demonstration of the capabilities of the LDA-driven embedding.

More comprehensive results are provided in Table [6.1b], where the mean AUC score over unseen datasets is reported. As with the PCA-driven embeddings, we note there is a drop in performance as the number of nodes, p, increases. However, the effect does not appear to be as dramatic in the case of the LDA-driven embedding. We attribute this to the supervised nature of this embedding. Formally, the objective of PCA-driven embedding is to learn a low-dimensional representation which captures maximal variance. A decrease in the ratio n/p leads to a corresponding increase in the variability of estimated networks. This may be partially responsible for the difference in embeddings shown in Figure [6.3] as p increases. On the other hand, the objective of the proposed LDA-driven embedding is to learn a linear combination of edges which is discriminative across multiple classes. As such, the drop in the ratio n/p does not result in significant changes to the magnitude of estimated embeddings.

6.3 Application

In this section we present an application of the proposed graph embedding techniques to task-based fMRI dataset taken from the HCP.

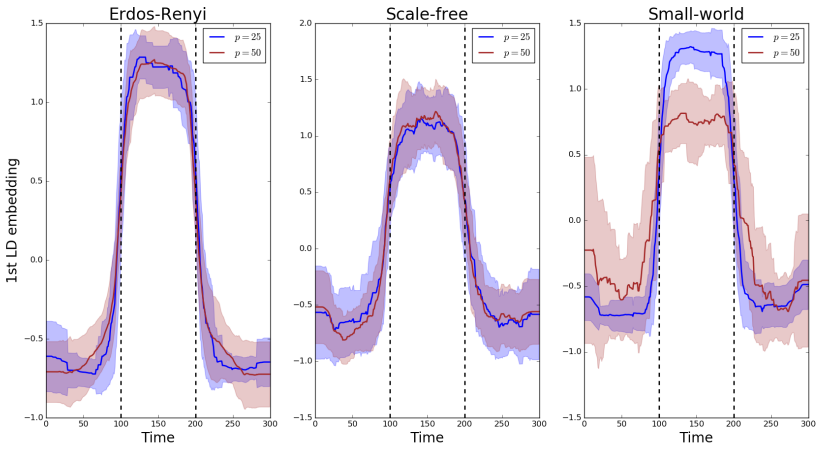


Figure 6.3: An example visualization of the results for the LDA-driven embedding for simulations with $p = 25$ and $p = 50$ nodes (results for $p = 10$ are omitted in the interest of clarity). Each panel shows the mean LDA-driven embedding over 10 unseen, simulated datasets (i.e., 10 unseen subjects). Standard deviations are indicated by the shaded regions. Results are shown when the underlying connectivity structure was simulated using three distinct graph algorithms: Erdős-Rényi, scale-free and small-world random graphs. Vertical dashed lines indicate a change in covariance structure. Vertical dashed lines indicate a change in covariance structure. Embeddings are based on unseen data.

6.3.1 HCP Working Memory task data

The data consisted of working memory task data taken from the Human Connectome Project [55]. During the tasks subjects where presented with blocks of trials consisting of either 0-back or 2-back working memory tasks. Two datasets were provided for each subject, corresponding to a left-right (LR) and right-left (RL) acquisitions. Throughout this work, they were treated as separate scans and studied independently. Data corresponding to $S = 206$ of the possible 500 subjects was selected at random. Thus a total of $2 \times 206 = 412$ datasets where studied.

Data pre-processing

Preprocessing involved regression of Fristons 24 motion parameters from the fMRI data. Sixty-eight cortical and 16 subcortical ROIs were derived from the Desikan-Killiany atlas and the ASEG atlas, respectively. Mean BOLD timeseries for each of these 84 ROIs were

extracted and further cleaned by regressing out timeseries sampled from white matter and cerebrospinal fluid. Finally, the extracted timecourses were high-pass filtered using a cut-off frequency of $\frac{1}{150}$ Hz.

Network estimation

As in the simulation study, time-varying functional connectivity networks were estimated for each subject using the SINGLE algorithm. This required the specification of three hyper-parameters: the width, h, of the Gaussian kernel as well as the regularization parameters, λ_1 and λ_2. A fixed kernel width of $h = 15$, selected via cross-validation, was employed across all subjects. The regularization parameter were selected on a subject-by-subject basis by minimizing AIC. This involved an extensive grid-search over all possible combinations of λ_1 and λ_2. In order to reduce the computational burden associated with selecting λ_1 and λ_2, an initial search was performed on a reduced subset of the subjects. This served to identify a region of the parameter space that was consistently selected across subjects, thereby greatly reducing the computational cost associated with the grid-search performed for each subject.

6.3.2 Results

The estimated functional connectivity networks produced by the SINGLE algorithm were subsequently analyzed using the proposed graph embedding methods.

Recall that the objective of the PCA-driven embedding was to provide a low-dimensional embedding which captures a large portion of the variability present in the data. This was achieved in an unsupervised manner by considering the embeddings associated with the $k = 2$ leading principal components. We note that both the LR and RL acquisitions for each subject where considered simultaneously as the goal was to understand variability across the entire population.

The left panel of Figure [6.4a)] shows the functional connectivity networks associated with each of the two principal component embeddings. Red edges indicate positive associations while blue edges indicate the opposite. The associated functional connectivity networks appear to reflect independent network dynamics. The network associated

with the first principal component displays strong interhemispheric coupling, especially across motor regions but also for other mid-range connections, such as between motor and frontal regions as well as between frontal and medial temporal regions. Decreased inter-hemispheric coherence has previously been linked to poor working working memory performance in patients with traumatic brain injury [98]. In addition, interactions between the medial temporal lobe and frontal areas has been demonstrated for working memory tasks [8]. On the other hand, the network associated with the second principal appears to show increased long-range coupling between frontal and parietal regions in the brain. This is in-line with the well-established engagement of the frontoparietal attention network during working memory tasks [36, 173, 174]. Finally, we note that the ordering of the embeddings is itself significant. In the context of resting state data we would expect the leading embedding to correspond to the DMN. However, from Figure [6.4a)] we note this is not the case, suggesting that the networks recovered are induced by the associated task.

Figure [6.4b)i)] shows the mean PCA-driven embeddings across all $S = 206$ subjects*. The background is colored to denote the task taking place at each point in time: red is used to denote 2-back working memory task while purple denotes a 0-back working memory task and a white background is indicative of rest. The embeddings associated with the first and second leading principal components display a clear oscillatory pattern which appears to be loosely correlated with the underlying task. Due to the unsupervised nature of the PCA-driven embedding, interpreting the root cause driving the embedding is non-trivial (we note the LDA-driven embedding does not share this deficiency). We hypothesize that the oscillatory nature of the PCA-driven embeddings may be the result of several factors: for example it may be associated with oscillations in the brain networks orthogonal to the underlying task. From a methodological perspective it is also important to consider the effect of the initial network estimation algorithms. In the context of this application the SINGLE algorithm was employed and it follows that the associated hyper-parameters will affect the resulting embeddings. An especially relevant hyper-parameter in the case of the SINGLE algorithm is the kernel bandwidth, h, discussed in Section 3.1.3. Finally, we note there is a lag between the 1st and 2nd principal component embeddings, suggesting that

*note that only LR acquisition datasets plotted here, as the task design varied from LR to RL acquisitions.

distinct dynamics in the connectivity structure may be capture by each.

In contrast to the PCA-driven embeddings, the LDA-driven embeddings are supervised methods which seek to identify a reduced subset of edges which are discriminative across tasks. In this section we study the contrast between 0-back and 2-back working memory tasks. As noted previously, two datasets where available for each subject. In such a supervised learning task care was taken to differentiate between the LR and RL acquisition datasets as there were small differences in task-design. The approach taken here was to build an LDA-driven embedding using only the LR acquisition datasets across all subjects and then validate this model using the unseen RL acquisition datasets. All $\binom{p}{2}$ potential edges were screened as described previously and only those selected over 60% of the time were studied. This reduced the number of candidate edges to $p' = 126 << \binom{p}{2}$.

The results for the LDA-driven embedding are shown in Figure [6.4b)ii)]. This corresponds to the results of applying the LDA-driven embedding to the unseen RL acquisitions, averaged across all S subjects. The resulting embedding is strongly correlated with the onset of the 0-back working memory task (denoted by purple shading in the figure). This serves as an empirical validation that the embedding is able to discriminate across the two classes. The discriminative performance of the LDA-driven embedding was subsequently studied on a subject by subject basis by calculating the AUC score over the unseen RL acquisition dataset. The mean AUC score across all subjects was 0.69 with a standard deviation of 0.14, indicating the robustness of the embedding.

We are also able to study the embedding in the context of the associated functional connectivity network, shown in Figure [6.4a)iii)]. While the networks associated with the PCA-driven embedding recovered edges which displayed high variability, the edges reported by the LDA-driven embedding are discriminative across the 0-back and 2-back working memory tasks. We find a distinct patterns that seems to separate between two conditions differing in their cognitive load. In particular, for the 2-back condition (corresponding to a high cognitive load and denoted by red edges) we observe stronger inter-hemispheric coupling across lateral prefrontal cortices. This result is in line with studies reporting that inter-hemispheric coupling improves the ability to perform high cognitive demand tasks [6]. This relationship has also been observed in patients suffering from schizophrenia, where disrupted prefrontal inter-hemispheric coupling was related to poor

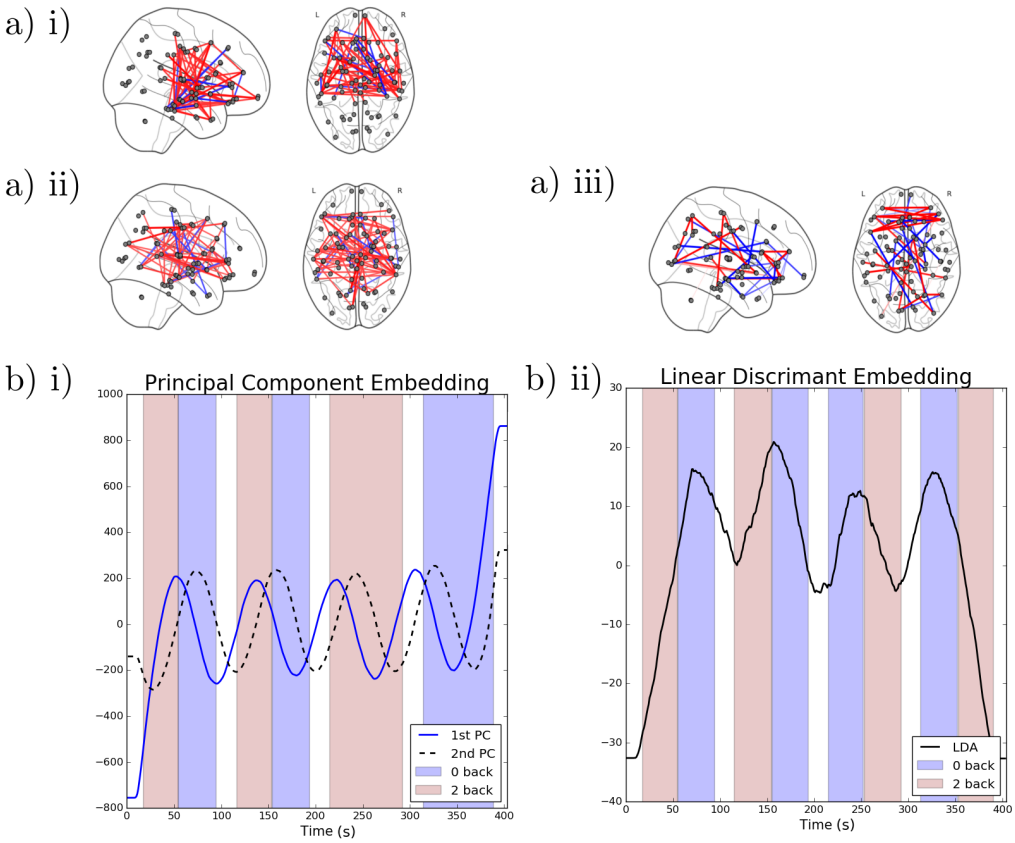


Figure 6.4: Visualization of results when linear graph embedding methods are applied to HCP data. **a)** The brain networks visualize the functional connectivity networks associated with each of the embeddings. The networks shown correspond to the following embeddings: **i)** 1st principal component embedding, **ii)** 2nd principal component embedding and **iii)** the LDA-driven embedding. **b)** Visualizations are provided for the PCA (left) and LDA (right) driven embeddings. The shaded background regions indicate the underlying cognitive task (blue indicates a 0-back task while red indicates a 2-back working memory task).

working memory performance [180].

6.4 Conclusion

The study of dynamic functional connectivity networks is a novel and important avenue of neuroscientific research [31]. As a result, many novel methodologies have been proposed through which to estimate time-varying connectivity networks. However, one aspect that has been overlooked has been how to effectively interpret and visualize the estimated

networks in order to thoroughly understand how such networks are modulated by the underlying task. In the past this issue has been partially addressed via the use of a wide methods including univariate testing on edges, tracking of graph metrics such as degree centrality and clustering methods.

In this work we look to address these issues via the use of graph embedding methods based on linear projections over the set of edges. The motivation behind the use of linear methods stems from the fact that they may subsequently be interpreted in the context of functional connectivity. As a result, such methods allow for the identification of entire networks (as opposed to only edges or nodes) which vary throughout a task. In this manner, we are able obtain a more holistic understanding of the dynamic reconfigurations which occur throughout a task.

Formally, the two embedding methods presented in this work are based on PCA and regularized LDA respectively. These two approaches correspond to unsupervised and supervised learning methods respectively, and can therefore be seen as complementary tools through which to understanding dynamic functional connectivity in further detail. The PCA-driven embedding presented is closely related to the eigen-connectivity approach introduced by [103]. Here PCA is employed to report a weighted combination of edges which demonstrates the largest variability over time. In the context of task-based fMRI, we hypothesize such edges will be related to the underlying task, however such an approach can also be applied in the context of resting-state data, indeed this is the original application presented by [103]. Conversely, the LDA-driven embedding corresponds to a novel supervised embedding algorithm which is explicitly designed for task-based fMRI data. First, a screening procedure is applied in order to weed out non-informative edges and yield sparse and interpretable networks. LDA is subsequently employed to learn an embedding which is discriminative across tasks.

While the proposed graph embedding methods have the advantage that they yield easily interpretable results, it is also important to consider their shortcomings. One significant limitation of the both graph embedding methods is that they fail to account for autocorrelation which may be present in the data over time. One potential way to address this issue may be consider dynamic PCA models, as proposed by [25, 151]. A further shortcoming of the proposed PCA-driven embedding is that the components recovered need not necessarily

be associated with temporal changes in the functional connectivity structure. For example, if the variability across subjects far exceeds any dynamic variability in the connectivity, the associated embedding will be difficult to interpret. As noted in the application to the HCP data, it is also important to consider the relationship between the graph embedding and network estimation methodologies. While this lies beyond the scope of this chapter, further work will look to provide a more detailed understanding of such a relationship and may provide guidance regarding the tuning of hyper-parameters.

We demonstrate that the empirical capabilities of the proposed embeddings methods using simulated data where the underlying network structure is simulated in a variety of different ways, each highlighting distinct properties which are frequently encountered in functional connectivity networks. The simulation study provides compelling empirical evidence demonstrating that the proposed methods are able to recover changes in the underlying connectivity structure and are robust when applied to unseen datasets. The capabilities of the proposed graph embedding methods are also highlighted in an application to task-based fMRI data.

Future work may consider more complex graph embeddings which further exploit the properties of networks, for example via the use of heat kernels [33, 34]. Moreover, graph embedding methods could be employed in a variety of contexts. An exciting potential application is in personalized neurofeedback based on functional connectivity [108, 109, 110, 126]. In such a setting, the use of graph embedding methods could potentially be employed to provide an easily interpretable score for subjects to optimize via the use of neurofeedback.

Chapter 7

Covariance selection in the context of heterogeneous data

The focus of preceding chapters has been largely associated with estimating time-varying Gaussian graphical models (GGMs). This has been motivated by the study of neuroimaging data where the underlying statistical dependencies across brain regions are assumed to vary over time [31]. More precisely, the methods described in previous chapters have been rooted on the introduction of regularization penalties, such as those introduced in Section 2.1, in order accurately quantify covariance structure at each observation.

In this chapter we consider a distinct problem, that of understanding heterogeneity in covariance structure across multiple GGMs. As a result, the methods presented in this chapter revolve around the estimation of multiple related GGMs. In many applied settings the data available corresponds to observations across several different classes. A pertinent example correspond to the study of functional connectivity networks across a population of subjects. In such a setting we may consider each subject as a distinct class and typical objectives include both estimating GGMs for each class as well as a population GGM. A third objective which is often overlooked corresponds to understanding and quantifying variability across estimated GGMs. In particular, it is imperative to understand variability on an edge-by-edge basis as this will allow researchers to distinguish edges shared across the entire population from subject-specific idiosyncrasies. It follows that quantifying variability across multiple subjects and relating this to physiological or genetic traits is an important

neuroscientific problem [51].

One of the hallmarks of neuroimaging data is its reproducible nature. Observed patterns in connectivity have been shown to demonstrate reproducible properties across subjects [41, 191]. This motivates the need for novel methodologies with two overriding objectives. First, there is a need to exploit the presence of shared connectivity structure in order to yield more accurate network estimates for each subject. Second, there is also a critical need to understand and quantify inter-subject variability in the context of functional connectivity [94, 127]. By quantifying variability across a cohort of subjects, such methods are able to untangle the characteristics which define a population from subject-specific idiosyncrasies. Such methods therefore open the door to a more intimate understanding of the properties of brain networks [60].

To date, the aforementioned challenges have not been simultaneously addressed in a comprehensive manner. Instead, previous work has considered one of two main approaches. The first involves learning a separate GGM for each subject. While methods such as the Graphical lasso are often employed to address the high-dimensional nature of the data, more sophisticated techniques are able to exploit the reproducible nature of connectivity via the introduction of novel regularization schemes [42, 172]. Such methods propose to jointly estimate networks across subjects under some constraints over edges. In this manner, the edge structure of each subject is informed by the estimated structure of all remaining subjects.

The second approach is to learn a single GGM that is representative of the entire population of brain networks. Such a strategy is able to alleviate issues caused by the high-dimensional nature of the data by combining observations across subjects (albeit in a potentially naïve manner). However, the question of understanding variability across the population is often sidelined [60].

The objective of the work presented in this chapter is to reconcile the two popular approaches presented above, thus allowing for accurate network estimation at subject-specific and population levels while also quantifying variability present across a cohort. The proposed methodology, named Mixed Neighborhood Selection (MNS), is based on the neighborhood selection method introduced in Section 2.2. By recasting covariance selection as a series of linear regression problems, neighborhood selection methods are able to learn the

local network topology of each region. MNS extends neighborhood selection by incorporating an additional random effect component. This corresponds to learning a novel model for covariance structure across a cohort of subjects. In the proposed model the conditional dependence structure for each subject is decomposed as the union of a population covariance structure together with subject-specific idiosyncrasies. This serves to directly model inter-subject variability and provides a much richer model of functional connectivity. In particular, the proposed method is able to partition edges according to their reproducibility across the cohort. In doing so, MNS provides an additional layer of information which can be exploited to further understand functional connectivity. Moreover, by effectively differentiating between reproducible edges present across the entire cohort and highly variable edges, the proposed method is able to share information across subjects in a discriminative manner, leading to more reliable network estimates.

In order to illustrate the capabilities of the proposed method we present a brief motivating example, shown in Figure [7.1]. We consider a scenario where the population consists of four individuals whose functional connectivity networks share a common structure but also demonstrate some variability. In particular, one edge varies across subjects such that two subjects exhibit the functional connectivity shown in Figure [7.1a] and the remaining two Figure [7.1b]; the edge in question (edge A) is shown to vary from positive to negative across groups. In such a scenario, it is of scientific interest both to uncover the correct functional connectivity networks as well as to correctly identify edges which are variable within the population. This is precisely what MNS is capable of achieving. The results are shown in Figure [7.1c] where the blue lines indicate edges shared across the entire population. The thick gray edges indicate *random effect* edges that demonstrate high variability. Figure [7.1d] shows the estimated edge coefficients for two edges of interest when estimated using the proposed method and the Graphical lasso. We note that in addition to correctly recovering the sparsity structure, the proposed method is able to discriminate edges according to their reproducibility over the cohort. This is in contrast to what could be achieved by studying the networks estimated for each subject independently. This point is demonstrated in Figure [7.1d] where the estimated edge coefficients for the Graphical lasso* are shown to

*The Graphical lasso was run independently for each subject. The regularization parameter for each subject was selected using cross-validation.

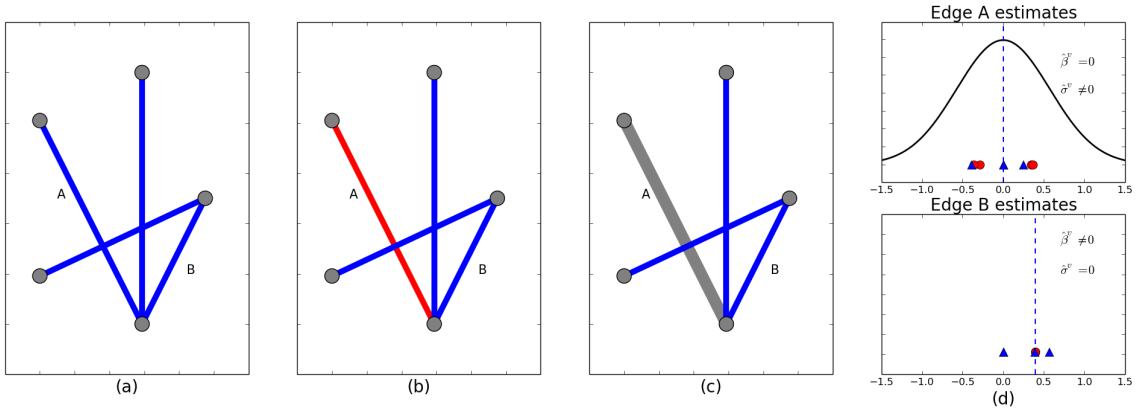


Figure 7.1: Toy motivating example to illustrate the capabilities of the proposed method. Networks were simulated with $p = 5$ nodes and with $n = 8$ observations per subject for $N = 4$ subjects. The networks for two of the subjects is shown in **(a)** while the networks for the remaining two is shown in **(b)**. Blue and red edges indicate positive and negative partial correlations respectively. A significant proportion of the edges are shared across subjects with a single variable edge. The results for our proposed method are shown in **(c)**: blue lines indicate edges shared by the entire population while thick gray edges indicate highly variable edges. Estimated edge coefficients for edges A and B are shown as obtained by the MNS algorithm as well as by applying the Graphical lasso to each dataset independently in **(d)**: Dashed blue lines indicate the estimated population edge value while the solid back line is the estimated probability density function of that edge based on the random effects. Blue, triangular points indicate edge values as estimated by the Graphical lasso while red, circular points indicate subject-specific MNS estimates.

be variable across both edges, one of which is does not vary across subjects. As a result, it follows that identifying variable edges in a two-step procedure is challenging, even in low dimensions.

The remainder of this chapter is organized as follows. The proposed method is detailed in Section 7.1. We present an extensive simulation study in Section 7.2. The proposed method is applied to resting-state fMRI data from the ABIDE consortium in Section 7.3.

7.1 Mixed neighborhood selection

To set notation, we assume we have access to fMRI time series across a cohort of S subjects. For the ith subject, it is assumed we observe an n-dimensional fMRI time series across p

fixed regions of interest. We write $V = \{1, \ldots, p\}$ to denote the set of regions or nodes and refer to the dataset for the ith subject by $X^{(i)} \in \mathbb{R}^{n \times p}$. Further, we write $X_v^{(i)} \in \mathbb{R}^n$ to denote the time-series for any node $v \in V$. Similarly, we let $X_{\setminus v}^{(i)} \in \mathbb{R}^{n \times (p-1)}$ denote the times-series across all remaining nodes.

Throughout this work it is assumed that the data of each subject follows a stationary multivariate Gaussian distribution. Since our primary interest is the estimation of functional connectivity networks, summarized in the inverse covariance matrix, we assume without loss of generality that each $X^{(i)}$ corresponds to n samples from a multivariate Gaussian distribution with zero mean and covariance given by $\Sigma^{(i)}$.

Under the assumption of Gaussianity, estimating functional connectivity networks based on partial correlations is equivalent to learning the conditional dependence structure for each subject. This can be succinctly represented as a graphical model, $G^{(i)} = \left(V, E^{(i)}\right)$, where the edge set, $E^{(i)}$, encodes conditional dependencies across a fixed set of nodes, V. Formally, the edge set summarizes the non-zero entries in the precision matrix, thus:

$$E^{(i)} = supp\left(\left(\Sigma^{(i)}\right)^{-1}\right) = \left\{(j,k) : \left(\Sigma^{(i)}\right)_{j,k}^{-1} \neq 0\right\}. \tag{7.1}$$

The objective of the proposed Mixed Neighborhood Selection (MNS) algorithm is to accurately infer the edge structure across a cohort of subjects. Due to the aforementioned properties of functional connectivity networks, we wish to exploit information across multiple subjects in order to obtain more accurate network estimates for each subject. We also wish to accurately identify the set of highly variable edges. This allows us to distill the set of stable edges, which are predominant across the entire cohort, from highly variable edges. In order to achieve this we introduce a model for the covariance structure across subjects, discussed in Section 7.1.1. The corresponding estimation framework and algorithm are discussed in Section 7.1.2.

7.1.1 A novel covariance model

We propose to model the covariance structure for each subject as the union of a shared covariance structure together with subject-specific idiosyncrasies. This corresponds to the assumption that there exists a shared covariance structure which manifests itself across all

subjects together with subject-specific deviations from this structure. The latter allows our model to accommodate inter-subject variability which cannot be ignored. As a result, we model the conditional dependence structure of each subject as the union of the support of a sparse population network and a subject-specific network. Formally, the support for each subject's conditional dependence structure, originally defined in equation (7.1), is modeled as:

$$E^{(i)} = E^{pop} \cup \tilde{E}^{(i)} \tag{7.2}$$

Here we interpret E^{pop} as the population edges which encode the conditional dependence structure shared across the entire population. Under the assumption of Gaussianity, it follows that E^{pop} is associated with a population precision matrix, $\Theta^{pop} \in \mathbb{R}^{p \times p}$. From the perspective of covariance structure, E^{pop} encodes the maximal conditional dependence structure shared across all subjects. On the other hand, it is $\tilde{E}^{(i)}$ which encodes subject-specific deviations from the population covariance structure. We define $\tilde{E} = \bigcup_{i=1}^{N} \tilde{E}^{(i)}$ as the set of edges demonstrating variability across the entire population of N subjects. This variability may either be attributed to the nature of the edge (i.e., positive or negative partial correlations as in the motivating example described in Figure [7.1]) or partial presence of the edge (i.e., the edge is only present in some subjects).

The objective of the proposed method therefore corresponds to accurately identifying both E^{pop} and $\tilde{E}^{(i)}$. Given E^{pop} and $\tilde{E}^{(i)}$, one can infer $E^{(i)}$ and $\tilde{E}$. However, by focusing on E^{pop} and $\tilde{E}^{(i)}$, as opposed to directly considering subject-specific edges, a far richer description of functional architecture is obtained. In the case of the motivating example presented in Figure [7.1], $\tilde{E} = \tilde{E}^{(i)} = \{A\}$ while the remaining edges are captured in E^{pop}. From the perspective of neuroimaging, partitioning edges in this manner is fundamental to further understanding the functional architecture of the brain [93].

It is useful to note that the model described in equation (7.2) generalizes two typical approaches in the study of functional connectivity. The traditional method of estimating a single population network, Θ^{pop}, by concatenating data across all subjects is equivalent to the assumption that $\tilde{E} = \emptyset$. This corresponds to the strong assumption that all observations across all subjects share an identical conditional dependence structure. Conversely, the approach of estimating a functional connectivity network for each subject independently

corresponds to the assumption that $E^{pop} = \emptyset$. In such a scenario, there is no advantage to be gained by sharing information across subjects. Typically, we would expect the true underlying network structure across subjects to lie somewhere along the spectrum between these two extremes; thus justifying the proposed model.

7.1.2 Estimation framework

The covariance model described in Section 7.1.1 provides a rich framework through which to understand connectivity across a cohort of subjects. In order to learn the associated parameters, we look to extend neighborhood selection. As a result, we consider learning the neighborhood of node $v \in V$ over a cohort of N subjects by studying the following linear mixed effect model:

$$X_v^{(i)} = X_{\setminus v}^{(i)} \beta^v + X_{\setminus v}^{(i)} \tilde{b}^{(i),v} + \epsilon^{(i),v} \text{ for } i = 1, \dots N \,. \tag{7.3}$$

Recall that $X_v^{(i)}$ denotes the time series at node v for subject i. The model described in equation (7.3) directly extends traditional neighborhood selection model by introducing random effect terms, $\tilde{b}^{(i),v}$, for each subject. We note that β^v corresponds to the shared population neighborhood.

The random effects are assumed to follow a multivariate Gaussian distribution, $\tilde{b}^{(i),v} \sim \mathcal{N}(0, \Phi^v)$, independently of $\epsilon^{(i),v}$. The choice of covariance structure for random effects is crucial to both estimating the model as well as to its interpretability. While it is possible to motivate many choices for $\Phi^v \in \mathbb{R}^{p-1 \times p-1}$, in this work we limit ourselves to the scenario where $\Phi^v = \sigma^2 \text{diag}(\sigma^{v\,2})$. Here $\sigma^v \in \mathbb{R}^{p-1}$ is a vector describing the standard deviation of the neighborhood of v across the cohort of N subjects. A large value of σ_u^v would be indicative of heterogeneity in the edge between nodes v and u.

For any node $v \in V$, the model described in equations (7.3) is easily interpretable. The population, or fixed effects, neighborhood is captured in β^v. These are the effects that are shared across the entire cohort of subjects and correspond to the set of edges in E^{pop}. Meanwhile, the random effects are able to capture subject-specific deviations from the population neighborhood and can thereby be employed to obtain a network for each subject. Formally, the random effects captured in σ^v correspond to the set of highly variable edges,

$\tilde{E}$. Finally, we are also able to obtain estimates of $\tilde{b}^{(i),v}$, which can be employed to obtain subject-specific networks. These values correspond to the subject-specific idiosyncrasies, $\tilde{E}^{(i)}$.

Estimation algorithm

The model described in Section 7.1.2 contains the following parameters, $\phi^v = (\beta^v, \sigma^v, \sigma^2) \in \mathbb{R}^{2(p-1)+1}$, which must be estimated for each node $v \in V$. Given ϕ^v we can subsequently obtain the best linear unbiased predictions (BLUPs) for each of the random effects, $\tilde{b}^{(i),v}$, across subjects [139]. In this work ϕ^v is estimated in a maximum likelihood framework, where the negative log-likelihood for node v is proportional to:

$$\mathcal{L}(\phi^v) = \sum_{i=1}^{N} \frac{1}{2} \log \det V_v^{(i)} + \frac{1}{2} \left(X_v^{(i)} - X_{\backslash v}^{(i)} \beta^v \right)^T V_v^{(i)^{-1}} \left(X_v^{(i)} - X_{\backslash v}^{(i)} \beta^v \right), \tag{7.4}$$

where we define $V_v^{(i)}$ to be the variance structure for node v at subject i:

$$V_v^{(i)} = \sigma^2 \left(X_{\backslash v}^{(i)} \text{diag}(\sigma^{v2}) \left(X_{\backslash v}^{(i)} \right)^T + I \right). \tag{7.5}$$

where we write I to denote the identity matrix.

We re-parameterize the random effects component of the mixed effect model, described in equation (7.3), as follows:

$$\tilde{b}^{(i),v} = \text{diag}(\sigma^v)\, b^{(i),v}, \tag{7.6}$$

where $b^{(i),v} \sim \mathcal{N}(0, \sigma^2 I)$. This serves to simplify the discussion of the estimation procedure.

In this work random effects are treated as latent variables and an EM algorithm is employed [115]. Fitting linear mixed effects models in this manner is a popular approach first posited by [45] and for which many efficient algorithms have been proposed [119]. In the context of this work, such an approach will prove beneficial when regularization constraints are introduced. Assuming the random effects, $b^{(i),v}$, are observed the complete data

log-likelihood is proportional to:

$$\mathcal{L}_c(\phi^v) = \sum_{i=1}^{N} \frac{n+p}{2} \log \sigma^2 + \frac{1}{2\sigma^2} \left(\left|\left| X_v^{(i)} - X_{\setminus v}^{(i)} \beta^v - X_{\setminus v}^{(i)} \text{diag}(\sigma^v) b^{(i),v} \right|\right|_2^2 + b^{(i),v^T} b^{(i),v} \right). \quad (7.7)$$

Regularization is introduced for two reasons. First, sparse solutions remain feasible when only a reduced number of observations or subjects are available. Second, parsimonious solutions remain easily interpretable even in the presence of many nodes. As a result, we impose an ℓ_1 penalty on both the fixed as well as random effects. In terms of the random effects we penalize the variance terms, σ^v. Should a variance be shrunk to zero, the resulting random effect is effectively removed from the model. The introduction of sparsity inducing penalties yields the following penalized complete-data log-likelihood:

$$\mathcal{L}_c^{\lambda_1,\lambda_2}(\phi^v) = \mathcal{L}_c(\phi^v) + \lambda_1 ||\beta^v||_1 + \lambda_2 ||\sigma^v||_1, \quad (7.8)$$

where λ_1 and λ_2 are regularization parameters. Sparsity at the population level is enforced by λ_1, while λ_2 encourages sparsity in the random effects by shrinking the standard deviation terms, σ^v.

The proposed EM algorithm involves iteratively computing the conditional expectation of latent variables, $Q(\phi; \phi^v)$, in our case the random effects, and minimizing the expected conditional log-likelihood with respect to parameters ϕ^v. The expectation step (E-step) can be computed in closed form as follows:

$$b^{(i),v} = \left(\text{diag}(\sigma^v) X_{\setminus v}^{(i)^T} X_{\setminus v}^{(i)} \text{diag}(\sigma^v) + I \right)^{-1} X_{\setminus v}^{(i)^T} \text{diag}(\sigma^v) \left(X_v^{(i)} - X_{\setminus v}^{(i)} \beta^v \right) \quad (7.9)$$

independently for each subject $i = 1, \ldots, N$. This follows from computing the conditional expectation of the latent random effect variables given the observed data and current parameter estimates [45]. It is clear from equation (7.9) that if σ_u^v is shrunk to zero then the uth entry of $b^{(i),v}$ will also be zero for all subjects.

In the minimization step (M-step) the latent variables, $b^{(i),v}$, are assumed to be observed. We therefore learn (β^v, σ^v) by solving the following convex problem:

$$(\beta^v, \sigma^v) = \underset{(\beta^v \in \mathbb{R}^{p-1}, \sigma^v \in \mathbb{R}_+^{p-1})}{\text{argmin}} \left\{ \left|\left| X_v^{(i)} - X_{\setminus v}^{(i)} \beta^v - X_{\setminus v}^{(i)} \text{diag}(b^{(i),v}) \sigma^v \right|\right|_2^2 + \lambda_1 ||\beta^v||_1 + \lambda_2 ||\sigma^v||_1 \right\}. \quad (7.10)$$

We note that equation (7.10) is a lasso problem with distinct regularization parameters applied to the fixed and random effects components respectively. A vast range of efficient algorithms can be employed to solve equation (7.10). In this work gradient descent algorithms [67], such as those discussed in Section 2.3, were employed. The motivation behind this choice was that due to the iterative nature of the EM algorithm employed, a lasso problem must be solved at each iteration. It follows that while solutions from one iteration to the next will typically not be identical they will be similar. As a result, computational gains can be obtained by using past solutions as warm-starts. Gradient descent algorithms are particularly well-suited for such tasks. Algorithm 4 details the proposed method.

Algorithm 4: Mixed neighborhood selection algorithm

Input: Data across N subjects, $\{X^{(i)} : i = 1, \ldots, N\}$
begin
  for v *in* $\{1, \ldots, V\}$ **do**
    Define initial estimates: $\beta^v = \boldsymbol{0}$, $\sigma^v = \boldsymbol{1}$, $\sigma = 1$ and $b^{(i),v} = \boldsymbol{0}$
    while *not converged* **do**
      Update (β^v, σ^v) by solving equation (7.10) `// M-step`
      Estimate latent variables using equation (7.9) `// E-step`
    Store β^v, σ^v and $\left\{b^{(i),v}\right\}_{i=1}^{N}$
  $E^{pop} = \{(u, v) : \beta_u^v \neq 0 \text{ and } \beta_v^u \neq 0\}$
  $\tilde{E} = \{(u, v) : \sigma_u^v \neq 0 \text{ and } \sigma_v^u \neq 0\}$
  $\tilde{E}^{(i)} = \{(u, v) : b_u^{(i),v} \neq 0 \text{ and } b_v^{(i),u} \neq 0\}$
return E^{pop}, $\tilde{E}$ *and* $\tilde{E}^{(i)}$ *for* $i = 1, \ldots, N$

7.1.3 Tuning parameters

The proposed method requires the tuning of two regularization parameters which govern the nature of the estimated population and subject-specific networks respectively. Large values of λ_1 will lead to sparse networks at the population level. Conversely, selecting large λ_2 will penalize the variance of the random effects leading to sparse subject-specific contributions to covariance structure.

Moreover, in the class of models considered in this work each covariate can contribute to the fixed as well as random effect structure. This can potentially lead to problems re-

garding the interpretability of estimated models. For example, over-penalizing the fixed effects may lead to over-estimation of the random effect variances as compensation [153]. The choice of regularization parameters is therefore a delicate issue which must be handled with care.

While information theoretic methods such as the AIC may be employed for the purpose of tuning regularization parameters, in this work we employ cross-validation. We note that such an approach is frequently employed within neuroimaging applications [172, 171]. Formally, the data across all subjects is divided into K folds. For each fold, the data from the remaining $K - 1$ folds is employed to fit the penalized linear mixed model described in Section 7.1.2. The resulting model is then used to predict the unseen data and the mean square error is noted. This procedure is repeated over all nodes and across all subjects, with the parameters minimizing total mean square error selected.

7.2 Simulation study

In this section we evaluate the performance of the proposed method using simulated data that is representative of functional imaging data. We assess the empirical performance of the MNS algorithm in three distinct settings which correspond to correctly reporting the edge structure of the population, subject-specific and highly variable network edges respectively. The first task corresponds to correctly recovering E^{pop} while the second requires learning subject-specific edge structure, $E^{(i)}$, defined in equation (7.1). Finally, the task of recovering variable edges is equivalent to learning the set of variable edges, $\tilde{E}$.

7.2.1 Simulation settings

In order to perform such a study we require a method through which to simulate population and subject-specific networks. While numerous algorithms have been proposed to generate random individual networks, there has been limited work on algorithms to simulate multiple clustered networks. Notably, there is no documented method through which to generate networks from a cohort of related subjects that demonstrate the characteristics observed in real fMRI data; namely a shared core structure which is reproducible across all subjects together with significant inter-subject variability in the remaining edges [30].

In order to address this issue we propose a novel method of simulating multiple, related networks. The proposed algorithm is motivated by an exploratory data analysis of resting state fMRI data. We briefly outline the proposed algorithm in this section with further details provided in Appendix C.2.

The underlying idea behind the proposed network simulation method is that key properties observed in fMRI data should be present. As such, the proposed method consists of a set of population edges E^{pop} which are sampled according to the preferential attachment model of [11]. These edges constitute the core, reproducible connectivity structure which will be present across all subjects. Thereafter, a set of variable edges, $\tilde{E}$, is selected uniformly at random across all edges. For each subject, edges in $\tilde{E}$, are included in the subject-specific network, $\tilde{E}^{(i)}$, with some fixed probability τ. This yields clustered networks where there is a clear shared structure together with diverse subject-specific idiosyncrasies.

The proposed method was employed to simulate synthetic data for a cohort of $N = 10$ subjects. The number of nodes was fixed at $p = 50$. For each subject, data consisted of n samples from a multivariate Gaussian with zero mean and covariance specified by $\tilde{E}^{(i)}$. Data was simulated with a varying number of observations per subject, $n \in \{50, 100, 200\}$.

7.2.2 Alternative methods

Throughout this simulation the performance of the MNS algorithm was benchmarked against the current state of the art in each of the three settings described above. In the case of estimating the population network, the Graphical lasso [65] was employed. Such an approach has been used extensively in the neuroimaging community to learn functional connectivity networks across populations [157]. An approach based on resampling and randomization was also employed. This approach, which we refer to as the *Stability* approach, is outlined in Appendix B.2. We note that while this approach is inspired by the recently proposed R^3 method of [129], the objective here is different.

The problem of estimating subject-specific functional connectivity networks has received considerable attention. In this simulation study we compare the performance of the proposed method with the two penalized likelihood methods presented in [172] and [42]. Each of these methods can be seen as a special case of the Joint Graphical lasso (JGL)

framework proposed by [42], as a result we refer to each as the the JGL-Group or the JGL-Fused algorithms respectively. The Graphical lasso algorithm is also employed in this context.

As far as we are aware there are no alternative methods available which address the problem of recovering highly variable edges. In order to provide a benchmark for the MNS algorithm, the aforementioned *Stability* approach was employed in this context.

7.2.3 Performance measures

Throughout this simulation the task of recovering covariance structure is treated as a binary classification task. Thus performance is measured according to the proportion of edges which are correctly reported as being either present or absent. In order to compare performance across various algorithms we employ receiver operating characteristic (ROC) curves, which illustrate the performance of a binary classifier by plotting the true positive rate against false positive rate across a range of regularization parameters [97].

The use of ROC curves requires a single, sparsity-inducing parameter to be varied across a range of possible values. In the case of the MNS algorithm both the population and subject-specific parameters can affect sparsity. As a result, we look to reparameterize the MNS penalty as follows:

$$\lambda_1 = \alpha\lambda \tag{7.11}$$

$$\lambda_2 = \sqrt{2}(1-\alpha)\lambda \tag{7.12}$$

where α controls the ratio of sparsity between the population and subject-specific contributions and λ the overall sparsity. Thus α is fixed allowing λ to vary. While no such adjustments are needed in the case of the JGL-Fused algorithm, we follow the same parameterization described in equations (7.11) and (7.12) in the case of the JGL-Group algorithm. We note this is the same parameterization employed by [42].

7.2.4 Results

In this section we present the results to the simulation study described above. We begin by first considering performance in the context of recovering the set of variable edges. Results for the more frequently studied problems of recovering population and subject covariance structure are presented thereafter.

Throughout this simulation the MNS algorithm was run with $\alpha = 0.25$ while sparsity parameter λ varied as described in equations (7.11) and (7.12). The same parameterization was employed for the JGL-Group algorithm with $\alpha = 0.15$ selected. In the case of the JGL-Fused algorithm, $\lambda_2 = 0.2$ was employed. Finally, the *Stability* algorithm was run with $B = 10,000$ bootstrap iterations per subject and $c = 0.25$.

Variable network recovery

Understanding variability in covariance structure across a cohort of subjects is a fundamental problem in neuroscience. In particular, understanding whether this variation can be attributed to phenotypic characteristics or other sources of noise is crucial in further understanding the human connectome.

The results shown in the top panel of Figure [7.2] demonstrate that the proposed MNS algorithm is able to accurately identify edges which demonstrate variability across a cohort of subjects. Recall that the MNS algorithm jointly estimates the population connectivity as well as the variance of a subject-specific random effect. Thus by considering the estimated variances for the random effects, the proposed MNS algorithm is able to discriminate between edges which are heterogeneous and homogeneous throughout the population. This is in contrast to the *Stability* method. Briefly, the *Stability* method (described in Appendix B.2) treats the presence or absence of edges at a subject level as a Bernoulli random variable. A hierarchical random effects model is then proposed to model the presence or absence of an edge across all subjects. The resulting estimate of the edge variability is then employed to discriminate between variable and non-variable edges. The *Stability* method therefore corresponds to a two-step procedure where variability is only studied *after* networks have been estimated for subjects independently. This is in contrast to the proposed method where subject-specific, population and variable networks are learnt *si-*

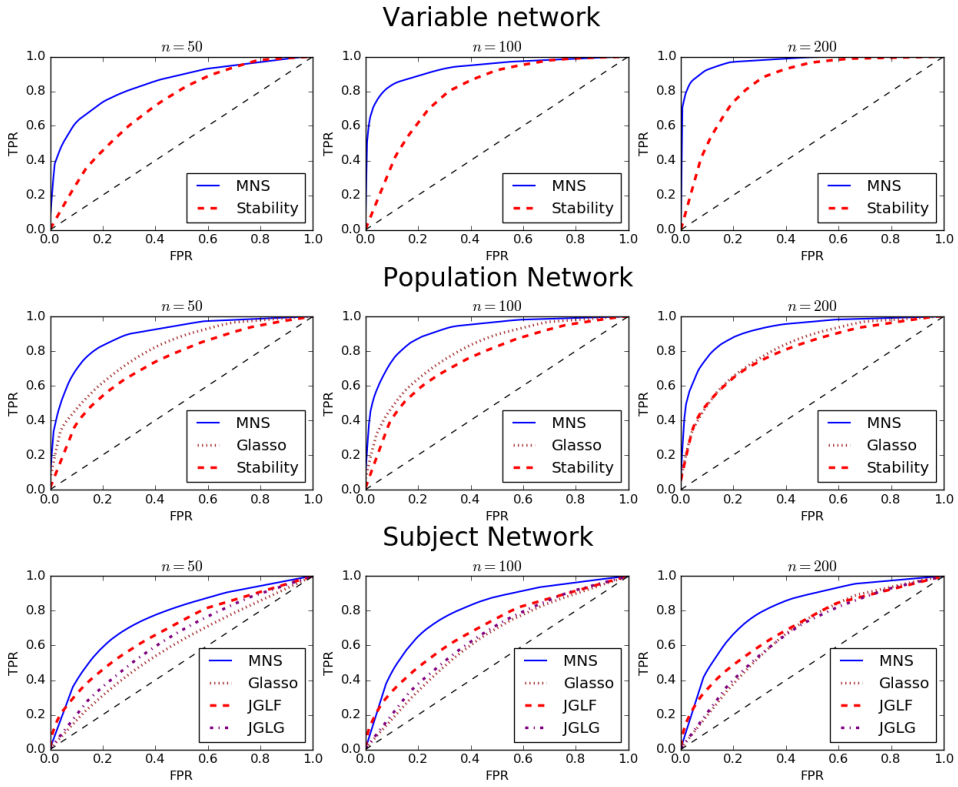


Figure 7.2: Simulation results for all five algorithms across all tasks. Recovery of variable edges is considered in the top panel, population network recovery in shown in the middle panel and finally the bottom panel shows subject-specific network recovery. This simulation was performed with $p = 50$ nodes and $n \in \{50, 100, 200\}$ observations.

multaneously, resulting in significant improvements in performance. Further results are given in Table [7.1] where the true positive rate (TPR) and false positive rate (FPR) are reported for selected regularization parameters.

Population network recovery

Obtaining an accurate understanding of a population level covariance structure is a challenging problem due to the high inter-subject variability. As mentioned previously, it is imperative to differentiate between subject-specific idiosyncrasies and behavior which is reproducible across the entire cohort. A popular approach often taken in neuroimaging

Algorithm	n	Population		Subject		Variable	
		TPR	FPR	TPR	FPR	TPR	FPR
MNS	50	0.76	0.12	0.75	0.33	0.54	0.06
	100	0.77	0.11	0.80	0.32	0.70	0.03
	200	0.75	0.11	0.82	0.30	0.79	0.02
Glasso	50	0.69	0.27	0.88	0.83		
	100	0.70	0.27	0.83	0.66	NA	
	200	0.68	0.27	0.85	0.58		
Stability	50	0.56	0.20			0.54	0.24
	100	0.59	0.20	NA		0.64	0.18
	200	0.78	0.35			0.71	0.15
JGL Group	50			0.86	0.71		
	100	NA		0.83	0.62	NA	
	200			0.82	0.57		
JGL Fused	50			0.78	0.51		
	100	NA		0.79	0.51	NA	
	200			0.79	0.50		

Table 7.1: Performance of all five algorithms. The true positive rate (TPR) and false positive rate (FPR) is reported for each of the three tasks: recovering population, subject and variable networks.

studies is to estimate a single network using data from all subjects [157], thus effectively concatenating all data. This corresponds to the tenuous assumption that $\tilde{E} = \emptyset$. Such an approach is included in this simulation together with the aforementioned *Stability* approach.

Results are shown in the middle panel of Figure [7.2]. It is interesting to note that for small sample sizes (i.e., $n = 50$ or $n = 100$) the *Stability* approach is out-performed by the Graphical lasso. As in the case of variable network recovery, we attribute this drop in performance to the two-step design of the *Stability* method where information is only shared across subjects *after* networks have been estimated. It is only when the number of observations increases that reliable estimates of uncertainty can be obtained. Conversely, the difference in performance between the Graphical lasso algorithm and the MNS algorithm is due to the presence of heterogeneous edges, implying $\tilde{E} \neq \emptyset$. Thus, by providing a more sophisticated model for inter-subject variability, the MNS algorithm is able to obtain more reliable population network estimates.

Subject-specific network recovery

Finally, we consider the recovery of subject-specific networks. This problem has received considerable attention in recent years and a range of methods have been proposed. The underlying theme in these methods revolves around effectively sharing information across subjects. In the case of the methods proposed by [172] and [42] this is achieved via the introduction of a regularization penalty over the edge structure. In this manner, the covariance structure of an individual subject is informed by the estimated covariance structure across all remaining subjects. However, a short coming of the aforementioned methods is that regularization is applied in an indiscriminate manner. By enforcing either a group or fused lasso penalty on all entries of precision matrices, such methods effectively encourage information to be shared homogeneously across all edges. We envisage a scenario where edges can be ordered according to their variability (or reproducibility). This is a well-documented phenomenon in neuroimaging. In particular for fMRI data there is evidence to suggest that variability in connectivity is directly modulated by factors such as the distance between regions [141, 152].

The proposed MNS algorithm is able to address precisely this issue. By discriminating between subject-specific and population edges, it is able to effectively vary the how extensively information is shared across subjects on an edge-by-edge basis. As a result, the MNS algorithm is able to more reliably recover subject-specific covariance structure.

7.2.4.1 Further experiments

One of the assumptions of the proposed method is that data follow a multivariate Gaussian distribution. While this assumption is commonplace in the analysis of fMRI data [105], we also consider the performance of the MNS algorithm in the context of non-Gaussian data. In order to study the robustness of the MNS algorithm, the simulation study presented above was repeated with data generated according to a multivariate t-distribution. Detailed results are reported in Appendix D. The results indicate that in comparison to alternative methods, the MNS algorithm is robust in the presence of non-Gaussian data. We attribute this behavior to the fact that the covariance model underlying the MNS algorithm explicitly models heterogeneity over subjects, thus allowing the MNS algorithm to better tolerate

contaminated data.

Furthermore, simulating networks as described in Section 7.2.1 is one of many possible methods which could be employed. In order to provide a thorough and fair comparison an additional simulation was also performed where networks were simulated as described in [42]. This simulation was proposed with the objective of providing empirical evidence regarding how accurately subject-specific networks could be reported. It is therefore not well suited for examining how reliably the population or variance networks can be reported. The results are presented in Appendix D.

7.3 Application

In this section the proposed MNS algorithm is applied to resting-state fMRI data from the ABIDE consortium [48]. While the ABIDE dataset contains data corresponding to healthy subjects and Autism Spectrum Disorder (ASD) subjects, we chose only to study healthy controls here as the focus of this work consisted in fully understanding uncertainty across a single population of subjects. The decision to study the ABIDE dataset in this manner was motivated by the fact that it is an open-source dataset which has been previously studied in the context of functional connectivity. Data from the University of Utah School of Medicine (USM) site was considered here, a choice motivated by results suggesting the USM site contained high-quality data [133]. The data therefore consisted of 43 healthy subjects with ages ranging from 8 to 40 years old.

7.3.1 ABIDE data

Data was downloaded from the Autism Brain Imaging Data Exchange (ABIDE) [48]. Data were preprocessed via a CPAC[†] pipeline from the ABIDE repository. Briefly, this involved slice time correction, motion correction and intensity normalization followed by regression of motion parameters. Linear and quadratic trends were removed from the time series to account for low frequency drifts. Mean time-courses were then extracted from 116 regions defined by the Automated Anatomical Labeling (AAL) atlas. This resulted in 200

[†]see `http://fcp-indi.github.com` for further details

observations over 116 nodes for each subject.

7.3.2 Results

The MNS algorithm requires the specification of two regularization parameters, each of which controls the population and subject-specific topology of each node. As discussed in Section 7.1.3, parameters were selected on the basis of a 10-fold cross-validation framework.

One of the advantages of the proposed MNS algorithm is that it is able to simultaneously estimate both a population network, corresponding to reproducible edges which are present across the entire cohort of subjects, as well as a network quantifying variability on an edge-by-edge basis. The latter network is able to succinctly summarize variability across a cohort of subjects. Finally, the MNS algorithm also yields estimates of subject-specific connectivity networks. This allows connectivity to be studied in three distinct yet complimentary approaches which we discuss below.

The top panel of Figure [7.3] shows the estimated population network, indicating the edges which were identified as being consistently present across the entire cohort. The network has an estimated edge density of around 10% and we note there is strong inter-hemispheric coupling as would be expected in resting-state connectivity. More importantly, the bottom panel of Figure [7.3] shows the estimated variability network. This corresponds to the collection of edges that were identified as demonstrating variability across the cohort of subjects. In the case of the variability network, the edge thickness is proportional to the estimated variance of the random effect. We note that is strong inter-hemispheric variability, in particular between the left and frontal gyrus as well as between the left and right postcentral gyrus. There also appears to be a region of variability centered around the cerebellum. We note that the aforementioned regions are in brain areas with relatively high susceptibility to artifact and sensitivity to changes in brain shape, such as the medial prefrontal cortex [133].

One of the strengths of the MNS algorithm is that this variability can be further studied to obtain a deeper understanding regarding the characteristics that define differences in connectivity over a cohort. In Figure [7.4] the variability of two edges is studied in

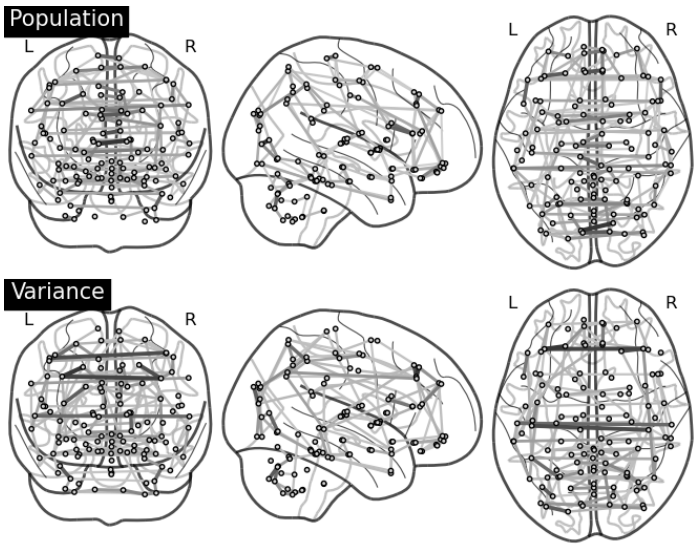


Figure 7.3: Estimated population network (top) and variable edge network (bottom). Edge thickness is proportional to the magnitude of the partial correlation across edges (or variance in the case of variance network). In the case of the population network we note there is high inter-hemispheric coupling which is to be expected in resting-state data. Similar patterns occur in the variable edge network.

detail. The edges correspond to inter-hemispheric connectivity between the left and right frontal gyrus and postcentral gyrus respectively. The histograms capture the distribution of estimated edges across the cohort of subjects. As these edges are estimated to be variable across subjects, the proposed method learns a distinct partial correlation for each subject. The color of histograms visualizes the mean age of subjects within each bin, thereby indicating that bilateral connectivity across frontal gyrus and postcentral gyrus was estimated to increase with age. This was further verified to be significant at the $\alpha = 1\%$ level using Spearman's rank correlation coefficient. At a higher level, these results are consistent with previous literature which suggests that connectivity increases across distant brain regions during development [58] and serve to highlight the maturation of a dual-control system within brain networks [59].

Finally, the MNS algorithm also provides estimates of subject-specific functional connectivity networks. As a result, the proposed method can be used to study connectivity on a subject-by-subject basis. Here we study various properties associated with the estimated functional connectivity network for each subject, in particular we look to study potential

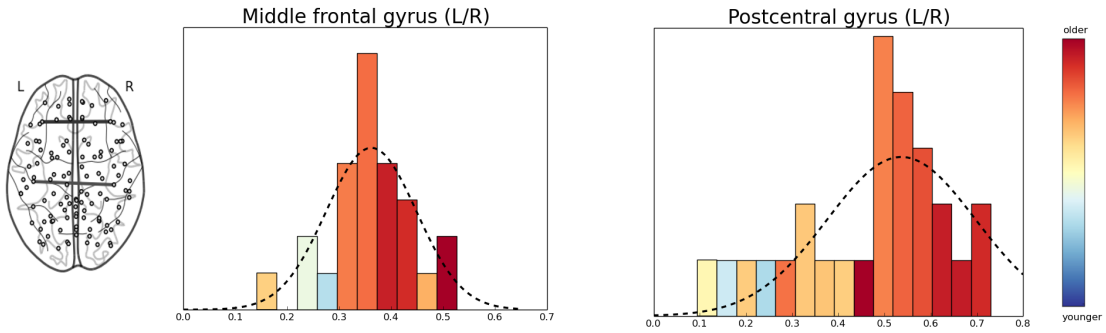


Figure 7.4: The two histograms show the estimated partial correlations across two edges highlighted on the left. Each of the histograms shows the distribution over estimated partial correlations between the left and right frontal gyrus and postcentral gyrus respectively. The color of bins is indicative of mean age of subjects within that bin.

changes in connectivity that are associated with the age of subjects.

It has been suggested that the structure of functional connectivity networks in children is driven by anatomical proximity, with a high connectivity between spatially adjacent regions, while the corresponding structure in adults reflects the integration of remote brain regions. In order to study this hypothesis the average distance between functionally connected brain regions was estimated on a subject-by-subject basis (i.e., using the subject-specific estimates of functional connectivity). The left panel of Figure [7.5] shows the average distance between functionally connected brain regions as a function of the subjects age. We note there is a significant positive correlation at the $\alpha = 1\%$ level using Spearman's rank-order correlation, placing the results in line with other results in the literature [59, 58].

In order to obtain a more detailed understanding of changes occurring in the functional connectivity two further network statistics are studied; the mean betweenness centrality of nodes and the transitivity of estimated networks. The betweenness centrality is a measure of node centrality or importance in a network [147] and is defined as the fraction of all shortest paths passing through a node. Nodes with high betweenness centrality are seen to be bridge connections across many nodes, thereby making their presence in a network important. The mean betweenness centrality across all nodes can be interpreted as a measure of the efficiency in a network. On the other hand, transitivity is a measure of network segre-

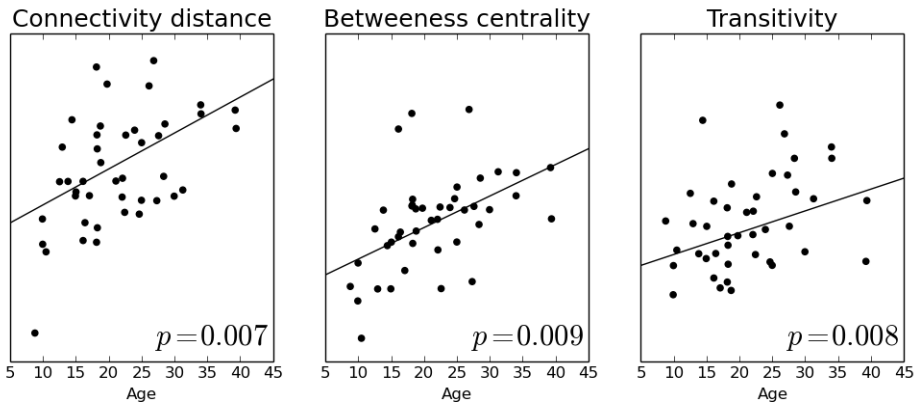


Figure 7.5: Multiple network properties are plotted as a function of the subject ages. Estimated p-values, obtained using Spearman's rank-order measure of correlation, are shown in bottom right corner of each plot. **Left**: the anatomical distance between functionally connected regions. **Middle**: the mean betweenness centrality of nodes. **Right**: the transitivity (i.e., clustering coefficient).

gation which quantifies the presence of clusters in the network. In the context of functional connectivity networks, high transitivity suggests an organization of statistical dependencies indicative of segregated neural processing [147].

The middle and right panels of Figure [7.5] shows the mean betweenness centrality and the transitivity of estimated networks as a function of age. These results would indicate an increase in network segregation and specialization during development, a finding that is consistent with previous literature [59, 58].

7.4 Conclusion

We have considered the task of estimating multiple related GGMs. In particular, we have focused on three closely related challenges: recovering population and subject-specific covariance structure as well as identifying the set of edges demonstrating heterogeneity across networks. The latter is fundamental in the context of many applications, yet it has received limited attention. The proposed methodology is able to simultaneously address all three aforementioned challenges by considering a novel model for covariance structure across a cohort of subjects. Formally, the proposed model looks to decompose covariance structure as the union of population effects, which are reproducible across subjects, with

subject-specific idiosyncrasies.

The underlying covariance model results in several important benefits, principal among which is the ability of the MNS algorithm to accurately identify heterogeneous edges. As a result, the MNS algorithm is able to borrow information across subjects in a discriminative manner. This is in contrast to many of the current methodologies which share information in an indiscriminate fashion (e.g., via the use of regularization penalties whose parameterization is fixed across edges).

The capabilities of the proposed MNS algorithm have been demonstrated using both simulated as well as resting-state data taken from the ABIDE consortium [48]. Throughout the simulation study, care was taken to ensure that the underlying covariance structure closely resembled the frequently reported properties of fMRI data as well as to consider the robustness of the proposed algorithm.

The MNS algorithm requires the specification of two regularization parameters, λ_1 and λ_2, each of which has a natural interpretation. The first parameter controls the sparsity in the population node topologies while the second controls the sparsity of the subject-specific edges. We employ a cross-validation to tune both parameters, as is frequently the case in the context neuroimaging data analysis [171]. The MNS algorithm, together with network simulation methods described in this work have been implemented as an R package named `MNS`. This can be downloaded from the Comprehensive R Archive Network (CRAN).

In conclusion, the MNS algorithm provides a novel methodology through which to understand variability across multiple related GGMs. This work corresponds to a first attempt to modeling functional connectivity in a hierarchical manner in the presence of regularization penalties. Furthermore, by providing a refined model for the covariance structure, the proposed method is also able to accurately recover both population and subject-specific functional connectivity networks.

Chapter 8

Conclusion

Covariance selection is a challenging statistical problem which is often studied under the assumption of stationarity. However, in many applied settings, such assumptions are not reasonable. This indicates the need for novel methodology. Throughout this thesis, we have employed the estimation of functional connectivity networks from fMRI as a motivating application. Such networks are now widely accepted to display non-stationary properties, especially during task-based studies.

The results presented in this thesis can broadly be split into studying non-stationarity across two distinct axes. Firstly, we focus on accurately quantifying non-stationarity over time. This is studied in Chapters 3 to 6. These chapters study non-stationary both in an offline as well as an online setting, where the latter setting is motivated by the study of fMRI in real-time. The second axis corresponds to non-stationarity in covariance structure across multiple related subjects. This work, presented in Chapter 7, is motivated by the need to understand variability in connectivity across a cohort of subjects. The contributions of this thesis can therefore be summarized as follows:

- We propose an algorithm through which to estimate time-varying GGMs in the context of non-stationary data in Chapter 3. The proposed SINGLE algorithm enforces both sparsity and temporal homogeneity constraints in order to ensure that the estimated graphs accurately reflect the true underlying covariance structure. Throughout a series of simulation studies, the SINGLE algorithm was shown to empirically outperform alternative methods based on sliding windows and change-point detection.

- We extend the proposed SINGLE algorithm to the real-time scenario in Chapter 4. This is motivated by an exciting avenue of modern neuroscientific research which involves the study of fMRI data in real-time. The proposed algorithm is based on the combination of adaptive filtering and convex optimization methods. The proposed rt-SINGLE algorithm thus allows for the estimation of GGMs in real-time, thereby facilitating the potential use of functional connectivity networks in the context of neurofeedback or *brain decoding* applications.

- A formal framework through which to tune sparsity inducing regularization parameters in the context streaming and potentially non-stationary data is presented in Chapter 5. Whilst regularized methods are frequently used in the context of streaming, non-stationary data, the choice of the associated regularization parameter has not been formally addressed to date. The proposed framework effectively recasts the selection of a sparsity parameter in the context of adaptive filtering, thereby relegating the choice of such a parameter to the data. This reformulation is the first of its kind in allowing for the tracking of a time-varying regularization parameter as well as the derivation of convergence guarantees in a non-stochastic setting.

- In order to provide robust and interpretable results for estimated graphical models, we derive and validate two graph embedding methods based on linear projections over the edge set. The proposed graph embedding algorithms are based on principal component analysis and regularized linear discriminant analysis respectively. The capabilities of the proposed embeddings are empirically validated throughout a series of simulation studies.

- We propose a novel algorithm through which to estimate multiple related graphical models inspired from the random effects hierarchical regression model. Moreover, unlike previous approaches, the proposed MNS algorithm is focused on understanding uncertainty in covariance structure across multiple graphs. It therefore provides a far more detailed insight into differences in graphical structure which may be present. This is achieved by proposing a novel model for covariance structure. In the proposed model, the set of edges is partitioned in to set of shared edges together with subject-

specific idiosyncrasies. This further allows for a population GGM to be accurately estimated. Throughout an extensive series of simulations, the MNS algorithm is shown to consistently outperform state-of-the-art competitors.

Future work

At a high level, the work presented in this thesis can be split into two distinct avenues of research: the first proposes machinery in order to estimate time-varying GGMs while the second is interested in jointly estimating multiple related graphical models and therefore studying heterogeneity across a cohort of subjects. While there are a number of potential future projects perhaps the most obvious would be to combine the two aforementioned avenues. This would involve jointly estimating multiple time-varying GGMs and could allow for the investigation of inter-subject variability in a task-based setting.

Further work could also be undertaken within each specific avenue. In the context of studying time-varying GGMs there is room for further work relating the tuning of regularization parameters in a streaming scenario. This would involve extending this work to consider alternative regularization penalties, for examples ridge penalization, or considering a more general family of likelihoods, such as exponential family models. In both cases, the derivative with respect to the regularization parameter can be computed or approximated in closed-form.

In the context of estimating multiple related GGMs, a future avenue for research would involve jointly estimating multiple clustered GGMs. In the context of functional connectivity, this could involve jointly estimating networks across a cohort that contained both healthy controls together with subjects suffering some neurological pathologies. This would be particularly interesting in the case of disorders such as Autism which are hypothesized to be related to differences in functional connectivity.

Furthermore, while the study of functional connectivity networks has provided a clear motivation for this work described in Chapter 7, it would be interesting to consider alternative applications. There are certainly many other biomedical applications such as the study of gene-gene networks, indeed this was the motivation behind the work of [42]. Looking beyond biomedical applications, an alternative application for such methodology could be

related to the study of sensor networks within electronic devices. Such devices, for example mobile phones or laptops, all feature data-generating processes which yield multivariate data similar to that studied in this thesis and which could potentially provide an exciting application.

Finally, it would also be interesting to consider the relationship between the methodology discussed throughout this thesis and its Bayesian counterparts. In recent years likelihood-based methods, such as the penalized likelihood methods considered throughout this thesis, have established themselves as competitors to Bayesian modeling approaches. Such methods are often preferred due to the fact that they can often be cast as optimization problems, thereby avoiding difficult integrals which are often associated with Bayesian approaches. This yields important advantages such as computational efficiency and numerical stability. However, due to recent advances in variation inference methods [16] as well as Markov Chain Monte Carlo (MCMC) [130], it is now possible to consider Bayesian models of increasing complexity. As such, it would be interesting to consider the Bayesian counterparts of each of the two aforementioned axes.

Bibliography

[1] S. Achard, R. Salvador, B. Whitcher, J. Suckling, and E. T. Bullmore. A resilient, low-frequency, small-world human brain function network with highly connected association of cortical hubs. *Journal of Neuroscience*, 26(1):63–72, 2006.

[2] C. Aggarwal. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.

[3] E. Allen, E. Damaraju, S. Plis, E. Erhardt, T. Eichele, and V. Calhoun. Tracking whole-brain connectivity dynamics in the resting state. *Cerebral Cortex*, 1:1–14, 2012.

[4] C. Anagnostopoulos, N. Adams, and D. J. Hand. Streaming Covariance Selection with Applications to Adaptive Querying in Sensor Networks. *The Computer Journal*, 53(9):1401–1414, 2010.

[5] C. Anagnostopoulos, D. K. Tasoulis, N. Adams, N. Pavlidis, and D. J. Hand. Online Linear and Quadratic Discriminant Analysis with adaptive forgetting for streaming classification. *Statistical Analysis and Data Mining*, 5(2):139–166, 2012.

[6] L. Ansado, J.and Collins, S. Joubert, V. Fonov, O. Monchi, S. Brambati, F. Tomaiuolo, M. Petrides, and Y. Faure, S.and Joanette. Interhemispheric coupling improves the brains ability to perform low cognitive demand tasks in alzheimers disease and high cognitive demand tasks in normal aging. *Neuropsychology*, 27(4):464, 2013.

[7] A. R. Aron, P. C. Fletcher, E. T. Bullmore, B. J. Sahakain, and T. W. Robbins.

Stop-signal inhibition disrupted by damage to right inferior frontal gyrus in humans. *Nature Neuroscience*, 6(2):115–116, 2003.

[8] N. Axmacher, D. Schmitz, T. Wagner, C. Elger, and J. Fell. Interactions between medial temporal lobe, prefrontal cortex, and inferior temporal regions during visual working memory: a combined intracranial eeg and functional magnetic resonance imaging study. *The Journal of Neuroscience*, 28(29):7304–7312, 2008.

[9] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[10] O. Banerjee, L. Ghaoui, and A. dAspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

[11] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[12] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.

[13] D. Bassett and E. Bullmore. Small-world brain networks. *The Neuroscientist*, 12(6):512–523, 2006.

[14] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39, 2011.

[15] D. P. Bertsekas. *Constrained Optimisation and Lagrange Multiplier Methods*. Athena Scientific, 1982.

[16] D. Blei, A. Kucukelbir, and J. McAuliffe. Variational Inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*, 2016.

[17] D. Bodenham. *Adaptive estimation with change detection for streaming data*. PhD thesis, Imperial College London, 2014.

[18] D. Bodenham and N. Adams. Continuous monitoring of a computer network using multivariate adaptive estimation. In *13th International Conference on Data Mining Workshops*, pages 311–318. IEEE, 2013.

[19] D. Bodenham and N. Adams. Adaptive change detection for relay-like behaviour. In *Joint Intelligence and Security Informatics Conference*, pages 252–255. IEEE, 2014.

[20] V. Bonnelle, T. E. Ham, R. Leech, K. M. Kinnunen, M. A. Mehta, R. J. Greenwood, and D. J. Sharp. Salience network integrity predicts default mode network function after traumatic brain injury. *Proceedings of the National Academy of Sciences of the United States of America*, 109(12):4690–4695, 2012.

[21] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

[22] L. Bottou. Stochastic learning. In *Advanced Lectures on Machine Learning*, pages 146–168. Springer, 2004.

[23] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT'2010*, pages 177–186. Springer, 2010.

[24] M. Botvinick, T. Braver, D. Barch, C. Carter, and J. Cohen. Conflict monitoring and cognitive control. *Psychological review*, 108(3):624, 2001.

[25] C. Boutsidis, D. Garber, Z. Karnin, and E. Liberty. Online principal components analysis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 887–901. Society for Industrial and Applied Mathematics, 2015.

[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[28] S. Bressler and V. Menon. Large-scale brain networks in cognition: emerging methods and principles. *Trends in Cognitive Sciences*, 14(6):277–290, 2010.

[29] P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[30] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature*, 10:186–198, 2009.

[31] V. Calhoun, R. Miller, G. Pearlson, and T. Adalı. The chronnectome: time-varying connectivity networks as the next frontier in fMRI data discovery. *Neuron*, 84(2):262–274, 2014.

[32] C. Chang and G. H. Glover. Time frequency dynamics of resting-state brain connectivity measured with fMRI. *NeuroImage*, 50(1):81 – 98, 2010.

[33] A. Chung, E. Pesce, R. P. Monti, and G. Montana. Classifying HCP task-fMRI networks using heat kernels. In *Pattern Recognition in NeuroImaging (PRNI), 2016 International Workshop on*. IEEE, 2016.

[34] A. Chung, M. Schirmer, M. Krishna, G. Ball, P. Aljabar, A. Edwards, and G. Montana. Characterising brain network topologies: a dynamic analysis approach using heat kernels. *NeuroImage*, 2016.

[35] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4):406–413, 2011.

[36] C. Constantinidis and T. Klingberg. The neuroscience of working memory capacity and training. *Nature Reviews Neuroscience*, 17(7):438–449, 2016.

[37] M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3(3):201–215, 2002.

[38] I. Cribben, R. Haraldsdottir, Y. L. Atlas, T. D. Wager, and M. A Lindquist. Dynamic Connectivity Regression: Determining state-related changes in brain connectivity. *NeuroImage*, 61(4):907–920, 2012.

[39] E. Damaraju, E. Allen, A. Belger, J. Ford, S. McEwen, D. Mathalon, B. Mueller, G. Pearlson, S. Potkin, A. Preda, and V. Calhoun. Dynamic functional connectivity analysis reveals transient states of dysconnectivity in schizophrenia. *NeuroImage: Clinical*, 5:298–308, 2014.

[40] J. Damoiseaux and M. Greicius. Greater than the sum of its parts: a review of studies combining structural connectivity and resting-state functional connectivity. *Brain Structure and Function*, 213(6):525–533, 2009.

[41] J. Damoiseaux, S. Rombouts, F. Barkhof, P. Scheltens, C. Stam, S. Smith, and C. Beckmann. Consistent resting-state networks across healthy subjects. *Proceedings of the National Academy of Sciences of the United States of America*, 103(37):13848–13853, 2006.

[42] P. Danaher, P. Wang, and D. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[43] C. deCharms. Applications of real-time fMRI. *Nature Reviews Neuroscience*, 9(9):720–729, 2008.

[44] M. Demirtaş, C. Tornador, C. Falcón, M. López-Solà, R. Hernández-Ribas, J. Pujol, P. Ritter, N. Cardoner, and C. Soriano-Mas. Dynamic functional connectivity reveals altered variability in functional connectivity among patients with major depressive disorder. *Human Brain Mapping*, 2016.

[45] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (Statistical Methodology)*, pages 1–38, 1977.

[46] A. P. Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

[47] R. Desikan, F. Ségonne, B. Fischl, B. Quinn, B. Dickerson, D. Blacker, R. Buckner, A. Dale, P. Maguire, and B. Hyman. An automated labeling system for subdivid-

ing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, 31(3):968–980, 2006.

[48] A. Di Martino, C. Yan, Q. Li, E. Denio, F. Castellanos, K. Alaerts, J. Anderson, M. Assaf, S. Bookheimer, and M. Dapretto. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667, 2014.

[49] M. Drton and M. Perlman. Model selection for gaussian concentration graphs. *Biometrika*, 91(3):591–602, 2004.

[50] M. Drton and M. Perlman. Multiple testing and error control in gaussian graphical model selection. *Statistical Science*, pages 430–449, 2007.

[51] J. Dubois and R. Adolphs. Building a science of individual differences from fMRI. *Trends in Cognitive Sciences*, 2016.

[52] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[53] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[54] V. M. Eguiluz, D. R. Chialvo, G. A. Cecchi, M. Baliki, and A. V. Apkarian. Scale-free brain functional networks. *Physical review letters*, 94(1):018102, 2005.

[55] J. S. Elam and D. Van Essen. Human connectome project. In *Encyclopedia of Computational Neuroscience*, pages 1–4. Springer, 2014.

[56] P. Erdos and A. Renyi. *On random graphs*. Publicationes Mathematicae Debrecen, 1959.

[57] F. Esposito, A. Bertolino, T. Scarabino, V. Latorre, G. Blasi, T. Popolizio, G. Tedeschi, S. Cirillo, R. Goebel, and F. Di Salle. Independent component model

of the default-mode brain function: Assessing the impact of active thinking. *Brain Research Bulletin*, 70(4):263–269, 2006.

[58] D. Fair, A. Cohen, J. Power, N. Dosenbach, J. Church, F. Miezin, B. Schlaggar, and S. Petersen. Functional brain networks develop from a local to distributed organization. *PLoS Computational Biology*, 5(5), 2009.

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

[60] F. Fallani, J. Richiardi, M. Chavez, and S. Achard. Graph analysis of functional brain networks: practical issues in translational neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1653):20130521, 2014.

[61] M. Finegold and M. Drton. Robust graphical modeling of gene networks using classical and alternative t-distributions. *The Annals of Applied Statistics*, 5(2):1057–1080, 2011.

[62] A. Fornito, B. J. Harrison, A. Zalesky, and J. S. Simons. Competitive and cooperative dynamics of large-scale brain functional networks supporting recollection. *Proceedings of the National Academy of Sciences of the United States of America*, 109(31):12788–12793, 2012.

[63] A. Fornito, A. Zalesky, and M Breakspear. Graph analysis of the human connectome: Promise, progress, and pitfalls. *NeuroImage*, 80:426–444, 2013.

[64] P. Fransson. How default is the default mode of brain function? Further evidence from intrinsic BOLD signal fluctuations. *Neuropsychologia*, 44(14):2836–2845, 2006.

[65] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[66] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

[67] T. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani. Pathwise Coordinate Optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[68] T. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation via the Graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

[69] K. J. Friston. Functional and effective connectivity in neuroimaging: A synthesis. *Human Brain Mapping*, 2(1):56–78, 1994.

[70] K. J. Friston. Function and effective connectivity: A review. *Brain Connectivity*, 1(1):13–36, 2011.

[71] W. Fu. Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.

[72] M. Greicius. Resting-state functional connectivity in neuropsychiatric disorders. *Current opinion in neurology*, 21(4):424–430, 2008.

[73] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J Honey, V. J. Wedeen, and O. Sporns. Mapping the structural core of human cerebral cortex. *PLoS Biology*, 6(7):e159, 2008.

[74] A. Hampshire, S. R. Chamberlain, M. M. Monti, J. Duncan, and A. M. Owen. The role of the right inferior frontal gyrus: inhibition and attentional control. *NeuroImage*, 50(3):1313–1319, 2010.

[75] T. Hastie, J. Taylor, R. Tibshirani, and G. Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1(1):1–29, 2007.

[76] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[77] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[78] S. Haykin. *Adaptive Filter Theory*. Prentice Hall Information and System Science Series, 2002.

[79] N. Heard, D. Weston, K. Platanioti, and D. Hand. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 4(2):645–662, 2010.

[80] G. Hein and R. Knight. Superior temporal sulcusit's my area: or is it? *Journal of Cognitive Neuroscience*, 20(12):2125–2136, 2008.

[81] M. Hinne, L. Ambrogioni, R. J. Janssen, T. Heskes, and M. van Gerven. Structurally-informed bayesian functional connectivity analysis. *NeuroImage*, 86:294–305, 2013.

[82] R. Hocking. The analysis and selection of variables in linear regression. *Biometrics*, 32(1):1–49, 1976.

[83] H. Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

[84] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[85] C. Hsieh, I. Dhillon, P. Ravikumar, and M. Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338, 2011.

[86] J. Huang, N. Liu, M. Pourahmadi, and L. Liu. Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, 93(1):85–98, 2006.

[87] S. Huang, L. Sun, J. Ye, A. Fleisher, T. Wu, K. Chen, and E. Reiman. Learning brain connectivity of alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 50(3):935 – 949, 2010.

[88] S. Huettel, A. Song, and G. McCarthy. *Functional Magnetic Resonance Imaging*. Sinauer Associates, 2004.

[89] M. Husain and P. Nachev. Space and the Parietal Cortex. *Trends in Cognitive Sciences*, 11(1):30–36, 2007.

[90] M. R. Hutchinson, T. Womelsdoft, P. A. Allen, E. A.Bandettini, V. D. Calhoun, M. Corbetta, S. D. Penna, J. Duyn, G. Glover, J. Gonzalez-Castillo, D. A. Handwerker, S. Keilholz, V. Kiviniemi, D. A. Leopold, F. de Pasquale, O. Sporns, M. Walterm, and C. Chang. Dynamic Functional Connectivity: Promise, issues, and interpretations. *NeuroImage*, 80:360–378, 2013.

[91] M. R. Hutchinson, T. Womelsdoft, J. S. Gat, S. Everling, and R. S. Menon. Resting-state networks show dynamic functional connectivity in awake humans and anesthetized macaques. *Human Brain Mapping*, 34:2154–2177, 2012.

[92] E. John and E. Yıldırım. Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, 41(2):151–183, 2008.

[93] R. Kanai and G. Rees. The structural basis of inter-individual differences in human behaviour and cognition. *Nature Reviews Neuroscience*, 12(4):231–242, 2011.

[94] C. Kelly, B. Biswal, C. Craddock, X. Castellanos, and M. Milham. Characterizing variation in the functional connectome: promise and pitfalls. *Trends in Cognitive Sciences*, 16(3):181–188, 2012.

[95] M. Kendall, A. Stuart, and J. Ord. *The advanced theory of statistics*. Oxford University Press, 1968.

[96] Y. Koush, M. Rosa, F. Robineau, K. Heinen, S. Rieger, N. Weiskopf, P. Vuilleumier, D. Van De Ville, and F. Scharnowski. Connectivity-based neurofeedback: dynamic causal modeling for real-time fMRI. *NeuroImage*, 81:422–430, 2013.

[97] W. Krzanowski and D. Hand. *ROC curves for continuous data*. CRC Press, 2009.

[98] S. Kumar, S. Rao, B. A Chandramouli, and S. Pillai. Reduction of functional brain connectivity in mild traumatic brain injury during working memory. *Journal of Neurotrauma*, 26(5):665–675, 2009.

[99] S. LaConte. Decoding fMRI brain states in real-time. *NeuroImage*, 56(2):440–454, 2011.

[100] S. L. Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.

[101] O. Ledoit and M. Wolf. Honey, I shrunk the sample covariance matrix. *UPF economics and business working paper*, 2003.

[102] C. Leng, Y. Lin, and G. Wahba. A note on the lasso and related procedures in model selection. *Statistica Sinica*, 16(4):1273–1284, 2006.

[103] N. Leonardi, J. Richiardi, M. Gschwind, S. Simioni, J-M. Annoni, M. Schluep, P. Vuilleumier, and D. Van De Ville. Principal components of functional connectivity: A new approach to study dynamic brain connectivity during rest. *NeuroImage*, 83:937–950, 2013.

[104] H. Li and J. Gui. Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317, 2006.

[105] M. Lindquist. The statistical analysis of fMRI data. *Statistical Science*, 23(4):439–464, 2008.

[106] M. Lindquist, Y. Xu, M. Nebel, and B. Caffo. Evaluating dynamic bivariate correlations in resting-state fMRI: A comparison study and a new approach. *NeuroImage*, 101:531–546, 2014.

[107] H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. In *Advances in Neural Information Processing Systems*, pages 1432–1440, 2010.

[108] R. Lorenz, R. P. Monti, A. Hampshire, Y. Koush, C. Anagnostopoulos, A. Faisal, D. Sharp, G. Montana, R. Leech, and I. Violante. Towards tailoring non-invasive brain stimulation using real-time fMRI and Bayesian optimization. In *Pattern Recognition in NeuroImaging (PRNI), 2016 International Workshop on*. IEEE, 2016.

[109] R. Lorenz, R. P. Monti, I. Violante, C. Anagnostopoulos, A. Faisal, G. Montana, and R. Leech. The automatic neuroscientist: A framework for optimizing experimental design with closed-loop real-time fMRI. *NeuroImage*, 129:320–334, 2016.

[110] R. Lorenz, R. P. Monti, I. Violante, A. Faisal, C. Anagnostopoulos, R. Leech, and G. Montana. Stopping criteria for boosting automatic experimental design using real-time fMRI with Bayesian optimization. *NIPS workshop on Machine Learning in Neuroimaging*, 2015.

[111] N. T. Markov, M. Ercsey-Ravasz, D. C. Van Essen, K. Knoblauch, Z. Toroczkai, and H. Kennedy. Cortical high-density counterstream architectures. *Science*, 342(6158):578–593.

[112] G. Marrelec, J. Kim, J. Doyon, and B. Horwitz. Large-scale neural model validation of partial correlation analysis for effective connectivity investigation in functional MRI. *Human Brain Mapping*, 30(3):941–950, 2009.

[113] J. B. Mattingley, M. Husain, C. Rorden, C. Kennard, and J. Driver. Motor role of human inferior parietal lobe revealed in unilateral neglect patients. *Nature*, 392(6672):179–182, 1998.

[114] M. McKerns, L. Strand, T. Sullivan, A. Fang, and M. Aivazis. Building a framework for predictive science. In *10th Python in Science Conference*, 2011.

[115] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007.

[116] B. McWilliams, C. Heinze, N. Meinshausen, G. Krummenacher, and H. Vanchinathan. LOCO: Distributing ridge regression with random projections. *arXiv preprint arXiv:1406.3469*, 2014.

[117] N. Meinshausen and P. Bühlmann. High-dimensional Graphs and Variable Selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

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

[119] X. Meng and D. Van Dyk. Fast EM-type implementations for mixed effects models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(3):559–578, 1998.

[120] R. P. Monti, C. Anagnostopoulos, and G. Montana. Learning population and subject-specific brain connectivity networks via mixed neighborhood selection. *arXiv preprint arXiv:1512.01947*, 2015.

[121] R. P. Monti, C. Anagnostopoulos, and G. Montana. A framework for adaptive regularization in streaming lasso models. *arXiv preprint arXiv:1610.09127*, 2016.

[122] R. P. Monti, P. Hellyer, D. Sharp, R. Leech, C. Anagnostopoulos, and G. Montana. Estimating time-varying brain connectivity networks from functional MRI time series. *NeuroImage*, 103:427–443, 2014.

[123] R. P. Monti, R. Lorenz, R. Braga, C. Anagnostopoulos, R. Leech, and G. Montana. Real-time estimation of dynamic functional connectivity networks. *Human Brain Mapping*, 38(1):202–220, 2017.

[124] R. P. Monti, R. Lorenz, P. Hellyer, R. Leech, C. Anagnostopoulos, and G. Montana. Graph embeddings of dynamic functional connectivity reveal discriminative patterns of task engagement in HCP data. In *Pattern Recognition in NeuroImaging (PRNI), 2015 International Workshop on*, pages 1–4. IEEE, 2015.

[125] R. P. Monti, R. Lorenz, P. Hellyer, R. Leech, C. Anagnostopoulos, and G. Montana. Decoding time-varying functional connectivity networks via linear graph embedding methods. *Frontiers in Computational Neuroscience*, 11, 2017.

[126] R. P. Monti, R. Lorenz, R. Leech, C. Anagnostopoulos, and G. Montana. Text-mining the NeuroSynth corpus using Deep Boltzmann Machines. In *Pattern Recognition in NeuroImaging (PRNI), 2016 International Workshop on*. IEEE, 2016.

[127] S. Mueller, D. Wang, M. Fox, T. Yeo, J. Sepulcre, M. Sabuncu, R. Shafee, J. Lu, and H. Liu. Individual variability in functional connectivity architecture of the human brain. *Neuron*, 77(3):586–595, 2013.

[128] M. Narayan and G. Allen. Randomized approach to differential inference in multi-subject functional connectivity. In *Pattern Recognition in Neuroimaging (PRNI), 2013 International Workshop on*, pages 78–81. IEEE, 2013.

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

[130] R. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

[131] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[132] B. Ng, G. Varoquaux, J. Poline, and B. Thirion. A novel sparse graphical approach for multimodal brain connectivity inference. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 707–714. Springer, 2012.

[133] J. Nielsen, B. Zielinski, T. Fletcher, A. Alexander, N. Lange, E. Bigler, J. Lainhart, and J. Anderson. Multisite functional connectivity MRI classification of autism: ABIDE results. 2013.

[134] J. Nocedal and S. J. Wright. *Numerical Optimisation*. Springer, 2006.

[135] A. S. Pandit, P. Expert, R. Lambiotte, V. Bonnelle, R. Leech, F. E. Turkheimer, and D. J. Sharp. Traumatic brain injury impairs small-world topology. *Neurology*, 80(20):1826–1833, 2013.

[136] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[137] J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

[138] K. Petersen and M. Pedersen. The Matrix Cookbook. *Technical University of Denmark*, 2008.

[139] J. Pinheiro and D. Bates. *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media, 2000.

[140] R. A. Poldrack, J. A. Mumford, and T. E. Nichols. *Handbook of Functional MRI Data Analysis*. Cambridge University Press, 2011.

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

[142] J. C. Príncipe, W. Liu, and S. Haykin. *Kernel Adaptive Filtering: A Comprehensive Introduction*, volume 57. John Wiley & Sons, 2011.

[143] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[144] J. Richiardi, D. Van De Ville, K. Riesen, and H. Bunke. Vector space embedding of undirected graphs with fixed-cardinality vertex sequences for classification. In *International Conference on Pattern Recognition*, pages 902–905. IEEE, 2010.

[145] W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–101, 1959.

[146] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, 35(3):1012–1030, 2007.

[147] M. Rubinov and O. Sporns. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage*, 52(3):1059–1069, 2010.

[148] S. Ruiz, K. Buyukturkoglu, M. Rana, N. Birbaumer, and R. Sitaram. Real-time fMRI brain computer interfaces: self-regulation of single brain regions to networks. *Biological Psychology*, 95:4–20, 2014.

[149] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[150] U. Sakoglu, G. D. Pearlson, K. A. Kiehl, Y. M. Wang, A. M. Micheal, and V. D. Calhoun. A method for evaluating dynamic functional network connectivity and task modulation: application to Schizophrenia. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 23:351–366, 2010.

[151] M. Salvador, J. Gallizo, and P. Gargallo. A dynamic principal components analysis based on multivariate matrix normal dynamic linear models. *Journal of Forecasting*, 22(6-7):457–478, 2003.

[152] T. Satterthwaite, D. Wolf, J. Loughead, K. Ruparel, M. Elliott, H. Hakonarson, R. Gur, and R. Gur. Impact of in-scanner head motion on multiple measures of functional connectivity: relevance for studies of neurodevelopment in youth. *NeuroImage*, 60(1):623–632, 2012.

[153] J. Schelldorfer, P. Bühlmann, and S. van de Geer. Estimation for high-dimensional linear mixed-effects models using l1-penalization. *Scandinavian Journal of Statistics*, 38(2):197–214, 2011.

[154] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[155] N. Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

[156] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. CRC press, 1986.

[157] S. Smith, K. Miller, G. Salimi-Khorshidi, M. Webster, C. Beckmann, T. Nichols, J. Ramsey, and M. Woolrich. Network modelling methods for fMRI. *NeuroImage*, 54(2):875–891, 2011.

[158] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. J. Behrens, H. Johansen-Berg, P. R. Bannister, De Luca M., I. Drobnjak, D. E. Flitney, R. K. Niazy, Saunders J., J. Vickers, Y. Zhang, N. De Stefano, M. J. Brady, and P. M.

Matthews. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23(1):208 – 219, 2004.

[159] S. M. Smith, K. L. Miller, S-K Gholamrez, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for fMRI. *NeuroImage*, 54(2):875 – 891, 2011.

[160] M. Sourty, L. Thoraval, D. Roquet, J. Armspach, J. Foucher, and F. Blanc. Identifying dynamic functional connectivity changes in dementia with lewy bodies based on product hidden markov models. *Frontiers in Computational Neuroscience*, 10, 2016.

[161] O. Sporns. *Discovering The Human Connectome*. MIT Press, 2012.

[162] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 8(9):418–425, 2004.

[163] N. Städler and S. Mukherjee. Penalized estimation in high-dimensional hidden markov models with state-specific graphical models. *The Annals of Applied Statistics*, 7(4):2157–2179, 2013.

[164] K. E. Stephan, C. Hilgetag, G. Burns, M. A. O'Neill, M. P. Young, and R. Kotter. Computational analysis of functional connectivity between areas of primate cerebral cortex. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 355(1393):111–126, 2000.

[165] F. T. Sun, L. M. Miller, A. A. Rao, and M. D'Esposito. Functional connectivity of cortical networks involved in bimanual motor sequence learning. *Cerebral Cortex*, 17(5):1227–1234, 2007.

[166] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 58(1):267–288, 1996.

[167] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[168] M. Van Den Heuvel and H. Pol. Exploring the brain network: a review on resting-state fMRI functional connectivity. *European Neuropsychopharmacology*, 20(8):519–534, 2010.

[169] C.J. Van Rijsbergen. *The geometry of information retrieval*. Cambridge University Press, 2004.

[170] L. Vandenberghe, S. Boyd, and S. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

[171] G. Varoquaux and C. Craddock. Learning and comparing functional connectomes across subjects. *NeuroImage*, 80:405–415, 2013.

[172] G. Varoquaux, A. Gramfort, J. Poline, and B. Thirion. Brain covariance selection: better individual functional connectivity models using population prior. In *Neural Information Processing Systems*, pages 2334–2342, 2010.

[173] S. Vossel, J. Geng, and G. Fink. Dorsal and ventral attention systems distinct neural circuits but collaborative roles. *The Neuroscientist*, 20(2):150–159, 2014.

[174] G. Wallis, M. Stokes, H. Cousijn, M. Woolrich, and A. Nobre. Frontoparietal and cingulo-opercular networks play dissociable roles in control of working memory. *Journal of Cognitive Neuroscience*, 27(10):2019–2034, 2015.

[175] L. Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[176] L. Wasserman and K. Roeder. High dimensional variable selection. *Annals of Statistics*, 37(5A):2178, 2009.

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

[178] N Weiskopf. Real-time fMRI and its application to neurofeedback. *NeuroImage*, 62(2):682–692, 2012.

[179] S. Weston and R. Calaway. Getting started with doparallel and foreach. 2015.

[180] Anne L Wheeler, M Mallar Chakravarty, Jason P Lerch, Jon Pipitone, Zafiris J Daskalakis, Tarek K Rajji, Benoit H Mulsant, and Aristotle N Voineskos. Disrupted prefrontal interhemispheric structural coupling in schizophrenia related to working memory performance. *Schizophrenia Bulletin*, 40(4):914–924, 2014.

[181] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley and Sons, 1990.

[182] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.

[183] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[184] Z. Yao, B. Hu, Y. Xie, F. Zheng, G. Liu, X. Chen, and W. Zheng. Resting-state time-varying analysis reveals aberrant variations of functional connectivity in autism. *Frontiers in Human Neuroscience*, 10, 2016.

[185] T. Yarkoni, R. Poldrack, T. Nichols, D. Van Essen, and T. Wager. Large-scale automated synthesis of human functional neuroimaging data. *Nature Methods*, 8(8):665–670, 2011.

[186] T. Yeo, F. Krienen, J. Sepulcre, M. Sabuncu, D. Lashkari, M. Hollinshead, J. Roffman, J. Smoller, L. Zöllei, and J. Polimeni. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3):1125–1165, 2011.

[187] S. Zhou, J. Lafferty, and L. Wasserman. Time Varying Undirected Graphs. *Machine Learning*, 80:295–319, 2010.

[188] A. Zilverstand, B. Sorger, J. Zimmermann, A. Kaas, and R. Goebel. Windowed correlation: a suitable tool for providing dynamic fMRI-based functional connectivity neurofeedback on task difficulty. *PLoS One*, 9(1):e85929, 2014.

[189] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

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

[191] X. Zuo, A. Di Martino, C. Kelly, Z. Shehzad, D. Gee, D. Klein, X. Castellanos, B. Biswal, and M. Milham. The oscillating brain: complex and reliable. *NeuroImage*, 49(2):1432–1445, 2010.

Appendices

Appendix A

Derivative of adaptive filtering gradient

In this section we derive the gradient of the adaptive filtering objective function detailed in equation (4.9). The results presented are taken from [5]. The log-likelihood for unseen observation, X_{t+1} is given by

$$C(X_{t+1}) = C((X_{t+1}; \bar{x}_t, S_t) = -\frac{1}{2} \log \det(S_t) - \frac{1}{2}(X_{t+1} - \bar{x}_t)^T S_t^{-1}(X_{t+1} - \bar{x}_t). \quad \text{(A.1)}$$

The approach taken here is to approximate the derivative of $C(X_{t+1})$ with respect to adaptive forgetting factor r_t by calculating the exact derivative of $\mathcal{L}_{t+1}$ with respect to a fixed forgetting factor r. Then under the assumption that changes in r_t occur sufficiently slowly, this will serve as a good approximation to the derivative of $\mathcal{L}_{t+1}$ with respect to r_t.

We begin by noting the following results from [138]:

$$\frac{\partial \log \det (S_t)}{\partial r} = \text{Trace } (S_t^{-1} S_t') \quad \text{(A.2)}$$

$$\frac{\partial (S_t^{-1})}{\partial r} = -S_t^{-1} S_t' S_t^{-1}. \quad \text{(A.3)}$$

Moreover, we note that we do not need to explicitly invert S_t. By noting that S_t is a rank one update of S_{t-1} we are able to directly obtain S_t^{-1} using the Sherman-Woodbury formula.

Further, from equations (4.2), (4.3), (4.4) and (4.5) we can see that:

$$\bar{x}'_t = \left(1 - \frac{1}{\omega_t}\right)\bar{x}'_{t-1} + \frac{\omega'_t}{\omega_t^2}\left(X_t - \bar{x}_{t-1}\right) \tag{A.4}$$

$$\omega'_t = r_{t-1}\omega'_{t-1} + \omega_t \tag{A.5}$$

$$\Pi'_t = \left(1 - \frac{1}{\omega_t}\right)\Pi'_{t-1} + \frac{\omega'_t}{\omega_t^2}\left(X_t X_t^T - \Pi_{t-1}\right) \tag{A.6}$$

$$S'_t = \Pi'_t - \bar{x}'_t\bar{x}_t^T - \bar{x}_t(\bar{x}'_t)^T, \tag{A.7}$$

where we have used the notation A' to denote the derivative of a vector or matrix A with respect to r. Using the results from equations (A.2) to (A.7) we can directly differentiate the $\mathcal{L}_{t+1}$ to obtain equation (4.9).

Appendix B

Alternative methods

In this appendix we provide details regarding alternative algorithms which have been presented throughout the thesis as benchmarks. In Appendix B.1 we detail the Dynamic Connectivity Regression (DCR) algorithm [38] which is provided as a benchmark for the SINGLE algorithm in Chapter 3. In Appendix B.2 we present a stability approach to estimating multiple related GGMs, which is presented as a benchmark in Chapter 7. Such an approach is inspired by the R^3 method of [129].

B.1 Dynamic connectivity regression algorithm

In this section a brief overview is provided of the Dynamic Connectivity Regression (DCR) algorithm proposed by [38]. The principal objective of the DCR algorithm is to provide a data-driven methodology through which to partition the experimental time-course associated with an fMRI experiment. In this manner, the data is split into distinct temporal intervals, each of which is associated with an estimated covariance structure. A greedy procedure based upon change-point detection is employed to segment the data.

The DCR algorithm begins by estimating the covariance structure (i.e., the functional connectivity) for the entire dataset under an ℓ_1 regularization penalty. The choice of regularization parameter is selected by minimizing the Bayesian Information Criterion (BIC).

The data, $\{X_i \in \mathbb{R}^{1\times p} : i = 1, \ldots, T\}$, is then segmented using a greedy partitioning scheme. The DCR algorithm proceeds to partition the data into subsets $A_\gamma = \{X_i : i =$

$1, \ldots, \gamma\}$ and $B_\gamma = \{X_i : i = \gamma + 1, \ldots, T\}$ for $\gamma \in \{\Delta + 1 \ldots, T - \Delta\}$. Thus Δ represents the minimum number of observations between change-points. For each of these partitions a network is estimated for A_γ and B_γ and their joint BIC is noted. This step therefore involves $\mathcal{O}(n)$ iterations of the Graphical lasso.

Subsequently, the value of γ resulting in the greatest reduction in BIC relative to the global network is proposed as a change-point. The DCR algorithm then employs a bootstrap permutation test in order to verify the statistical significance of such a change-point. Under the null hypothesis, no change-point occurs and observations are independent and identically distributed. The data can therefore be repeatedly permuted in order to obtain a non-parametric pivotal quantity. In order to account for the auto-correlated nature of fMRI data, a block bootstrap permutation test is employed. The $1 - \frac{\alpha}{2}$ and $\frac{\alpha}{2}$ quantiles of the permutation distribution are calculated and interpreted as confidence bounds, allowing for the proposed change-point to be accepted or rejected in a traditional hypothesis test framework. The procedure described above is repeated recursively until no further partitions are reported.

B.1.1 Computational cost

We begin by noting that the computational complexity of the Graphical lasso is $\mathcal{O}(p^3)$. While it is possible to reduce the computational complexity in some special cases we do not consider this below.

The first stage of the DCR algorithm involves the proposal of a change-point which will be subsequently tested using a non-parametric hypothesis test. Points are proposed based on an extensive search across all candidate change-points, thereby incurring a computational cost of $\mathcal{O}(np^3)$.

In order to check the statistical significance of the proposed change-point a block bootstrap permutation permutation test is performed. This step involves a further b iterations of the Graphical lasso where b is the number of bootstrap permutations performed. As a result this step has a computational complexity of $\mathcal{O}(bp^3)$.

This procedure is repeated until all significant change-points have been reported. We therefore conclude that the computational complexity of the DCR algorithm is $\mathcal{O}((n +$

$b)p^3)$.

B.2 A stability approach to estimating a cohort of related networks

In this section we briefly overview a stability selection (i.e., bootstrap) approach for studying multiple, related graphical models. This approach is inspired by the R^3 approach proposed in [129], however, it has a fundamentally different objective. As a result, some adjustments are introduced.

As in the R^3 method, this approach is based upon resampling, randomized penalizations and random effects. The method, described in Algorithm 5, proceeds by iteratively obtaining bootstrapped estimates of covariance structure for each subject. These results are subsequently incorporated into a Beta-Binomial random effects model. Each of these steps is described below, for further discussion and motivation of these steps we refer the reader to [128] and [129].

B.2.1 Resampling

In order to obtain reliable estimates of covariance structure the bootstrap is employed; resulting in B bootstrap estimates of connectivity structure per subject. Recall that the dataset for the ith subject, $X^{(i)} \in \mathbb{R}^{n \times p}$, consists of n observations across p nodes. At the bth bootstrap iteration, n observations are sampled with replacement in order to form a bootstrapped dataset, $X^{(i),b} \in \mathbb{R}^{n \times p}$, which is subsequently used to obtain an estimate for the covariance structure using the Graphical lasso, as described in Section B.2.2.

B.2.2 Randomization penalization

In order to alleviate possible bias introduced by the use of an ℓ_1 penalty we employ randomized penalization techniques, an approach first introduced by [118]. The objective of randomized penalization schemes is to reduce the influence of inclusion/exclusion of any edge on the presence of remaining edges. Thus when estimating the network for the bth bootstrap sample, a random, symmetric penalty matrix, $\Lambda^{(i),b} \in \mathbb{R}^{p \times p}$, is employed.

In order to obtain $\Lambda^{(i),b}$, we first estimate the regularization parameter for the ith subject using the StARS method of [107]. This is performed only once using the entire (non-bootstrapped) dataset, $X^{(i)}$, and is denoted by $\lambda^{(i)} \in \mathbb{R}$. The randomized penalization matrix is defined as follows:

$$\left(\Lambda^{(i),b}\right)_{k,j} = \left(\Lambda^{(i),b}\right)_{j,k} = \lambda^{(i)} + c\lambda^{(i)}_{max} W_{j,k} \quad \forall j < k, \tag{B.1}$$

where $\lambda^{(i)}_{max}$ is the value of sparsity parameter leading to a null model and $W \in \{-1, +1\}^{p\times p}$ is defined as:

$$W_{j,k} = \begin{cases} +1, & \text{w.p. } 0.5 \\ -1, & \text{w.p. } 0.5 \end{cases}.$$

We are then able to obtain a penalized estimate of the precision as follows:

$$\Theta^{(i),b} = \underset{\Theta}{\operatorname{argmin}} \left\{ -\log\det\Theta + \operatorname{trace}\left(\frac{1}{n} X^{(i),b^T} X^{(i),b}\Theta\right) + ||\Lambda^{(i),b} \circ \Theta||_1 \right\}, \tag{B.2}$$

where $\circ$ denotes element-wise multiplication.

B.2.3 Random effects

Finally, we look to formally quantify the presence or absence of edges at a population level. In order to achieve this a Beta-Binomial model is employed. For the ith subject we treat the presence of any given edge at each bootstrap iteration as a Binomial random variable. We thus define $Y^{(i),B} \in \mathbb{R}^{p\times p}$ such that

$$Y^{(i),B}_{j,k} = \frac{1}{B}\sum_{b=1}^{B} \mathbb{I}\left(\Theta^{(i),b}_{j,k} \neq 0\right). \tag{B.3}$$

Following [129], $Y^{(i),B}_{j,k}$ is modeled as follows:

$$Y^{(i),B}_{j,k} | \mu^{(i)}_{j,k} \sim \text{Binomial}\left(\mu^{(i)}_{j,k}, B\right) \quad \text{and} \quad \mu^{(i)}_{j,k} \sim \text{Beta}\left(\mu^{pop}_{j,k}, \rho^{pop}_{j,k}\right), \tag{B.4}$$

where $\mu_{j,k}^{pop}$ is the population mean and $\rho_{j,k}^{pop}$ the variance. They can be estimated as follows:

$$\mu_{j,k}^{pop} = \frac{1}{N}\sum_{i=1}^{N} Y_{j,k}^{(i),B} \tag{B.5}$$

$$\rho_{j,k}^{pop} = \frac{B}{B-1}\,\frac{\sum_{i=1}^{N}\left(\mu_{j,k}^{pop} - Y_{j,k}^{(i),B}\right)^2}{\mu_{j,k}^{pop}(1-\mu_{j,k}^{pop})(N-1)} - \frac{1}{B-1} \tag{B.6}$$

These parameters are subsequently used to infer population networks (using μ^{pop}) as well as report highly variable edges (using ρ^{pop}). Pseudo-code for the *Stability* approach is provided in Algorithm 5.

Algorithm 5: Stability algorithm for estimating multiple related GGMs

Input: Data across N subjects, $\{X^{(i)}\}$, number of bootstrap samples to perform, B.

begin
  for $i \in \{1, \ldots, N\}$ **do**
    Select $\lambda^{(i)}$ using the StARS method [107]
    for $b \in \{1, \ldots, B\}$ **do**
      Obtain $X^{(i),b}$ be sampling n times with replacement from $X^{(i)}$
      Set randomization penalization matrix, $\Lambda^{(i),b}$, as in equation (B.1)
      Estimate penalization precision matrix, $\Theta^{(i),b}$, as in equation (B.2)
  Estimate μ^{pop}, ρ^{pop} using equations (B.5) and (B.6)
return μ^{pop}, ρ^{pop}

Appendix C

Network simulation methods

Producing synthetic data where the true underlying covariance structure is known is fundamental to providing an empirical validation of any algorithm. In this section we focus on a host of distinct methods through which to generate synthetic network structures which accurately reproduce many of the empirical properties reported in fMRI datasets.

We begin by discussing the wide variety of algorithms which have been proposed through which to simulate individual networks in Appendix C.1. Each of the algorithms discussed demonstrates some of the properties known to be present in functional connectivity networks. A brief overview is provided for three such algorithms. A related problem corresponds to simulating multiple related random networks. This is an issue which has not been addressed in the literature to date. As a result, a novel algorithm is proposed to this end in Appendix C.2. This algorithm is motivated by an exploratory data analysis of resting state fMRI, which is also presented.

C.1 Simulating individual networks

In this section we outline three algorithms which are frequently employed to simulate functional connectivity networks and which are subsequently employed throughout this thesis. They correspond to the Erdős-Rényi, Barabási and Albert and Watts-Strogatz models. We briefly detail each model and provide a detail the properties displayed by the resulting simulated networks.

C.1.1 Erdős-Rényi model

Perhaps the simplest and most intuitive form of random network model is the Erdős-Rényi model [56]. Such a model is parameterized by a single parameter, $\alpha \in [0, 1]$, which dictates the probability of an edge being present between any pair of nodes. As a result, all edges are equally likely to occur. Such a model therefore does not demonstrate many of the empirical properties typically reported of functional connectivity networks such as high-clustering across nodes and the presence of hub nodes [13, 30, 54]. In particular, the use of such a model results in random networks with a low clustering coefficient and fails to generate graphs where the degree distribution follows a power law. Nonetheless, such a model in included throughout the simulations presented in this thesis as it corresponds to the simplest and most widely studied random network.

It follows that the value of α dictates the density of the simulated network. Large values of α (i.e., close to 1) will result in simulated networks with a high edge density while the converse is true if α is close to 0. The left panel of Figure [C.1] contains an example visualization of a random network generated using the Erdős-Rényi model. There is no clear structure present across the nodes due to the fact that edges are added independently.

C.1.2 Barabási-Albert model

A more realistic model for generating synthetic networks is the Barabási and Albert model [11]. Such a model is able to produce scale-free networks where the degree distribution of nodes follows a power law. From a biological perspective, this implies that there exists a small number of hub regions which have access to most other regions [54]. This is in contrast to Erdős-Rényi random graphs, which may be conceptually simple but fail to generate networks where the degree distribution follows a power law.

The generating mechanism behind such a model is fundamentally different to that of the Erdős-Rényi model. While the latter model proceeded by randomly added edges with a fixed probability, the Barabási and Albert model is based on a preferential attachment mechanism. In this mechanism, nodes are iteratively added to the network. At each iteration a new node is added and edges between to the previous nodes are added with probability that is proportional to the current number of edges at each node. This encourages

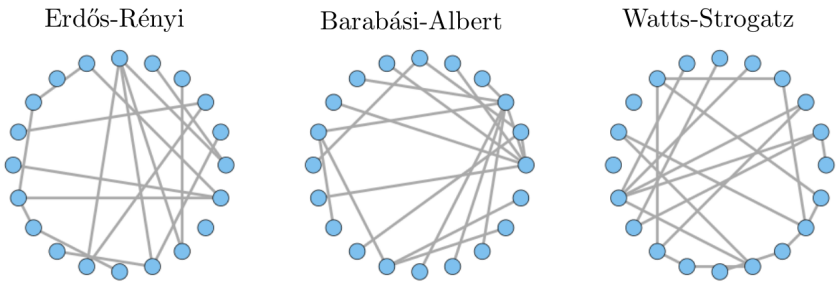


Figure C.1: Visualizations of three random graph generated each of the three simulation models discussed. The left network visualizes a sample from the Erdős-Rényi model, where edges are each independently added with a fixed probability. As a result there is no clear edge structure present. The middle panel visualizes a Barabási-Albert model where a preferential attachment mechanism is employed. This yields networks with hubs but where there is a high average path length across nodes. Finally, on the right panel contains a random network generated according to the Watts-Strogatz model. Such a model is able to generate random networks where there is a high clustering across nodes together with a low average path length.

the presence of hub nodes, which contain a large number of edges and therefore continue to attract more edges as the network grows. The middle panel of Figure [C.1] shows a visualization produced using the Barabási and Albert model. We note the presence of hub nodes on the right of the network which contain the majority of edges across the network while connectivity across the remaining nodes is sparse.

C.1.3 Watts-Strogatz model

While the Barabási and Albert model is able to produce random networks where the degree distribution follows a power law, the resulting networks also result in large average path lengths between nodes. This is in contrast to the empirical properties of brain networks, which are reported to display low average path lengths across nodes [162, 164]. In particular, brain networks are characterized by a low average path length together with a high clustering coefficient across nodes. Networks displaying such characteristics are often said to display a small-world topology [13].

The Watts-Strogatz model is one candidate method which can be employed to simulate networks which display a small-world topology [177]. The model is initialized with a

regular lattice, which consists of a graph where each node is connected to its K nearest neighbors. Each of the edges in then randomly *re-weired* with a fixed probability $\beta \in [0, 1]$. It follows that as β tends to 1 the majority of edges are randomly re-wired and an Erdős-Rényi random graph is obtained. Conversely, as β tends towards 0 fewer edges are re-wired and a highly structured lattice network is obtained. For intermediate value of β, such an algorithm is able to generate random networks where there is a tendency for nodes to form clusters, formally referred to as a high clustering coefficient. This is desirable as both anatomical as well as the functional brain networks have been reported as exhibiting such a network topology [13, 162]. The right panel of Figure [C.1] contains a random network produced via the Watts-Strogatz model. We note there are hub nodes present which contain a large number of edges. In addition to this, there is also a low average path length across nodes. This is a result of the lattice initialization which is employed.

C.2 Simulating a cohort of networks

In order to validate the methodology presented in Chapter 7, it is important to be able to simulate multiple related networks which recreate many of the reproducible properties often encountered in fMRI data analysis. However, the problem of simulating networks for multiple related subjects has not been thoroughly considered in the literature. While there is a vast literature on the properties which can be expected for subject-specific networks (see e.g., [30]), there is limited knowledge of the behavior which can be expected across a cohort of related subjects. In this work we look to address this issue by empirically studying resting state data from healthy subjects taken from the ABIDE consortium [48].

In this section we present an exploratory data analysis of the ABIDE dataset presented in Section 7.3.1. This exploratory analysis is employed to propose a novel algorithm through which to generate multiple related random networks.

C.2.1 Exploratory data analysis

The data employed consisted of a resting state scan for $N = 43$ healthy subjects taken from the USM site. For each subject, $n = 230$ BOLD measurements were collected over $p = 92$ regions. We consequently estimated functional connectivity networks for each

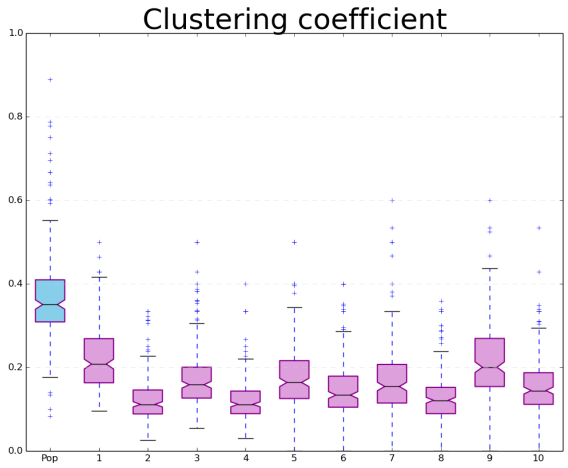


Figure C.2: Clustering coefficients for the population network (in blue) as well as for 10 randomly selected subjects. We note there is a clear drop in clustering coefficient from the population network to the subject-specific networks.

subject independently while employing a Graphical lasso penalty. A stability selection procedure was employed, whereby a block bootstrap was used to resample the data for each subject multiple times. Randomized penalization was also employed to further correct for systematic bias in model selection. At each iteration, selected edges were recorded. This allowed us to obtain a network per subject by selecting only edges that where consistently present across all iterations. A population network was also obtained by selecting the edges that where consistently present across all subjects.

The properties of the resulting networks where studied in the hope of obtaining notable properties which could subsequently be re-created synthetically. In particular, we chose to focus on graph theoretic measures as these can be easily interpreted and calculated on simulated data [147]. Specifically, the clustering coefficient was studied across all networks. This provides a measure of how tightly nodes in a network tend to group together, thereby expressing the cohesiveness of a network [12].

The results, shown in Figure [C.2], show that the clustering coefficient is significantly larger in the population network when compared the each of the subject-specific networks. We hypothesize that this is a manifestation of the fact that there is a highly structured population network present. We further hypothesize that it is the large inter-subject variability

Algorithm 6: Generate multiple related random networks

Input: Number of nodes p, number of subjects N, size of random effects network $e_{ran} = |\tilde{E}|$, a random effects edge probability $\tau \in [0, 1]$ and connectivity strength $r \in \mathbb{R}_+$

begin
  Simulate E^{pop} according to [11] model;
  Build Θ^{pop} by uniformly sampling edge weights from the interval $[-r, -\frac{r}{2}] \cup [\frac{r}{2}, r]$;
  Simulate $\tilde{E}$ according to [56] model with e_{ran} edges;
  for $i \in \{1, \ldots, N\}$ **do**
    for *each edge* (j, k) **do**
      if $(j, k) \in \tilde{E}$ **then**
        $E^{(i)} \leftarrow E^{(i)} \cup (j, k)$ with probability τ
    Randomly select edge weights and signs for $\Theta^{(i)}$
return E^{pop}, $\tilde{E}$, $\{E^{(i)}\}$ *and* Θ^{pop}, $\{\Theta^{(i)}\}$

which accounts for some of the drop in clustering coefficient at the subject-specific level.

C.2.2 Proposed algorithm

In order to generate synthetic networks we demonstrate the aforementioned empirical properties we proceed as follows. We begin by first simulating a population network according to the Barabási and Albert model [11]. This results in a highly structure population network which also demonstrates many of the properties known to be present in neuroimaging data (e.g., power law distribution and the presence of hub nodes). A subset of highly variable edges, denoted by $\tilde{E}$, is then randomly selected according to the Erdős-Rényi model [56]. For each subject, each edge in $\tilde{E}$ is added to the subject-specific network with a given probability, τ. This yields variable edges that are only present across a subset of the population. The introduction of these random edges serves to reduce the clustering coefficient of the subject-specific network, thereby recreating the properties observed in our exploratory analysis. Pseudo-code is provided in Algorithm 6.

Ensuring positive definiteness

Through algorithm 6 we are able to simulate a population precision, Θ^{pop}, together with subject-specific deviations, $\Theta^{(i)}$. We define the precision for each subject to be $\Theta^{pop} + \Theta^{(i)}$, however, care must be taken to ensure this sum is positive-definite. In this work we follow [42] and ensure the subject-specific precision matrices are positive definite by rescaling the matrix. Formally, each off-diagonal element is divided by the sum of the absolute values of all off-diagonal elements in its row. This yields a non-symmetric matrix which is subsequently averaged with its transpose.

Appendix D

Sensitivity analysis for Mixed Neighbourhood Selection

D.1 Sensitivity analysis

The proposed method is based upon several assumptions, the most significant of which is the assumption of that observations across all subjects follow a multivariate Gaussian distribution. While such an assumption is typically made in the context of fMRI data [105], it is important to acknowledge that deviations from normality may impact performance of the proposed method. This problem has been studied extensively by [61], however, they only consider the task of estimating a single GGM.

In this section we perform a sensitivity analysis of the proposed method. We follow experimental setup of [61] and simulate data according to a multivariate t-distribution. This is achieved by first simulating multivariate Gaussian random variables, $X \sim \mathcal{N}(0, \Sigma)$ for some given covariance $\Sigma \in \mathbb{R}^{p \times p}$. A Gamma random variable, $\tau \sim \Gamma(v/2, v/2)$, is then simulated independently. It then follows that $X/\sqrt{\tau}$ follows a p-dimensional t-distribution with τ degrees of freedom. As in the Gaussian case, the covariance structure is specified by Σ^{-1}. In this manner, we are able to generate datasets following multivariate t-distributions where the covariance structure is known.

The sensitivity analysis performed in this section proceeds as follows: data is first simulated as described in Section 6.2. This yields a dataset for each subject, $X^{(i)} \in \mathbb{R}^{n \times p}$, fol-

lowing a multivariate Gaussian distribution. An n-dimensional vector, $\tau^{(i)}$, is the generated independently for each subject where each entry follows a Gamma($v/2, v/2$) distribution. Each row of $X^{(i)}$ is subsequently divided by the square-root of the corresponding entry of $\tau^{(i)}$, as described above. As a result, we obtain data for each subject following a multivariate t-distribution with a known covariance structure. The proposed method, together with various alternatives, was then employed in an attempt to recover underlying GGMs.

The procedure described was employed to simulate synthetic data for a cohort of $N = 10$ subjects. The number of nodes was fixed at $p = 50$. Data was simulated with varying numbers of observations per subject, $n \in \{50, 100, 200\}$. The degrees-of-freedom of the t-distribution was fixed at $v = 3$ throughout this study. We note that the simulated data in this sensitivity analysis shares the same covariance structure as data employed for the simulation study of Section 7.2. The only difference is the nature in which multivariate observations were generated. This allows us to directly compare the results of the sensitivity analysis with those presented in Section 7.2.

As in Section 7.2, the MNS algorithm was run with $\alpha = 0.25$ while sparsity parameter λ varied as described in equations (7.11) and (7.12). The same parameterization was employed for the JGL-Group algorithm with $\alpha = 0.15$ selected. In the case of the JGL-Fused algorithm, $\lambda_2 = 0.2$ was employed. Finally, the *Stability* algorithm was run with $B = 10,000$ bootstrap iterations per subject and $c = 0.25$.

Results are shown in Figure [D.1] and detailed results are provided in Table [D.1]. As expected, the performance of all algorithms considered was adversely affected by the departure from Gaussianity. This is most notable in the estimation of variable edges, where the performance of both the MNS and *Stability* algorithms suffered. While we attribute this drop in performance to the added variability and heavy-tailed nature of the data generation mechanism, it is reassuring to note that proposed algorithm is still able to identify variable edges with high accuracy.

In the context of recovering population networks, all algorithms proved to be relatively robust. As in the previous simulation, the proposed MNS algorithm is able to comfortably outperform alternative approaches. Finally, a drop in performance was also observed across all methods in the context of recovering the covariance structure for each subject. However, the proposed method is still able to compete at a high level. We attribute this behavior to

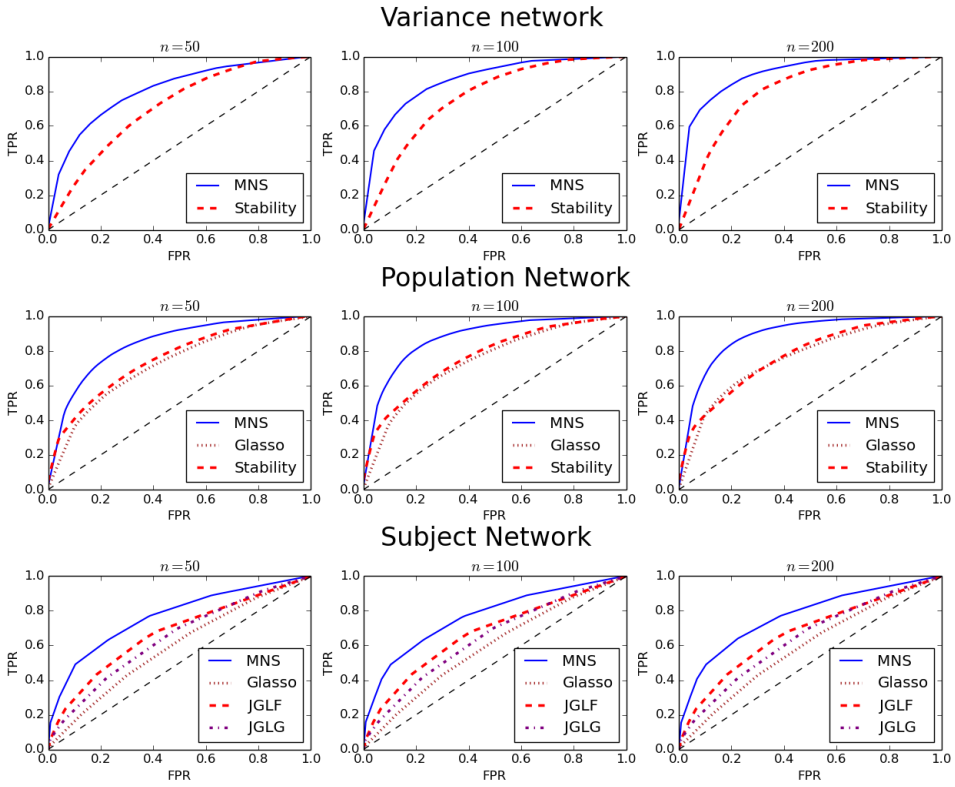


Figure D.1: Sensitivity analysis results for all five algorithms across all tasks. Data were generated following multivariate t-distributions with known covariance structure. Recovery of variable edges is considered in the top panel, population network recovery in shown in the middle panel and finally the bottom panel shows subject-specific network recovery. This simulation was performed with $p = 50$ nodes and $n \in \{50, 100, 200\}$ observations.

the fact that the covariance model underlying the MNS algorithm explicitly models heterogeneity over subjects. This allows the MNS algorithm to better tolerate contaminated data.

In conclusion, the sensitivity analysis presented provides compelling empirical evidence that while the proposed method may be adversely affected when the assumption of Gaussianity is not valid, its performance remains robust. It is also important to note that the performance of the proposed method remains competitive when compared to alternative algorithms, whose performance is equally affected by non-Gaussian data.

Algorithm	n	Population		Subject		Variance	
		TPR	FPR	TPR	FPR	TPR	FPR
MNS	50	0.60	0.11	0.60	0.20	0.49	0.08
	100	0.71	0.11	0.62	0.21	0.60	0.08
	200	0.76	0.10	0.73	0.22	0.70	0.08
Glasso	50	0.63	0.26	0.68	0.60		
	100	0.63	0.27	0.74	0.69	NA	
	200	0.61	0.24	0.79	0.55		
Stability	50	0.55	0.21			0.43	0.16
	100	0.57	0.22	NA		0.51	0.15
	200	0.61	0.22			0.59	0.14
JGL Group	50			0.82	0.77		
	100	NA		0.80	0.61	NA	
	200			0.81	0.53		
JGL Fused	50			0.71	0.48		
	100	NA		0.73	0.38	NA	
	200			0.77	0.35		

Table D.1: Sensitivity analysis performance of all five algorithms. Data were generated following multivariate t-distributions with known covariance structure. The true positive rate (TPR) and false positive rate (FPR) is reported for each of the three tasks: recovering population, subject and variance networks.

D.1.1 Further simulations

In Section 6.2 networks were simulated as described in Algorithm 6. While this algorithm was derived from an exploratory analysis of resting-state fMRI data, a wide range of alternative algorithms could also be proposed. In this section we look to provide further empirical evidence by recreating the simulation study of [42].

While [42] are able to simulate networks where variability is present, their proposed method is designed primarily to provide empirical evidence on how accurately subject-specific networks could be recovered. We therefore follow [42] and focus exclusively on recovering subject-specific covariance structure here.

Two simulations where performed where data was simulated for $N = 3$ subjects and $p = 100$ and $p = 250$ nodes respectively. Within each simulation, nodes where divided into 10 equally sized and unconnected components. The connectivity structure within each component was simulated according to scale-free model of [11], resulting in 10 scale-free

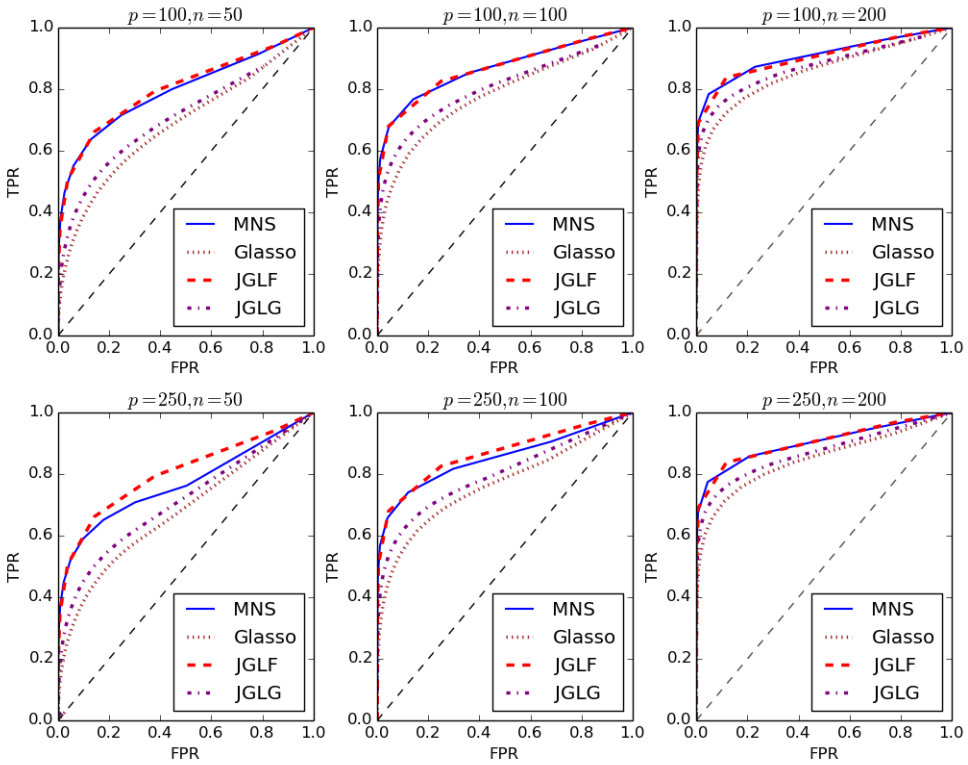


Figure D.2: Results for simulations 1 and 2. These simulations sought to re-create the simulation study presented in [42] with $p = 100$ and $p = 250$ nodes respectively.

sub-networks. Of the 10 sub-networks, eight were shared across all three subjects. Of the remaining two sub-networks, one was present in two out of three subjects and the final sub-network was only present in the first subject. For further details see [42].

D.1.1.1 Simulation 1: $p = 100$, $n \in \{50, 100, 200\}$

In the first simulation $p = 100$ nodes were employed resulting in 10 sub-networks each with 10 nodes. The number of observations per subject was allowed to vary from $n = 50$ through to $n = 200$. The results over 100 simulations are shown in top row of Figure [D.2]. The MNS algorithm performs competitively with respect to the JGL-Fused algorithm and outperforms both the JGL-Group and graphical lasso algorithms across all values of n. In particular, we note that the MNS algorithm remains competitive even as the number of

p	n	MNS		Glasso		JGL Fused		JGL Group	
100	50	0.346	0.006	0.175	0.016	0.343	0.007	0.221	0.012
	100	0.477	0.003	0.282	0.008	0.503	0.005	0.353	0.005
	200	0.594	0.002	0.429	0.004	0.632	0.005	0.514	0.003
250	50	0.287	0.002	0.125	0.007	0.292	0.003	0.161	0.005
	100	0.443	0.002	0.215	0.003	0.451	0.002	0.295	0.003
	200	0.573	0.001	0.370	0.002	0.584	0.002	0.465	0.002

Table D.2: Performance of all four algorithms when recovering subject specific functional connectivity structure

observations, n, falls drastically. Detailed results are provided in Table D.2.

D.1.2 Simulation 2: $p = 250$, $n \in \{50, 100, 200\}$

The second simulation employed $p = 250$ nodes which were divided into 10 sub-networks of 25 nodes each. The number of observations per subject was allowed to vary from $n = 50$ through to $n = 200$ as before. The results over 100 simulations are shown in bottom row of Figure [D.2]. As before, the MNS algorithm performs competitively against alternative algorithms. As with the previous simulation, we note there is a trend for ROC curves to improve as the number of observations, n, increases. Detailed results are provided in Table D.2.