Sparse Estimation of Structured Inverse Covariance Matrices

Annaliza McGillivray

Department of Mathematics and Statistics
McGill University, Montreal

August 2016

A thesis submitted to McGill University in partial fulfillment
of the requirements of the degree of Doctor of Philosophy

Abstract

In recent years, the problem of estimating a sparse inverse covariance matrix in the moderate-to-large dimensional setting has been an important and challenging task in many fields, including genomics, finance and earth sciences. To achieve sparsity in the inverse covariance matrix, methods based on L_1 regularization are widely used, but fail to incorporate rich structural information known *a priori*. In this thesis, we study the problem of sparse inverse covariance estimation in three different settings in which L_1 penalization is inappropriate and alternative penalties must be considered.

First, we consider the problem of estimating a sparse inverse covariance matrix in the time-ordered data context. L_1-penalized likelihood methods penalize the elements of the inverse equally and independently of each other without taking into account the positive-definiteness constraint. We propose a penalized likelihood approach based on the partial autocorrelation (PAC) parametrization. The novelty of this approach lies in the use of the PACs, which allow for shrinkage in an unconstrained setting and offer greater interpretability in the ordered data context. The performance of the proposed PAC-based penalized likelihood method is assessed in a simulation study and with two real data sets.

Next, we explore inverse covariance estimation in the case where variables are *unordered*. Under multivariate normality, estimation of a sparse inverse covariance matrix can be thought of as a way of estimating a graphical model for the data, where each variable corresponds to a node in the graph and each non-zero element represents an edge between the corresponding pair of nodes. We focus on the case where the underlying graphical model has hubs, which are highly connected nodes, and introduce a weighted

lasso approach that takes into account hub structure. Some asymptotic properties are established and the finite-sample performance of the method is illustrated with simulated data and two microbiome data sets.

Finally, we study the problem of estimating time-varying networks in the context of a longitudinal study. We propose two penalized likelihood approaches for estimating time-varying networks with a penalty based on a Wishart prior for the precision matrix. We introduce a sequential approach, where the estimated precision matrix at time point t is taken to be the maximizer of a penalized log-likelihood that encourages sparsity but also shrinkage towards the estimated precision matrix at the previous time point. We also introduce a joint estimation approach, where we estimate multiple graphical models by jointly maximizing a penalized log-likelihood with an L_1 penalty to encourage sparsity and a Wishart-type penalty to promote similarity between precision matrix estimates at adjacent time points. We present a computational procedure for solving the resulting convex optimization problem and assess the performance of the methods in simulation.

Résumé

Ces dernières années, le problème d'estimation de l'inverse d'une matrice de covariance en dimension modérée à grande a été une tâche importante et difficile dans de nombreux domaines, incluant la génomique, la finance et les sciences de la terre. Des méthodes de régularisation L_1 sont souvent utilisées pour obtenir une inverse de la matrice de covariance creuse, mais elles ne permettent pas d'incorporer les informations structurelles connues *a priori*. Dans cette thèse, nous étudions le problème d'estimation éparse de l'inverse d'une matrice de covariance dans trois contextes différents où la pénalisation L_1 n'est pas appropriée et d'autres pénalités doivent être considérées.

Premièrement, nous considérons le problème d'estimation de l'inverse de la matrice de covariance dans le contexte de données ordonnées dans le temps. Les méthodes de vraisemblance pénalisée réduisent les éléments de l'inverse également et indépendamment les uns des autres, sans tenir compte de la contrainte que la matrice doit être définie positive. Nous proposons une approche de vraisemblance pénalisée basée sur la paramétrisation autocorrélation partielle. La nouveauté de cette approche réside dans l'utilisation des autocorrélations partielles, qui permettent la pénalisation dans un cadre sans contrainte et offrent une plus grande interprétabilité dans le contexte de données ordonnées. Nous évaluous la performance de la méthode proposée à l'aide de simulations et de l'analyse de deux jeux de données réelles.

Ensuite, nous étudions l'estimation de l'inverse de la matrice de covariance dans le cas où les variables sont non ordonnées. Dans le cas gaussien, l'estimation éparse de l'inverse de la matrice de covariance peut être considérée comme un moyen d'estimation d'un modèle graphique pour les données, où chaque variable correspond à un sommet

dans le graphe, et chaque élément non nul représente une arête entre la paire de sommets correspondants. Nous nous concentrons sur le cas où le modèle graphique sous-jacent possède des *supernœuds*, qui sont des nœuds avec un grand nombre de liaisons, et nous introduisons une approche lasso qui tient compte de cette structure. Nous établissons certaines propriétés asymptotiques et illustrons la performance de la méthode à l'aide de simulations et de l'analyse de deux jeux de données réelles sur le microbiome.

Finalement, nous étudions le problème d'estimation des modèles graphiques dynamiques dans le cadre d'une étude longitudinale. Nous proposons deux méthodes de vraisemblance pénalisée pour l'estimation des réseaux temporels avec une pénalité basée sur une loi de Wishart pour l'inverse de la matrice de covariance. Nous présentons une approche séquentielle, où l'inverse de la matrice de covariance estimée au temps t est celle qui maximise une log-vraisemblance pénalisée encourageant à la fois l'éparsité et le rétrécissement vers la matrice de précision estimée au temps précédent. Nous présentons également une approche d'estimation conjointe, où nous estimons plusieurs modèles graphiques en maximisant conjointement une log-vraisemblance pénalisée avec une pénalité L_1 pour encourager l'éparsité et une pénalité de type Wishart pour promouvoir la similitude entre les estimations de la matrice de précision à des temps adjacents. Nous présentons un algorithme pour résoudre le problème d'optimisation convexe résultant et évaluons la performance des méthodes proposées au moyen d'études de simulation.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Abbas Khalili and Professor David A. Stephens, for their invaluable support and guidance over the years of my doctoral research. This thesis would not have been possible without their continued feedback and encouragement.

Thank you to the faculty of the Department of Mathematics and Statistics at McGill University for providing me with a solid foundation in mathematics and statistics over the years of my graduate studies.

I must also extend my gratitude to the Fonds de recherche du Québec - Nature et technologies (FRQNT) for their generous financial support.

Finally, I would like to thank my friends and family - my parents, my sister, and my brother - for supporting me throughout my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Inverse covariance matrix estimation in the high dimensional setting, in which the dimension of the data p is comparable to or larger than the sample size n, has generated a great deal of interest among researchers in recent years. This interest has focused in particular on estimating a *sparse inverse covariance matrix*, that is, obtaining an estimate of the inverse covariance matrix, also known as the *precision matrix*, in which some elements are equal to zero. With this goal in mind, the penalized likelihood method with an appropriately constructed penalty has emerged as a favourable approach, but still presents some challenges.

Penalized likelihood methods were first introduced in the regression context as means of performing variable selection and parameter estimation simultaneously. Popular penalties include the least absolute shrinkage and selection operator (lasso; Tibshirani, 1996), the adaptive lasso (Zou, 2006) and the smoothly clipped absolute deviation (SCAD; Fan and Li, 2001). The theoretical properties and computational algorithms of the corresponding penalized likelihood problems have been well studied. In the context of estimating an inverse covariance matrix Θ, additional challenges arise due to the positive-definiteness constraint on Θ and in the case where Θ is structured. Friedman et al. (2008) studied L_1-penalization of Θ. Under this framework, however, the elements of Θ are penalized equally and independently of each other without taking into account the underlying structure of Θ. In this thesis, we study inverse covariance estimation by sparse selection where this type of penalization is inappropriate and alternative penalties must

be considered.

For the first part of this thesis, we investigate inverse covariance estimation in the case where variables have a natural ordering. We propose a penalized likelihood approach to estimate Θ, based on the partial autocorrelation (PAC) parametrization. The novelty of this method lies in the use of the PACs, which vary freely over the interval (-1,1), removing the positive-definiteness constraint on the inverse, and allow for greater interpretation compared to the partial correlations in the ordered data setting. We consider this PAC-based penalized likelihood methodology in the context of inverse covariance estimation, but also for estimating the order of a stationary Gaussian autoregressive (AR) process.

For the second part of this thesis, we investigate estimation of high-dimensional inverse covariance matrices in the case where variables are unordered. This is of particular interest because if $\mathbf{X}_1 \ldots, \mathbf{X}_n$ are independent and identically distributed multivariate normal random vectors with mean $\mathbf{0}$ and covariance matrix $\Sigma = \Theta^{-1}$, a non-zero in the off-diagonals of Θ corresponds to a pair of variables that are conditionally dependent. Thus, under multivariate normality, estimation of a sparse inverse covariance matrix can be thought of as a way of estimating a graphical model for the data, where each variable corresponds to a node in the graph and each non-zero element of Θ represents an edge between the corresponding pair of nodes. We focus on the case where the underlying graphical model has *stars* or *hubs*, which are highly connected nodes, inspired by microbiome data. We propose a penalized likelihood approach for estimating networks with hubs, referred to as the *hubs weighted graphical lasso* (HWGL), and provide a simulation study, demonstrating its superior finite-sample performance compared to competing network estimation methods. We also establish asymptotic properties for the proposed hubs weighted graphical lasso estimator.

For the third and final part of this thesis, we consider the problem of estimating time-varying networks in the context of a longitudinal study, where multiple measurements at each time point taken under similar experimental conditions are available. We investigate the performance of the method of Zhou et al. (2010) in this context and also introduce two new penalized likelihood approaches that make use of a Wishart-type

penalty that encourages similar structure for networks at consecutive time points. The first is a sequential penalized likelihood approach that allows for the borrowing of strength from reconstructed networks at previous time points. From a modelling perspective, this would be suitable for many real-world applications, where data arrive sequentially over time. The second is a penalized likelihood approach that jointly estimates the T graphical models by imposing sparsity through the use of an L_1 penalty and by shrinking networks at adajcent time points toward each other through the use of a Wishart-type penalty.

1.1 Thesis Contributions

The specific contributions of this thesis can be summarized as follows.

- When it comes to inverse covariance estimation for ordered data, the modified Cholesky decomposition (Pourahmadi, 1999) is often used. It converts the constrained entries of Θ into unconstrained parameters and reduces the task of modelling a $p \times p$ covariance matrix to that of modelling $p - 1$ regression problems. In Chapter 3, we consider the PAC parametrization, which is an alternative reparametrization of a covariance matrix. The PAC parametrization has mainly been used in a Bayesian setting for constructing priors for the correlation matrix R, but has not been considered in the frequentist penalized likelihood framework. Therefore, we introduce a new PAC-based penalized likelihood approach that makes use of the nested lasso penalty of Levina et al. (2008).

- The PAC-based penalized likelihood methodology developed in Chapter 3 can also be used for autoregressive process modelling. In Chapter 4, we focus on the specific task of estimating the order of a stationary Gaussian AR process. For this purpose, the lasso methodology has been used (Wang et al., 2007b; Nardi and Rinaldo, 2011), where a lasso penalty is applied to the AR coefficients. However, such a procedure ignores the temporal dependence information embedded in AR time series. Rather than imposing shrinkage on the AR coefficients, we instead introduce shrinkage via the PACs, which vary on the same scale, free of constraints, and better reflect the

temporal dependence of the AR process.

- There have been a few procedures developed in the literature for estimating graphical models with hubs, such as the hubs graphical lasso (HGL) of Tan et al. (2014) and the reweighted L_1 regularization approach of Liu and Ihler (2011). The former is designed for estimating networks with very densely connected nodes, while the latter is designed for estimating scale-free networks, for which there may be no clear distinction between *hub* and *non-hub* nodes. In Chapter 5, we introduce a procedure that allows for more flexible and general modelling of networks with hubs. The procedure makes use of a weighted lasso approach with novel row/column sum weights, introduced as a finite-sample correction to the adaptive lasso (Fan et al., 2009) procedure in the case where the underlying network has hubs. The asymptotic properties of this weighted lasso estimator are also provided.

- In Chapter 5, we develop methodology for estimating networks with hubs, motivated by an application to microbiome data. While sparse network selection methods have been widely applied to genomic data sets, the use of such procedures in microbiome data analysis is relatively recent. Thus, in Section 5.6, we explore a relatively new application of the statistical methodology developed for estimating high-dimensional networks.

- In Chapter 6, we introduce two new penalized likelihood approaches for estimating T time-varying networks in the context of a longitudinal study. The first is a sequential approach, estimating the precision matrix at each time point t separately, while the other is a joint approach, but both make use of a Wishart-type penalty that encourages similar structure for networks at consecutive time points. We also provide the computational algorithms for solving the resulting convex optimization problems.

1.2 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, we provide a literature review of sparsity-based regularization in linear regression and inverse covariance estimation with a focus on penalized likelihood methodology. This will provide the theoretical basis for our work in subsequent chapters. In Chapters 3 and 5, we study penalized likelihood methods in the context of inverse covariance estimation. In particular, in Chapter 3, we introduce our PAC-based penalized likelihood method for estimating an inverse covariance matrix in the case where variables are ordered. In Chapter 4, we consider another application of our PAC-based penalized likelihood methodology, where we consider the problem of estimating the order of a stationary Gaussian AR process. In Chapter 5, we then study high-dimensional network estimation in the case where the networks have hubs, and introduce our HWGL procedure. In Chapter 6, we investigate the problem of estimating dynamic networks in a longitudinal setting. This thesis then concludes with a discussion of our results and future research directions in Chapter 7.

Chapter 2

A Literature Review: Regularization in Regression and Covariance Estimation

In broad terms, regularization is the class of methods that are used for solving ill-posed problems, yielding stable solutions of unstable problems. The need for regularization in statistics arose in large part due to the growing complexity of datasets available. Datasets with a large number of variables and only a small number of observations have now become a common occurrence in statistics. To accommodate the high-dimensionality of observations, the tendency is to fit more and more complex models to the data. The fitting of models with a large number of parameters, however, is inherently unstable (Bickel and Li, 2006, and references therein). This phenomenon, known as *overfitting*, occurs when a statistical model captures the noise of the data and thus exhibits low bias but high variance. This can be remedied by imposing explicit constraints on model complexity, such as bounds on the L_q norm of model parameters. Regularization thus serves to improve estimation in ill-posed or overparametrized problems by making additional assumptions or introducing suitable *a priori* knowledge. From a Bayesian perspective, many regularization methods correspond to imposing certain prior distributions on model parameters.

One of the earliest examples of regularization in statistics arose in high-dimensional regression, where it was suggested to constrain or bound the L_2 norm of the regression parameter. This method is known as ridge regression (Hoerl and Kennard, 1970), and since

its introduction, a number of other constraints or bounds on the regression parameter have been considered.

Another statistical problem that has relied heavily on regularization is covariance matrix estimation, as the number of parameters grows rapidly with the number of variables. As the maximum likelihood estimator (MLE) of the $p \times p$ covariance matrix Σ, the sample covariance matrix behaves optimally if p is smaller than sample size n, converging to Σ at rate $n^{-1/2}$. However, when the ratio p/n is large, it is well known that the sample covariance matrix S is a poor estimator, as its eigenstructure tends to be systematically distorted; the largest eigenvalue tends to be overestimated and the smallest eigenvalue tends to be underestimated (see Pourahmadi, 2011 and references therein). Therefore, regularization in the covariance matrix setting began with the goal of obtaining estimators that are better-conditioned than the sample covariance matrix.

To this end, regularization by Steinian shrinkage was proposed early on and is achieved by either shrinking the eigenvalues of S toward a central value (Haff, 1980; Dey and Srinivasan, 1985), or by replacing S with a linear combination of itself and the identity matrix (Ledoit and Wolf, 2004). The proposed estimators affect the eigenvalues of the sample covariance matrix, but not the eigenvectors, and are also not sparse. Lately, regularization has been employed with parsimony as its guiding principle. Such regularization procedures are the focus of this thesis.

In the context of inverse covariance estimation, achieving sparsity in Θ is of particular interest because when the p-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_p)^T$ follows a multivariate normal distribution with mean zero and covariance matrix $\Sigma = \Theta^{-1}$, a zero entry in Θ corresponds to a conditional independence relationship. More precisely, the $(i,j)^{\text{th}}$ entry of Θ is zero if and only if X_i and X_j are conditionally independent, given the remaining variables. The conditional independence structure of $\mathbf{X}$ can represented by an undirected graph $\mathcal{G} = (V, E)$, where $V = \{1, 2, \ldots, p\}$ is the set of vertices and E is the edge set defined as

$$E = \left\{(i,j) : X_i \text{ and } X_j \text{ are dependent given } X_{-(i,j)}, 1 \leq i, j \leq p\right\},$$

where $X_{-(i,j)} = \{X_k : k \neq i, j, 1 \leq k \leq p\}$. The goal of network selection is then to identify the edges in the set E.

Methods for sparse (inverse) covariance estimation make use of the machinery developed for regression analysis. Therefore, in this chapter, we review existing regularization methods in the literature both for estimating the parameters of linear regression models with a large number of predictors and large inverse covariance matrices.

In Section 2.1, we begin by reviewing popular penalization methods in the regression context that perform variable selection and parameter estimation simultaneously, namely the lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001) and adaptive lasso (Zou, 2006). In Section 2.2, we then review existing regularization procedures for sparse estimation of large inverse covariance matrices. We start with a discussion of more general methods, such as penalized maximum likelihood methods, including the graphical lasso (Friedman et al., 2008) and the graphical SCAD (Fan et al., 2009) in Section 2.2.1 as well as banding (Bickel and Levina, 2008a) and thresholding (Bickel and Levina, 2008b) in Section 2.2.2. The penalization approaches and thresholding work for unordered variables and provide permutation-invariant inverse covariance estimators. We then review in Section 2.2.3 existing methods in the literature that utilize the modified Cholesky decomposition, where it is assumed that there is a natural ordering among the variables. In Section 2.2.4, we focus on Bayesian methods that have been proposed in the literature for estimating covariance matrices, leading us to the use of the partial autocorrelation parametrization of the covariance matrix, which will be important in subsequent chapters of this thesis.

2.1 Regularized Linear Regression

Methods for estimating a covariance matrix or its inverse have brought to use the tools developed for regression analysis (e.g., penalized likelihood estimation, nonparametric methods, Bayesian analysis). In this section, we provide a literature review of regularization in linear regression with a focus on exisiting penalized likelihood methodology.

Suppose that data $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, is available, where y_i is the i^{th} observation of

the response variable and $\mathbf{x}_i$ is its associated p-dimensional vector of covariates, typically assumed to be a random sample from the population (X, Y), where the conditional mean of Y given X, $\mathbb{E}(Y|X)$, depends on the linear predictor $X^T\boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$. The model that links the conditional mean of y_i given $\mathbf{x}_i$ to the linear predictor $\mathbf{x}_i^T\boldsymbol{\beta}$ is

$$g\left\{\mathbb{E}(y_i|\mathbf{x}_i)\right\} = \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, ,$$

where $g(\cdot)$ is a one-to-one *link function*. The covariates x_{ij} can be continuous, binary or categorical, and the function $g(\cdot)$ has different forms depending on whether the problem at hand is one of classification or one of regression. A linear regression model corresponds to the case where y_i is continuous and $g(\cdot)$ is the identity function. A logistic regression model corresponds to the case where y_i is binary and $g(\cdot)$ is the logit function. If y_i are count data, the Poisson regression model is often used with the log function for $g(\cdot)$.

To perform sparse selection, it is assumed that most regression coefficients β_j are zero. The goal of variable selection is then to identify all important variables whose regression coefficients are significant and to provide estimates of those coefficients. Sparsity is assumed for the following two reasons:

- *Model interpretability:* By removing irrelevant variables, we retain the variables that have the strongest effects on the outcome, resulting in a model that is more easily interpretable.

- *Prediction accuracy:* Prediction accuracy may be improved by removing insignificant variables. While shrinking or setting some coefficients to zero may increase the bias of the estimates, it may reduce the variance of the predicted values. For the full model, the estimators for the parameters have low bias but large variance. In contrast, for a parsimonious model, the estimators may have larger bias but smaller variance. Therefore, by incorporating shrinkage, we sacrifice a little bias to reduce the prediction variance, which has the net effect of reducing the mean squared error (MSE) of prediction. This is called the *bias-variance tradeoff*.

Subset selection methods, such as forward stepwise selection, backward stepwise selec-

tion, the Akaike information criterion (AIC; Akaike, 1973) and the Bayesian information criterion (BIC; Schwarz, 1978) were initially proposed for performing variable selection. These procedures are discrete in the sense that variables are either retained or discarded, and can often have high variability (Breiman, 1996). AIC and BIC suggest a framework for performing variable selection that involves maximizing the penalized log-likelihood

$$\ell_n(\boldsymbol{\beta}) - \lambda\|\boldsymbol{\beta}\|_0, \tag{2.1}$$

where $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^{p} I(\beta_j \neq 0)$ counts the number of non-zero parameters in $\boldsymbol{\beta}$ and $\lambda > 0$ is a tuning parameter. In terms of computational complexity, the problem with the L_0 norm, however, is NP-hard. Methods using *shrinkage* or *regularization* that use alternative penalties in (2.1) were thus proposed as more stable procedures that are computationally efficient when p is large.

A natural generalization of the L_0-penalized likelihood is the L_q-penalized likelihood, called bridge regression (Frank and Friedman, 1993), where $p_\lambda(|\beta|) = \lambda|\beta|^q$ for $0 < q \leq 2$. The bridge penalty includes a few well known penalty functions as special cases. For $q \in (0, 1]$, $p_\lambda(\cdot)$ is known as the *soft-thresholding* penalty (Donoho and Johnstone, 1994). For $q = 1$, it is known as the least absolute shrinkage and selection operator (lasso; Tibshirani, 1996). For $q = 2$, it is the penalty in ridge regression (Hoerl and Kennard, 1970).

In this section, we discuss the various penalty functions in (2.1) that were proposed, including the lasso (Tibshirani, 1996), adaptive lasso (Zou, 2006), and smoothly clipped absolute deviation (SCAD; Fan and Li, 2001) penalties, as well as the important task of choosing the tuning parameter in these penalized likelihood methods.

2.1.1 Shrinkage and Selection Estimators

In this section, we review the various penalty functions $p_{\lambda_n}(\cdot)$ used in the penalized likelihood problem

$$\arg\min_{\boldsymbol{\beta}} \left\{ -\ell_n(\boldsymbol{\beta}) + p_{\lambda_n}(\boldsymbol{\beta}) \right\}.$$

We focus this review on the lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), and adaptive lasso (Zou, 2006), which all lead to simultaneous variable selection and parameter estimation.

When studying the properties of a shrinkage and selection estimator, two important ideas emerge: (1) whether the estimator can identify the true support, asymptotically, if the true parameter is sparse, and (2) to assess the performance of the estimator with respect to the estimates of the true non-zero parameters. In particular, it is of interest to study whether the penalized estimator behaves as well as the unpenalized estimator with respect to the non-zero coefficients.

Let $\mathcal{A}$ denote the support of the true parameter $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)$ and $\mathcal{A}_n$ denote the support of the penalized estimator $\hat{\boldsymbol{\beta}}_n = (\hat{\beta}_{n,1}, \ldots, \hat{\beta}_{n,p})$. Two important properties that any penalized estimator $\hat{\boldsymbol{\beta}}_n$ should possess, which make up the so-called *oracle property*, are the following:

- *Variable selection consistency:* $\lim_{n\to\infty} P(\mathcal{A}_n = \mathcal{A}) = 1$ and
- $\sqrt{n}$*-estimation consistency:* $\sqrt{n}(\hat{\boldsymbol{\beta}}_{n_{|\mathcal{A}}} - \boldsymbol{\beta}^*_{|\mathcal{A}}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma^*)$, where Σ^* is the covariance matrix knowing the true subset model.

2.1.1.1 Least Absolute Shrinkage and Selection Operator (LASSO)

The least absolute shrinkage and selection operator, called the *lasso* (Tibshirani, 1996), is the most widely used shrinkage and selection method. It was originally proposed in the regression context and has since been used in a variety of settings. Its popularity can be attributed to two characteristics. Firstly, it performs variable selection and parameter

estimation simultaneously. Secondly, it leads to a convex optimization problem for which a number of efficient algorithms have been proposed. We begin by reviewing the lasso and computational algorithms for solving the lasso optimization problem.

Consider the linear regression model $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ and $\epsilon_1, \ldots, \epsilon_n$ are independent and identically distributed random errors with mean 0 and variance σ^2.

Definition 1. (The lasso estimator) The lasso estimator, denoted by $\hat{\boldsymbol{\beta}}_n^{\text{lasso}}$, is defined as

$$\hat{\boldsymbol{\beta}}_n^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.2}$$

or, equivalently,

$$\hat{\boldsymbol{\beta}}_n^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t \tag{2.3}$$

for some $t \geq 0$.

In general, there is no closed form solution to (2.2). However, an analytical formula exists in the orthonormal design case. To gain further insight into the shrinkage mechanism, we study the lasso in the orthonormal design case in Proposition 1.

Proposition 1. (The lasso estimator in the orthonormal design case) Suppose that $\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = I$, where I is the identity matrix, then the lasso estimator takes the form

$$\hat{\beta}_j(\lambda_n) = \text{sgn}(\hat{\beta}_j^{\text{ols}}) \left(|\hat{\beta}_j^{\text{ols}}| - \lambda_n \right)_+, \tag{2.4}$$

where $(z)_+ = \max(z, 0)$ and $\hat{\beta}_j^{\text{ols}}$ is the ordinary least squares (OLS) estimator of β_j.

Asymptotic Properties:

The asymptotic properties of the lasso estimator for fixed p were studied in Knight and Fu (2000), Fan and Li (2001), Zhao and Yu (2006), and Zou (2006). These authors showed that the lasso estimator is estimation consistent, but the optimal rate of estimation is available only when $\lambda_n = O(\sqrt{n})$. However, this leads to inconsistent variable selection.

The question of interest then becomes whether consistency in variable selection can be achieved if we are willing to sacrifice the convergence rate in estimation. Zou (2006), Zhao and Yu (2006), and Meinshausen and Bühlmann (2006) all independently investigated this issue. It turns out that a slower rate of convergence does not guarantee variable selection consistency. A necessary condition for the lasso to be variable selection consistent is the *irrepresentable condition* (Zhao and Yu, 2006), which concerns the design matrix $\mathbf{X}$.

Zhao and Yu (2006) showed that the irrepresentable condition is almost necessary and sufficient for lasso to be variable selection consistent both for fixed p and diverging p as the sample size n increases. We provide definitions for the *strong* and *weak irrepresentable conditions* in Definitions 3 and 4 from Zhao and Yu (2006). We let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_s, \beta_{s+1}, \ldots, \beta_p)^T$, $\beta_j \neq 0$ for $j = 1, \ldots, s$ and $\beta_j = 0$ for $j = s+1, \ldots, p$. Further, we let $C^{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{x}_i$,

$$C^{(n)} = \begin{pmatrix} C_{11}^{(n)} & C_{12}^{(n)} \\ C_{21}^{(n)} & C_{22}^{(n)} \end{pmatrix},$$

where $C_{11}^{(n)}$ is an $s \times s$ matrix, and write $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T)^T$.

Definition 2. (Strong irrepresentable condition; Zhao and Yu, 2006) There exists a positive constant vector $\boldsymbol{\eta}$ such that

$$|C_{21}^{(n)}(C_{11}^{(n)})^{-1}\mathrm{sgn}(\boldsymbol{\beta}_1)| \leq \mathbf{1} - \boldsymbol{\eta},$$

where $\mathbf{1}$ is a $(p-s) \times 1$ vector of 1's and the inequality holds componentwise.

Definition 3. (Weak irrepresentable condition; Zhao and Yu, 2006)

$$|C_{21}^{(n)}(C_{11}^{(n)})^{-1}\text{sgn}(\boldsymbol{\beta}_1)| < \mathbf{1},$$

where the inequality holds componentwise.

Zhao and Yu (2006), and Zou (2006) showed that the weak irrepresentable condition is necessary for variable selection consistency of the lasso, and the strong irrepresentable condition is sufficient for selection consistency of the lasso. These conditions, however, can be restrictive in high dimensions. The weak irrepresentable condition states that the lasso is variable selection consistent if and (almost) only if the variables that are not in the true model are "irrepresentable" by variables that are in the true model (Zhao and Yu, 2006). In other words, if an irrelevant variable is highly correlated with variables in the true model, the lasso may fail to distinguish it from the true variables even with large n. To improve the performance of the lasso, a variety of penalties have been proposed, such as the smoothly clipped absolute deviation (SCAD) penalty of Fan and Li (2001), and the adaptive lasso of Zou (2006), which we discuss in Sections 2.1.1.2 and 2.1.1.3.

Computational Algorithms:

The objective function in the lasso procedure to be optimized is convex. Therefore, the global minimum can be found efficiently using a variety of algorithms. These algorithms include the quadratic programming algorithm (Tibshirani, 1996), the least angle regression and shrinkage (LARS; Efron et al., 2004), and coordinate descent (Friedman et al., 2007). In this section, we focus on the coordinate descent procedure of Friedman et al. (2007).

By the Karush-Kuhn-Tucker conditions (KKT conditions; see, e.g. Boyd and Vandenberghe, 2004), a necessary and sufficient condition for $\hat{\boldsymbol{\beta}}_\lambda$ to be the global minimizer of the lasso objective function, written as

$$Q_\lambda(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2.5}$$

is that the subdifferential of $Q_\lambda(\boldsymbol{\beta})$ at $\hat{\boldsymbol{\beta}}_\lambda$ is zero. Consider the following two cases:

Case 1: $\hat{\beta}_j(\lambda) \neq 0$

The first derivative of $Q_\lambda(\boldsymbol{\beta})$ at $\hat{\beta}_j(\lambda)$ must be 0:

$$\frac{\partial Q_\lambda(\boldsymbol{\beta})}{\partial \beta_j}\bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda)} = -\mathbf{X}_j^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \text{sgn}(\beta_j)\big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda)} = 0$$

$$\iff$$

$$G_j(\hat{\boldsymbol{\beta}}_\lambda) = -\mathbf{X}_j^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda) = -\lambda \text{sgn}(\hat{\beta}_j(\lambda)) \text{ if } \hat{\beta}_j(\lambda) \neq 0.$$

Case 2: $\hat{\beta}_j(\lambda) = 0$

The subdifferential of $Q_\lambda(\boldsymbol{\beta})$ at $\hat{\beta}_j(\lambda)$ must include the zero element:

$$G_j(\hat{\boldsymbol{\beta}}_\lambda) + \lambda\gamma = 0 \text{ for some } \gamma \in [-1, 1] \iff |G_j(\hat{\boldsymbol{\beta}}_\lambda)| \leq \lambda \text{ if } \hat{\beta}_j(\lambda) = 0.$$

Coordinate Descent: The lasso objective function to be minimized is

$$f(\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2.6}$$

for $\lambda > 0$.

With a single predictor, the lasso solution is simply a soft-thresholded version of the least squares estimate $\hat{\beta}^{\text{ols}}$:

$$\hat{\beta}_\lambda = S(\hat{\beta}^{\text{ols}}, \lambda) = \text{sgn}(\hat{\beta}^{\text{ols}})(|\hat{\beta}^{\text{ols}}| - \lambda)_+ = \begin{cases} \hat{\beta}^{\text{ols}} - \lambda & \text{if } \hat{\beta}^{\text{ols}} > 0 \text{ and } \lambda < |\hat{\beta}^{\text{ols}}|, \\ \hat{\beta}^{\text{ols}} + \lambda & \text{if } \hat{\beta}^{\text{ols}} < 0 \text{ and } \lambda < |\hat{\beta}^{\text{ols}}| \\ 0 & \text{if } \lambda \geq |\hat{\beta}^{\text{ols}}|. \end{cases}$$

With more than one predictor in the case where the predictors are uncorrelated, the lasso solutions are again soft-thresholded versions of the least squares estimates. For general

predictors, Friedman et al. (2007) write the objective function in (2.6) as follows

$$f(\tilde{\boldsymbol{\beta}}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k - x_{ij}\beta_j\right)^2 + \lambda\sum_{k\neq j}|\tilde{\beta}_j| + \lambda|\beta_j|, \tag{2.7}$$

where all β_k's for $k \neq j$ are held fixed at values $\tilde{\beta}_k$.

Minimizing the objective function in (2.7) with respect to β_j yields the update

$$\tilde{\beta}_{j,\lambda} \leftarrow S\left(\sum_{i=1}^{n} x_{ij}(y_i - \tilde{y}_i^{(j)}), \lambda\right), \tag{2.8}$$

where $\tilde{y}_i^{(j)} = \sum_{k\neq j} x_{ik}\tilde{\beta}_{k,\lambda}$ and $S(z,\lambda) = \text{sgn}(z)(|z|-\lambda)_+$ is the soft-thresholding operator. At the m^{th} iteration, the update is repeated for $j = 1, 2, \ldots, p$ to obtain an estimate $\tilde{\boldsymbol{\beta}}_\lambda^{(m)}$ and this is repeated until convergence. The sequence of estimates $\left\{\tilde{\boldsymbol{\beta}}_\lambda^{(m)}\right\}$ was shown to converge to the lasso estimate $\hat{\boldsymbol{\beta}}^{\text{lasso}}$.

To find lasso solutions over a range of possible values of λ, Friedman et al. (2007) begin with a value of λ large enough so that the only optimal solution is the vector of zeroes. This value is equal to $\lambda_{\max} = \max_j \sum x_{ij}y_i$. Then λ is progressively decreased and each time the coordinate descent procedure is run until convergence, using the previous solution as a "warm start". This procedure allows for the efficient computation of solutions over a grid of λ values and is referred to as *pathwise coordinate descent* (Friedman et al., 2007; Hastie et al., 2015).

2.1.1.2 Smoothly Clipped Absolute Deviation (SCAD)

Several alternative penalties have been introduced, designed to remedy some of the drawbacks of the lasso penalty. The popularity of the lasso can be attributed to its convexity, but it is known to produce biased estimates of the regression coefficients due to the linear increase of the penalty function. In this section, we review one such alternative penalty known as the *smoothly clipped absolute deviation* (SCAD) penalty (Fan and Li, 2001), but first we present three desirable properties identified by Fan and Li (2001) that any penalized estimator should possess in the penalized least squares or penalized likelihood

framework:

(i) *Unbiasedness:* The resulting estimator is nearly unbiased when the true unknown parameter is large.

(ii) *Sparsity:* The estimator is a thresholding rule, which automatically sets small estimated coefficients to zero to reduce model complexity.

(iii) *Continuity:* The estimator is continuous in the data to avoid instability in prediction.

The L_q, $q > 0$, penalty functions do not result in estimators that simultaneously satisfy the mathematical conditions for unbiasedness, sparsity and continuity. The bridge (Frank and Friedman, 1993) solution is only continuous when $0 < q < 1$, but when $q > 1$, sparse solutions are not produced. Therefore, Fan and Li (2001) proposed the smoothly clipped absolute deviation (SCAD) penalty, which is a quadratic spline, given by

$$p_\lambda(\beta) = \begin{cases} \lambda|\beta| & \text{if } |\beta| \leq \lambda \\ \frac{-(\beta^2 - 2a\lambda|\beta| + \lambda^2)}{2(a-1)} & \text{if } \lambda < |\beta| \leq a\lambda \\ \frac{(a+1)\lambda^2}{2} & \text{if } |\beta| > a\lambda \end{cases} \tag{2.9}$$

with derivative

$$p'_\lambda(\beta) = \lambda \cdot \text{sgn}(\beta) \left\{ I(|\beta| \leq \lambda) + \frac{(a\lambda - |\beta|)_+}{(a-1)\lambda} I(|\beta| > \lambda) \right\}$$

for some $a > 2$ and where $(\cdot)_+$ denotes the positive truncation function. The SCAD penalty function is continuously differentiable everywhere except at 0 and its derivative vanishes outside $[-a\lambda, a\lambda]$. Thus, it can produce continuous and sparse solutions and unbiased estimates for large coefficients, satisfying properties (i)-(iii). We provide a plot of the SCAD penalty function in Figure 2.1 along with the lasso penalty function. The lasso and SCAD penalty functions behave similarly for small coefficients. For larger coefficients, SCAD applies a constant penalty, whereas the lasso penalty increases linearly

with the coefficient. Hence, the SCAD penalty function results in asymptotically unbiased estimators while the lasso penalty function does not.

The SCAD-penalized least-squares estimator is defined in Definition 4 (2.10). We also consider the case of an orthonormal design matrix in Proposition 2 to gain some insight into the shrinkage mechanism.

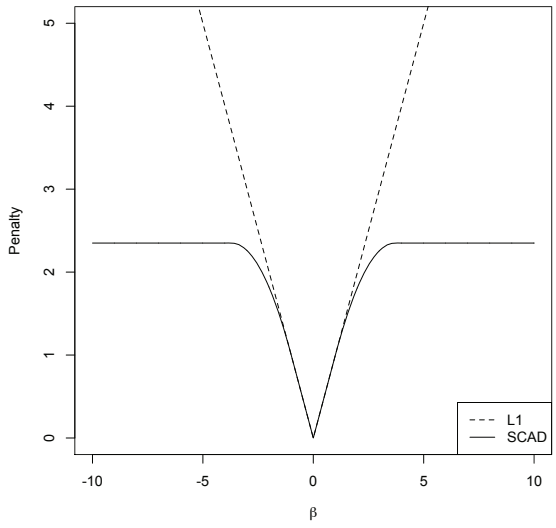

Figure 2.1: Plot of the lasso and SCAD penalty functions with $\lambda = 1$ and $a = 3.7$.

Definition 4. (The SCAD-penalized estimator) The SCAD-penalized estimator, denoted by $\hat{\boldsymbol{\beta}}_n^{\text{SCAD}}$, is defined as

$$\hat{\boldsymbol{\beta}}_n^{\text{SCAD}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \sum_{j=1}^{p} p_{\lambda_n}(\beta_j) \right\}, \tag{2.10}$$

where $p_{\lambda_n}(\cdot)$ is defined in (2.9).

Proposition 2. (The SCAD-penalized estimator in the orthonormal design case) Suppose that $\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = I$, where I is the identity matrix, then the SCAD-penalized least-squares estimator takes the form

$$\hat{\beta}_j^{\text{SCAD}}(\lambda) = \begin{cases} \text{sgn}(\hat{\beta}_j^{\text{ols}})(|\hat{\beta}_j^{\text{ols}}| - \lambda)_+ & \text{if } \hat{\beta}_j^{\text{ols}} \leq 2\lambda, \\ \left\{ (a-1)\hat{\beta}_j^{\text{ols}} - \text{sgn}(\hat{\beta}_j^{\text{ols}}) a\lambda \right\} / (a-2) & \text{if } 2\lambda < |\hat{\beta}_j^{\text{ols}}| \leq a\lambda, \\ \hat{\beta}_j^{\text{ols}} & \text{if } |\hat{\beta}_j^{\text{ols}}| > a\lambda, \end{cases}$$

where $(z)_+ = \max(z, 0)$, $a > 2$, and $\hat{\beta}_j^{\text{ols}}$ is the ordinary least squares (OLS) estimator of β_j.

Asymptotic Properties:

Fan and Li (2001) studied the asymptotic properties of nonconcave penalized likelihood estimators, obtained by maximizing

$$\ell_n(\boldsymbol{\beta}) - n\sum_{j=1}^{p} p_{\lambda_n}(|\beta_j|) \tag{2.11}$$

for fixed p as $n \to \infty$, and showed that there exists a penalized likelihood estimator that converges at the rate

$$O_p(n^{-1/2} + a_n),$$

where $a_n = \max\{p'_{\lambda_n}(|\beta_{j0}|) : \beta_{j0} \neq 0\}$. Therefore, for the SCAD penalty function, if $\lambda_n \to 0$, the penalized likelihood estimator is $\sqrt{n}$-consistent. They also showed that under regularity conditions (A)-(C) in Fan and Li (2001), which are needed to guarantee asymptotic normality of the ordinary MLEs, if $\lambda_n \to 0$ and $\sqrt{n}\lambda_n \to \infty$ as $n \to \infty$, the SCAD-penalized estimator has the *oracle property*. In other words, the $\sqrt{n}$-consistent estimator $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_1^T, \hat{\boldsymbol{\beta}}_2^T)^T$ has the sparsity property, that is, if $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)^T = (\boldsymbol{\beta}_1^{*T}, \boldsymbol{\beta}_2^{*T})^T$ denotes the true parameter and $\boldsymbol{\beta}_2^* = \mathbf{0}$, then $\hat{\boldsymbol{\beta}}_2 = \mathbf{0}$ with probability approaching 1 as $n \to \infty$, and $\hat{\boldsymbol{\beta}}_1$ is asymptotically normal with the same covariance matrix knowing the true subset model.

Remarks:

- For the lasso penalty in (2.11), $a_n = \lambda_n$ and $\lambda_n = O_p(n^{-1/2})$ is required for $\sqrt{n}$-consistency. However, the oracle property requires that $\sqrt{n}\lambda_n \to \infty$. These conditions cannot be simultaneously satisfied.

- The results of Fan and Li (2001) are for the case where p is fixed as $n \to \infty$. Fan and Peng (2004) extended the results of Fan and Li (2001) to the case where p

diverges with the sample size n. Under regularity conditions, they established an oracle property and the asymptotic normality of the nonconcave penalized likelihood estimator in the moderate dimensional setting with $p = o(n^{1/5})$ or $o(n^{1/3})$ (see Fan and Lv, 2010 for a review of variable selection in this setting).

Computational Algorithm:

Fan and Li (2001) had proposed an algorithm for optimizing the nonconcave penalized log-likelihood function. Since the L_1 and SCAD penalty functions are singular at the origin and do not have continuous second order derivatives, Fan and Li (2001) used a local quadratic approximation (LQA) approach. Assuming that $\boldsymbol{\beta}^{(0)} = (\beta_1^{(0)}, \ldots, \beta_p^{(0)})^T$ is an initial value of $\boldsymbol{\beta}$, they suggest that the penalty function be locally approximated by a quadratic function as follows

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_j^{(0)}|) + \frac{1}{2}\frac{p'_\lambda(|\beta_j^{(0)}|)}{|\beta_j^{(0)}|}\ (\beta_j^2 - \beta_j^{(0)2})$$

for $\beta_j \approx \beta_j^{(0)}$. If $\beta_j^{(0)} \approx 0$, they set $\hat{\beta}_j = 0$.

For the penalized least squares problem, the solution can be found by iteratively minimizing

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\sum_{j=1}^{p}\frac{1}{2}\frac{p'_\lambda(|\beta_j^{(m)}|)}{|\beta_j^{(m)}|}\beta_j^2 \tag{2.12}$$

for $m = 1, 2, \ldots$. Using Newton-Raphson, this amounts to iteratively computing the following ridge regression

$$\boldsymbol{\beta}^{(m)} = \left\{\mathbf{X}^T\mathbf{X} + n\Sigma_\lambda(\boldsymbol{\beta}^{(m-1)})\right\}^{-1}\mathbf{X}^T\mathbf{y}$$

for $m = 1, 2, \ldots$, where $\Sigma_\lambda(\boldsymbol{\beta}^{(m)}) = \text{diag}\left(\frac{p'_\lambda(|\beta_1^{(m)}|)}{|\beta_1^{(m)}|}, \ldots, \frac{p'_\lambda(|\beta_p^{(m)}|)}{|\beta_p^{(m)}|}\right)$.

One drawback of LQA is that once a coefficient is set to zero, it remains at zero. As an improvement over LQA, Zou and Li (2008) suggested using a local linear approximation

(LLA), where

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_j^{(0)}|) + p'_\lambda(|\beta_j^{(0)}|)\ (\beta_j - \beta_j^{(0)}).$$

2.1.1.3 Adaptive Lasso

In this section, we study another procedure that yields consistent estimators and also selects variables consistently without stringent conditions on the design matrix, namely the *adaptive lasso* (Zou, 2006). The adaptive lasso is a modified version of the lasso, where the L_1 norms on the regression coefficients are reweighted by data-dependent weights. We review the definition, computational algorithm, and asymptotic properties of the adaptive lasso.

Definition 5. (The adaptive lasso estimator) The adaptive lasso estimator, denoted by $\hat{\boldsymbol{\beta}}_n^{\text{alasso}}$, is defined as

$$\hat{\boldsymbol{\beta}}_n^{\text{alasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j |\beta_j| \right\},$$

where $\hat{w}_j = \frac{1}{|\hat{\beta}_j|^\gamma}$ for some $\gamma > 0$ and a $\sqrt{n}$-consistent estimator $\hat{\beta}_j$ of β_j.

By allowing larger penalties for zero coefficients and smaller penalties for non-zero coefficients, the adaptive lasso aims to reduce the estimation bias and improve variable selection accuracy, compared with the standard lasso.

Remarks:

- As $n \to \infty$, the weights corresponding to insignificant variables tend to infinity, while the weights corresponding to significant variables converge to a finite constant. Thus, large coefficients can be estimated unbiasedly (asymptotically) and small coefficients can be thresholded, simultaneously (Zou, 2006).

- The adaptive lasso solution is continuous. Fan and Li (2001) had identified continuity as an important property of any variable selection procedure because discontinuities result in instability in model prediction.

Proposition 3. (The adaptive lasso estimator in the orthonormal design case) Suppose that $\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = I$, where I is the identity matrix, then the adaptive lasso estimator takes the form

$$\hat{\beta}_j^{\text{alasso}}(\lambda_n) = \text{sgn}(\hat{\beta}_j^{\text{ols}}) \left(|\hat{\beta}_j^{\text{ols}}| - \frac{\lambda_n}{|\hat{\beta}_j^{\text{ols}}|^{\gamma}} \right)_+ , \tag{2.13}$$

where $(z)_+ = \max(z, 0)$, $\gamma > 0$ and $\hat{\beta}_j^{\text{ols}}$ is the ordinary least squares (OLS) estimator of β_j.

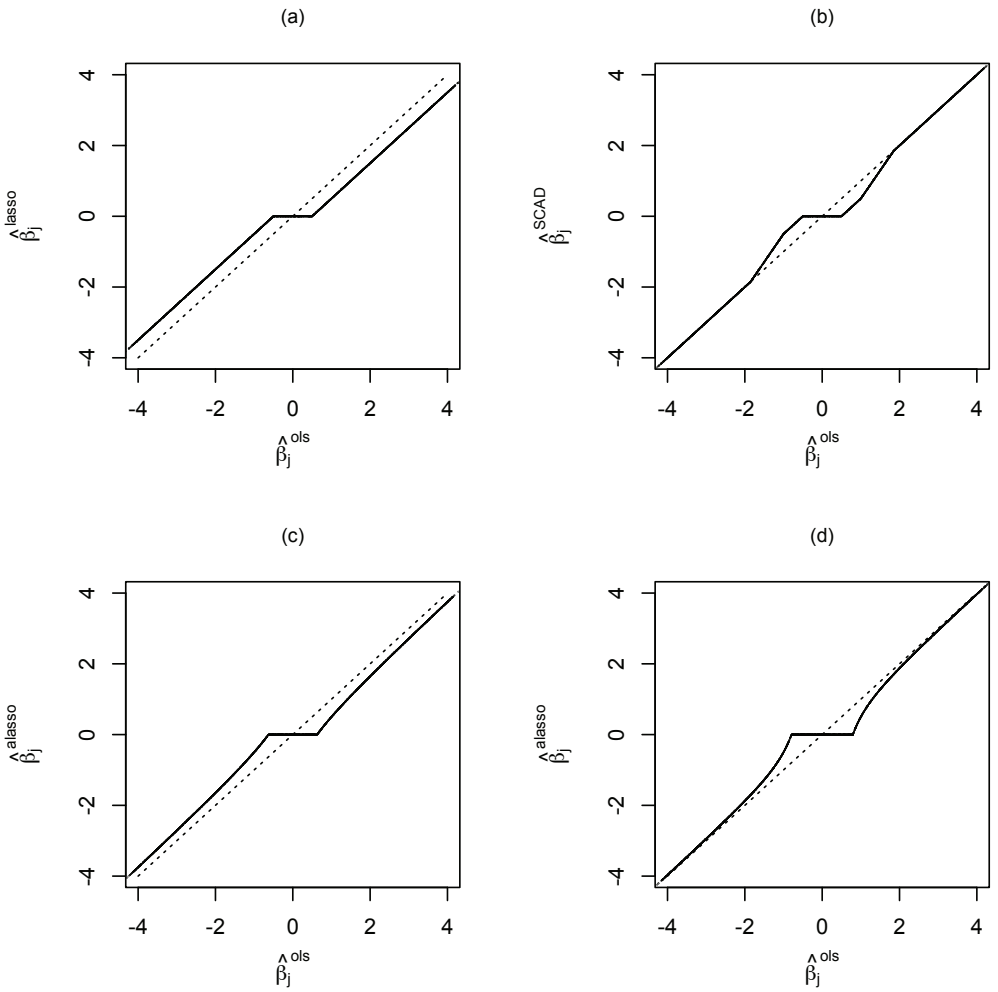

Figure 2.2: Plot of thresholding functions with $\lambda = 1$ for (a) lasso, (b) SCAD, (c) adaptive lasso with $\gamma = 0.5$, and (d) adaptive lasso with $\gamma = 2$.

Figure 2.2 (c)-(d) gives us insight into the shrinkage mechanism of the adaptive lasso. We observe that the adaptive lasso shrinkage still causes the estimate of a non-zero coefficient to be biased towards zero, but unlike the lasso (Figure 2.2 (a)), the bias becomes

smaller for coefficients that are larger in magnitude.

Asymptotic Properties:

Zou (2006) studied the asymptotic properties of the adaptive lasso estimator for fixed p as $n \to \infty$. He showed that, under certain regularity conditions, if $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$, then the adaptive lasso estimator possesses the oracle property.

It should be noted that the initial estimator $\hat{\boldsymbol{\beta}}$ used in the weights is not required to be $\sqrt{n}$-consistent; the condition can be weakened. If there is a sequence $\{a_n\}$ such that $a_n \to \infty$ and $a_n(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) = O_p(1)$, $\lambda_n/\sqrt{n} \to 0$ and $a_n^\gamma \lambda_n/\sqrt{n} \to \infty$, then the oracle property still holds.

Computational Algorithm:

Using the adaptive lasso penalty leads to a convex optimization problem and the efficient algorithms for solving the lasso can be used to compute the adaptive lasso estimates. In what follows, we provide the algorithm of Zou (2006).

Algorithm 1:

1. Define $x_{ij}^* = x_{ij}/\hat{w}_j$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$.

2. Solve the lasso problem

$$\hat{\boldsymbol{\beta}}_n^* = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}^* \beta_j \right)^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| \right\},$$

   for all λ_n.

3. Set $\hat{\beta}_j^{\text{alasso}} = \hat{\beta}_j^*/\hat{w}_j$, $j = 1, 2, \ldots, p$.

2.1.1.4 Group Lasso

Yuan and Lin (2006) proposed the group lasso procedure, which allows for whole groups of covariates to be selected for inclusion or exclusion from the model. A leading example is when we have dummy variables encoding a multilevel categorical predictor. In this

case, the lasso would select individual dummy variables instead of including the group of variables together. There are many biological applications with a natural group structure among the variables. For example, genes do not work in isolation but rather operate within known pathways (Sokolov, 2016), and often it is of interest to establish which pathways are related to a response rather than the individual genes. In what follows, we provide the definition of the group lasso estimator.

Definition 6. Definition (The group lasso estimator): Consider a linear regression model with J groups of covariates, where for $j = 1, \ldots, J$, $\mathbf{X}_j$ represents the covariates in group j of size p_j. The group lasso estimator is obtained by solving

$$\underset{\beta_1,\ldots,\beta_J}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{y} - \sum_{j=1}^{J} \mathbf{x}_j \beta_j\|^2 + \lambda \sum_{j=1}^{J} \sqrt{p_j} \|\beta_j\|_2 \right\}, \tag{2.14}$$

where $\lambda > 0$ is a tuning parameter and the term $\sqrt{p_j}$ accounts for the varying group sizes.

The group lasso procedure acts like the lasso at the group level: a group of covariates may be knocked out, depending on the choice of λ (Friedman et al., 2010). The group lasso objective function is convex and can be optimized using a block coordinate descent procedure. Since the penalty is also block separable, the algorithm is guaranteed to converge to the optimal solution (Hastie et al., 2015).

2.1.1.5 Criticisms of the Oracle Property

Penalized likelihood methodology, such as lasso and SCAD, have been widely used in high-dimensional data analysis due to the fact that they perform model selection and parameter estimation simultaneously. In most existing work, the focus has been on studying their prediction, estimation and selection consistency properties, but other important questions remain to be answered. Procedures like lasso and SCAD set a parameter directly to zero as a result of the optimization of a penalized objective function, which makes challenging the task of performing statistical inference (e.g. obtaining valid confidence intervals for the true parameter when model selection precedes parameter estimation). A naive use of inference procedures that do not take into account the model selection step can be highly

misleading (see e.g. Leeb and Pötscher, 2005).

Other questions that arise relate to the oracle property, as defined in Fan and Li (2001). Leeb and Pötscher (2008) related the oracle property of shrinkage estimators to the superefficiency property of Hodges' estimator. For $X_1 \ldots, X_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, 1)$, Hodges' estimator for the mean is $T_n = \bar{X}_n$ if $|\bar{X}_n| > n^{-1/4}$ and is 0 otherwise, which is a hard-thresholding estimator exhibiting the sparsity and oracle property. As the sample size increases, however, its maximal (scaled) mean squared error grows without bound, whereas the standard maximum likelihood estimator $\bar{X}_n$ has constant finite quadratic risk. Leeb and Pötscher (2008) showed that the unbounded risk result is true for any estimator possessing the sparsity property. They showed that any estimator satisfying a sparsity property has maximal risk that converges to the supremum of the loss function; in particular, the maximal risk diverges to infinity whenever the loss function is unbounded. Their result is in the linear regression setting. They find that the SCAD estimator can perform rather poorly in finite samples and that its worst-case performance relative to maximum likelihood deteriorates with increasing sample size when the estimator is tuned to sparsity. The bad risk behaviour is a local phenomenon and occurs at points in the parameter space that are sparse in the sense that some of the components are equal to 0. They argue that the oracle property is an asymptotic feature that holds only pointwise in the parameter space and gives a misleading picture of the actual finite-sample performance of the estimator.

2.1.2 Tuning Parameter Selection

The asymptotic properties of the penalized likelihood estimators discussed in Section 2.1.1 depend on the choice of the tuning parameter $\lambda > 0$, which controls the balance between model fit and model complexity. In what follows, we provide a literature review on the selection of the tuning parameter in the penalized maximum likelihood problem

$$\hat{\boldsymbol{\beta}}_\lambda = \arg\max_{\boldsymbol{\beta}} \left\{ \ell_n(\boldsymbol{\beta}) - n \sum_{j=1}^{p} p_\lambda(|\beta_j|) \right\}$$

for some penalty function $p_\lambda(\cdot)$ with tuning parameter λ. First, we discuss two desirable asymptotic properties of a tuning parameter selection procedure, namely *consistency* and *efficiency*, in Section 2.1.2.1. Then, in Section 2.1.2.2, we review existing tuning parameter selection procedures, focusing our discussion on cross-validation (CV), generalized cross-validation (GCV), and the generalized information criterion (GIC), which has as special cases the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).

2.1.2.1 Consistency and Efficiency

In model selection, one goal is to identify important predictors that are relevant to the response. Another goal is to construct a model with strong predictive power. Accordingly, model selection criteria are assessed from two different perspectives: identification of the true model and accurate prediction. In the literature, the asymptotic properties of model selection criteria that are studied from these two perspectives are *consistency* and *efficiency*. A tuning parameter selection procedure is said to be *consistent* if the true model is identified with probability approaching 1 in large samples when the set of candidate models contains the true model. A tuning parameter selection procedure is said to be *efficient* if it selects the model so that its average squared error is asymptotically equivalent to the minimum among the candidate models when the true model is approximated by a family of candidate models (Zhang et al., 2010). Both consistency and efficiency are pointwise asymptotic properties.

To formally define these categories of selection criteria in the context of tuning parameter selection, we first introduce some notation. Let $\mathcal{A}$ denote the collection of all candidate models α and α_0 denote the unique true model. Further, let $\hat{\lambda}$ denote the selected tuning parameter and $\alpha_{\hat{\lambda}}$ the corresponding selected model.

Definition 1 (Consistency): A tuning parameter selection procedure is said to be *consistent* if

$$P(\alpha_{\hat{\lambda}} = \alpha_0) \to 1 \text{ as } n \to \infty.$$

Definition 2 (Asymptotically loss efficient): A tuning parameter selection procedure is said to be asymptotically loss efficient if

$$\frac{L(\hat{\boldsymbol{\beta}}_{\hat{\lambda}})}{\inf_{\lambda \in [0,\lambda_{\max}]} L(\hat{\boldsymbol{\beta}}_{\lambda})} \to 1 \tag{2.15}$$

as $n \to \infty$ in probability, where $\hat{\boldsymbol{\beta}}_{\hat{\lambda}}$ is associated with the tuning parameter $\hat{\lambda}$ selected by this procedure, and L is some loss function. In the literature, the L_2 norm has been commonly used to assess the efficiency of the classical AIC procedure in linear regression models.

Asymptotic properties of some model selection criteria have been established in different settings. The traditional model selection criterion AIC is an *efficient* selection criterion in that it selects the best finite-dimensional candidate model in terms of prediction accuracy when the true model is only approximated by a family of candidate models (Wang, 2007a). However, it is an *inconsistent* selection criterion (Shao, 1997) since it does not select the correct model with probability approaching 1 in large samples when the true model is among the set of candidate models. The traditional model selection criterion BIC, on the other hand, is *consistent* under some assumptions. In Section 2.1.2.2, we review tuning parameter selectors in the penalized likelihood framework and include in our discussion their asymptotic properties.

2.1.2.2 Existing Methods for Tuning Parameter Selection

Cross-Validation and Generalized Cross-Validation:

Cross-validation (CV) and generalized cross-validation (GCV) are nonparametric methods for estimating prediction error, and had been used by Fan and Li (2001) for selecting the tuning parameter in their nonconcave penalized likelihood methods. In K-fold CV, part of the available data is used to fit the model, while the remaining part is used to test it. It first involves randomly splitting the data into K roughly equal-sized parts. Then for the k^{th} part, the model is fitted to the other $K-1$ parts of the data, which make up the *training* data set, and the prediction error of the fitted model when predicting the

k^{th} part of the data, called the *test* or *validation* data set, is calculated. This process is repeated K times, with each of the K parts used exactly once as the validation data, and the K estimates of prediction error are then combined. The case $K = n$ is called leave-one-out CV. For the i^{th} observation, the fit is computed using all the data except the i^{th}. With $K = n$, the CV estimator is approximately unbiased for the true (expected) prediction error. See Hastie et al. (2009) for more details.

Fan and Li (2001) had used 5-fold CV for selecting the tuning parameter. In this case, if D denotes the full data set, and D_{-k} and D_k denote the training and test data sets, respectively, then the selected λ is then taken to be the minimizer of

$$\text{CV}(\lambda) = \sum_{k=1}^{5} \sum_{(y_k, \mathbf{x}_k) \in D_k} \left\{ y_k - \mathbf{x}_k^T \hat{\boldsymbol{\beta}}_\lambda^{(-k)} \right\}^2,$$

where, for each λ and k, $\hat{\boldsymbol{\beta}}_\lambda^{(-k)}$ is the estimator of $\boldsymbol{\beta}$ using the training set D_{-k}.

Fan and Li (2001) had also used GCV, which provides a convenient approximation to leave-one-out cross-validation for linear fitting under squared-error loss (Friedman et al, 2009). Viewing

$$P_{\mathbf{X}}(\hat{\boldsymbol{\beta}}_\lambda) = \mathbf{X} \left\{ \mathbf{X}^T \mathbf{X} + n \Sigma_\lambda(\hat{\boldsymbol{\beta}}_\lambda) \right\}^{-1} \mathbf{X}^T$$

as a projection matrix, where $\Sigma_\lambda(\hat{\boldsymbol{\beta}}_\lambda) = \text{diag}\left\{ p'_\lambda(|\hat{\beta}_{\lambda 1}|)/|\hat{\beta}_{\lambda 1}|, \ldots, p'_\lambda(|\hat{\beta}_{\lambda p}|)/|\hat{\beta}_{\lambda p}| \right\}$, Fan and Li (2001) defined the number of effective parameters in the penalized least squares fit to be $\text{df}_\lambda = \text{tr}\left\{ P_{\mathbf{X}}(\hat{\boldsymbol{\beta}}_\lambda) \right\}$ so that the GCV statistic is

$$\text{GCV}(\lambda) = \frac{1}{n} \frac{\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda\|^2}{\left(1 - \frac{\text{df}_\lambda}{n}\right)^2}.$$

The selected tuning parameter λ is then taken to be the minimizer of GCV. Wang et al. (2007a) investigated tuning parameter selection for the penalized least squares method with the SCAD penalty. They found that the resulting model selected by GCV tends to overfit.

The Akaike Information Criterion:

Another possible method for selecting the tuning parameter in penalized likelihood approaches is the Akaike information criterion (Akaike, 1973), which was derived as an estimator of the Kullback-Leibler information discrepancy. It aims to minimize the Kullback-Leibler divergence between the true distribution and the estimate from a candidate model. In the penalized likelihood framework in the regression setting, AIC is computed as twice the negative log-likelihood evaluated at the penalized MLE $\hat{\boldsymbol{\beta}}_\lambda$, penalized by the number of non-zero coefficients in $\hat{\boldsymbol{\beta}}_\lambda$. When the penalized least squares function

$$Q_\lambda(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\sum_{j=1}^{p} p_\lambda(|\beta_j|)$$

is used, AIC becomes

$$\mathrm{AIC}(\lambda) = \log \hat{\sigma}_\lambda^2 + \frac{2\mathrm{df}_\lambda}{n},$$

where $\hat{\sigma}_\lambda^2 = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda\|^2$ and $\hat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}} Q_\lambda(\boldsymbol{\beta})$. The traditional model selection criterion AIC is an efficient but inconsistent selection criterion. As pointed out in Wang et al. (2007a), since $\mathrm{AIC}(\lambda)$ can be viewed as an approximation of the log-transformation of

$$\mathrm{GCV}(\lambda) = \frac{\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda\|^2}{n\left(1 - \frac{\mathrm{df}_\lambda}{n}\right)^2} = \frac{\hat{\sigma}_\lambda^2}{\left(1 - \frac{\mathrm{df}_\lambda}{n}\right)^2},$$

as follows

$$\log \mathrm{GCV}(\lambda) = \log \hat{\sigma}_\lambda^2 - 2\log\left(1 - \frac{\mathrm{df}_\lambda}{n}\right) \approx \log \hat{\sigma}_\lambda^2 + \frac{2\mathrm{df}_\lambda}{n} = \mathrm{AIC}(\lambda),$$

AIC and $\log \mathrm{GCV}$ are very similar. Thus, AIC also suffers from an overfitting effect and GCV, like AIC, may not identify the true model consistently.

The Bayesian Information Criterion:

Shown by Shao (1997) to be a *consistent* model selection criterion in the classical regression setting, Wang et al. (2007a) employed the Bayesian information criterion (Schwarz, 1978) to select the tuning parameter λ in the penalized least squares method with the SCAD penalty. They selected the optimal tuning parameter by minimizing

$$\mathrm{BIC}(\lambda) = \log \hat{\sigma}_\lambda^2 + \frac{\mathrm{df}_\lambda \log n}{n}.$$

The BIC arises from the Bayesian approach to model selection; choosing the model with minimum BIC is equivalent to choosing the model with largest (approximate) posterior probability. Wang et al. (2007a) showed that BIC is able to identify the true model consistently when the penalized least squares approach is used with the SCAD penalty.

Generalized Information Criterion:

When performing classical variable selection for the normal linear regression model $y_i = \mathbf{x}_i^T \boldsymbol{\beta}_\alpha + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$, Nishii (1984) proposed the generalized information criterion (GIC), which is given by

$$\mathrm{GIC}_{\kappa_n}(\alpha) = \log \hat{\sigma}_\alpha^2 + \frac{1}{n}\kappa_n d_\alpha,$$

where $\boldsymbol{\beta}_\alpha$ is the parameter of the candidate model α, $\hat{\sigma}_\alpha^2$ is the MLE of σ^2, and κ_n is a positive number that controls properties of variable selection. The GIC has AIC and BIC as special cases; when $\kappa_n = 2$, GIC becomes AIC, while $\kappa_n = \log n$ leads to BIC.

In the context of linear and generalized linear models with a nonconcave penalty function, Zhang et al. (2010) proposed to use a GIC-type criterion. Their GIC-type tuning parameter selector is given by

$$\mathrm{GIC}_{\kappa_n}(\lambda) = \frac{1}{n}\left\{ G(\mathbf{y}, \hat{\boldsymbol{\beta}}_\lambda) + \kappa_n \mathrm{df}_\lambda \right\}, \tag{2.16}$$

where $G(\mathbf{y}, \hat{\boldsymbol{\beta}}_\lambda)$ measures the fitting of model α_λ and df_λ is the degrees of freedom of

model α_λ. Fan and Li (2001) proposed that the degrees of freedom in (2.16) be the trace of the approximate linear projection matrix

$$\mathrm{df}_{L,\lambda} = \mathrm{tr}\left\{(\bigtriangledown_\lambda^{\otimes 2} Q^*(\hat{\boldsymbol{\beta}}_\lambda))^{-1} \bigtriangledown_\lambda^{\otimes 2} \ell(\hat{\boldsymbol{\beta}}_\lambda)\right\},$$

where

$$Q^*(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - n\sum_{j=1}^{p} q_\lambda(|\beta_j|),$$

and $q_\lambda(\cdot)$ is the LQA of $p_\lambda(\cdot)$, $\left[\bigtriangledown_\lambda^{\otimes 2} Q^*(\boldsymbol{\beta})\right]_{jj'} = \frac{\partial^2}{\partial\beta_j\beta_{j'}} Q^*(\boldsymbol{\beta})$, and $\left[\bigtriangledown_\lambda^{\otimes 2}\ell(\boldsymbol{\beta})\right]_{jj'} = \frac{\partial^2}{\partial\beta_j\beta_{j'}}\ell(\boldsymbol{\beta})$ for j, j' such that $\hat{\beta}_j \neq 0$ and $\hat{\beta}_{j'} \neq 0$.

Assuming that the penalized likelihood estimator $\hat{\boldsymbol{\beta}}_\lambda$ is sparse and consistent, and under certain conditions on the penalty function, Zhang et al. (2010) showed that

$$P(\mathrm{df}_{L,\lambda} = d_{\alpha_\lambda}) \to 1$$

as $n \to \infty$, where d_{α_λ} is the size of model α_λ. In linear regression models, Zou et al. (2007) also suggested using d_{α_λ} as an estimator of the degrees of freedom for the lasso. They showed that d_{α_λ} is an asymptotically unbiased estimator.

In (2.16), when $\kappa_n = 2$, GIC is referred to as the AIC selector, while when $\kappa_n \to 2$, GIC is referred to as the AIC-type selector. GIC with $\kappa_n \to \infty$ and $\kappa_n/\sqrt{n} \to 0$ is called the BIC-type selector. In the linear and generalized linear modelling context, Zhang et al. (2010) found that when the true model is among a set of candidate models, the BIC-type selector identifies the true model consistently, whereas the AIC-type selector tends to overfit. On the other hand, in the linear modelling context, if the true model is approximated by a set of candidate models, the AIC-type selector is asymptotically loss-efficient, which is a property not shared by the BIC-type selector in general. Zhang et al. (2010) focused on the efficiency of linear model selections via the L_2 norm.

2.2 Regularized Inverse Covariance Estimation

In Section 2.1, we reviewed penalized likelihood methods for simultaneously selecting important variables and estimating parameters in a linear regression model with a large number of predictors. In this section, we review these same penalized likelihood methods in the context of inverse covariance estimation as well as discuss other regularization procedures for estimating a sparse inverse covariance matrix.

2.2.1 Penalized Maximum Likelihood Estimation

When it comes to estimating a precision matrix on the basis of a sample of vectors drawn from a multivariate Gaussian distribution, the most widely used approach is L_1 regularization. Various authors (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008) have investigated L_1-penalized likelihood methods for sparse estimation of precision matrices. These methods work within the *covariance selection* framework, where the goal is to identify the zero elements of Θ. In the graphical modelling context, *neighbourhood selection* (Meinshausen and Bühlmann, 2006) is also used. Neighbourhood selection aims to estimate (individually) the neighbourhood of any given variable (or node), rather than to produce an estimate of the inverse covariance matrix. It can be cast as a standard regression problem and be solved efficiently with the lasso. We provide a review of these penalization methods, but focus our discussion on inverse covariance selection methods. In Section 2.2.1.3, we also discuss the selection of the tuning parameter in the penalized maximum likelihood problem.

2.2.1.1 Neighbourhood Selection

Meinshausen and Bühlmann (2006) proposed performing a neighbourhood selection at each node in the graph using the lasso penalty. The neighbourhood $\mathcal{N}_i$ of a node $i \in V$ consists of all nodes $j \in V \setminus \{i\}$ such that $(i, j) \in E$. Neighbourhood selection aims to estimate the set of neighbours of a node. It does so by fitting a lasso regression using each node (variable) as the response and the others as predictors. More specifically, let $\mathbf{X}_j$

denote the j^{th} column of $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{X}_{-j} \in \mathbb{R}^{n \times (p-1)}$. For each node j, Meinshausen and Bühlmann (2006) solve the following optimization problem

$$\hat{\boldsymbol{\beta}}^j(\lambda) = \arg\min_{\boldsymbol{\beta}} \left(\frac{1}{n} \|\mathbf{X}_j - \mathbf{X}_{-j}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1 \right), \tag{2.17}$$

where $\lambda > 0$. The element θ_{ij} is then estimated to be nonzero if either the estimated coefficient of variable i on j is nonzero or the estimated coefficient of variable j on i is nonzero (Hastie et al., 2009). The authors show that asymptotically this procedure consistently estimates the support of Θ, even when the number of variables is allowed to grow as rapidly as the sample size raised to an arbitrarily large power, but it is not guaranteed to produce a positive definite estimate $\widehat{\Theta}$.

2.2.1.2 Sparse Inverse Covariance Selection

While Meinshausen and Bühlmann (2006) only estimate which θ_{ij}'s are nonzero, Yuan and Lin (2007), Banerjee et al. (2008) and Friedman et al. (2008) try to estimate the sparsity pattern of the underlying graph as well as obtain a regularized estimate of the precision matrix. They consider maximizing the penalized log-likelihood

$$\log\det\Theta - \operatorname{tr}(S\Theta) - \lambda\|\Theta\|_1 \tag{2.18}$$

over non-negative definite matrices Θ, and where tr denotes the trace, $\lambda > 0$ is a regularization parameter and $\|\Theta\|_1 = \sum_{ij} |\theta_{ij}|$ is the L_1 norm. The diagonal entries may also be omitted from the penalty. The objective function (2.18) was shown to be convex in Banerjee et al. (2008). Due to the convexity of (2.18), Friedman et al. (2008) solve this optimization problem using a coordinate descent procedure, which is remarkably fast.

While the L_1 penalty is convex and leads to a desirable convex optimization problem when the log-likelihood function is convex, even in the simple regression setting, the lasso penalty produces biases in the estimates for large coefficients. This occurs since the L_1 penalty increases linearly with the magnitude of its argument. This problem also arises for precision matrix estimation (see Lam and Fan, 2009).

As a remedy to the bias issue, Fan et al. (2009) and Lam and Fan (2009) introduce nonconcave penalties, such as the smoothly clipped absolute deviation (SCAD; Fan and Li, 2001) penalty. The SCAD penalty is symmetric and a quadratic spline on $[0, \infty)$, whose first order derivative is given by

$$p'_\lambda(x) = \lambda \left\{ I(|x| \leq \lambda) + \frac{(a\lambda - |x|)_+}{(a-1)\lambda} I(|x| > \lambda) \right\},$$

where $\lambda > 0$ and $a > 2$ are two tuning parameters. Fan and Li (2001) recommend the choice $a = 3.7$, based on an argument of minimizing certain Bayes risk criteria. Therefore, Fan et al. (2009) seek to solve the following optimization problem:

$$\widehat{\Theta}^{\text{SCAD}} = \underset{\Theta \succ 0}{\arg\max} \left\{ \log\det\Theta - \text{tr}(S\Theta) - \sum_{i=1}^{p}\sum_{j=1}^{p} p_\lambda(|\theta_{ij}|) \right\}. \tag{2.19}$$

To take advantage of the graphical lasso algorithm of Friedman et al. (2008), Fan et al. (2009) use a local linear approximation (LLA) of the penalty function, proposed by Zou and Li (2008). Specifically, given the current estimate $\widehat{\Theta}^{(m)} = (\hat{\theta}_{ij}^{(m)})$, they approximate $p_\lambda(|\theta_{ij}|)$ in a neighbourhood of $|\hat{\theta}_{ij}^{(m)}|$ as follows:

$$p_\lambda(|\theta_{ij}|) \approx p_\lambda(|\hat{\theta}_{ij}^{(m)}|) + p'_\lambda(|\hat{\theta}_{ij}^{(m)}|)(|\theta_{ij}| - |\hat{\theta}_{ij}^{(m)}|), \tag{2.20}$$

where $p'_\lambda(x) \geq 0$. Therefore, using this approximation, the expression in (2.19) becomes

$$\underset{\Theta \succ 0}{\arg\max} \left\{ \log\det\Theta - \text{tr}(S\Theta) - \sum_{i=1}^{p}\sum_{j=1}^{p} p'_\lambda(|\hat{\theta}_{ij}^{(m)}|)|\theta_{ij}| \right\}, \tag{2.21}$$

which is optimized at each iteration m. The penalty term in (2.21) is the adaptive lasso penalty (Zou, 2006) with weights $w_{ij} = p'_\lambda(|\hat{\theta}_{ij}^{(m)}|)$ specified at each iteration m. In other words, the weighting scheme is governed by the derivative of the SCAD penalty function, evaluated at the magnitude of the current estimate $\hat{\theta}_{ij}^{(m)}$; the larger the magnitude, the smaller the weight. This optimization problem can be solved using the graphical lasso algorithm of Friedman et al. (2008). The objective function in (2.21) was shown in Fan

et al. (2009) to increase at every iteration.

Fan et al. (2009) also consider the adaptive graphical lasso with weights $w_{ij} = 1/|\tilde{\theta}_{ij}|^{\gamma}$ for some $\gamma > 0$ and any consistent estimate $\tilde{\Theta} = (\tilde{\theta}_{ij})$ of Θ. This optimization problem can be solved by the graphical lasso algorithm proposed by Friedman et al. (2008) as well. One advantage of the SCAD penalty is that an entry in the precision matrix estimated as zero can escape from zero in the next iteration, which is not the case for the adaptive lasso penalty.

Fan et al. (2009) studied the asymptotic properties of their proposed penalized likelihood estimators. One desirable property of an estimator is the oracle property. Let $\mathcal{A} = \{(i,j) : \theta_{ij} \neq 0\}$. The estimator $\widehat{\Theta} = (\hat{\theta}_{ij})$ of the precision matrix $\Theta = (\theta_{ij})$ is said to possess the *oracle property* if the following conditions are satisfied.

Oracle Property:

1. *Sparsity*: If $\theta_{ij} = 0$, then $\mathrm{P}(\hat{\theta}_{ij} = 0) \to 1$ as $n \to \infty$.

2. *Asymptotic normality:* For $(i,j) \in \mathcal{A}$, the entries $\hat{\theta}_{ij}$ of $\hat{\Theta}$ are $\sqrt{n}$-consistent and asymptotically normal.

In other words, the true zero entries of the precision matrix are estimated as zero with probability tending to one, and the estimators of the entries θ_{ij} for $(i,j) \in \mathcal{A}$ of the precision matrix have the same limiting distribution as the maximum likelihood estimator, knowing the true sparsity pattern. The first property is also referred to as *sparsistency* in Lam and Fan (2009).

Under certain conditions and assuming that p is fixed as $n \to \infty$, Fan et al. (2009) established that for both the SCAD and adaptive lasso penalties, the optimizer of the penalized likelihood function has the oracle property. Specifically, they showed that if $\lambda \to 0$ and $\sqrt{n}\lambda \to \infty$, then the oracle property holds for the SCAD-penalized estimator. Also, if $\sqrt{n}\lambda = O_p(1)$ and $\lambda\sqrt{n}a_n^{\gamma} \to \infty$, then the oracle property holds for the graphical adaptive lasso estimator with weights $w_{ij} = 1/|\tilde{\theta}_{ij}|^{\gamma}$ for some $\gamma > 0$ and any a_n-consistent estimator $\tilde{\Theta}$ of Θ, i.e. $a_n(\tilde{\Theta} - \Theta) = O_p(1)$.

In Yuan and Lin (2007), they studied the asymptotic properties of the graphical lasso estimator as well as the non-negative garrote-type estimator. They assumed that p is

held fixed as the sample size $n \to \infty$. They showed that the non-negative garrote-type (Breiman, 1995) estimator possesses the oracle property.

Lam and Fan (2009) studied the rates of convergence for penalized likelihood estimation of sparse precision matrices via the L_1 penalty and a general penalty function satisfying the properties in Fan and Li (2001). They showed that the rate for estimating Θ under the Frobenius norm is of order $(s_n \log p_n/n)^{1/2}$, where s_n is the number of non-zero elements, p_n is the dimension, and n is the sample size. Their result demonstrates how the number of non-zero elements and dimensionality affect the convergence rate: there are s_n non-zero parameters and each one of them can be estimated at best with rate $n^{-1/2}$. The contribution of high-dimensionality to the convergence rate is merely a logarithmic factor $\log p_n$. They showed that for the L_1 penalty to guarantee sparsistency and the optimal rate of convergence, the number of non-zero elements in Θ should be small; specifically, the number of non-zero elements in the off-diagonals of Θ should be at most $O(p_n)$, among the $O(p_n^2)$ parameters. However, for the SCAD penalty function, there is no such restriction. Their results allow for $p_n \gg n$ as long as $\log p_n/n = o(1)$.

Bien and Tibshirani (2011) used the L_1 penalty to impose sparsity in the covariance matrix itself, rather than its inverse. Under the normality assumption, zeros in a covariance matrix correspond to marginal independence relationships between variables. Gaussian graphical models for marginal independence are known as covariance graph models, which are popular in genomics (Hastie et al., 2009 and references therein).

2.2.1.3 Tuning Parameter Selection

A challenging problem in high-dimensional inverse covariance estimation via the penalized likelihood method is the data-dependent selection of the tuning parameter λ, which has the important role of controlling the sparsity of Θ. There are two standard approaches for selecting the optimal tuning parameter, namely information criteria and resampling schemes. In what follows, we review these existing methods for selecting the tuning parameter λ.

Information Criteria and Resampling Schemes:

Information criteria that have been used in the literature for selecting the tuning parameter include the Bayesian information criterion (BIC; Gao et al., 2012), the extended Bayesian information criterion (EBIC; Gao et al., 2012), and the Akaike information criterion (AIC; Lian, 2011). Resampling schemes such as cross-validation (CV; Fan et al., 2009) and generalized approximate cross-validation (GACV; Lian, 2011) have also been used.

The two most widely used information criteria for selecting the tuning parameter in any penalized likelihood method are AIC and BIC. In this context, they are given by

$$\text{AIC}(\lambda) = -2\ell_n(\widehat{\Theta}_\lambda) + 2\sum_{i<j} I(\hat{\theta}_{ij,\lambda} \neq 0)$$

and

$$\text{BIC}(\lambda) = -2\ell_n(\widehat{\Theta}_\lambda) + \log n \sum_{i<j} I(\hat{\theta}_{ij,\lambda} \neq 0),$$

respectively, where $\ell_n(\widehat{\Theta}_\lambda)$ is the multivariate Gaussian log-likelihood, evaluated at $\widehat{\Theta}_\lambda$, the penalized maximum likelihood estimator of Θ for a given λ. The optimal value of the tuning parameter in either case is taken to be the minimizer of the criterion.

Gao et al. (2012) studied the BIC-selection of the tuning parameter for penalized likelihood estimation of Θ. Gao et al. (2012) showed that, for fixed p, the optimal tuning parameter selected by BIC with either the SCAD or adaptive lasso penalties leads to consistency in model selection. In other words, the BIC identifies the sparsity pattern of the true precision matrix with probability approaching one in large samples. They also showed that if p diverges to infinity with the sample size, a modified BIC with an extra penalty on the dimension p of the precision matrix is model selection consistent when the number of true non-zeros is bounded. Their modified BIC is equivalent to the extended BIC (EBIC) of Foygel and Drton (2010) with $\gamma = 1$, adapted from Chen and Chen (2008), who had studied the EBIC for Gaussian linear models. The theoretical results of Chen and Chen (2008) implied that the traditional BIC is likely to be inconsistent when p is

of a larger order than $\sqrt{n}$. The EBIC is given by

$$\text{EBIC}(\lambda) = -2\ell_n(\widehat{\Theta}_\lambda) + \log n \sum_{i<j} I(\hat{\theta}_{ij,\lambda} \neq 0) + 4\gamma \log p \sum_{i<j} I(\hat{\theta}_{ij,\lambda} \neq 0)$$

for some $\gamma > 0$. Gao et al. (2012) take $\gamma = 1$.

Gao et al. (2012) compared the empirical performance of BIC and EBIC to cross-validation and demonstrated the advantageous performance of BIC for sparse precision matrix estimation through simulation studies. It is important to note that, as in the regression context, the tuning parameter selection procedure used should depend on one's statistical goal. If the aim is to correctly identify the zeros and non-zeros of the precision matrix, then BIC and EBIC are appropriate because of their selection consistency properties. If, on the other hand, one's concern is prediction performance, then CV and AIC are better options as they are both estimators of the Kullback-Leibler information and are equivalent asymptotically under certain assumptions.

Stability Approach to Regularization Selection (StARS):

Procedures AIC, BIC and cross-validation for selecting the tuning parameter λ have desirable theoretical properties in low dimensions, but they do not perform satisfactorily for high-dimensional problems. Liu et al. (2010) proposed a stability-based method for choosing the tuning parameter in the high-dimensional setting. Their method, which makes use of subsampling and is called Stability Approach to Regularization Selection (StARS), has the goal of using the least amount of shrinkage that results in a sparse network (precision matrix) that is reproducible under random sampling. Their method repeatedly takes random subsamples of the data and estimates for each subsample the entire solution path indexed by tuning parameter λ. For each tuning parameter, the selection frequencies of individual edges are calculated and a measure of overall stability is obtained. StARS then selects the value of λ at which subsampled (non-empty) graphs are the most stable in terms of edge selection frequencies (Kurtz et al., 2015).

The authors show that StARS is partially sparsistent in terms of graph estimation

under mild conditions; i.e. StARS selects all true edges with high probability even when the dimension p diverges with the sample size n.

2.2.2 Banding and Thresholding

The simplest approaches to regularized (inverse) covariance estimation are banding, tapering (Bickel and Levina, 2008a) and thresholding (Bickel and Levina, 2008b). In Bickel and Levina (2008a), they proposed two methods of regularization for the case where both p and n tend to infinity. Their first method is to band the sample covariance matrix $S = (s_{ij})$, given by $S = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i \mathbf{X}_i^T$, assuming that $\bar{\mathbf{X}} = 0$ so that the columns of $\mathbf{X}$ are centered. Given any $0 \leq k \leq p$, they define the k-banded version of the sample covariance matrix as

$$B_k(S) = [s_{ij} I(|i-j| \leq k)]_{1 \leq i,j \leq p},$$

where $I(\cdot)$ is the indicator function and estimate $\Sigma = (\sigma_{ij})$ by $\widehat{\Sigma}_k = B_k(S)$. This method assumes that the indices are such that if $|i-j| > k$, then $\sigma_{ij} = 0$. However, banding does not guarantee positive-definiteness of the estimated covariance matrix. Therefore, Bickel and Levina (2008a) also considered tapering the covariance matrix; that is, replacing S by $S * P$, where $*$ denotes Schur (coordinate-wise) matrix multiplication and P is a positive definite symmetric matrix. This would guarantee positive-definiteness of the estimated covariance matrix since the Schur product of positive definite matrices is also positive definite. Banding is a special case of tapering, where $P = (p_{ij})_{1 \leq i,j \leq p} = [I(|i-j| \leq k)]_{1 \leq i,j \leq p}$, which is not positive definite.

Their second method of regularization involves banding the Cholesky factor of the inverse covariance matrix. Note the modified Cholesky decomposition of the inverse will be introduced in Section 2.2.3.1. Given a sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$, they estimated a k-banded inverse by taking the elements in the first k bands of the Cholesky factor to be the ordinary least-squares (OLS) estimates of the coefficients and setting the elements in the remaining bands to zero. A resampling scheme is used to select the number of non-zero

bands k for both methods.

Both types of banding yield similar results. The authors showed that in the Gaussian case, the banded estimators are consistent in the operator norm (also known as the spectral norm or matrix L_2 norm), uniformly over a class of approximately "bandable" matrices, provided $(\log p)/n \to 0$. The operator norm of a matrix A is the largest singular value of A, that is, the square root of the largest eigenvalue of the positive-semidefinite matrix $A^T A$:

$$\|A\| = \sup\{\|A\mathbf{x}\| : \|\mathbf{x}\| = 1\} = \sqrt{\lambda_{\max}(A^T A)},$$

which reduces to $\|A\| = \max_i |\lambda_i(A)|$ for symmetric matrices. Bickel and Levina (2008a) obtained explicit rates of convergence depending on how fast $k \to \infty$. The rate of k that guarantees convergence of the banded estimator depends not only on n and p, but on the dependence structure as well. Convergence in the operator norm implies convergence of eigenvalues and eigenvectors (see Bickel and Levina, 2008b, and references therein), making this norm important for PCA applications.

Bickel and Levina (2008b) also proposed thresholding the sample covariance matrix S. They defined the thresholding operator by

$$T_\lambda(S) = [s_{ij} I(|s_{ij}| \geq \lambda)]_{1 \leq i,j \leq p},$$

where we see that matrix S is thresholded at λ. Note that T_λ preserves symmetry and is permutation-invariant. However, it does not necessarily preserve positive-definiteness. The threshold λ is chosen by cross-validation. The authors showed that for a suitably sparse class of matrices, the estimator is consistent in the operator norm, provided $(\log p)/n \to 0$. Therefore, the estimator will be positive-definite with probability tending to 1.

2.2.3 Cholesky-Based Regularization

In this section, we present methods for estimating a covariance matrix or its inverse when there is a natural ordering among the variables. These methods make use of the modified Cholesky decomposition of the inverse and introduce regularization through the Cholesky factor in this decomposition. We provide a derivation of the modified Cholesky decomposition in Section 2.2.3.1. We then discuss smoothing-based regularization of the Cholesky factor in Section 2.2.3.2, followed by penalized maximum likelihood estimation of the Cholesky factor in Section 2.2.3.3.

2.2.3.1 The Modified Cholesky Decomposition

We begin this section by reviewing the modified Cholesky decomposition of the inverse covariance matrix $\Theta = \Sigma^{-1}$, which relies on the assumption that variables have a natural ordering. A review can also be found in Levina et al. (2008). Let $\mathbf{X} = (X_1, \ldots, X_p)^T$ be a random vector with mean 0 and covariance matrix Σ. Wu and Pourahmadi (2003) think of $\mathbf{X}$ as the time-ordered observations of one subject in a longitudinal study. The inverse covariance matrix of $\mathbf{X}$ has the following unique modified Cholesky decomposition

$$\Sigma^{-1} = L^T D^{-1} L, \tag{2.22}$$

where L is a lower triangular matrix with ones on its diagonal and D is a diagonal matrix. The Cholesky factor L and the diagonal matrix D can be constructed by regressing a variable X_j on its predecessors. Let $X_1 = \epsilon_1$ and, for $j > 1$,

$$X_j = \sum_{t=1}^{j-1} \phi_{jt} X_t + \epsilon_j, \quad j = 2, \ldots, p, \tag{2.23}$$

where the ϕ_{jt}'s are the coefficients of the linear least-squares predictor of X_j from $X_1, \ldots, X_{j-1}$ and $\sigma_j^2 = \mathrm{Var}(\epsilon_j)$ are the corresponding residual variances. Then $-\phi_{jt}$ is the $(j,t)^{\text{th}}$ entry of L for $j > t$ and $D = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$.

To see this, let Φ be the lower triangular matrix with j^{th} row containing the coefficients

ϕ_{jt}, $t = 1, \ldots, j-1$ of the j^{th} regression and Φ has zeros on the diagonal. Let $\epsilon = (\epsilon_1, \ldots, \epsilon_p)^T$. Then converting (2.23) to matrix form,

$$\epsilon = (I - \Phi)\mathbf{X}, \tag{2.24}$$

which is the vector of successive uncorrelated prediction errors with $\text{Cov}(\epsilon) = D$, and where I is the $p \times p$ identity matrix. Therefore, taking covariance of both sides of (2.24), we have that

$$D = (I - \Phi)\Sigma(I - \Phi)^T.$$

Finally, letting $L = I - \Phi$, we can write $\Theta = L^T D^{-1} L$.

Therefore, the modified Cholesky decomposition converts the constrained entries of Θ into two groups of unconstrained parameters, namely the log of the variance parameters, $\{\log \sigma_1^2, \ldots, \log \sigma_p^2\}$, and the subdiagonal entries of L, $\{\phi_{jt} : j = 2, \ldots, p, t = 1, \ldots, j-1\}$. If we denote the estimators of L and D by $\hat{L}$ and $\hat{D}$, respectively, an estimator of Θ is given by $\hat{\Theta} = \hat{L}^T \hat{D}^{-1} \hat{L}$, which is guaranteed to be positive-definite. This approach reduces the challenging task of modelling a covariance matrix to that of modelling $p - 1$ regression problems (Pourahmadi, 2011).

2.2.3.2 Smoothing-Based Regularization of the Cholesky Factor

In this section, we first review the nonparametric smoothing method of Wu and Pourahmadi (2003) to regularize the estimation of covariance matrices. They proposed to estimate a banded inverse covariance matrix by smoothing along the first few subdiagonals of L using local polynomial smoothing (Fan and Gijbels, 1996) and setting the rest to zero. Their two-step estimation procedure proceeds as follows. The first step is to obtain the ordinary least-squares estimates $\hat{L}$ and $\hat{D}$ of L and D in (2.22), respectively. The second step is to apply local polynomial smoothing to the diagonal elements of $\hat{D}$ and the subdiagonals of $\hat{L}$. The number of diagonals to be smoothed is chosen using an information criterion like AIC or BIC. The authors established elementwise consistency of their

nonparametric estimator, but that is a property also shared by the sample covariance matrix (Levina et al., 2008).

Huang et al. (2007) proposed regularization of maximum likelihood estimation (MLE) by applying spline smoothing to the diagonal elements of D and subdiagonals of L. In their approach, which they call regularized MLE by basis expansion, they model $\log \sigma_j^2$ and ϕ_{jt} as realizations of smooth functions, which are each approximated as spline functions, which in turn are represented by basis expansion, and then maximize the likelihood with respect to the spline coefficients. In their implementation, they use quadratic splines and the number of knots and the number of subdiagonals in L to smooth are determined using BIC. The advantage of their method over that of Wu and Pourahmadi (2003) is that it does not rely on a first-step estimator of the Cholesky factor. Their estimator can be computed even when the first-step estimator is not well-defined (for example, when $p > n$).

To compare the performance of their covariance matrix estimator to that of Wu and Pourahmadi (2003), they considered the entropy loss for the covariance matrix and the quadratic loss for the covariance matrix. Three different covariance structures are considered. The authors take $\Sigma = I_p$ for the first case, a varying-coefficient AR(1) covariance structure for the second case, and an inverse covariance matrix with a non-sparse Cholesky factor that has many small entries for the third case. The authors find that both smoothed covariance estimators outperform the sample covariance matrix for every combination of Σ, n and p considered under both loss functions, and the improvement increases as p increases. They also find that the spline-smoothed covariance estimators significantly improve upon the two-step local polynomial smoothed estimator of Wu and Pourahmadi (2003). They reasoned that the two-step method of Wu and Pourahmadi (2003) does not perform as well as their method because their first-step raw estimator is too noisy.

2.2.3.3 Penalized Likelihood Estimation of the Cholesky Factor

To impose sparsity in the Cholesky factor L of the inverse covariance matrix Θ, penalized likelihood methods have also been used. In this section, we review the penalization schemes of Huang et al. (2006) and Levina et al. (2008) for introducing zeros in the Cholesky factor L of the inverse.

Penalization via L_1 and L_2 penalties: The regression interpretation of the Cholesky factor of the inverse covariance matrix, outlined in Section 2.2.3.1, suggests that the familiar variable selection techniques in regression analysis can be applied to covariance matrix estimation. With two such techniques in mind, namely ridge regression (Hoerl and Kennard, 1970) and the lasso (Tibshirani, 1996), Huang et al. (2006) introduced shrinkage to the Cholesky factor L by applying to the Gaussian log-likelihood L_1 and L_2 penalties to the entries ϕ_{jt}, $j > t$, of L. The Gaussian log-likelihood, up to a constant, is given by

$$\begin{aligned}
\ell(\Sigma; \mathbf{x}_1, \ldots, \mathbf{x}_n) &= n \log |\Sigma^{-1}| - \sum_{i=1}^{n} \mathbf{x}_i^T \Sigma^{-1} \mathbf{x}_i \\
&= n \log |D^{-1}| - \sum_{i=1}^{n} \mathbf{x}_i^T L^T D^{-1} L \mathbf{x}_i \\
&= -\left\{ n \sum_{j=1}^{p} \log \sigma_j^2 + \sum_{j=2}^{p} \sum_{i=1}^{n} \frac{1}{\sigma_j^2} \left(x_{ij} - \sum_{t=1}^{j-1} \phi_{jt} x_{it} \right)^2 \right\}.
\end{aligned}$$

Therefore, Huang et al. (2006) proposed to estimate Σ by minimizing the following objective function

$$-\ell(\Sigma; \mathbf{x}_1, \ldots, \mathbf{x}_n) + \lambda \sum_{j=2}^{p} P(\phi_j),$$

where $\phi_j = (\phi_{j1}, \ldots, \phi_{j,j-1})$ and $P(\phi_j) = \|\phi_j\|_q^q$ with $\|\phi_j\|_q^q$ as the L_q vector norm for $q = 1, 2$. The L_2 penalty ($q = 2$) results in shrinkage of the Cholesky factor L, but does not set elements of L to zero. The L_1 penalty, on the other hand, introduces zeros in arbitrary places in L, making their method more flexible than that of Wu and Pourahmadi

(2003), discussed in Section 2.2.3.2. However, a zero entry in the Cholesky factor L does not generally imply a zero entry in the inverse Θ, and so even with a sparse Cholesky factor, the resulting inverse may not be sparse at all.

The authors compared the performance of their penalized maximum likelihood estimator using the L_1 penalty to that using the L_2 penalty, and they also compared these methods to the sample covariance matrix and two minimax estimators (Muirhead, 1982). To gauge the performance of these methods, they considered two loss functions, the Kullback-Leibler loss and the quadratic loss for the covariance matrix (see Section 3.10 for a detailed discussion of loss functions). In simulation with $n = 100$ and $p = 30$, they observe that when the Cholesky factor L has many zeros, the L_1 penalty does better than the L_2 penalty, and that when the Cholesky factor L has many small values, the L_2 penalty does better than the L_1 penalty, as expected. To select the tuning parameter, they considered 5-fold cross-validation (CV) and generalized cross-validation (GCV). They found that both tuning parameter selection procedures perform similarly in the case of the L_2 penalty, while 5-fold CV performs better than GCV in the case of the L_1 penalty. Finally, they observed that their penalized likelihood estimators outperform the sample covariance matrix and the minimax estimator in nearly all cases considered.

Penalization via a nested lasso penalty: When components have a natural ordering, for longitudinal data or time series, for example, it can be assumed that variables far apart in the ordering are only weakly correlated. In this case, there is the need to impose some structure on the covariance matrix. While the penalized likelihood estimator proposed by Huang et al. (2006) is more stable than the sample covariance matrix, no structure on the covariance matrix is imposed. Furthermore, any sparsity achieved in the Cholesky factor due to the L_1 penalty may be lost in the inverse.

Levina et al. (2008) thus proposed a penalized likelihood method with a penalty that imposes a banded structure in the Cholesky factor L, and such a structure is preserved in the resulting inverse. They introduced a novel penalty, called the nested lasso penalty,

applied to the entries of the Cholesky factor L. Their nested lasso penalty is given by

$$J_0(\phi_j) = \lambda \left(|\phi_{j,j-1}| + \frac{|\phi_{j,j-2}|}{|\phi_{j,j-1}|} + \frac{|\phi_{j,j-3}|}{|\phi_{j,j-2}|} + \ldots + \frac{|\phi_{j,1}|}{|\phi_{j,2}|} \right),$$

where $0/0$ is defined as 0. It can be seen that if $\phi_{jt} = 0$, then $\phi_{j,t-1} = 0$. In other words, if the t^{th} variable is excluded from the j^{th} regression, then all preceding variables are also excluded from the j^{th} regression. This approach is more flexible than regular banding of the Cholesky factor (Bickel and Levina, 2008a) since the nested lasso penalty can be decomposed as separate penalties on each row of the Cholesky factor, allowing varying row band lengths, where the band length of row j is defined as the smallest integer k_j such that $\phi_{jt} = 0$ for all $t < j - k_j$.

Penalizing the coefficient $\phi_{j,j-1}$ and the ratios $\frac{|\phi_{j,t}|}{|\phi_{j,t+1}|}$ with the same tuning parameter may be inappropriate if they are on different scales. Therefore, to address the potential issue of difference of scales, the authors proposed two modified versions J_1 and J_2 of the penalty J_0:

$$J_1(\phi_j) = \lambda \left(\frac{|\phi_{j,j-1}|}{|\hat{\phi}^*_{j,j-1}|} + \frac{|\phi_{j,j-2}|}{|\phi_{j,j-1}|} + \frac{|\phi_{j,j-3}|}{|\phi_{j,j-2}|} + \ldots + \frac{|\phi_{j,1}|}{|\phi_{j,2}|} \right),$$

and

$$J_2(\phi_j) = \lambda_1 \sum_{t=1}^{j-1} |\phi_{jt}| + \lambda_2 \sum_{t=1}^{j-2} \frac{|\phi_{jt}|}{|\phi_{j,t+1}|},$$

where $\hat{\phi}^*_{j,j-1}$ is the coefficient from regressing X_j on X_{j-1} alone. In their simulations, the authors found that J_2 tends to perform better than J_0 and J_1. We therefore use the penalty J_2 in our simulation studies in Section 3.10. The minimization of the negative Gaussian log-likelihood, penalized with the nested lasso penalty on the Cholesky factor is a nonconvex problem and so the authors proposed a two-step iterative procedure that uses a local quadratic approximation algorithm.

The authors compared their penalized likelihood estimator to the sample covariance matrix, the L_1-penalized likelihood estimator of Huang et al. (2006), the banded estima-

tor of Bickel and Levina (2008a), and the shrinkage estimator of Ledoit and Wolf (2004), which is not sensitive to the ordering of the variables. To assess the performance of these methods in simulation, they used the Kullback-Leibler loss of the precision matrix. They found that in terms of the Kullback-Leibler loss, banding and adaptive banding, in general, outperform the sample covariance matrix, the Ledoit-Wolf estimator, and the lasso estimator. In the case where the inverse is banded with a fixed band length, banding and adaptive banding perform similarly, as expected. In the case where the true inverse is banded with varying band lengths, adaptive banding outperforms regular banding, and the difference becomes more prominent as p grows. To demonstrate that their method is able to preserve sparsity in the inverse, unlike the Huang et al. (2006) method, they computed average percentages of true zeros in the Cholesky factor and in the inverse that were estimated as zero. For the cases where the true inverse is banded both with fixed and varying band lengths, adaptive banding, banding and the standard lasso are all reasonably able to identify the true zeros in the Cholesky factor, but the zeros are lost in the inverse in the case of the lasso applied to the Cholesky factor. To select the tuning parameters, the authors used a resampling scheme, such as 5-fold CV.

Levina et al. (2008) did not discuss the theoretical properties of their estimator. While the nested lasso penalty is not convex, the theory developed by Fan and Li (2001) for nonconvex penalized maximum likelihood estimation, in the case of fixed p and $n \to \infty$, cannot be directly applied since the penalty cannot be decomposed as the sum of identical penalties on the individual coefficients (Levina et al., 2008).

Note the nested lasso penalty was also used by Rothman et al. (2010), who presented a new regression interpretation of the Cholesky factor of the covariance matrix (and not the inverse) and proposed to estimate a banded covariance matrix by banding the Cholesky factor of the covariance matrix.

2.2.4 Bayesian Estimation

Bayesian graphical lasso: The graphical lasso (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008) is a popular method for estimating graphical models (sparse

precision matrices). Wang (2012) considered the Bayesian version of the graphical lasso. Since maximum penalized likelihood estimation with the lasso penalty on the elements of Θ is equivalent to maximum *a posteriori* estimation when independent, double exponential priors are placed on the elements of the inverse, Wang (2012) had assumed the following model

$$\begin{aligned} p(\mathbf{x}_i \mid \Theta) &= \mathcal{N}(\mathbf{x}_i \mid \mathbf{0}, \Theta^{-1}) \text{ for } i = 1, \ldots, n, \\ p(\Theta|\lambda) &= C^{-1} \prod_{i<j} \left\{ \text{DE}(\theta_{ij} \mid \lambda) \right\} \prod_{i=1}^{p} \left\{ \text{Exp}\left(\theta_{ij} \mid \frac{\lambda}{2}\right) \right\} 1_{(\Theta \succ 0)}, \end{aligned} \tag{2.25}$$

where $\mathcal{N}(\mathbf{x} \mid \mathbf{0}, \Theta^{-1})$ represents the density function of a multivariate normal random variable with mean $\mathbf{0}$ and covariance matrix Θ^{-1}, evaluated at $\mathbf{x}$, $\text{DE}(x \mid \lambda)$ represents the double exponential density function of the form $p(x) = (\lambda/2)\exp(-\lambda|x|)$, $\text{Exp}(x \mid \lambda)$ represents the exponential density function of the form $p(x) = \lambda e^{-\lambda x}$, $x > 0$, and C is a normalizing constant not involving $\lambda > 0$. The posterior mode of Θ is the graphical lasso estimate with tuning parameter $\rho = \lambda/n$ for any fixed $\lambda > 0$. Wang (2012) developed a framework for efficient Bayesian inference for the graphical lasso model (2.25). They introduced a novel block Gibbs sampler for sampling Θ from model (2.25). They then generalized the Bayesian graphical lasso to the Bayesian adaptive graphical lasso.

Sparsity inducing priors based on the partial autocorrelation parametrization: In the case where there is an ordering among the variables, Gaskins et al. (2014) considered the problem of correlation matrix estimation in a Bayesian framework, where they focused on developing appropriate priors for the correlation matrix R. By considering the decomposition $\Theta = MR^{-1}M$, where M is diagonal with the partial standard deviations of $\mathbf{X}$ and R^{-1} is the matrix of full partial correlations, and reparametrizing $R = (\rho_{ij})$ in terms of the matrix of partial autocorrelations $\Pi = (\pi_{ij})$, the positive-definiteness constraint on R is removed. Now rather than imposing sparsity on R^{-1} directly, the authors imposed sparsity through the PAC's. A zero entry in the partial autocorrelation matrix Π implies that X_i and X_j for $i < j$ are uncorrelated given the

intervening variables $(X_{i+1}, \ldots, X_{j-1})$. Under multivariate normality, this implies that X_i and X_j are conditionally independent, given $(X_{i+1}, \ldots, X_{j-1})$. However, this does not mean that X_i and X_j are uncorrelated given the *remaining* variables and is thus not equivalent to conditional independence as in the normal case. The authors proposed two new prior distributions on the set of correlation matrices for ordered data through the partial autocorrelations.

Their first prior shrinks PACs toward zero with the strength of the shrinkage depending on lag. They used independent beta priors, shifted to the support (-1,1) with shape parameters depending on lag. More specifically, if $\text{SBeta}(\alpha, \beta)$ denotes the beta distribution with support shifted to (-1,1), then the shrinkage priors are formed by taking

$$\pi_{ij} \overset{\text{indep.}}{\sim} \text{SBeta}(\alpha_{ij}, \beta_{ij}),$$

where $\alpha_{ij} = \beta_{ij}$ so that $\mathbb{E}(\pi_{ij}) = 0$. Then setting

$$\xi_{ij} = \text{Var}(\pi_{ij}) = \frac{4\alpha_{ij}\beta_{ij}}{(\alpha_{ij} + \beta_{ij})^2(\alpha_{ij} + \beta_{ij} + 1)},$$

one finds $\alpha_{ij} = \beta_{ij} = (\xi_{ij}^{-1} - 1)/2$ so that the distribution of π_{ij} is determined by ξ_{ij}. The authors then parametrized ξ_{ij} as

$$\xi_{ij} = \epsilon_0 |j - i|^{-\gamma} \tag{2.26}$$

for $\epsilon_0 \in (0, 1)$ and $\gamma > 0$. Therefore, ξ_{ij} is now decreasing in lag so that higher lag terms will be shrunk toward zero. To complete the Bayesian specification, the authors used a uniform prior for ϵ_0 and a gamma prior for γ. Since γ is required to be positive, they take $\gamma \sim \text{Gamma}(5, 5)$ so that γ has a prior mean of 1 and a prior variance of $1/5$.

Their second prior is a selection prior that assigns positive probability to the event that π_{ij} is zero. The selection priors are formed by taking the prior for each π_{ij} as a mixture

of a degenerate distribution δ_0 with point mass at zero and a continuous distribution:

$$\pi_{ij} \overset{\text{indep.}}{\sim} \epsilon_{ij}\,\text{SBeta}(\alpha_{ij}, \beta_{ij}) + (1 - \epsilon_{ij})\delta_0.$$

In this case, Gaskins et al. (2014) let $\alpha = \alpha_{ij}$ and $\beta = \beta_{ij}$ so that the shape parameters do not depend on lag. Structure is imposed through ϵ_{ij} instead. The authors recommended either a uniform distribution on (-1,1) for α and β so $\alpha = \beta = 1$ or the triangular prior of Wang and Daniels (2013) with $\alpha = 2$ and $\beta = 1$. The PAC will be zero with probability $1 - \epsilon_{ij}$. As the values of the ϵ's decrease, more weight will be placed on the point mass at zero by the selection prior, yielding sparse partial autocorrelation matrices. Imposing structure through ϵ_{ij}, they let $\epsilon_{ij} = \epsilon_0|j - i|^{-\gamma}$. Note $\text{P}(\pi_{ij} = 0)$ increases with increasing lag since ϵ_{ij} decreases with increasing lag. As with the shrinkage prior, they chose hyperpriors $\epsilon_0 \sim \text{Unif}(0, 1)$ and $\gamma \sim \text{Gamma}(5, 5)$.

To gain a better understanding of the behaviour of their proposed priors, the authors assessed through simulation the (frequentist) risk of their posterior estimators. They considered two loss functions, the Kullback-Leibler loss of the covariance matrix,

$$\mathcal{L}_1(\hat{R}, R) = \text{tr}(\hat{R}R^{-1}) - \log|\hat{R}R^{-1}| - p$$

and

$$\mathcal{L}_2(\hat{\Pi}, \Pi) = \sum_{i<j}(\hat{\pi}_{ij} - \pi_{ij})^2.$$

They compared their shrinkage and selection priors to four competing priors: (1) a flat prior on R (Barnard et al., 2000), where each ρ_{ij} is uniform on (-1,1), (2) a flat prior on Π, where each π_{ij} is uniform on (-1,1), (3) the triangular prior of Wang and Daniels (2013), where each $\pi_{ij} \sim \text{SBeta}(2, 1)$, and (4) a naive shrinkage prior where $\gamma = 0$ in (2.26) so that all π_{ij}'s are shrunk independently of the lag. They considered four correlation structures: an autoregressive (AR) structure of order 1, an independent correlation structure ($R = I_p$), a nonzero decaying structure in Π, and a banded structure in Π. They considered

samples of size $n = 20, 50, 200$ and considered only low dimensions $p = 6, 10$. For the first two correlation structures, which have the most sparsity in Π, they found that their shrinkage and selection priors outperform the other four priors. In the independence case, the naive shrinkage prior performs better than the triangular prior and the two flat priors. For the third correlation structure, which has a non-sparse Π, all priors perform comparably. Finally, for the fourth correlation structure, which has zero π_{ij}'s, the shrinkage and selection priors outperform the other four priors.

2.3 Conclusion

In this chapter, we provided a review of existing methods in the literature for performing shrinkage and selection in the regression and inverse covariance estimation context. Working in the penalized likelihood framework, we reviewed various penalties that have been proposed, along with computational algorithms for solving the resulting penalized likelihood problems, as well as asymptotic properties of the penalized likelihood estimators. We also reviewed procedures for selecting the tuning parameter in the penalized likelihood, which controls the amount of regularization, and studied some asymptotic properties of these procedures. The material presented in this chapter will help us to understand the work presented in subsequent chapters.

Chapter 3

Inverse Covariance Estimation for Ordered Data via Banding the Partial Autocorrelation Matrix

3.1 Introduction

The problem of (inverse) covariance matrix estimation has been an active area of research, particularly in the moderate-to-high dimensional setting, where the dimension of the data p is comparable to or larger than the sample size n. With advancements in computing, high-dimensional data have become increasingly common in domains such as genetics, finance, spectroscopy and climatology. Several areas of multivariate statistical analysis require the estimation of a covariance matrix, including dimensionality reduction by principal component analysis (PCA) and establishing conditional independence relationships between components in graphical models (Bickel and Levina, 2008b). Although the sample covariance matrix is unbiased and positive definite, it is a poor estimator when $p \gg n$ (see Johnstone, 2001, and references therein). In this case, better-conditioned covariance estimators are desired.

In addition to the issue of high-dimensionality, another major challenge in covariance matrix estimation is the positive-definiteness constraint, which makes the elements of the covariance matrix algebraically dependent. Changing one element of the covariance matrix generally affects the values of the other elements as well.

To address the issue of high-dimensionality, where the number of parameters grows

quadratically in p, many of the methods that have been proposed thus far have utilized sparsity assumptions about the covariance matrix Σ or its inverse $\Theta = \Sigma^{-1}$, known as the precision matrix. The idea of setting elements of the precision matrix to zero, known as *covariance selection*, was discussed by Dempster (1972), who argued that, for the sake of parsimony and to decrease estimation error, sparsity in Θ is preferred.

Identifying the zero entries of the precision matrix Θ is also of particular importance in graphical modelling. For Gaussian graphical models, where $\mathbf{X} = (X_1, \ldots, X_p)^T$ is a p-dimensional random vector having a multivariate normal distribution with mean vector μ and covariance matrix Σ, a zero in the $(i, j)^{\text{th}}$ entry of Θ implies a conditional independence relationship between the variables X_i and X_j, given all other variables. For a proof of this result, see Lauritzen (1996). The conditional independence structure can be represented by an undirected graph $\mathcal{G}$, consisting of a set of vertices V and a set of edges E, where an edge connects a pair of variables if and only if they are conditionally dependent. Thus, by identifying the sparsity pattern of Θ, the graph structure is obtained.

To obtain sparse estimates of Θ, the most widely used approach is to penalize the log-likelihood with an L_1 penalty on the elements of the inverse covariance matrix (Banerjee et al., 2006; Yuan and Lin, 2007; Friedman et al., 2008). While the L_1 penalty leads to an advantageously convex optimization problem, it introduces bias in the estimation (see Lam and Fan, 2009). As a remedy to the bias issue, Fan et al. (2009) considered nonconcave penalties, such as the SCAD (Fan and Li, 2001) penalty. The problem of estimating the precision matrix then becomes equivalent to solving a sequence of weighted L_1-penalized likelihood problems, which can be solved by taking advantage of the efficient graphical lasso algorithm. While both methods obtain positive definite solutions, provided that the procedures are initialized with positive definite matrices, the penalties themselves do not incorporate the dependence between the θ_{ij}'s arising from the positive-definiteness constraint on Θ, as they are sums of identical penalties on the individual entries of Θ. While the L_1-penalized likelihood method has become popular for estimating a precision matrix - in part because of the many efficient algorithms developed for solving the optimization problem - in this thesis, we hope to highlight

the deficiencies of penalized likelihood methods that penalize the entries of the precision matrix independently of each other. Such penalization is often inappropriate for problems involving estimating a covariance matrix or its inverse, especially when the matrix of interest is structured. The choice of the appropriate penalty largely depends on the problem at hand.

To remove the positive-definiteness constraint, reparametrizations of the covariance matrix or its inverse have been used, but come at the expense of imposing an ordering among the variables in $\mathbf{X}$. The most commonly used is the modified Cholesky decomposition of either the covariance matrix or its inverse (Pourahmadi, 1999), reviewed in Section 2.2.3.1. The modified Cholesky decomposition of the inverse is given by

$$\Theta = L^T D^{-1} L,$$

where lower triangular matrix L and diagonal matrix D are defined in Section 2.2.3.1. The elements in the subdiagonals of L and the logarithm of the diagonal elements of D are unconstrained and so any estimate $(\hat{L}, \hat{D}^{-1})$ yields a positive definite estimated precision matrix $\hat{\Theta} = \hat{L}^T \hat{D}^{-1} \hat{L}$.

Therefore, in the case where variables have a natural ordering, sparsity in the inverse is introduced by imposing sparsity in the Cholesky factor L. In particular, some authors (Wu and Pourahmadi, 2003; Bickel and Levina, 2008a) estimated a banded inverse covariance matrix by banding the Cholesky factor L. Huang et al. (2006), on the other hand, assumed a sparse Cholesky factor without assuming a banded structure in L. They proposed penalizing the Gaussian log-likelihood by either an L_1 or L_2 penalty on the elements of the Cholesky factor L, which would help to produce more stable estimators by introducing shrinkage in L. For the L_1 penalty, however, zeros are introduced in arbitrary locations in the Cholesky factor and the resulting estimated inverse may not be sparse at all. Therefore, Levina et al. (2008) introduced a method called adaptive banding, which places a nested lasso penalty on the elements of the Cholesky factor that imposes a banded structure in L. Their approach is more flexible than regular banding as the nested lasso penalty allows the rows of the Cholesky factor to have varying band lengths.

Furthermore, unlike the method of Huang et al. (2006), their method preserves sparsity in the inverse since the inverse itself is banded if the Cholesky factor is banded.

The modified Cholesky decomposition is not the only reparametrization that removes the positive-definiteness constraint on a covariance matrix and its inverse. An alternative reparametrization of the covariance matrix that removes the positive-definiteness constraint is the partial autocorrelation (PAC) parametrization. Starting with the variance-correlation decomposition $\Sigma = VRV$, where V is a diagonal matrix with the marginal standard deviations of $\mathbf{X}$ and R is the correlation matrix, the precision matrix can be decomposed as

$$\Theta = MR^{-1}M \tag{3.1}$$

so that the diagonal elements of M give the partial standard deviations and $R^{-1} = (\rho^{ij})$ is the matrix of (full) partial correlations. The covariance selection problem is thus equivalent to identifying the zero elements of R^{-1}. For a random vector $\mathbf{X} = (X_1, \ldots, X_p)^T$, the partial autocorrelation between X_i and X_j for $i < j$, which we denote by π_{ij}, is the correlation between the two, after controlling for the effects of the *intervening* variables $X_{i+1}, \ldots, X_{j-1}$. Note that this is in contrast to the (full) partial correlation ρ^{ij}, which is defined as the correlation between X_i and X_j, after controlling for the effects of the *remaining* variables. The correlation matrix R can be reparametrized in terms of the symmetric matrix of partial autocorrelations $\Pi = (\pi_{ij})$, where $\pi_{ii} = 1$ and the entries on the off-diagonals of Π vary freely in the interval (-1,1). With a one-to-one correspondence between the matrices R and Π (Joe, 2006), the complex constraints on the correlation matrix can therefore be avoided by considering this alternative parametrization.

The PAC parametrization has been largely used in a Bayesian setting for constructing priors for the correlation matrix R. In this chapter, we work within the frequentist penalized likelihood framework and propose to estimate the inverse covariance matrix for ordered data by maximizing the Gaussian log-likelihood with a nested lasso penalty (Levina et al., 2008) on the matrix of partial autocorrelations Π. The advantage of the PACs is that they allow for penalization under a more natural parametrization in an

unconstrained setting. While an L_1 penalty applied to the partial autocorrelation matrix does not result in the preservation of zeros in the corresponding inverse covariance matrix, the banded structure in the partial autocorrelation matrix imposed by the nested lasso penalty corresponds to a banded structure in the resulting inverse so that sparsity is preserved. A banded structure in Π is a reasonable assumption in the time-ordered setting that we are considering, where it is expected that PACs of large lags are small.

The remainder of this chapter is organized as follows. In Section 3.2, we discuss our motivation for considering an alternative parametrization of a covariance matrix. This will be followed by a review of the PAC parametrization in Section 3.3. We then outline our proposed penalized maximum likelihood method for estimating Θ in Section 3.4, based on penalization of the PACs through the use of the nested lasso penalty (Levina et al., 2008). Section 3.5 provides details of our coordinate descent procedure for maximizing the penalized log-likelihood. The main competitor of our proposed PAC-based nested lasso method is the adaptive banding method of Levina et al. (2008) that uses a nested lasso penalty on the Cholesky factor. In Section 3.6, we thus review the connections established by Pourahmadi (2001), and Daniels and Pourahmadi (2009) between the PACs and the parameters in the modified Cholesky decomposition of the inverse. In Section 3.7, we then identify cases where using the PAC paramaterization would be advantageous over the modified Cholesky decomposition. In Section 3.8, we discuss the selection of the tuning parameter in our penalized maximum likelihood method. We then assess the performance of our method through simulation in Section 3.10 and analyze a real dataset in Section 3.11 to illustrate the methodology developed in this chapter. We conclude with a discussion in Section 3.12.

3.2 Motivation for Alternative Parametrization

Sparsity of the precision matrix Θ is often studied via penalized likelihood methods. These methods estimate Θ by maximizing the following penalized log-likelihood function

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \sum_{i\neq j} p_{\lambda_{ij}}(|\theta_{ij}|) \tag{3.2}$$

over the set of positive definite matrices Θ. Here $S = \mathbf{X}^T\mathbf{X}/n$ is the sample covariance matrix, θ_{ij} is the $(i,j)^{\text{th}}$ entry of Θ, and $p_{\lambda_{ij}}(\cdot)$ denotes a generic penalty function on θ_{ij} with corresponding tuning parameter $\lambda_{ij} > 0$. The tuning parameters λ_{ij}, $i,j = 1,\ldots,p$, control the level of sparsity of Θ. The most widely known penalized likelihood method for estimating the precision matrix Θ is the graphical lasso algorithm of Friedman et al. (2008). In this case, the penalty function in (3.2) is the L_1 penalty function $p_\lambda(|x|) = \lambda|x|$ for some $\lambda > 0$. To remedy the well-known bias issue resulting from the L_1-penalty, Fan et al. (2009) consider alternative penalty functions, such as the SCAD penalty (Fan and Li, 2001) and the adaptive lasso penalty (Zou, 2006). Note that some authors consider a slightly modified version of (3.2), where the diagonal elements of Θ are also penalized. We focus this discussion on the graphical lasso, where the function in (3.2) becomes

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \lambda\sum_{i\neq j}|\theta_{ij}|. \tag{3.3}$$

We make a few observations regarding the graphical lasso. Firstly, the graphical lasso penalizes the entries of Θ independently. Due to the positive-definite constraint on Θ, the set of values that any particular element θ_{ij} can take depends on the choice of the remaining θ_{ij}'s. While this constraint is taken into account by the block coordinate descent algorithm of Friedman et al. (2008) for maximizing (3.3), provided the procedure is initialized with a positive definite matrix, the complex relationship between the θ_{ij}'s is not reflected in the penalty itself.

Secondly, the graphical lasso is not scale-invariant. This can be observed by first considering the variance-correlation decomposition of Σ, which allows us to write $\Theta =$

$MR^{-1}M$, as shown in (3.1) When one applies the graphical lasso with a given tuning parameter $\lambda > 0$ to the sample correlation matrix (in effect standardizing the variables to unit variance) to obtain $\hat{R}^{-1}$, the reconstruction $\hat{\Theta} = M\hat{R}^{-1}M$ is not the same as the estimated precision matrix obtained by the graphical lasso applied to the sample covariance matrix S (see Section 3.2.1). This suggests that working on the correlation scale, where all entries lie in the interval [-1,1], would be more appropriate so that the penalization is done on the same scale.

3.2.1 Using the Sample Correlation Matrix Instead of the Sample Covariance Matrix in the Graphical Lasso

Let s_{ii} for $i = 1, \ldots, p$, be the diagonal elements of the sample covariance matrix S, and let $\hat{M} = \text{diag}(1/\sqrt{s_{11}}, \ldots, 1/\sqrt{s_{pp}})$ so that $\hat{R} = \hat{M}S\hat{M}$ is the sample correlation matrix. With $\Omega = R^{-1}$, consider maximizing the objective function

$$\log\det\Omega - \text{tr}(\hat{R}\Omega) - \lambda\sum_{i\neq j}|\omega_{ij}| \tag{3.4}$$

to obtain the estimator $\hat{\Omega}$ of Ω and then taking $\hat{\Theta} = \hat{M}\hat{\Omega}\hat{M}$ as an estimator of the precision matrix Θ. Writing (3.4) as a function of $\Theta = M\Omega M$ and using the fact that for n large, s_{ii} is approximately equal to the diagonal entries of Σ, we find that the objective function is approximately given by

$$\sum_{i=1}^{p}\log(s_{ii}) + \log\det\Theta - \text{tr}(S\Theta) - \lambda\sum_{i\neq j}\sqrt{s_{ii}s_{jj}}\,|\theta_{ij}|. \tag{3.5}$$

Therefore, for n large, using the graphical lasso based on the sample correlation matrix is equivalent to using the graphical adaptive lasso based on the sample covariance matrix with weights given by $w_{ij} = \sqrt{s_{ii}s_{jj}}$ and tuning parameter λ.

3.3 A Review of the Partial Autocorrelation Parametrization

One of the main challenges when modelling the $p \times p$ correlation matrix $R = (\rho_{ij})$ is the constraint of positive-definiteness. This difficulty can be circumvented by reparametrizing the correlation matrix in terms of the matrix of partial autocorrelations $\Pi = (\pi_{ij})$, which is a symmetric matrix with $\pi_{ii} = 1$ and for $i < j$, π_{ij} is the correlation between X_i and X_j, adjusted for the intervening variables $X_{i+1}, \ldots, X_{j-1}$. Note that this is in contrast to the (full) partial correlation ρ_{ij}, which is defined as the correlation between X_i and X_j, conditional on *all* the other variables. If we let $U = \{i+1, \ldots, j-1\}$ and if we denote the linear least squares predictor of X_i based on X_t, $t \in U$ by $\hat{X}_{i|U}$, then $\pi_{ij} = \text{Corr}(X_i - \hat{X}_{i|U}, X_j - \hat{X}_{j|U})$ and so π_{ij} can be interpreted as the correlation between X_i and X_j after correcting for X_t, $t \in U$.

The advantage of working with the partial autocorrelations π_{ij} is that they can vary independently of each other in the interval $(-1, 1)$. With a one-to-one correspondence between the matrices R and Π found in Joe (2006), which we detail below, the complex constraints on the correlation matrix can be avoided by considering this alternative parametrization.

Using the following recursion formula (Joe, 2006), the partial autocorrelations (π_{ij}) can be computed in terms of the marginal correlations (ρ_{ij}). For the lag-1 partial autocorrelations, $\pi_{i,i+1} = \rho_{i,i+1}$ for $i = 1, \ldots, p-1$. From Joe (2006), higher lag $(j - i > 1)$ partial autocorrelations can be computed using the expression

$$\pi_{ij} = D_{ij}^{-1}\Big\{\rho_{ij} - r_1^T(i,j)R_2(i,j)^{-1}r_3(i,j)\Big\}, \tag{3.6}$$

where $r_1^T(i,j) = (\rho_{i,i+1}, \ldots, \rho_{i,j-1})$, $r_3^T(i,j) = (\rho_{j,i+1}, \ldots, \rho_{j,j-1})$, $R_2(i,j)$ is the correlation matrix corresponding to the variables $(X_{i+1}, \ldots, X_{j-1})$, and

$$D_{ij} = [1 - r_1^T(i,j)R_2(i,j)^{-1}r_1(i,j)]^{1/2}[1 - r_3^T(i,j)R_2(i,j)^{-1}r_3(i,j)]^{1/2}.$$

Note the function in (3.6) that maps the correlation matrix R to the partial autocor-

relation matrix Π is invertible; one obtains the marginal correlations from the partial autocorrelations using the expression

$$\rho_{ij} = r_1^T(i,j)R_2(i,j)^{-1}r_3(i,j) + D_{ij}\pi_{i,j} \tag{3.7}$$

for $j - i > 1$.

Due to the positive-definiteness constraint on R, each marginal correlation ρ_{ij} takes a value from a subset of (-1,1) that depends on the structure of the remaining entries of R. On the other hand, each π_{ij} can take any value in the interval (-1,1) irrespective of the values of the other partial autocorrelations while maintaining positive-definiteness of R. Since the parameters π_{ij} are restricted to the interval (-1,1), Fisher's z-transform can be used to map the π_{ij}'s for $i \neq j$ to the entire real line.

3.4 The Proposed Method

In this section, we introduce our proposed PAC-based penalized likelihood method for estimating the inverse covariance matrix of ordered data. Suppose that we observe a sample of n independently drawn multivariate Gaussian random vectors, $\mathbf{X}_1, \ldots, \mathbf{X}_n \sim \mathcal{N}(0, \Sigma)$, where Σ is the unknown $p \times p$ covariance matrix. Let $\Theta = \Sigma^{-1}$, which we can write as $\Theta = M\Omega M$, where Ω is the matrix of partial correlations R^{-1}. Further, let $\Omega = g(\Pi)$, where g maps the partial autocorrelation matrix Π to Ω. We propose to estimate the matrix of partial autocorrelations Π by maximizing the Gaussian log-likelihood function subject to penalty $p_\lambda(\cdot)$. In other words, we solve the optimization problem

$$\hat{\Pi} = \underset{\Pi \in \mathcal{D}}{\arg\max} \left\{ \log\det\left[g(\Pi)\right] - \mathrm{tr}[\hat{R}\, g(\Pi)] - p_\lambda(\Pi) \right\}, \tag{3.8}$$

where $\mathcal{D} = \{\Pi = (\pi_{ij}) : \pi_{ii} = 1, \pi_{ij} = \pi_{ji} \in (-1,1), i,j = 1,\ldots,p\}$. Various penalty functions have been proposed in the penalized likelihood framework; the most widely used is the L_1 penalty. However, while the L_1 penalty on the PACs would help to produce more

stable estimators by introducing shrinkage to the elements of the partial autocorrelation matrix, a zero π_{ij} does not generally imply a zero partial correlation. Therefore, it would be best to consider a penalty that would impose some kind of structure, in particular, a banded structure, on the partial autocorrelation matrix. This would then lead to a banded structure in the inverse. Wang and Daniels (2014) showed that the precision matrix corresponding to a k-banded partial autocorrelation matrix is also k-banded. To impose a banded structure on the partial autocorrelation matrix Π, we use the nested lasso penalty of Levina et al. (2008). Let $\boldsymbol{\pi}_i = (\pi_{i,i+1}, \pi_{i,i+2}, \ldots, \pi_{i,p})^T$ for $i = 1, \ldots, p-1$. The nested lasso penalty function is given by $p_\lambda(\Pi) = \lambda \sum_{i=1}^{p-1} q(\boldsymbol{\pi}_i)$, where

$$\begin{aligned} q(\boldsymbol{\pi}_i) =& |\pi_{i,i+1}| + \frac{|\pi_{i,i+2}|}{|\pi_{i,i+1}|} + \frac{|\pi_{i,i+3}|}{|\pi_{i,i+2}|} + \ldots + \frac{|\pi_{i,p}|}{|\pi_{i,p-1}|} \\ =& |\pi_{i,i+1}| + \sum_{j=i+2}^{p} \frac{|\pi_{i,j}|}{|\pi_{i,j-1}|} \\ =& \sum_{j=i+1}^{p} \frac{|\pi_{i,j}|}{|\pi_{i,j-1}|} \end{aligned} \tag{3.9}$$

and the last equality follows since $\pi_{ii} = 1$ for all $i = 1, \ldots, p$.

Penalizing the PACs instead of the full partial correlations has three main advantages. Firstly, the PACs are functionally independent of each other, allowing the positive-definiteness constraint on R^{-1} to be avoided. Secondly, the PACs vary on the same scale, allowing for them to be penalized with the same tuning parameter. Not only do the π_{ij}'s vary on the same scale, but the ratios $\psi_{ij} = \pi_{ij}/\pi_{i,j-1}$ as well, making the penalization of the ratios with a single tuning parameter appropriate. Finally, since the PACs are not conditional on future values, they offer greater interpretability than the full partial correlations in applications with time-dependent data (Gaskins et al., 2014).

We let $\hat{\Omega} = g(\hat{\Pi})$ denote the penalized maximum likelihood estimator of $\Omega = R^{-1}$. Taking $\hat{M}$ to be the diagonal matrix with the MLE's of the marginal variances along its diagonal, an estimator $\hat{\Theta}$ of the inverse covariance matrix is then $\hat{\Theta} = \hat{M}\hat{\Omega}\hat{M}$.

3.5 Computational Considerations

The maximization of (3.8) with the nested lasso penalty in (3.9) is a non-convex problem. Therefore, rather than finding a global optimum, we seek a local optimum using an iterative procedure. The algorithm requires the specification of an initial estimate $\hat{\Pi}^{(0)}$. For $p < n$, one could take the sample partial autocorrelation matrix, obtained by transforming the sample correlation matrix. For $p > n$, the sample correlation matrix is singular and so we shrink the inverse sample correlation matrix using the graphical lasso until a non-singular starting value is obtained, which is then transformed to the corresponding partial autocorrelation matrix.

To solve the optimization problem in (3.8) with the nested lasso penalty, we use a coordinate descent procedure, where we optimize the objective function in (3.8) with respect to one entry π_{ij} at a time, while holding the others fixed. We then cycle through the entries π_{ij} several times until convergence. Since the PACs vary independently of each other in the interval (-1,1), we can update each π_{ij} one at a time (while holding the others fixed) without needing to take into account any positive-definiteness constraint on R^{-1}. Each update must be done numerically, using an optimization algorithm, such as `optimize` in `R`. Once $\hat{\pi}_{ij} = 0$, we set $\hat{\pi}_{i,j+1} = \ldots = \hat{\pi}_{ip} = 0$ because of the nature of the nested lasso penalty.

Another approach would be to reparametrize the partial autocorrelations π_{ij} to the ratios $\alpha_{ij} = \pi_{i,j+1}/\pi_{ij}$ and view the nested lasso penalty as an L_1 penalty on the ratios α_{ij}. The α_{ij}'s are not free of constraints because

$$\alpha_{ii} = \pi_{i,i+1} \in (-1, 1)$$
$$\alpha_{ij} = \frac{\pi_{i,j+1}}{\pi_{ij}} \Longrightarrow \pi_{i,j+1} = \alpha_{ij}\pi_{ij}$$

and since $-1 < \pi_{i,j+1} < 1$, we have that $-\frac{1}{\pi_{ij}} < \alpha_{ij} < \frac{1}{\pi_{ij}}$. A coordinate descent procedure can then be used to solve for the α_{ij}'s. Each update must also be done numerically. During the procedure, we track the $\hat{\alpha}_{ij}$'s and the $\hat{\pi}_{ij}$'s and, for row i, once $\hat{\pi}_{ij} = 0$ for some j, we

set $\hat{\pi}_{i,j+1} = \hat{\pi}_{i,j+2} = \ldots = \hat{\pi}_{i,p} = 0$.

Since the optimization problem is not convex, it is not guaranteed to converge to the global optimum, but it is an ascent algorithm in which each iteration increases the objective. Therefore, if the procedure is initialized with suitable estimates, the algorithm should still yield good empirical results.

For numerical stability, we threshold the absolute values of the partial autocorrelations at some pre-specified $\epsilon > 0$. We take $\epsilon = 10^{-6}$. Once convergence is achieved, we set all estimates equal to ϵ to zero. We also make use of "warm starts" to improve the efficiency of this procedure.

3.6 Connections Between the Partial Autocorrelation Parametrization and the Modified Cholesky Decomposition

In Section 2.2.3.1, we presented the modified Cholesky decomposition of the inverse covariance matrix Θ. In what follows, we review the connections established in Pourahmadi (2001) and Daniels and Pourahmadi (2009) between the partial autcorrelations and the parameters in the modified Cholesky decomposition. Recall, if $\mathbf{X}$ is a mean-zero random vector with covariance matrix Σ and $\hat{X}_j$ is the linear least-squares predictor of X_j based on its predecessors $X_1, \ldots, X_{j-1}$, where $\epsilon_j = X_j - \hat{X}_j$ is its prediction error with variance $\sigma_j^2 = \text{Var}(\epsilon_j)$, then there are unique scalars ϕ_{jt} such that

$$X_j = \sum_{t=1}^{j-1} \phi_{jt} X_t + \epsilon_j, \qquad j = 2, \ldots, p.$$

Now as detailed in Section 2.2.3.1, we can write the modified Cholesky decompositions of Σ and Σ^{-1} as

$$\Sigma = L^{-1} D (L^{-1})^T, \qquad \Sigma^{-1} = L^T D^{-1} L,$$

where $D = \text{diag}(\sigma_1^2, \ldots \sigma_p^2)$ and L is the following lower triangular matrix:

$$L = \begin{pmatrix} 1 & & & & & \\ -\phi_{21} & 1 & & & & \\ -\phi_{31} & -\phi_{32} & 1 & & & \\ -\phi_{41} & -\phi_{42} & -\phi_{43} & 1 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ -\phi_{p1} & -\phi_{p2} & -\phi_{p3} & \ldots & -\phi_{p,p-1} & 1 \end{pmatrix}.$$

Lemma 1 and Theorem 1 from Daniels and Pourahmadi (2009), which we reproduce below, demonstrate that only the entries in the first column of L are multiples of the partial autocorrelations in the first column of Π (first proved in Pourahmadi, 2001). In Lemma 1(a), it can be seen that for $t > 1$, ϕ_{jt} is not of the form of the entries of Π since it is a multiple of the partial correlation between X_j and X_t, adjusted for $\{X_i : i \in U\}$, where $U = \{1, \ldots, j-1\} \setminus \{t\}$.

Lemma 1. (Daniels and Pourahmadi, 2009) Let $\mathbf{X} = (X_1, \ldots, X_p)^T$ be a mean-zero random vector with covariance matrix Σ that can be decomposed as in (2.22). For j, t such that $j > t$, let $U = \{1, \ldots, j-1\} \setminus \{t\}$ and $\pi_{jt|U}$ stand for the partial correlation between X_j and X_t adjusted for $X_l, l \in U$. Denote the linear least squares predictor of X_j based on $X_l, l \in U$ by $\hat{X}_{j|U}$. Then the following results hold.

(a) $\hat{X}_{j|\{1,\ldots,j-1\}} = \hat{X}_{j|U} + \phi_{jt}\left(X_t - \hat{X}_{t|U}\right), \qquad \phi_{jt} = \pi_{jt|U}\sqrt{\frac{\text{Var}(X_j - \hat{X}_{j|U})}{\text{Var}(X_t - \hat{X}_{t|U})}}.$

(b) $\text{Var}(X_j - \hat{X}_{j|\{1,\ldots,j-1\}}) = (1 - \pi_{jt|U}^2)\text{Var}(X_t - \hat{X}_{t|U})$.

Theorem 1. (Daniels and Pourahmadi, 2009) Let $\mathbf{X} = (X_1, \ldots, X_p)^T$ be a mean-zero random vector with covariance matrix $\Sigma = (\sigma_{ij})$ that can be decomposed as in (2.22). Then the following results hold.

(a) For $j = 2, \ldots, p$ and $t = 1, \ldots, j-1$, $\sigma_j^2 = \sigma_{jj}\prod_{t=1}^{j-1}(1 - \pi_{tj}^2)$.

(b) The determinant of Σ can be simply expressed in terms of the marginal variances σ_{jj} and the partial autocorrelations π_{tj}:

$$|\Sigma| = \left(\prod_{j=1}^{p}\sigma_{jj}\right)\prod_{j=2}^{p}\prod_{t=1}^{j-1}(1 - \pi_{tj}^2).$$

(c) For $j = 2, \ldots, p$ and $U = \{2, \ldots, j-1\}$,

$$\phi_{j1} = \pi_{1j} \sqrt{\frac{\text{Var}(X_j - \hat{X}_{j|U})}{\text{Var}(X_1 - \hat{X}_{1|U})}}. \tag{3.10}$$

(d) For $t = 2, \ldots, j-1$, $\phi_{jt} = \phi_{jt|U} - \phi_{j1}\phi_{1,j-t|U}$, where $\phi_{jt|U}$ and $\phi_{1,j-t|U}$ are, respectively, the forward and backward predictor coefficients of X_j and X_1 based on $X_k, k \in U$, defined by

$$\hat{X}_{j|U} = \sum_{t=2}^{j-1} \phi_{jt|U} X_t, \qquad \hat{X}_{1|U} = \sum_{t=2}^{j-1} \phi_{1,j-t|U} X_t.$$

It should be noted that the Cholesky factor L and the diagonal matrix D of prediction variances are not fully unconstrained in the case where $\Sigma = R$ and R is a stationary (Toeplitz) correlation matrix. In this case, the diagonal elements of D, given by σ_j^2 for $j = 1, \ldots, p$, in the modified Cholesky decomposition of the inverse are monotone decreasing with $\sigma_1^2 = 1$. There are also additional constraints on L, for example, $\phi_{21} \in (-1, 1)$. When considering the partial autocorrelation parametrization, if R is stationary, then $\pi_{i,i+k}$ depends only on the lag k. Therefore, stationarity is a simplifying assumption in the case of the partial autocorrelation parametrization. It thus seems easier to use than the Cholesky decomposition when estimating a stationary correlation matrix R.

3.7 A Comparison of the Nested Lasso Penalty on the Cholesky Factor and the Partial Autocorrelation Matrix

One drawback of the nested lasso penalty is that it does not allow for the elimination of weaker signals in between strong signals. Applying the nested lasso penalty to the Cholesky factor of the inverse would therefore not be suitable in the case where each X_i follows a stationary subset AR(q) process, where the autoregressive coefficient corre-

sponding to lag q is non-zero and one or more intermediate AR coefficients are zero. The Cholesky factor would then be q-banded with one or more of the first $q-1$ bands having entries that are zero, while the corresponding partial autocorrelation matrix would also be q-banded, but all of its first q bands may be non-zero. For example, suppose that X is the following stationary Gaussian AR(3) process

$$X_t = 0.6X_{t-1} + 0.3X_{t-3} + e_t, \qquad t = 4, \ldots, p,$$

where $e_t \sim \mathcal{N}(0, \sigma_e^2)$. This implies a Cholesky factor with non-zero elements in the first and third sub-diagonals and zero elements everywhere else. On the other hand, the corresponding partial autocorrelation matrix is tri-diagonal with a non-zero constant along each of the first 3 off-diagonal bands (although the constants are not necessarily decaying with lag). Applying the nested lasso penalty to the Cholesky factor in this case would either set all off-diagonal bands after the first three to zero without setting the second off-diagonal band to zero, or set all off-diagonal bands after the first band to zero, wrongly eliminating the non-zero entries in the third sub-diagonal. In Figure 3.1, we plot the theoretical ACF and PACF of this AR(3) process. It can be seen that the first three lags of the PACF are non-zero and the remaining lags of the PACF are zero, and therefore applying the nested lasso penalty to the partial autocorrelation matrix would rightfully set all off-diagonal bands after the first three to zero. It is this observation that motivated us to consider the penalization of the PACs with the nested lasso penalty in a likelihood-based method as a means of identifying the order of an autoregressive process in Chapter 4.

3.8 Tuning Parameter Selection

The tuning parameter in the PAC-based penalized likelihood method can be selected using standard information criteria, such as AIC and BIC. In this context, AIC and BIC

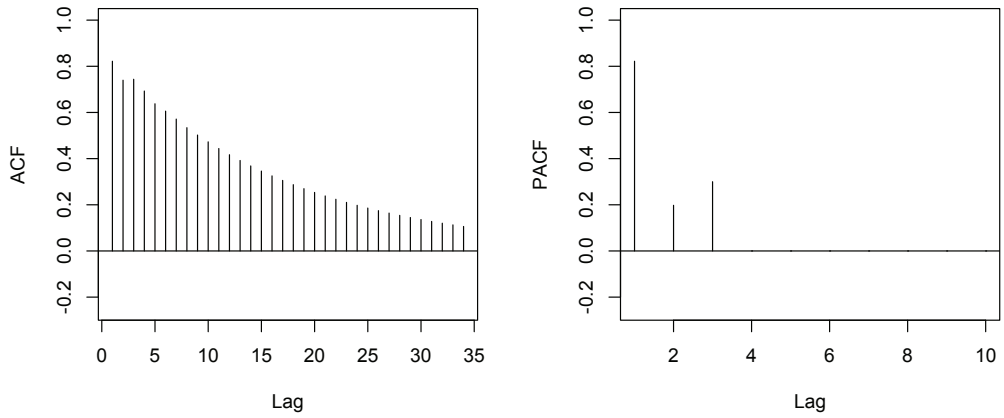

Figure 3.1: Theoretical autocorrelations (left) and partial autocorrelations (right) for the AR(3) process $X_t = 0.6X_{t-1} + 0.3X_{t-3} + e_t$ for $t = 4, \ldots, p$ and $e_t \sim \mathcal{N}(0, \sigma_e^2)$.

are given by

$$\text{AIC}(\lambda) = -2\ell_n(\hat{\Pi}_\lambda) + 2\sum_{i<j} I(\hat{\pi}_{ij,\lambda} \neq 0)$$

and

$$\text{BIC}(\lambda) = -2\ell_n(\hat{\Pi}_\lambda) + \log n \sum_{i<j} I(\hat{\pi}_{ij,\lambda} \neq 0),$$

respectively, where $\ell_n(\hat{\Pi}_\lambda)$ is the multivariate Gaussian log-likelihood, evaluated at $\hat{\Pi}_\lambda$, the penalized maximum likelihood estimator of Π for a given λ. The optimal value of the tuning parameter in either case is taken to be the minimizer of the criterion. In our simulation studies, EBIC was also considered, but was found to be too heavy for the sample sizes and dimensions considered.

3.9 A Discussion of Loss Functions

Regularization is introduced with the goal of minimizing suitable norms, risks or objective functions. If the inverse covariance matrix is the parameter of interest, the two most

commonly used loss functions are

$$\mathcal{L}_1(\hat{\Theta}, \Theta) = \text{tr}(\hat{\Theta}\Theta^{-1}) - \log\det(\hat{\Theta}\Theta^{-1}) - p,$$
$$\mathcal{L}_2(\hat{\Theta}, \Theta) = \text{tr}(\Theta^{-1}\hat{\Theta} - I)^2 = \|\Theta^{-1}\hat{\Theta} - I\|_F^2,$$

where $\|A\|_F = \sqrt{\text{tr}(AA^T)}$ is the Frobenius norm of matrix A. The first loss function, $\mathcal{L}_1$, is the Kullback-Leibler loss for the precision matrix. It is the Kullback-Leibler divergence of two multivariate Gaussian densities corresponding to the two precision matrices Θ and $\hat{\Theta}$, and was used in Yuan and Lin (2007), and Levina et al. (2008). Interchanging the roles of the covariance matrix and its inverse in the Kullback-Leibler loss yields the entropy loss for the covariance matrix, which was used by Huang et al. (2006) and Gaskins et al. (2014). The entropy loss is used when the covariance matrix is the primary matrix of interest. The second loss function, $\mathcal{L}_2$, is the quadratic loss function, which is the squared Frobenius norm of the matrix $\Theta^{-1}\hat{\Theta} - I$. It favours "smaller" estimates compared to the $\mathcal{L}_1$ loss.

The corresponding risk functions are

$$\mathcal{R}_i(\hat{\Theta}, \Theta) = \mathbb{E}_\Theta\left[\mathcal{L}_i(\hat{\Theta}, \Theta)\right], \quad i = 1, 2.$$

An estimator $\hat{\Theta}_1$ is considered better than another estimator $\hat{\Theta}_2$ is its risk function is smaller.

Other loss functions can be used to evaluate the performance of an estimator $\hat{\Theta}$ of Θ. Considering the Frobenius norm of the difference $\hat{\Theta} - \Theta$, one obtains the loss function

$$\mathcal{L}_3(\hat{\Theta}, \Theta) = \frac{1}{p}\|\hat{\Theta} - \Theta\|_F^2 = \frac{1}{p}\text{tr}(\hat{\Theta} - \Theta)^2 = \frac{1}{p}\sum_{i,j}(\hat{\theta}_{ij} - \theta_{ij})^2.$$

By dividing by p, the norm of the identity matrix is one. Such a loss function was considered by Ledoit and Wolf (2004), but with the precision matrix replaced by the covariance matrix.

We consider these loss functions to evaluate the performance of our proposed penalized

likelihood estimators in Section 3.10. Even though a precision matrix estimate can be close to the true precision matrix in terms of the Kullback-Leibler loss, the corresponding sparsity structure (or graph) can be completely different from the true one. We therefore assess the performance of the penalized likelihood estimators under consideration in terms of sensitivity (true positive rate) and specificity (true negative rate) in Section 3.10.

3.10 Simulation Studies

In this section, the performance of our proposed method for inverse covariance estimation is studied via simulation. We compare our PAC-based penalized likelihood method to adaptive banding of the Cholesky factor proposed in Levina et al. (2008). As a benchmark, we also include the standard graphical lasso of Friedman et al. (2008) and the graphical SCAD of Fan et al. (2009). In this section, we hope to highlight some of the shortcomings of the graphical lasso and the graphical SCAD for estimation of structured inverse covariance matrices.

The performance of the methods are assessed for six different dependence structures. We focus mainly on AR covariance structures of varying orders. For the AR(1) case, where correlation matrix R has entries $\rho_{ij} = \rho^{|i-j|}$ for some $\rho > 0$ and the corresponding inverse is tri-diagonal, the partial autocorrelation matrix Π is sparse with $\pi_{i,i+1} = \pi_{i+1,i} = \rho$ for the lag-1 terms and $\pi_{ij} = 0$ for $|i-j| > 1$. In this case, the Cholesky factor L has $\phi_{i+1,i} = \rho$ for the terms in the first subdiagonal and $\phi_{ij} = 0$ for $i - j > 1$. Therefore, since the nested lasso penalty is the same when applied to the Cholesky factor L and the partial autocorrelation matrix Π, the two methods are the same in the AR(1) case. We thus consider AR covariance structures of orders 2, 4, 8 and 15.

For Simulation 1, we consider an AR(2) model with $\pi_{i,i+1} = 0.8$ for the lag-1 terms and $\pi_{i,i+2} = 0.4$ for the lag-2 terms, and $\pi_{ij} = 0$ for $|i-j| > 2$. For Simulation 2, another AR(2) model is considered with $\pi_{i,i+1} = 0.7$ for the lag-1 terms and $\pi_{i,i+2} = -0.5$ for the lag-2 terms, and $\pi_{ij} = 0$ for $|i-j| > 2$. The corresponding inverse covariances matrices are banded with $k = 2$ non-zero bands. The theoretical autocorrelations for the AR(2)

process in Simulation 1 are positive and decaying with lag, while those for the AR(2) process in Simulation 2 are positive and negative, decaying with lag. For Simulations 1 and 2 with $p = 10$, both correlation matrices are not sparse, but the correlation matrix in Simulation 2 has a few small entries.

For Simulation 3, we consider a banded, non-stationary partial autocorrelation matrix with varying band lengths. It is a model taken from Gaskins at al. (2014), where for $p = 10$,

$$\Pi^{(5)}_{10\times 10} = \begin{pmatrix} 1 & 0.9 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9 & 1 & 0.8 & 0.4 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0.8 & 1 & 0.6 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.6 & 1 & 0.8 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0.2 & 0.8 & 1 & 0.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.7 & 1 & 0.8 & 0.4 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 1 & 0.6 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 & 1 & 0.8 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.8 & 1 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 1 \end{pmatrix}.$$

We also consider dimension $p = 30$ and take

$$\Pi^{(5)}_{30\times 30} = \begin{pmatrix} A & B \\ B^T & A \end{pmatrix},$$

where $A = \Pi^{(5)}_{10\times 10}$ and B is a 10×10 matrix with $b_{10,1} = 0.7$ and $b_{ij} = 0$ otherwise.

For Simulation 4, we consider an AR(8) covariance structure with $\pi_{i,i+1} = 0.5$, $\pi_{i,i+2} = -0.45$, $\pi_{i,i+3} = -0.4$, $\pi_{i,i+4} = 0.35$, $\pi_{i,i+5} = 0.3$, $\pi_{i,i+6} = -0.25$, $\pi_{i,i+7} = -0.2$ and $\pi_{i+8} = 0.15$, while for Simulation 5, we take $\pi_{i,i+1} = 0.8$ and $\pi_{ij} = 0.65^{|i-j|}$ for $1 < |i - j| \leq 15$.

In all simulations, we take $\sigma_{ii} = 1$ for $i = 1, \ldots, p$ for the the marginal variances. For each of the dependence structures, we simulate samples of mean-zero multivariate Gaussian vectors. We take $p = 10, 30, 50$ and 100 and consider samples of size $n = 50$ and 100. We consider both AIC and BIC to select the tuning parameters used in the methods under consideration. Even though $p > \sqrt{n}$ for $p = 30, 50$ and $n = 50, 100$, suggesting that EBIC would be more appropriate, we found in our simulations that EBIC had the tendency to select models that were too sparse. While Levina et al. (2008) recommended using 5-fold cross-validation to select the tuning parameters used in their adaptive banding method, we also found BIC to perform better in practice.

For the graphical lasso and graphical SCAD, we set $\hat{\theta}_{ij} = 0$ if $|\hat{\theta}_{ij}| < 10^{-3}$ since the threshold of convergence for all methods is taken to be 10^{-4}, which is also the default threshold of convergence for the graphical lasso algorithm in `R`. For both the proposed method and the adaptive banding method of Levina et al. (2008), we set entries in the final estimated partial autocorrelation matrix and the final estimated Cholesky factor that are less than 10^{-3} in absolute value to 0, and then reconstruct the final estimated precision matrix $\hat{\Theta}$.

To assess the performance of each of the methods, we evaluate specificity and sensitivity, defined as follows:

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\#\left\{\hat{\theta}_{ij} = 0, \theta_{ij} = 0\right\}}{\#\left\{\theta_{ij} = 0\right\}} \qquad \text{and}$$
$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\#\left\{\hat{\theta}_{ij} \neq 0, \theta_{ij} \neq 0\right\}}{\#\left\{\theta_{ij} \neq 0\right\}},$$

where TP, TN, FP and FN are the numbers of true positives, true negatives (or true zero entries), false positives, and false negatives. We also compare the performance of our inverse covariance estimators in terms of the following three loss functions, presented in Section 3.9:

$$\begin{aligned}
\mathcal{L}_1(\hat{\Theta}, \Theta) &= \text{tr}(\hat{\Theta}\Theta^{-1}) - \log\det(\hat{\Theta}\Theta^{-1}) - p, \\
\mathcal{L}_2(\hat{\Theta}, \Theta) &= \text{tr}(\Theta^{-1}\hat{\Theta} - I)^2, \\
\mathcal{L}_3(\hat{\Theta}, \Theta) &= \frac{1}{p}\sqrt{\sum_{i,j}(\hat{\theta}_{ij} - \theta_{ij})^2},
\end{aligned}$$

where $\mathcal{L}_1$ is the Kullback-Leibler loss, $\mathcal{L}_2$ is the quadratic loss, and $\mathcal{L}_3$ will be referred to as the Frobenius norm loss. The corresponding risk functions are

$$\mathcal{R}_i(\hat{\Theta}, \Theta) = \mathbb{E}_\Theta\left[\mathcal{L}_i(\hat{\Theta}, \Theta)\right], \quad i = 1, 2, 3.$$

An estimator $\hat{\Theta}_1$ is considered better than another estimator $\hat{\Theta}_2$ is its risk function is smaller. We approximate the risk function of each estimator by Monte Carlo simulation.

To produce the results presented in Tables 3.1-3.4, $N = 500$ simulation runs are used for each setup and we estimate the risk by averaging the loss over the $N = 500$ datasets. The sensitivity and specificity reported are also averaged over the $N = 500$ datasets. The standard errors for the means are reported in parentheses.

Discussion of Simulation Results:

For Simulation 1 (see Table 3.1), we find that our PAC-based nested lasso method using the BIC tuning parameter selector performs the best in terms of correct sparsity (specificity or true negative rate), Kullback-Leibler loss and the Frobenius norm error. The graphical lasso with BIC does better in terms of the quadratic loss. All methods perform comparably in terms of sensitivity (true positive rate). For Simulation 2 (see Table 3.2), again our PAC-based nested lasso method performs the best in terms of correct sparsity. It also has the smallest quadratic loss. The graphical lasso performs poorly in terms of specificity. It is not able to effectively identify the banded structure in the inverse covariance matrix, setting entries to zero in arbitrary locations in Θ. The graphical SCAD improves upon the graphical lasso in terms of specificity and Kullback-Leibler loss, but sets more true positives to zero. As expected, for the PAC-based nested lasso methods, the BIC-selector results in a higher true negative rate compared to the AIC-selector, but both tuning parameter selectors perform well in terms of the true positive rate.

For the non-stationary covariance structure in Simulation 3 (see Table 3.3), we see that the nested lasso penalty applied to the PACs again performs the best in terms of correct sparsity, Kullback-Leibler loss and Frobenius norm loss. The graphical lasso with BIC performs the best in terms of quadratic loss, but only sightly compared to our proposed method with the nested lasso penalty.

For Simulation 4 (see Table 3.3), applying the nested lasso penalty to the PACs yields the smallest Kullback-Leibler loss and Frobenius norm error among the methods under consideration. The graphical lasso with BIC again has the best quadratic loss. The nested lasso methods perform similarly in terms of specificity, but using the PAC parametrization results in a higher sensitivity, lower Kullback-Leibler loss and lower Frobenius norm error.

In Simulation 5 (see Table 3.4), we provide an example where the PAC-based nested lasso method does not perform very well. The graphical lasso performs the best for all sample sizes and dimensions considered in terms of sensitivity (true positive rate), Kullback-Leibler loss, quadratic loss and Frobenius norm error (or sum of squared errors). None of the methods perform well in terms of sensitivity (even with the larger sample size $n = 1000$). Their poor performance is likely due to the selection of the tuning parameter. While the BIC-selector was shown to be model selection consistent in Gao et al. (2012), in finite sample it may still exhibit a high false negative rate. The graphical lasso performs better compared to the other methods in terms of sensitivity, but it performs the worst in terms of specificity. The nested lasso methods are better able to identify the zeros in the precision matrix.

Method	Tuning Parameter Selector	Sensitivity	Specificity	Kullback-Leibler Loss	Quadratic Loss	Frobenius Norm
Simulation 1						
$n = 100, p = 10$						
S^{-1}		100 (0)	0 (0)	0.69 (0.01)	6.50 (0.13)	3.44 (0.06)
Nested lasso - MCD	AIC	100 (0)	71.89 (0.50)	0.36 (0.01)	4.03 (0.08)	1.27 (0.03)
Nested lasso - MCD	BIC	100 (0)	84.31 (0.24)	0.35 (0.01)	3.83 (0.08)	1.15 (0.02)
Nested lasso - PAC	AIC	100 (0)	92.71 (0.47)	0.31 (0.004)	3.54 (0.07)	0.98 (0.02)
Nested lasso - PAC	BIC	99.98 (0.01)	98.67 (0.11)	0.30 (0.004)	3.41 (0.07)	0.93 (0.02)
Lasso	AIC	99.84 (0.04)	43.90 (0.72)	0.41 (0.01)	3.86 (0.08)	1.25 (0.02)
Lasso	BIC	99.93 (0.02)	61.44 (0.42)	0.39 (0.004)	3.10 (0.06)	1.23 (0.01)
SCAD	AIC	97.90 (0.13)	90.11 (0.41)	0.38 (0.01)	4.16 (0.09)	1.18 (0.02)
SCAD	BIC	94.69 (0.17)	96.79 (0.18)	0.40 (0.01)	4.31 (0.09)	1.32 (0.02)
$n = 100, p = 30$						
S^{-1}		99.99 (0.01)	0.09 (0.01)	8.50 (0.04)	135.83 (1.70)	25.77 (0.25)
Nested lasso - MCD	AIC	100 (0)	93.44 (0.10)	1.15 (0.01)	19.92 (0.27)	1.39 (0.02)
Nested lasso - MCD	BIC	100 (0)	96.23 (0.04)	1.13 (0.01)	18.85 (0.25)	1.31 (0.01)
Nested lasso - PAC	AIC	100 (0)	98.10 (0.06)	0.98 (0.01)	17.25 (0.23)	1.07 (0.01)
Nested lasso - PAC	BIC	99.98 (0.01)	99.63 (0.02)	1.01 (0.01)	16.70 (0.22)	1.08 (0.01)
Lasso	AIC	99.88 (0.02)	70.52 (0.24)	1.74 (0.01)	23.27 (0.29)	1.53 (0.01)
Lasso	BIC	99.94 (0.01)	79.89 (0.11)	1.87 (0.01)	16.37 (0.19)	2.20 (0.02)
SCAD	AIC	98.39 (0.07)	89.05 (0.24)	1.82 (0.02)	32.69 (0.55)	1.60 (0.02)
SCAD	BIC	94.07 (0.12)	98.16 (0.05)	1.48 (0.02)	22.76 (0.32)	1.68 (0.02)
$n = 50, p = 50$						
Nested lasso - MCD	AIC	99.92 (0.03)	97.61 (0.03)	4.00 (0.03)	84.04 (1.00)	3.15 (0.04)
Nested lasso - MCD	BIC	99.88 (0.01)	99.03 (0.02)	4.06 (0.03)	81.03 (0.97)	2.99 (0.03)
Nested lasso - PAC	AIC	99.89 (0.01)	98.99 (0.03)	3.29 (0.02)	68.69 (0.83)	2.15 (0.02)
Nested lasso - PAC	BIC	99.73 (0.02)	99.65 (0.01)	3.37 (0.02)	66.80 (0.81)	2.16 (0.02)
Lasso	AIC	99.08 (0.03)	78.45 (0.14)	6.52 (0.03)	100.55 (1.36)	2.87 (0.02)
Lasso	BIC	99.42 (0.03)	85.13 (0.07)	6.85 (0.03)	59.48 (0.69)	4.24 (0.03)
SCAD	AIC	92.70 (0.12)	91.09 (0.14)	7.23 (0.09)	172.11 (3.24)	3.54 (0.04)
SCAD	BIC	86.87 (0.11)	96.47 (0.05)	6.25 (0.04)	107.67 (1.31)	3.94 (0.03)
$n = 100, p = 100$						
Nested lasso - MCD	AIC	100 (0)	99.54 (0.01)	4.08 (0.04)	71.45 (1.05)	1.36 (0.02)
Nested lasso - MCD	BIC	100 (0)	99.38 (0.004)	3.73 (0.02)	72.32 (0.49)	1.34 (0.01)
Nested lasso - PAC	AIC	100 (0)	99.54 (0.01)	3.35 (0.01)	65.96 (0.47)	1.09 (0.01)
Nested lasso - PAC	BIC	100 (0)	99.82 (0.01)	3.46 (0.01)	64.68 (0.45)	1.12 (0.01)
Lasso	AIC	99.92 (0.01)	82.72 (0.04)	8.88 (0.02)	156.97 (0.91)	1.97 (0.01)
Lasso	BIC	99.97 (0.004)	91.34 (0.02)	10.32 (0.02)	65.00 (0.36)	3.98 (0.01)
SCAD	AIC	96.95 (0.08)	95.71 (0.06)	5.28 (0.03)	107.86 (0.72)	1.45 (0.01)
SCAD	BIC	93.79 (0.05)	98.15 (0.01)	6.01 (0.03)	97.22 (0.67)	2.03 (0.01)

Table 3.1: Sensitivity, specificity, Kullback-Leibler loss, quadratic loss and Frobenius norm error, averaged over $N = 500$ replications of size $n = 50, 100$, for the inverse sample covariance matrix, the nested lasso method of Levina et al. (2008) based on the modified Cholesky decomposition (MCD), the PAC-based nested lasso, the graphical lasso of Friedman et al. (2008) and the graphical SCAD of Fan et al. (2009). The standard errors for the means over the 500 replications are reported in parentheses.

Method	Tuning Parameter Selector	Sensitivity	Specificity	Kullback-Leibler Loss	Quadratic Loss	Frobenius Norm
Simulation 2						
$n = 100, p = 10$						
S^{-1}		100 (0)	0 (0)	0.69 (0.01)	8.22 (0.15)	6.35 (0.19)
Nested lasso - MCD	AIC	100 (0)	72.81 (0.69)	0.37 (0.01)	3.95 (0.08)	1.97 (0.07)
Nested lasso - MCD	BIC	100 (0)	83.08 (0.24)	0.36 (0.01)	3.40 (0.07)	1.56 (0.04)
Nested lasso - PAC	AIC	100 (0)	91.78 (0.42)	0.34 (0.01)	3.07 (0.06)	1.78 (0.04)
Nested lasso - PAC	BIC	99.99 (0.01)	97.86 (0.14)	0.35 (0.01)	2.82 (0.05)	2.01 (0.04)
Lasso	AIC	98.98 (0.09)	20.64 (0.39)	0.55 (0.01)	4.84 (0.09)	3.33 (0.06)
Lasso	BIC	95.53 (0.28)	32.22 (0.48)	0.62 (0.01)	3.95 (0.07)	6.12 (0.18)
SCAD	AIC	93.34 (0.30)	71.07 (0.64)	0.53 (0.01)	4.74 (0.09)	3.16 (0.08)
SCAD	BIC	90.64 (0.34)	82.01 (0.47)	0.57 (0.01)	4.57 (0.08)	4.02 (0.10)
$n = 100, p = 30$						
S^{-1}		100 (0)	0 (0)	8.52 (0.04)	178.58 (1.61)	50.07 (0.61)
Nested lasso - MCD	AIC	100 (0)	94.20 (0.09)	1.14 (0.01)	13.33 (0.16)	1.93 (0.03)
Nested Lasso - MCD	BIC	100 (0)	96.43 (0.04)	1.13 (0.01)	11.82 (0.12)	1.69 (0.02)
Nested lasso - PAC	AIC	100 (0)	97.65 (0.06)	1.09 (0.01)	10.52 (0.11)	2.09 (0.02)
Nested Lasso - PAC	BIC	100 (0)	99.29 (0.03)	1.17 (0.01)	9.85 (0.10)	2.59 (0.03)
Lasso	AIC	94.89 (0.20)	36.87 (0.39)	3.75 (0.02)	34.12 (0.49)	7.71 (0.15)
Lasso	BIC	72.29 (0.14)	67.80 (0.17)	4.77 (0.02)	12.36 (0.07)	23.97 (0.12)
SCAD	AIC	89.93 (0.24)	75.28 (0.26)	3.13 (0.03)	33.53 (0.47)	4.61 (0.06)
SCAD	BIC	84.48 (0.28)	90.57 (0.13)	2.68 (0.02)	19.92 (0.17)	7.78 (0.11)
$n = 50, p = 50$						
Nested Lasso - MCD	AIC	88.19 (1.44)	96.05 (0.13)	4.63 (0.04)	62.66 (0.79)	14.15 (0.96)
Nested Lasso - MCD	BIC	93.79 (1.08)	96.79 (0.10)	4.36 (0.03)	58.20 (0.68)	9.65 (0.73)
Nested Lasso - PAC	AIC	99.96 (0.01)	98.58 (0.03)	3.89 (0.02)	37.17 (0.34)	4.22 (0.03)
Nested Lasso - PAC	BIC	99.88 (0.01)	99.45 (0.01)	4.17 (0.02)	34.95 (0.30)	5.10 (0.04)
Lasso	AIC	73.37 (0.15)	62.78 (0.24)	15.05 (0.06)	83.24 (1.30)	22.71 (0.18)
Lasso	BIC	67.59 (0.02)	87.90 (0.18)	19.39 (0.09)	32.39 (0.17)	42.70 (0.13)
SCAD	AIC	74.81 (0.15)	81.25 (0.15)	9.91 (0.05)	57.73 (0.73)	15.37 (0.09)
SCAD	BIC	68.59 (0.07)	87.83 (0.11)	10.56 (0.06)	44.46 (0.31)	21.69 (0.13)
$n = 100, p = 100$						
Nested Lasso - MCD	AIC	100 (0)	97.88 (0.05)	4.15 (0.03)	52.51 (0.47)	2.55 (0.04)
Nested Lasso - MCD	BIC	100 (0)	98.38 (0.04)	3.92 (0.02)	48.39 (0.40)	2.18 (0.03)
Nested Lasso - PAC	AIC	100 (0)	99.31 (0.04)	3.70 (0.03)	36.48 (0.44)	2.21 (0.03)
Nested Lasso - PAC	BIC	100 (0)	99.80 (0.01)	4.10 (0.03)	34.34 (0.37)	2.87 (0.05)
Lasso	AIC	73.07 (0.12)	69.31 (0.15)	22.97 (0.05)	106.27 (1.06)	21.60 (0.11)
Lasso	BIC	67.01 (0.001)	92.90 (0.07)	33.72 (0.09)	45.69 (0.09)	42.46 (0.07)
SCAD	AIC	74.78 (0.16)	88.05 (0.11)	13.01 (0.03)	55.97 (0.40)	15.28 (0.06)
SCAD	BIC	67.40 (0.02)	94.56 (0.03)	14.61 (0.04)	47.24 (0.19)	20.12 (0.04)

Table 3.2: Sensitivity, specificity, Kullback-Leibler loss, quadratic loss and Frobenius norm error, averaged over $N = 500$ replications of size $n = 50, 100$, for the inverse sample covariance matrix, the nested lasso method of Levina et al. (2008) based on the modified Cholesky decomposition (MCD), the PAC-based nested lasso, the graphical lasso of Friedman et al. (2008) and the graphical SCAD of Fan et al. (2009). The standard errors for the means over the 500 replications are reported in parentheses.

Method	Tuning Parameter Selector	Sensitivity	Specificity	Kullback-Leibler Loss	Quadratic Loss	Frobenius Norm
Simulation 3						
$n = 100, p = 10$						
S^{-1}		100 (0)	0 (0)	0.68 (0.01)	6.04 (0.13)	3.07 (0.08)
Nested lasso - MCD	AIC	96.17 (0.12)	85.69 (0.28)	0.37 (0.01)	3.47 (0.08)	1.04 (0.03)
Nested lasso - MCD	BIC	96.04 (0.12)	86.33 (0.26)	0.37 (0.01)	3.46 (0.08)	1.04 (0.03)
Nested lasso - PAC	AIC	95.90 (0.16)	93.39 (0.43)	0.31 (0.004)	2.90 (0.07)	0.86 (0.02)
Nested lasso - PAC	BIC	91.94 (0.17)	98.70 (0.10)	0.32 (0.004)	2.85 (0.07)	0.83 (0.02)
Lasso	AIC	95.59 (0.15)	45.06 (0.73)	0.41 (0.01)	3.31 (0.08)	1.18 (0.02)
Lasso	BIC	95.56 (0.15)	62.34 (0.47)	0.41 (0.01)	2.57 (0.05)	1.56 (0.03)
SCAD	AIC	87.19 (0.19)	87.42 (0.43)	0.38 (0.01)	3.53 (0.08)	0.96 (0.02)
SCAD	BIC	83.59 (0.20)	94.24 (0.23)	0.39 (0.01)	3.54 (0.08)	1.02 (0.02)
$n = 100, p = 30$						
S^{-1}		100 (0)	0 (0)	8.59 (0.05)	124.30 (1.32)	21.70 (0.24)
Nested lasso - MCD	AIC	97.66 (0.07)	93.60 (0.09)	1.13 (0.01)	12.67 (0.16)	1.08 (0.02)
Nested lasso - MCD	BIC	95.21 (0.08)	96.51 (0.04)	1.17 (0.01)	12.65 (0.16)	1.02 (0.02)
Nested lasso - PAC	AIC	96.29 (0.10)	98.01 (0.06)	0.96 (0.01)	10.53 (0.14)	0.85 (0.01)
Nested lasso - PAC	BIC	92.36 (0.11)	99.40 (0.02)	1.03 (0.01)	10.43 (0.13)	0.84 (0.01)
Lasso	AIC	96.09 (0.08)	68.34 (0.26)	1.90 (0.01)	16.28 (0.20)	1.97 (0.02)
Lasso	BIC	96.80 (0.08)	81.45 (0.12)	2.15 (0.01)	10.35 (0.09)	3.57 (0.03)
SCAD	AIC	88.47 (0.12)	87.17 (0.24)	1.73 (0.02)	20.58 (0.35)	1.10 (0.01)
SCAD	BIC	83.90 (0.12)	96.32 (0.06)	1.37 (0.01)	13.75 (0.17)	1.10 (0.01)
Simulation 4						
$n = 100, p = 30$						
S^{-1}		100 (0)	0 (0)	8.51 (0.04)	182.10 (1.87)	59.51 (0.88)
Nested lasso - MCD	AIC	82.57 (0.31)	99.57 (0.04)	5.69 (0.04)	33.33 (0.31)	31.38 (0.34)
Nested lasso - MCD	BIC	81.48 (0.32)	99.63 (0.04)	5.74 (0.04)	33.19 (0.31)	31.81 (0.34)
Nested lasso - PAC	AIC	97.98 (0.08)	93.87 (0.17)	2.86 (0.02)	37.65 (0.37)	6.04 (0.06)
Nested lasso - PAC	BIC	90.98 (0.16)	99.47 (0.03)	3.29 (0.02)	33.53 (0.32)	11.55 (0.15)
Lasso	AIC	94.94 (0.11)	14.93 (0.25)	5.21 (0.03)	61.50 (0.72)	12.54 (0.14)
Lasso	BIC	72.69 (0.42)	56.88 (0.57)	6.81 (0.06)	21.93 (0.32)	49.16 (0.50)
SCAD	AIC	78.85 (0.14)	68.65 (0.30)	4.79 (0.03)	60.21 (0.61)	9.56 (0.08)
SCAD	BIC	67.79 (0.15)	91.36 (0.15)	4.22 (0.02)	37.49 (0.34)	17.46 (0.16)
$n = 50, p = 50$						
Nested lasso - MCD	AIC	50.78 (0.48)	100 (0)	22.79 (0.18)	127.79 (1.30)	63.76 (0.46)
Nested lasso - MCD	BIC	50.90 (0.48)	100 (0)	22.60 (0.17)	126.21 (1.09)	63.37 (0.46)
Nested lasso - PAC	AIC	91.47 (1.35)	98.05 (0.46)	10.53 (0.25)	75.57 (18.78)	8.20 (2.06)
Nested lasso - PAC	BIC	75.23 (1.83)	99.97 (0.02)	12.81 (0.53)	64.47 (16.43)	16.25 (4.21)
Lasso	AIC	78.04 (0.41)	44.76 (0.67)	27.06 (0.36)	350.33 (12.64)	47.22 (0.54)
Lasso	BIC	51.77 (0.08)	81.31 (0.10)	21.92 (0.05)	56.18 (0.44)	76.26 (0.08)
SCAD	AIC	52.56 (0.23)	75.64 (0.25)	20.43 (0.09)	161.15 (2.16)	54.83 (0.17)
SCAD	BIC	45.81 (0.07)	84.37 (0.08)	18.70 (0.05)	103.62 (0.87)	62.01 (0.10)

Table 3.3: Sensitivity, specificity, Kullback-Leibler loss, quadratic loss and Frobenius norm error, averaged over $N = 500$ replications of size $n = 50, 100$, for the inverse sample covariance matrix, the nested lasso method of Levina et al. (2008) based on the modified Cholesky decomposition (MCD), the PAC-based nested lasso, the graphical lasso of Friedman et al. (2008) and the graphical SCAD of Fan et al. (2009). The standard errors for the means over the 500 replications are reported in parentheses.

Method	Tuning Parameter Selector	Sensitivity	Specificity	Kullback-Leibler Loss	Quadratic Loss	Frobenius Norm
Simulation 5						
$n = 100, p = 30$						
S^{-1}		99.92 (0.01)	0.09 (0.01)	8.57 (0.05)	138.08 (2.28)	24.10 (0.22)
Nested lasso - MCD	AIC	54.44 (0.45)	99.44 (0.27)	1.74 (0.04)	36.64 (0.93)	2.35 (0.06)
Nested lasso - MCD	BIC	44.65 (0.32)	100 (0)	1.89 (0.04)	35.72 (0.88)	2.07 (0.04)
Nested lasso - PAC	AIC	49.11 (0.26)	100 (0)	2.03 (0.01)	42.65 (0.61)	2.72 (0.03)
Nested lasso - PAC	BIC	36.92 (0.12)	100 (0)	2.34 (0.01)	41.44 (0.57)	2.39 (0.02)
Lasso	AIC	58.94 (0.14)	71.16 (0.37)	1.78 (0.01)	34.02 (0.49)	1.54 (0.01)
Lasso	BIC	56.53 (0.09)	83.60 (0.21)	1.69 (0.01)	28.53 (0.39)	1.40 (0.01)
SCAD	AIC	35.25 (0.10)	94.93 (0.12)	2.53 (0.01)	48.74 (0.68)	3.16 (0.02)
SCAD	BIC	31.43 (0.08)	97.14 (0.08)	2.97 (0.02)	51.84 (0.72)	3.83 (0.03)
$n = 1000, p = 30$						
S^{-1}		99.73 (0.01)	0.53 (0.03)	0.49 (0.002)	4.42 (0.04)	0.68 (0.003)
Nested lasso - MCD	AIC	69.43 (0.47)	99.64 (0.17)	0.25 (0.002)	3.45 (0.08)	0.31 (0.004)
Nested Lasso - MCD	BIC	56.97 (0.09)	100 (0)	0.29 (0.002)	3.67 (0.04)	0.28 (0.002)
Nested lasso - PAC	AIC	66.55 (0.57)	99.99 (0.01)	0.25 (0.002)	3.40 (0.08)	0.30 (0.004)
Nested lasso - PAC	BIC	52.26 (0.09)	100 (0)	0.33 (0.002)	3.76 (0.04)	0.31 (0.002)
Lasso	AIC	71.91 (0.10)	70.20 (0.34)	0.25 (0.001)	3.29 (0.03)	0.27 (0.001)
Lasso	BIC	70.02 (0.07)	85.33 (0.34)	0.27 (0.002)	3.06 (0.03)	0.30 (0.002)
SCAD	AIC	54.72 (0.10)	95.77 (0.15)	0.29 (0.001)	3.67 (0.04)	0.35 (0.002)
SCAD	BIC	47.22 (0.09)	97.86 (0.06)	0.40 (0.003)	4.15 (0.04)	0.51 (0.004)
$n = 50, p = 50$						
Nested lasso - MCD	AIC	23.60 (0.35)	100 (0)	12.53 (0.25)	213.16 (3.64)	6.69 (0.10)
Nested lasso - MCD	BIC	24.27 (0.43)	100 (0)	12.67 (0.25)	218.24 (3.77)	6.55 (0.09)
Nested lasso - PAC	AIC	34.34 (0.05)	100 (0)	6.15 (0.03)	213.80 (3.32)	4.61 (0.04)
Nested lasso - PAC	BIC	29.05 (0.11)	100 (0)	7.01 (0.04)	202.33 (3.05)	4.21 (0.03)
Lasso	AIC	48.95 (0.07)	82.54 (0.16)	5.61 (0.03)	167.65 (2.65)	2.20 (0.01)
Lasso	BIC	48.71 (0.06)	88.74 (0.10)	5.36 (0.02)	131.74 (1.88)	2.14 (0.01)
SCAD	AIC	29.55 (0.09)	93.09 (0.11)	9.83 (0.07)	344.91 (5.76)	7.14 (0.07)
SCAD	BIC	26.07 (0.06)	96.07 (0.06)	9.61 (0.05)	293.71 (4.28)	7.03 (0.05)

Table 3.4: Sensitivity, specificity, Kullback-Leibler loss, quadratic loss and Frobenius norm error, averaged over $N = 500$ replications of size $n = 50, 100$, for the inverse sample covariance matrix, the nested lasso method of Levina et al. (2008) based on the modified Cholesky decomposition (MCD), the PAC-based nested lasso, the graphical lasso of Friedman et al. (2008) and the graphical SCAD of Fan et al. (2009). The standard errors for the means over the 500 replications are reported in parentheses.

To gain further insight into the selection of the tuning parameter, we plot sensitivity and specificity against BIC, averaged over 100 samples, for the graphical lasso precision matrix estimates (see Figure 3.2). The samples were generated from a multivariate normal distribution with mean zero and covariance structure as specified in Simulation 5 with $n = 1000$ and $p = 30$. It can be seen that the tuning parameter yielding the largest BIC has the highest specificity, but lowest sensitivity, while the tuning parameter yielding the smallest BIC has a high sensitivity, but may have a low specificity. In Figure 3.3, we plot the Kullback-Leibler loss, averaged over 100 samples, against the mean BIC values. It can be seen that the minimum BIC corresponds to the minimum Kullback-Leibler loss.

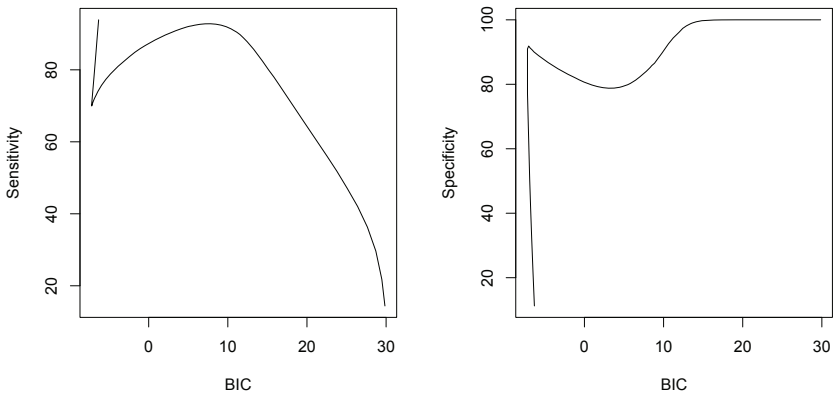

Figure 3.2: Plot of mean sensitivity (left) and mean specificity (right) against mean BIC using the graphical lasso, averaged over 100 samples generated from a multivariate normal distribution with mean zero and covariance structure as specified in Simulation 5 with $n = 1000$ and $p = 30$.

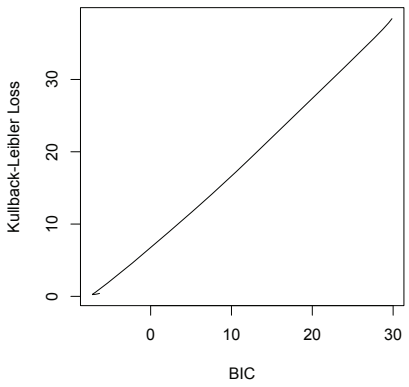

Figure 3.3: Plot of Kullback-Leibler loss against mean BIC using the graphical lasso, averaged over 100 samples generated from a multivariate normal distribution with mean zero and covariance structure as specified in Simulation 5 with $n = 1000$ and $p = 30$.

We also provide similar plots for the graphical SCAD precision matrix estimates (see Figures 3.4 and 3.5). It can be seen that smaller BIC values tend to correspond to high specificities but low sensitivities. Therefore, if the goal of the statistical analysis is to identify the zeros of the precision matrix, then the graphical SCAD with the BIC-selector would perform satisfactorily. If, on the other hand, the goal of the statistical analysis is to identify the non-zeros of the precision matrix, then using BIC would not yield satisfactory results.

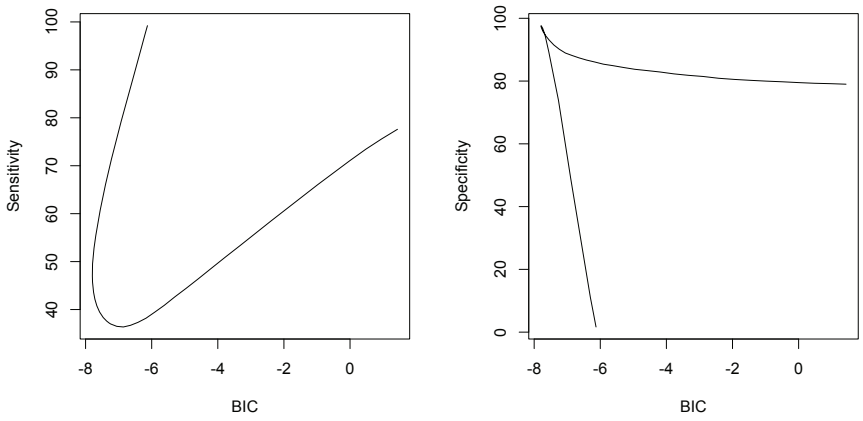

Figure 3.4: Plot of mean sensitivity (left) and mean specificity (right) against mean BIC using the graphical SCAD, averaged over 100 samples generated from a multivariate normal distribution with mean zero and covariance structure as specified in Simulation 5 with $n = 1000$ and $p = 30$.

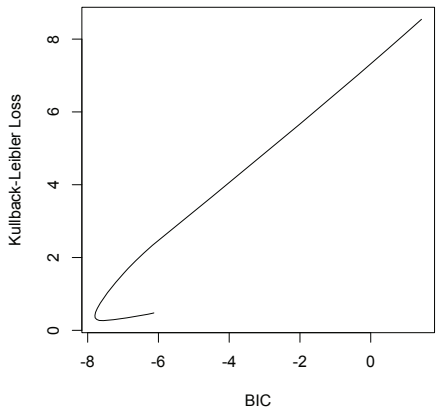

Figure 3.5: Plot of mean Kullback-Leibler loss against mean BIC using the graphical SCAD, averaged over 100 samples generated from a multivariate normal distribution with mean zero and covariance structure as specified in Simulation 5 with $n = 1000$ and $p = 30$.

3.11 Real Data Analysis

As an illustration of our method for estimating a $p \times p$ precision matrix in the case where $n < p$, we consider the change in Canadian monthly unemployment rate from January 2005 to December 2012. The data set, obtained from Statistics Canada[1], consists of monthly seasonally adjusted unemployment rates for 55 employment insurance (E.I.) economic regions across Canada.

The estimated precision matrix can be used for forecasting the change in unemployment rate. Let $\mathbf{x}_i$ denote the data for E.I. region i and write $\mathbf{x}_i = (x_{i1}, \ldots, x_{i,95})^T$. We form the partition $\mathbf{x}_i = (\mathbf{x}_i^{(1)T}, \mathbf{x}_i^{(2)T})^T$, where $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ are the changes in unemployment rates in the first 48 months and the last 47 months, respectively, of E.I. region i.

The corresponding partition of the mean and covariance matrix is

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}.$$

Assuming multivariate normality, the best mean-squared error forecast of $\mathbf{x}_i^{(2)}$ using $\mathbf{x}_i^{(1)}$ is

$$\mathbb{E}(\mathbf{x}_i^{(2)} | \mathbf{x}_i^{(1)}) = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}_i^{(1)} - \mu_1), \tag{3.11}$$

which is also the best linear predictor for non-Gaussian data.

To compare the forecast performance using different precision matrix estimates, we use 50 splits of the 55 regions into training and test datasets. For each split, 40 regions will form the training set that is used to estimate the covariance structure. The estimates are then applied for forecasting using formula (3.11) for the 15 regions in the test set. We used the changes in unemployment rates in the first 48 months to forecast the changes in unemployment rates in the last 47 months. For each $t = 49, \ldots, 95$, we can define the

[1]Statistics Canada. (2015). Monthly Seasonal Adjusted Unemployment Rates by EI Economic Region. http://open.canada.ca/data/en/dataset/aad2bcd4-9f45-4013-b2a6-8367106dc0b2.

average absolute forecast error by

$$\mathrm{AE}_t = \frac{1}{50}\sum_{k=1}^{50}\left\{\frac{1}{15}\sum_{i\in M_k}|\hat{x}_{it} - x_{it}|\right\}, \tag{3.12}$$

where x_{it} and $\hat{x}_{it}$ are the observed and forecast values, respectively, and M_k is the set of indices of observations in the k^{th} test data set. The results are reported in Table 3.5. It can be seen that the nested lasso method based on the partial autocorrelation parametrization yields the smallest prediction error. As expected, the EBIC selector results in sparser models compared to AIC and BIC.

Method	Tuning Parameter Selector		
	AIC	BIC	EBIC
Nested lasso - MCD	0.342	0.341	0.340
Nested lasso - PAC	0.294	0.294	0.295
Lasso	0.302	0.340	0.340

Table 3.5: Average absolute forecast error for the graphical lasso, nested lasso (MCD), and nested lasso (PAC) methods, applied to the changes in monthly unemployment rates in Canada, corresponding to 50 splits of the data into training sets of size 40 and test sets of size 15.

Method	Tuning Parameter Selector		
	AIC	BIC	EBIC
Nested lasso - MCD	518.1 (7.54)	376.9 (12.73)	99.1 (0.71)
Nested lasso - PAC	362.3 (3.89)	299.5 (28.40)	170.0 (0.72)
Lasso	764.8 (31.39)	95.7 (0.22)	95.1 (0.045)

Table 3.6: Number of non-zero elements in the upper triangular part (including the diagonal) of the estimated precision matrix, averaged over 50 training sets for the changes in Canadian monthly unemployment rates, for the graphical lasso, nested lasso (MCD), and nested lasso (PAC) methods

3.12 Discussion

In this chapter, we proposed a penalized likelihood method that makes use of the partial autocorrelation (PAC) parametrization for estimating large precision matrices in the case where variables have a natural ordering. The PAC parametrization allows for shrinkage in an unconstrained setting and is a more natural parametrization to use in the ordered data context. Expecting PACs at large lags to be negligible in longitudinal/ordered data contexts, we imposed a banded structure in the matrix of partial autocorrelations through the use of a nested lasso penalty, which was introduced by Levina et al. (2008) as a penalty

on the Cholesky factor of the modified Cholesky decomposition of the inverse covariance matrix. While the Cholesky decomposition provides a convenient representation of the inverse covariance matrix in which parameters are also unconstrained, sparsity under the PAC parametrization is more interpretable. We identified cases where the PAC-based nested lasso method is advantageous over its Cholesky counterpart. Under the banded assumption, the PAC-based method was shown to perform well in simulation and with a real data example. The procedure for optimizing the PAC-based penalized log-likelihood with the nested lasso penalty, however, is computationally intensive. In Chapter 4, we introduce another application of the PAC-based penalized likelihood methodology in which the resulting computational problem is more feasible. The asymptotic properties of the PAC-based method remain for future work.

Chapter 4

Autoregressive Order Estimation via Penalization of the Partial Autocorrelations

In Chapter 3, we introduced our proposed PAC-based penalized likelihood method for inverse covariance estimation for ordered data. In this chapter, we consider another application of our PAC-based penalized likelihood methodology, where we consider the problem of estimating the order of a stationary Gaussian autoregressive (AR) process. To this end, the lasso methodology has been used (Wang et al., 2007b; Nardi and Rinaldo, 2011), where an L_1 penalty is applied to the AR coefficients. However, such a procedure ignores the temporal dependence information embedded in the AR time series. Rather than imposing shrinkage on the AR coefficients, we instead introduce shrinkage via the PACs, which vary on the same scale, free of constraints, and better reflect the characteristics of underlying AR processes, especially the AR order. For AR processes, it is well known that the partial autocorrelation function (PACF) identifies the AR order as the lag beyond which the PACF vanishes. Therefore, for n observations generated from an AR(p) model, the corresponding partial autocorrelation matrix is p-banded. The PAC-based penalized likelihood method that applies the nested lasso penalty to the PACs, introduced in Chapter 3, can thus be used to estimate the bandwidth p of the partial autocorrelation matrix. The nested lasso penalty applied to the PACs is designed for the class of stationary AR processes of order p, where the PACs at all intermediate lags are non-zero. To handle the general class of stationary AR processes, shrinkage is also

applied to the PACs through the use of the lasso penalty. To solve the resulting maximization problems, we adopt a cyclic coordinate descent procedure that was found to perform well in practice. Empirically, we show that our proposed PAC-based penalized likelihood methods perform better those based on penalization of the AR coefficients in terms of AR order estimation. The performance of our proposed AR order estimators are also compared to a number of other AR order estimators, both in simulation and on a real data example.

4.1 Introduction

Autoregressive (AR) models have been applied in various fields, including finance, engineering, and the natural sciences, such as geophysics and hydrology. In particular, they have been used in applications such as spectral estimation, speech processing, and radar and sonar signal processing (Khorshidi et al., 2011, and references therein), as well as for the modelling of river flows, sunspot numbers and various other geophysical phenomena (Hipel and McLeod, 1994).

When the true order p_0 of the autoregressive process is given, there are many ways of estimating the AR coefficients, such as conditional least-squares, maximum likelihood estimation, where the likelihood function considered is that corresponding to the unconditional joint distribution of the observed values, and the Burg (1967) and Yule-Walker methods. In practice, however, the assumption that the AR order is known in advance is often unrealistic and so p_0 must be estimated from the data. It is well known that making a model unnecessarily complex can degrade the efficiency of the resulting parameter estimates, while oversimplifying a model can yield less accurate predictions. Thus, order estimation is a task of critical importance.

The two most commonly used approaches for order selection are the Akaike information criterion (AIC; Akaike, 1973) and the Bayesian information criterion (BIC; Schwarz, 1978). Various authors have applied AIC and BIC for selecting the order of an AR model, including Brockwell and Davis (2002), Shumway and Stoffer (2006), and Tsay

(2010). However, for an upper bound q on the AR order p_0, AIC and BIC require the fitting of q candidate AR models.

Exploiting the success of the lasso (Tibshirani, 1996) for performing simultaneous variable selection and parameter estimation in the context of linear and generalized linear modelling, Wang et al. (2007b) and Nardi and Rinaldo (2011) applied shrinkage to the AR coefficients through the L_1 penalty, allowing for subset AR models. For autoregressive modelling, the lasso features are particularly attractive as the AR order and the corresponding AR coefficients can be estimated simultaneously.

In this chapter, we propose an alternative penalized likelihood method for estimating the AR order. Rather than imposing shrinkage on the AR coefficients, we instead introduce shrinkage via the partial autocorrelations, which were shown to vary independently over the interval (-1,1) and to be in a one-to-one, continuously differentiable correspondence with the AR coefficients (Barndorff-Nielsen and Schou, 1973). The advantage of using the PAC parametrization is that the partial autocorrelations vary on the same scale, free of constraints. Furthermore, while the AR coefficients provide a convenient representation of the autoregressive process, the structural dependence of the process is better captured through the PACF. We thus consider regularized maximum likelihood estimation of the partial autocorrelations as a means of selecting the order of an autoregressive process.

The rest of this chapter is organized as follows. In Section 4.2, we begin with a brief literature review of existing methods for estimating the order of an AR model. We then introduce in Section 4.3 our proposed method for AR order estimation, which involves maximizing the penalized Gaussian log-likelihood using the nested lasso penalty of Levina et al. (2008) on the partial autocorrelations. The nested lasso penalty applied to the PACs is designed for the class of stationary AR processes of order p, where the the PACs at all lags $j \leq p$ are non-zero. To handle the general class of stationary AR processes, we also consider applying the lasso (Tibshirani, 1996) penalty to the PACs. We assess the finite sample performance of our proposed method in simulation studies in Section 4.5, comparing it to various information criteria as well as the lasso and modified

lasso methods of Wang et al. (2007b). We then illustrate the methodology developed in this chapter through an application to a time series of wave heights in Section 4.7.

4.2 Autoregressive Order Estimation

We begin by introducing the autoregressive (AR) model of order p. Suppose that $X_1, \ldots, X_n$ are n observations from the AR(p) process

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + e_t, \qquad t = p+1, \ldots, n,$$

where $\phi = (\phi_1, \ldots, \phi_p)$ is the autoregressive coefficient and e_t are independent Gaussian random variables with mean 0 and variance σ_e^2. We assume that $\{X_t\}$ is stationary and causal. This is equivalent to the condition that all roots of the characteristic polynomial $\Phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$, where B is the backshift operator, lie outside the unit circle. The process $\{X_t\}$ is said to be an AR(p) process with mean μ if $\{X_t - \mu\}$ is an AR(p) process. In this section, we assume without loss of generality that $\mu = 0$.

Recall that for a stationary process $\{X_t\}$, the autocovariance between X_t and X_{t+j} is

$$\gamma(j) = \mathrm{Cov}(X_t, X_{t+j}) = \mathbb{E}\left[(X_t - \mu)(X_{t+j} - \mu)\right],$$

and the autocorrelation between X_t and X_{t+j} is $\rho(j) = \gamma(j)/\gamma(0)$, where

$$\gamma(0) = \mathrm{Var}(X_t) = \frac{\sigma_e^2}{1 - \phi_1\gamma(1) - \ldots - \phi_p\gamma(p)}.$$

The partial autocorrelation coefficient (PAC) at lag j, π_j, is the autocorrelation between X_t and X_{t+j} after removing their dependency on the intervening variables $X_{t+1}, \ldots, X_{t+j-1}$, that is,

$$\pi_j = \mathrm{Cor}(X_t, X_{t+j} | X_{t+1}, \ldots, X_{t+j-1}).$$

The defining feature of the partial autocorrelation function of an AR process of order

p is that it cuts off after lag p; in other words, $\pi_j = 0$ for all $j > p$. Therefore, the order of the AR model is identified as the maximal lag j such that $\pi_j \neq 0$.

There are two standard approaches used for determining the order. For $\{X_t\}$ following an autoregressive process, the first approach involves looking at the sample partial autocorrelation function (PACF). The sample PACF may be obtained by successively fitting the following autoregressive models

$$\begin{aligned} X_t &= \phi_{11} X_{t-1} + e_{1t}, \\ X_t &= \phi_{12} X_{t-1} + \phi_{22} X_{t-2} + e_{2t}, \\ X_t &= \phi_{13} X_{t-1} + \phi_{23} X_{t-2} + \phi_{33} X_{t-3} + e_{3t}, \\ &\vdots \end{aligned}$$

where ϕ_{ij} and e_{jt} are the coefficient of X_{t-i} and the error term of an AR(j) model, respectively. The estimate $\hat{\phi}_{jj}$ of the last coefficient at each stage is the lag-j sample PACF of X_t.

For a stationary Gaussian AR(p) model, Quenouville (1949) showed that the sample partial autocorrelations of lags $p + 1$ and higher are approximately independently and normally distributed with zero mean and variance $1/n$ for lags $j > p$. Thus, the standard error (SE) of the sample PACF $\hat{\phi}_{jj}$ is

$$\mathrm{SE}(\hat{\phi}_{jj}) \simeq \frac{1}{\sqrt{n}}, \quad j > p,$$

and so if we observe a sample PACF satisfying $|\hat{\phi}_{jj}| > 1.96/\sqrt{n}$ for $0 \leq j \leq p$ and $|\hat{\phi}_{jj}| < 1.96/\sqrt{n}$ for $j > p$, this suggests an AR model of order p for the data.

The second approach uses likelihood-based information criteria, such as AIC and BIC. For a Gaussian AR(j) model, AIC and BIC reduce to

$$\mathrm{AIC}(j) = \log\left(\hat{\sigma}^2_{e,j}\right) + \frac{2j}{n} \quad \text{and} \quad \mathrm{BIC}(j) = \log\left(\hat{\sigma}^2_{e,j}\right) + \frac{j \log(n)}{n},$$

respectively, where $\hat{\sigma}^2_{e,j}$ is the maximum likelihood estimate of σ^2_e. When considering the

AIC procedure, Akaike (1969) and Shibata (1976) had used the Yule-Walker estimate of σ_e^2 rather than the MLE. While maximum likelihood, Burg (1967) and Yule-Walker estimation are asymptotically equivalent for finite AR models, it has been established that for small to moderate sample sizes, the ML and Burg estimates tend to have less bias than the Yule-Walker estimates. However, Chen et al. (1993) showed, with an example, that while the Yule-Walker method may not provide the best estimates of the parameters when the order of the model is known, it can be more reliable for fitting overparametrized models than ML and Burg estimation, and would therefore be better suited for order determination. We investigate this further in our simulation studies in Section 4.5. Comparing AIC and BIC, it is well known that BIC tends to select a lower order AR model than AIC when the sample size is moderate or large. Shibata (1976) showed that AIC is not consistent, but rather overestimates the true AR order, asymptotically, with a non-zero probability. Hannan and Quinn (1979) focused on finding a criterion of the form $\log(\hat{\sigma}_{e,j}^2) + jC_n$ that would be strongly consistent for the true order. They invoked the law of the iterated logarithm to show that any $C_n > 2\log(\log n)/n$ leads to a strongly consistent order estimate. Therefore, Hannan and Quinn (1979) considered estimating the AR order by minimizing

$$\text{HQC}(j) = \log(\hat{\sigma}_{e,j}^2) + \frac{2jc\log(\log n)}{n}, \quad c > 1. \tag{4.1}$$

They showed that if $\hat{p}$ is chosen to minimize HQC(j) over $j \leq q$, then for $p_0 \leq q$, $\hat{p}$ is strongly consistent for p_0. Since $2\log(\log n)/n < (\log n)/n$, BIC is strongly consistent. This also implies that the method of Hannan and Quinn (1979) will underestimate the order in large samples less than BIC. In practice, the performance of these criteria for moderately large n is in agreement with the asymptotic theory, but not for small n. Therefore, Hurvich and Tsai (1989) proposed that the AR order be selected by minimizing the bias-corrected AIC criterion

$$\text{AIC}_c(j) = \text{AIC}(j) + \frac{2(j+1)(j+2)}{n-j-2}. \tag{4.2}$$

It can be seen that for j fixed, $\text{AIC}_c(j) \to \text{AIC}(j)$ as $n \to \infty$ and so AIC and AIC_c are asymptotically equivalent.

In the time series context, AIC and BIC have been used for estimating the AR order, but they are more commonly applied for selecting variables in the linear and generalized linear modelling context. It has been well documented that the statisical performance of AIC and BIC in this context can be unstable (Breiman, 1996) as parameter estimation and model selection are two different processes. Therefore, penalized likelihood methods, such as the lasso (Tibshirani, 1996) and adaptive lasso (Zou, 2006), which estimate parameters while simultaneously selecting important variables, emerged to overcome the deficiencies of traditional methods. These penalization methods have also been considered in the literature for fitting autoregressive processes. Wang et al. (2007b) studied linear regression with autoregressive errors. They used the lasso procedure to simultaneously estimate the regression coefficients and the autoregressive coefficients under the assumption that the autoregressive order is fixed. Nardi and Rinaldo (2011), on the other hand, used the lasso procedure for autoregressive process modelling in the case where the number of parameters, or equivalently, the maximal possible lag, grows with the sample size. These methods allow for subset AR models, where the order of the AR model, which is the maximal lag p of the AR coefficients, does not correspond to the number of non-zero AR coefficients. To remedy the well-known bias issue of the lasso, Wang et al. (2007b) also considered a modified lasso penalty that allows for different tuning parameters for different coefficients. Rather than selecting the p tuning parameters λ_j, the authors fix the tuning parameter λ and weight each coefficient by $1/|\tilde{\phi}_j|$, where $\tilde{\phi}_j$ is the unpenalized least squares estimator. This allows a larger amount of shrinkage to be applied to insignificant coefficients, while a smaller amount of shrinkage can be applied to the significant coefficients. They show that, for fixed p, the modified lasso possesses the oracle property under certain conditions on the tuning parameters. Nardi and Rinaldo (2011) established model selection consistency, estimation consistency, and prediction consistency for the lasso estimator under the conditions that the maximal lag p grows with n as $p = o(n)$, $p = o(n^{1/2})$ and $p = o(n^{1/5})$, respectively. For selection of the

tuning parameter, Wang et al. (2007b) had considered both cross-validation (CV) and BIC, and found that BIC performed better in practice, while Nardi and Rinaldo (2011) had simply employed CV. The simulations of Wang et al. (2007b) demonstrate that the lasso with CV as the tuning parameter selector performs the worst, while the modified lasso with BIC as the tuning parameter selector offers the best performance.

4.3 The Proposed Method for Autoregressive Order Estimation

We adopt a penalized likelihood approach for estimating the order of an AR model. Rather than applying shrinkage to the AR coefficients in (4.1), we instead introduce shrinkage through the partial autocorrelations, defined as the conditional correlation between X_t and X_s, given the intervening variables $X_{t+1}, \ldots, X_{s-1}$. It is well known that the distinguishing feature of the partial autocorrelation function of an AR(p) process is that it cuts off after lag p, and so for n observations, generated from a stationary AR(p) model, their corresponding partial autocorrelation matrix is banded with a non-zero band at lag p. Therefore, we propose to estimate the AR order by maximizing the Gaussian log-likelihood subject to a nested lasso penalty on the partial autocorrelations, which would impose a banded structure on the partial autocorrelation matrix. The estimated order of the AR model would then correspond to the bandwidth of the penalized maximum likelihood estimate of the partial autocorrelation matrix. In what follows, we outline the proposed method.

Suppose that $\{X_t\}$ is a zero-mean Gaussian stationary process with autocovariance function $\gamma(|s-t|) = \mathbb{E}(X_s X_t)$. Let $\mathbf{X}_n = (X_1, \ldots, X_n)^T$ and let $\Sigma_n = \mathbb{E}(\mathbf{X}_n \mathbf{X}_n^T)$ denote the covariance matrix. Using the variance-correlation decomposition, we write $\Sigma_n = V_n R_n V_n$, where V_n is a diagonal matrix with the marginal standard deviations of $\mathbf{X}_n$ and R_n is the correlation matrix.

To remove the positive-definiteness constraint on the correlation matrix, we reparametrize R_n in terms of the symmetric matrix Π_n of partial autocorrelations, which is not required

to be positive-definite. Since $\{X_t\}$ is stationary, Π_n is a stationary (Toeplitz) matrix. We let π_j denote the constant value along the j^{th} diagonal of Π_n. Unlike the marginal correlations, the partial autocorrelations vary freely in the interval (-1,1); each π_j can take any value in (-1,1), regardless of the choice of the remaining partial autocorrelations. If $\{X_t\}$ is a stationary Gaussian AR(q) process, then $\pi_{q+1} = \ldots = \pi_{n-1} = 0$.

Rather than working with the $n \times n$ matrix of partial autocorrelations Π_n, which for a stationary AR(q) model, is q-banded with a constant π_j along the j^{th} diagonal, we instead work with the vectorized form of the partial autocorrelations, which we denote by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_q)$.

To reparametrize the correlation matrix R_n to the partial autocorrelations $(\pi_1, \ldots, \pi_q)$, we use the recursive Levinson-Durbin algorithm. Starting from $\rho_{11} = \rho(1)$, we compute recursively

$$\rho_{jj} = \frac{\rho(j) - \sum_{k=1}^{j-1} \rho_{j-1,k}\rho(j-k)}{1 - \sum_{k=1}^{j-1} \rho_{j-1,k}\rho(k)},$$
$$\rho_{jk} = \rho_{j-1,k} - \rho_{jj}\rho_{j-1,j-k}, \quad k = 1, \ldots, j-1,$$

where $\pi_j = \rho_{jj}$ for $j = 1, \ldots, n-1$. Note that the function that maps the autocorrelations to the partial autocorrelations is indeed invertible and its inverse can be easily computed.

Let Y_i denote the standardized variable, standardized by the sample variance, which we take as an estimate of $\sigma^2 = \gamma(0) = \text{Var}(X_t)$. Then given data $\mathbf{y}_n = (y_1, \ldots, y_n)$, the log-likelihood function (apart from a constant) is given by

$$\ell_n(\boldsymbol{\pi}) = -\frac{1}{2}\log\det R_n - \frac{1}{2}\mathbf{y}_n^T R_n^{-1}\mathbf{y}_n.$$

Therefore, we propose to estimate the AR order by solving

$$\hat{\boldsymbol{\pi}}_n = \arg\max_{\boldsymbol{\pi}\in\mathcal{D}}\Big\{\ell_n(\boldsymbol{\pi}) - p_\lambda(\boldsymbol{\pi})\Big\}, \tag{4.3}$$

where $\mathcal{D} = \{\boldsymbol{\pi} = (\pi_1, \ldots, \pi_q) : \pi_j \in (-1, 1)\}$ and $\lambda > 0$ is a tuning parameter. The estimated AR order is then taken to be $\hat{p}_n = \max(j : \hat{\pi}_j \neq 0)$. We consider two penalties

for $p_\lambda(\cdot)$ in (4.3). For the first penalty, we take $p_\lambda(\boldsymbol{\pi}) = \lambda \sum_{j=1}^{q} \frac{|\pi_j|}{|\pi_{j-1}|}$, which is the nested lasso penalty of Levina et al. (2008), where $q < n$ is a known upper bound on the true order p_0, $\pi_0 = 1$, and $0/0$ is defined as 0. The effect of the penalty is that if $\pi_{j-1} = 0$, then $\pi_j = 0$. Thus, if the partial autocorrelation at lag j is zero, then the partial autocorrelations at all subsequent lags are zero.

One drawback of the nested lasso penalty is that if the PAC at lag j is non-zero, then it cannot set PACs at intermediate lags to zero. McLeod and Zhang (2006, 2008) distinguished between three types of autoregressive models. The first type is the non-subset AR(p) model with p non-zero AR coefficients and p non-zero PACs. The second type is the usual subset AR(p) model, where the AR coefficient of lag p is non-zero with some intermediate AR coefficients being zero (in this case, all of the PACs at the first p lags may be non-zero or there may be some PACs at lags $j < p$ that are zero). The last type is a family of subset AR(p) models, where all the AR coefficients of lags $j \leq p$ are non-zero, while the PAC at lag p is non-zero with some PACs at intermediate lags constrained to zero.

The PAC-based nested lasso method is designed to perform best for identifying the order of AR models of the first type since then the partial autocorrelation matrix is banded with p non-zero bands. To estimate the order of AR models of the second and third types, we use a lasso (Tibshirani, 1996) penalty so that PACs are penalized independently of each other. That is, we take $p_\lambda(\boldsymbol{\pi}) = \lambda \sum_{j=1}^{q} |\pi_j|$. The computational procedure presented in Section 4.4 for maximizing the penalized log-likelihood (4.3) is general enough to accommodate both the lasso and nested lasso penalties. The tuning parameter $\lambda > 0$ in (4.3) can be selected using a data-driven method, such as cross-validation, AIC or BIC.

4.4 Computational Procedure

To solve the optimization problem in (4.3), we adopt an iterative procedure, similar to that in Section 3.5. The procedure requires the specification of the initial estimate $\hat{\boldsymbol{\pi}}^{(0)} = (\hat{\pi}_1^{(0)}, \ldots, \hat{\pi}_q^{(0)})$, which we take to be the sample partial autocorrelations with $q = n - 1$. If

an upper bound $q < n-1$ on the true AR order is known, then the MLEs of the partial autocorrelations corresponding to the AR(q) model may be used. We adopt a cyclic coordinate descent procedure, which starts with the initial estimate $\hat{\boldsymbol{\pi}}^{(0)}$ and produces a sequence $\left\{\hat{\boldsymbol{\pi}}^{(m)}\right\}$. At each iteration m, $\hat{\boldsymbol{\pi}}^{(m+1)}$ is obtained by sequentially updating each component of $\hat{\boldsymbol{\pi}}^{(m)}$ while the other components are held fixed. Componentwise updating is possible as the PACs are unconstrained. We let $f(\cdot)$ denote the penalized log-likelihood function in (4.3). Then at iteration m, $\hat{\pi}_j^{(m)}$ is updated by solving

$$\hat{\pi}_j^{(m+1)} = \underset{\pi_j \in (-1,1)}{\arg\max}\ f(\pi_1^{(m+1)}, \ldots, \pi_{j-1}^{(m+1)}, \pi_j, \pi_{j+1}^{(m)}, \ldots, \pi_q^{(m)}).$$

The first term in the penalized log-likelihood function (4.3) can be simplified by using the fact that

$$\det R_n = \prod_{j=1}^{n-1}(1-\pi_j^2)^{n-j} = \prod_{j=1}^{q}(1-\pi_j^2)^{n-j}$$

for a stationary AR(q) model, while the second term in (4.3) is a polynomial in the π_j's. Each component update must be done numerically, using an algorithm such as `optimize` in `R`, which employs Brent's method.

After each $\hat{\pi}_j^{(m)}$ is updated to $\hat{\pi}_j^{(m+1)}$, we set $\hat{\boldsymbol{\pi}}^{(m+1)} = (\hat{\pi}_1^{(m+1)}, \ldots, \hat{\pi}_q^{(m+1)})$ and then repeat the process until the sequence $\hat{\boldsymbol{\pi}}^{(0)}, \hat{\boldsymbol{\pi}}^{(1)}, \ldots$ converges. To ensure numerical stability when the nested lasso penalty is used, we threshold the absolute values of the partial autocorrelations at some pre-specified $\epsilon > 0$. Once convergence is achieved, we set all estimates equal to ϵ to zero. The procedure was found to perform well in our simulation studies (see Section 4.5).

Remark: Note that in this implementation, we standardized the variables by the sample variance, which was taken as an estimate of $\sigma^2 = \mathrm{Var}(X_t)$. In other implementations, σ^2 or the prediction variance σ_e^2 may be estimated.

- *Jointly estimating* $\boldsymbol{\pi}$ *and* σ^2*:* For the former case, we first consider the joint log-

likelihood function

$$\ell(\boldsymbol{\pi}, \sigma^2) = -\frac{1}{2}\log\det\Sigma_n - \frac{1}{2}\mathbf{x}_n^T\Sigma_n\mathbf{x}_n = -\frac{1}{2}\log\det\left(V_nR_nV_n\right) - \frac{1}{2}\mathbf{x}_n^TV_n^{-1}R_n^{-1}V_n^{-1}\mathbf{x}_n$$
$$= -\frac{1}{2}\log\det R_n - \log\det V_n - \mathbf{x}_n^TV_n^{-1}R_n^{-1}V_n^{-1}\mathbf{x}_n$$
$$= -\frac{1}{2}\log\det R_n - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\mathbf{x}_n^TR_n^{-1}\mathbf{x}_n,$$

and then maximize the joint log-likelihood function with respect to σ^2 to get the concentrated log-likelihood (after dropping constant terms)

$$\ell_c(\boldsymbol{\pi}) = -\frac{1}{2}\log\det R_n - \frac{n}{2}\log\hat{\sigma}^2,$$

where $\hat{\sigma}^2 = \frac{1}{n}\mathbf{x}_n^TR_n^{-1}\mathbf{x}_n$. The penalized concentrated log-likelihood can then be maximized to obtain an estimate of Π.

- *Jointly estimating $\boldsymbol{\pi}$ and σ_e^2:* In the latter case for the estimation of σ_e^2, we first consider the joint log-likelihood function

$$\ell(\boldsymbol{\pi}, \sigma_e^2) = -\frac{1}{2}\log\det\Sigma_n - \frac{1}{2}\mathbf{x}_n^T\Sigma_n\mathbf{x}_n = -\frac{1}{2}\log\sigma_e^2 + \frac{1}{2}\log\det M_n^{(p)} - \frac{1}{2\sigma_e^2}\mathbf{x}_n^TM_n^{(p)}\mathbf{x}_n,$$

where $M_n^{(p)} = \sigma_e^2\Sigma_n^{-1}$. Now since $\sigma_e^2 = \sigma^2\prod_{j=1}^p(1-\pi_j^2)$ and $|R_n| = \prod_{j=1}^p(1-\pi_j^2)^{n-j}$,

$$M_n^{(p)} = \sigma_e^2\Sigma_n^{-1} = \frac{\sigma_e^2}{\sigma^2}R_n^{-1} = \prod_{j=1}^p(1-\pi_j^2)R_n^{-1}$$

and

$$|M_n^{(p)}| = \prod_{j=1}^p(1-\pi_j^2)^n|R_n|^{-1} = \prod_{j=1}^p(1-\pi_j^2)^j = h(\boldsymbol{\pi}).$$

Therefore, as in McLeod and Zhang (2006), letting $S(\boldsymbol{\pi}) = \mathbf{x}_n^TM_n^{(p)}\mathbf{x}_n$, the log-

likelihood of $(\boldsymbol{\pi}, \sigma_e^2)$ is

$$\ell(\boldsymbol{\pi}, \sigma_e^2) = -\frac{n}{2}\log(\sigma_e^2) + \frac{1}{2}\log h(\boldsymbol{\pi}) - \frac{1}{2\sigma_e^2}S(\boldsymbol{\pi}).$$

Then maximizing the joint log-likelihood function with respect to σ_e^2, one obtains the concentrated log-likelihood (after dropping constant terms)

$$\ell_c(\boldsymbol{\pi}) = \frac{1}{2}\log h(\boldsymbol{\pi}) - \frac{n}{2}\log\hat{\sigma}_e^2,$$

where $\hat{\sigma}_e^2 = \frac{1}{n}S(\boldsymbol{\pi})$. The penalized concentrated log-likelihood can then be maximized to obtain an estimate of $\boldsymbol{\pi}$. We use this procedure to implement the lasso method based on the PAC parametrization.

The maximization of the penalized concentrated log-likelihood in either case may also be done using a cyclic coordinate descent procedure.

4.5 Simulation Studies

In this section, we provide a simulation study to evaluate the performance of our proposed PAC-based methods for estimating the order of an AR model. We compare their performance to the standard information criteria AIC and BIC, the bias-corrected AIC of Hurvich and Tsai (1989), the criterion of Hannan and Quinn (1979) as well as the lasso and modified lasso methods of Wang et al. (2007b). For the information criteria, we consider both ML and Yule-Walker parameter estimation. The performance of the methods will be assessed for the following three Gaussian autoregressive models:

Simulation 1: $X_t = 0.48X_{t-1} + 0.40X_{t-2} + e_t, \qquad t = 3, \ldots, n,$

Simulation 2: $X_t = 0.455X_{t-1} - 0.2015X_{t-2} - 0.182X_{t-3} - 0.30X_{t-4} + e_t, \qquad t = 5, \ldots, n,$

Simulation 3: $X_t = 0.52X_{t-1} + 0.2078X_{t-2} - 0.2526X_{t-3} - 0.4707X_{t-4} + 0.184X_{t-5}$
$+ 0.2X_{t-6} + e_t, \qquad t = 7, \ldots,$

where $e_t \sim \mathcal{N}(0, \sigma_e^2)$ with $\sigma_e^2 = 0.01$. For Simulation 1, the corresponding partial autocorrelations are $\pi_1 = 0.8$, $\pi_2 = 0.4$ and $\pi_j = 0$ for $j > 2$. For Simulation 2, the partial autocorrelations are $\pi_1 = 0.5$, $\pi_2 = -0.4$, $\pi_3 = -0.35$, $\pi_4 = -0.3$ and $\pi_j = 0$ for $j > 4$. For Simulation 3, the partial autocorrelations are $\pi_1 = 0.6$, $\pi_2 = -0.4$, $\pi_3 = -0.5$, $\pi_4 = -0.3$, $\pi_5 = 0.3$, $\pi_6 = 0.2$ and $\pi_j = 0$ for $j > 6$.

To obtain the estimated order for the PAC-based methods, we set $\hat{\pi}_j = 0$ if $|\hat{\pi}_j| < 10^{-3}$ for $j = 1, \ldots, q$ and take $\hat{p} = \max(j : \hat{\pi}_j \neq 0)$. For the PAC-based nested lasso method, we did not set an upper bound $q < n - 1$ on the true AR order. We found that the method still performs well even when $q = n - 1$. However, such a large upper bound does increase the computational burden. For the lasso and modified lasso methods of Wang et al. (2007b) as well as the lasso method based on the PAC parametrization, we shrink from AR models of order $q = 15$ to obtain subset AR models. For the PAC-based methods, we consider both AIC and BIC for selecting the tuning parameter. For the lasso and modified lasso methods of Wang et al. (2007b), we also use BIC for tuning parameter selection as well as 5-fold cross-validation. The simulations corresponding to the lasso and modified lasso methods were conducted using the `R` package `glmnet`. If $\tilde{\phi}_j$ is the unpenalized least squares estimator of ϕ_j, then the weights used in the modified lasso method of Wang et al. (2007b) are $1/|\tilde{\phi}_j|$. We also consider the weights $1/|\tilde{\phi}_j|^\gamma$, where we take $\gamma = 0.5$ and 2, and use the value of γ that yields optimal order selection performance.

For each model, we report the percentage of times out of $N = 500$ replications that the estimated order equals a given value of p with sample sizes $n = 100$ and 200. The results can be found in Tables 4.1 to 4.4.

For Simulation 1 (see Table 4.1), it was found that the PAC-based nested lasso method with the BIC-selector and the classical BIC with the MLEs performed the best. For both sample sizes under consideration, the bias-corrected AIC, AIC_c, performed slightly better than AIC, while the HQ criterion overestimated less than AIC and AIC_c but overestimated more than BIC. For the methods in Wang et al. (2007b), the modified lasso method with the BIC-selector performed the best. The lasso had a greater tendency to

overfit than the modified lasso. This is because each autoregressive coefficient in the lasso is penalized with the same tuning parameter and so the insignificant coefficients cannot be effectively shrunk to 0. This was first observed in Wang et al. (2007b), who recommended using the modified lasso with the BIC-selector. When it comes to estimating the tuning parameter in the L_1-penalized or weighted L_1-penalized likelihood, they found that BIC performs better than cross-validation, which was also observed in our simulation studies. The PAC-based methods offer improved performance over the lasso methods of Wang et al. (2007b), which are based on penalization of the AR coefficients.

For Simulation 2 (see Table 4.2) with $n = 100$, the bias-corrected AIC using Yule-Walker estimation and HQC using ML estimation performed the best. The methods of Wang et al. (2007b) performed the worst. The lasso and modified lasso with 5-fold CV as the tuning parameter selector had strong tendencies to overfit, and they did not significantly - if at all - improve with an increased sample size. Using instead BIC as the tuning parameter selector resulted in smaller overestimating rates for both these methods. The PAC-based lasso method performed better than the modified lasso method; it had a smaller percentage of overestimated orders. For $n = 200$, the BIC performed the best with either Yule-Walker or ML estimation, selecting the correct order at a rate of roughly 95%. The PAC-based nested lasso method with the BIC-selector was not far behind, selecting the correct order at a rate of 91.0%.

For Simulation 3 (see Table 4.3) with $n = 100$, the bias-corrected AIC and the PAC-based nested lasso method with the AIC selector performed the best. The information criteria using the Yule-Walker parameter estimates performed very poorly with strong tendencies to underestimate, while the same information criteria using the MLEs performed satisfactorily. This is not surprising since the Yule-Walker estimates are biased in small samples. While Yule-Walker estimation was found to be advantageous over ML estimation for fitting overparametrized models when performing order estimation (Chen et al., 1993), its advantage disappears when the maximum candidate order q is not very large relative to the true order p_0. For the penalized likelihood methods, the lasso with either the CV or BIC-selector also performed poorly, with strong tendencies to overes-

timate. For sample size $n = 200$, the method of Hannan and Quinn (1979), and the PAC-based nested lasso method with the BIC-selector performed the best. Comparing the two tuning parameter selectors for the PAC-based nested lasso method, we see more underestimation by BIC relative to AIC for smaller n, but better results for the BIC than AIC for larger n. This is not surprising since AIC was shown to be an inconsistent criterion in a variety of settings (see, for example, Zhang et al., 2010). The modified lasso procedure performed much better than the lasso procedure, while the PAC-based methods performed much better than the modified lasso procedure, which is based on penalization of the AR coefficients.

As discussed in Section 4.3, the drawback of the nested lasso penalty is that it cannot eliminate bands of weak signals in between bands of strong signals. We verify this in simulation by investigating the performance of our method when applied to samples generated from the following stationary Gaussian AR(3) model

$$\text{Simulation 4:} \quad X_t = 0.80X_{t-1} - 0.32X_{t-2} + 0.4X_{t-3} + e_t, \qquad t = 4, \ldots, n,$$

where $e_t \sim \mathcal{N}(0, \sigma_e^2)$ with $\sigma_e^2 = 0.01$, which has $\pi_1 = 0.8$, $\pi_2 = 0$ and $\pi_3 = 0.4$. From Table 4.4 with sample size $n = 100$, it can be seen that the nested lasso methods do not perform well in identifying the true order, but neither does the adaptive lasso method. For $n = 100$, the PAC-based nested lasso method with the BIC-selector wrongly selects an AR(1) model 23.2% of the time, but it also overselects quite often. For $n = 200$, it appears to be split between AR(3) and AR(4) models. As the sample size increases, the underestimating rate of the PAC-based nested lasso method decreases, but it still has a strong tendency to overestimate. It is clear that a lasso penalty applied to the partial autocorrelations is more appropriate, as the partial autocorrelations in the second band of the PAC matrix could be shrunk without wrongly shrinking the third band of partial autocorrelations. For both sample sizes, the PAC-based lasso method with the BIC-selector performs significantly better than the other methods under consideration.

Simulation 1	Method	Order				
		1	2*	3	4	>4
$n = 100$	AIC (YW)	4.0	87.4	6.2	1.8	0.6
	AIC_c (YW)	4.0	88.4	5.6	1.6	0.4
	BIC (YW)	12.4	87.0	0.4	0.2	0
	HQC (YW)	7.8	88.4	2.4	1.0	0.4
	AIC (MLE)	1.4	67.6	12.4	4.8	13.8
	AIC_c (MLE)	1.8	73.2	11.6	3.8	9.6
	BIC (MLE)	7.0	89.2	2.6	0.8	0.4
	HQC (MLE)	2.8	85.8	6.6	2.6	2.2
	Lasso (CV)	0	15.2	3.4	6.0	75.4
	Lasso (BIC)	1.2	44.8	8.0	6.4	39.6
	Modified lasso (CV)	3.2	28.8	1.4	2.4	64.2
	Modified lasso (BIC)	9.2	60.6	2.2	2.2	25.8
	Lasso - PAC (AIC)	5.8	49.4	2.0	3.6	39.2
	Lasso - PAC (BIC)	22.6	70.8	0.8	0.2	5.6
	Nested lasso - PAC (AIC)	1.4	78.0	14.0	2.4	4.2
	Nested lasso - PAC (BIC)	2.6	92.8	3.6	0.8	0.2
$n = 200$	AIC (YW)	0	89.6	6.2	2.4	1.8
	AIC_c (YW)	0	90.2	6.4	2.0	1.4
	BIC (YW)	0	99.0	1.0	0	0
	HQC (YW)	0	96.8	2.4	0.6	0.2
	AIC (MLE)	0	71.6	9.6	5.8	13.0
	AIC_c (MLE)	0	74.0	9.2	5.4	11.4
	BIC (MLE)	0	97.6	1.6	0.6	0.2
	HQC (MLE)	0	90.8	6.4	1.8	1.0
	Lasso (CV)	0	10.6	5.6	3.8	80.0
	Lasso (BIC)	0	45.0	12.2	5.6	37.2
	Modified lasso (CV)	0.2	44.2	1.0	1.4	53.2
	Modified lasso (BIC)	0.4	83.2	1.0	1.4	14.0
	Lasso - PAC (AIC)	0	51.2	1.2	2.4	45.2
	Lasso - PAC (BIC)	1.2	92.0	0.2	0.6	6.0
	Nested lasso - PAC (AIC)	0	75.4	16.4	2.8	5.4
	Nested lasso - PAC (BIC)	0	97.4	2.0	0.4	0.2

Table 4.1: The percentage of times in which order p was estimated by AIC, AIC_c, BIC, HQC, lasso (CV, BIC, $q = 15$), modified lasso (CV, BIC, $q = 15$), PAC-based lasso (AIC, BIC), and PAC-based nested lasso (AIC, BIC), where $n = 100, 200$ observations are generated from the stationary Gaussian AR(2) model $X_t = 0.48X_{t-1} + 0.4X_{t-2} + e_t$, where $e_t \sim \mathcal{N}(0, 0.01)$. The true order is denoted by *.

Simulation 2	Method	Order						
		1	2	3	4*	5	6	>6
$n = 100$	AIC (YW)	0.4	0.2	10.2	78.6	6.2	2.6	1.8
	AIC_c (YW)	0.4	0.6	11.4	79.6	5.2	1.8	1.0
	BIC (YW)	2.2	4.4	21.0	71.4	1.0	0	0
	HQC (YW)	0.4	1.4	16.4	78.4	2.4	0.6	0.4
	AIC (MLE)	0	0.2	3.6	68.4	12.4	4.4	11.0
	AIC_c (MLE)	0	0.4	4.4	73.8	11.0	3.6	6.8
	BIC (MLE)	0.6	1.4	16.6	78.2	2.2	0.8	0.2
	HQC (MLE)	0.4	0.2	8.0	79.6	6.2	2.6	3.0
	Lasso (CV)	0	0	0	8.4	2.0	1.0	88.6
	Lasso (BIC)	0.4	0	0.8	35.2	3.2	1.2	59.2
	Modified lasso (CV)	0.2	0.2	3.6	21.0	2.2	2.6	70.2
	Modified lasso (BIC)	1.2	0.6	9.0	51.2	2.0	2.4	33.6
	Lasso - PAC (AIC)	0.2	0	5.2	35.0	2.6	2.2	54.8
	Lasso - PAC (BIC)	4.0	0.2	17.8	69.0	1.0	0.2	7.8
	Nested lasso - PAC (AIC)	0	0	2.6	62.0	24.2	4.8	6.2
	Nested lasso - PAC (BIC)	0.6	1.8	8.8	76.4	10.0	1.6	0.4
$n = 200$	AIC (YW)	0	0	0.6	88.4	6.2	4.2	0.6
	AIC_c (YW)	0	0	0.6	89.4	5.6	3.8	0.6
	BIC (YW)	0	0	3.6	95.4	1.0	0	0
	HQC (YW)	0	0	1.4	94.0	2.8	1.6	0.2
	AIC (MLE)	0	0	0.2	71.6	9.6	8.2	10.4
	AIC_c (MLE)	0	0	0.2	74.4	10.0	7.4	8
	BIC (MLE)	0	0	1.8	95.0	2.6	0.6	0
	HQC (MLE)	0	0	1.0	89.6	5.2	3.4	0.8
	Lasso (CV)	0	0	0	4.8	0.4	1.4	93.4
	Lasso (BIC)	0	0	0	33.8	2.0	2.0	62.2
	Modified lasso (CV)	0	0	0.8	28.2	1.6	2.4	67.0
	Modified lasso (BIC)	0	0	2.8	70.6	1.6	3.0	22.0
	Lasso - PAC (AIC)	0	0	0.4	28.8	1.6	2.0	67.2
	Lasso - PAC (BIC)	0	0	3.2	69.6	1.2	2.8	23.2
	Nested lasso - PAC (AIC)	0	0	0	65.6	23.4	6.8	4.2
	Nested lasso - PAC (BIC)	0	0	0.4	91.0	7.4	1.0	0.2

Table 4.2: The percentage of times in which order p was estimated by AIC, AIC_c, BIC, HQC, lasso (CV, BIC, $q = 15$), modified lasso (CV, BIC, $q = 15$), PAC-based lasso (AIC, BIC), and PAC-based nested lasso (AIC, BIC), where $n = 100, 200$ observations are generated from the stationary Gaussian AR(4) model $X_t = 0.455X_{t-1} - 0.2015X_{t-2} - 0.182X_{t-3} - 0.30X_{t-4} + e_t$, where $e_t \sim \mathcal{N}(0, 0.01)$. The true order is denoted by *.

Simulation 3	Method	Order						
		<4	4	5	6*	7	8	>8
$n = 100$	AIC (YW)	3.8	20.8	46.0	26.6	2.0	0.4	0.4
	AIC_c (YW)	4.6	23.8	46.4	23.8	1.2	0.2	0
	BIC (YW)	19.2	33.8	40.0	7.0	0	0	0
	HQC (YW)	7.8	31.0	44.6	16.2	0.4	0	0
	AIC (MLE)	0.2	7.2	28.4	40.6	8.0	4.4	11.2
	AIC_c (MLE)	0.4	9.2	34.6	41.8	5.2	3.8	5.0
	BIC (MLE)	6.2	25.2	41.2	25.4	1.2	0.4	0.4
	HQC (MLE)	1.8	16.0	37.6	36.6	4.8	1.4	1.8
	Lasso (CV)	0	1.0	0	0.8	3.4	2.0	92.8
	Lasso (BIC)	0	8.4	0.6	1.8	11.0	9.6	68.6
	Modified lasso (CV)	2.2	5.6	5.2	12.4	2.2	2.0	70.4
	Modified lasso (BIC)	5.0	23.0	10.2	22.4	3.4	2.6	33.4
	Lasso - PAC (AIC)	1.6	7.6	5.8	23.4	4.8	4.6	52.2
	Lasso - PAC (BIC)	2.6	15.6	9.6	38.2	5.4	4.4	24.2
	Nested lasso - PAC (AIC)	0.4	5.0	21.0	41.2	16.6	5.8	10.0
	Nested lasso - PAC (BIC)	5.8	17.6	33.2	31.6	8.6	1.8	1.4
$n = 200$	AIC (YW)	0	1.2	27.8	63.0	5.2	2.0	0.8
	AIC_c (YW)	0	1.2	29.4	62.0	5.0	1.8	0.6
	BIC (YW)	0.6	9.6	52.6	36.6	0.6	0	0
	HQC (YW)	0.2	3.8	41.6	51.6	2.0	0.6	0.2
	AIC (MLE)	0	0.2	11.4	61.6	10.6	6.6	9.6
	AIC_c (MLE)	0	0.2	12.8	63.8	10.8	5.8	6.6
	BIC (MLE)	0.2	5.0	36.2	56.4	1.8	0.4	0
	HQC (MLE)	0	1.4	22.4	68.6	4.6	1.8	1.2
	Lasso (CV)	0	0	0	0.4	0.8	2.6	96.2
	Lasso (BIC)	0	2.4	0	1.8	11.4	11.6	72.8
	Modified lasso (CV)	0.2	0.8	5.2	14.4	2.4	3.6	73.4
	Modified lasso (BIC)	0.8	7.2	17.4	41.4	5.6	1.6	26.0
	Lasso - PAC (AIC)	0	0.4	3.0	25.6	3.8	4.6	62.6
	Lasso - PAC (BIC)	0.4	5.2	8.4	63.0	5.6	2.8	14.6
	Nested lasso - PAC (AIC)	0	0.2	4.8	53.8	26.2	7.4	7.6
	Nested lasso - PAC (BIC)	0	2.2	16.6	66.6	12.4	2.0	0.2

Table 4.3: The percentage of times in which order p was estimated by AIC, AIC_c, BIC, HQC, lasso (CV, BIC, $q = 15$), modified lasso (CV, BIC, $q = 15$), PAC-based lasso (AIC, BIC), and PAC-based nested lasso (AIC, BIC), where $n = 100, 200$ observations are generated from the stationary Gaussian AR(6) model $X_t = 0.52X_{t-1} + 0.2078X_{t-2} - 0.2526X_{t-3} - 0.4707X_{t-4} + 0.184X_{t-5} + 0.2X_{t-6} + e_t$, where $e_t \sim \mathcal{N}(0, 0.01)$. The true order is denoted by *.

Simulation 4	Method	Order						
		1	2	3*	4	5	6	>6
$n = 100$	Modified lasso ($\gamma = 1$, BIC)	20.8	0.8	40.6	4.0	2.6	2.2	29.0
	Adaptive lasso ($\gamma = 2$, BIC)	21.6	0.8	43.6	3.8	2.6	2.6	25.0
	Lasso - PAC (AIC)	8.4	0.2	52.6	2.6	3.2	2.8	30.2
	Lasso - PAC (BIC)	13.2	0.2	71.4	2.0	2.0	1.8	9.4
	Nested lasso - PAC (AIC)	8.0	1.0	30.8	34.8	13.4	4.0	8.0
	Nested lasso - PAC (BIC)	23.2	5.2	40.6	22.8	5.4	1.4	1.4
$n = 200$	Modified lasso ($\gamma = 1$, BIC)	3.0	0	55.2	3.0	1.8	3.6	33.4
	Adaptive lasso ($\gamma = 2$, BIC)	2.6	0	68.6	2.2	1.4	1.8	23.4
	Lasso - PAC (AIC)	0.2	0	57.6	1.4	3.2	2.0	35.6
	Lasso - PAC (BIC)	0.2	0	87.2	1.4	0.4	1.0	9.8
	Nested lasso - PAC (AIC)	2.2	0	34.0	46.4	10.4	3.2	3.8
	Nested lasso - PAC (BIC)	17.4	4.0	51.4	24.4	2.6	0.2	0

Table 4.4: The percentage of times in which order p was estimated by modified lasso ($\gamma = 1$, BIC, $q = 15$), adaptive lasso ($\gamma = 2$, BIC, $q = 15$), PAC-based lasso (AIC, BIC), and PAC-based nested lasso (AIC, BIC), where $n = 100, 200$ observations are generated from the stationary Gaussian AR(3) model $X_t = 0.80X_{t-1} - 0.32X_{t-2} + 0.4X_{t-3} + e_t$, $t = 4, \ldots, n$, where $e_t \sim \mathcal{N}(0, 0.01)$. Note that this model has partial autocorrelations $\pi_1 = 0.8$, $\pi_2 = 0$, $\pi_3 = 0.4$, and $\pi_j = 0$ for $j > 3$. The true order is denoted by *.

4.6 Standard Errors for Selection-Based Estimators by Bootstrapping

In this section, we study how to obtain standard error estimates for our maximum penalized likelihood estimates $\hat{\pi}_j$ by bootstrapping. We refer to the literature on bootstrap methods for computing standard errors and confidence intervals for selection-based estimators in the linear regression context. In what follows, we review some of these bootstrap procedures.

Given correct model specification, a number of procedures are available for obtaining reliable parameter estimates along with reliable variance and confidence interval estimates. When model selection precedes parameter estimation, however, inference is usually done conditional on the selected model and does not take into account any uncertainty in model selection. As a result, variance estimates are too small and confidence interval estimates have less than nominal coverage. Various authors have discussed the need to incorporate model selection uncertainty into statistical inference whenever data-

based model selection precedes parameter estimation (Chatfield, 1995; Buckland et al., 1997; Leeb and Pötscher, 2006). The bootstrap (Efron, 1979), which is a data resampling procedure, is one approach for estimating variance and generating robust confidence intervals that avoids conditioning on a selected model. In this setting, the bootstrap involves generating resamples and then applying the model selection procedure separately to each resample (see, for example, Buckland et al., 1997). The sample variance of the bootstrap estimates then provides an estimated variance for the original estimate from the real data. When using AIC to select covariates in a Poisson regression model, Buckland et al. (1997) found that the performance of the bootstrap procedure depends on the resampling method used. Generating the bootstrap resamples from the AIC-selected model, for example, would bias the results in favour of the model with the smallest AIC value. When considering estimation accuracy after performing model selection in a regression context, Efron (2014) suggested generating bootstrap samples not from the selected model, but from the full model instead. Applying the lasso, for example, to bootstrap samples generated from the lasso-selected model, would result in double shrinkage.

Even after careful consideration of the manner in which bootstrap samples are generated, Efron (2014) found that when a model selection procedure is applied to each bootstrap sample, the resulting bootstrap replications of an estimate can be very different from the original selection-based estimate. He observes that model selection can produce "jumpy and erratic" estimates, and therefore considers a method called bootstrap smoothing (Efron and Tibshirani, 1996), which is a form of model averaging that reduces variability and eliminates discontinuities of selection-based estimators. His paper focuses on attaching standard error estimates to selection-based estimators of the mean in regression models. In one of his examples, he uses the lasso to obtain an estimate $\hat{\mu}_{\hat{\lambda}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\hat{\lambda}}$ of mean μ in a standard normal linear regression model and wishes to attach standard error estimates to $\hat{\mu}_{\hat{\lambda}}$. His procedure involves two levels of bootstrapping. At the first level, B_1 bootstrap samples are drawn from the full ordinary least squares (OLS) model. At the second level, for each bootstrap sample b, $1 \leq b \leq B_1$, the parameter estimates of the full OLS model are obtained, and then another B_2 bootstrap samples are drawn

from the fitted OLS model. The lasso is then applied to each of the B_2 bootstrap samples (including tuning parameter selection) to obtain the lasso-estimated mean $\hat{\mu}^*_{\hat{\lambda}^*}$, and then a smoothed estimate of the mean is obtained by averaging over the B_2 bootstrap replications $\hat{\mu}^*_{\hat{\lambda}^*}$. However, there is no guarantee that the bootstrap replications $\hat{\boldsymbol{\beta}}^*_{\hat{\lambda}^*}$ will have the same number of zero coordinate estimates as the original lasso estimate $\hat{\boldsymbol{\beta}}_{\hat{\lambda}}$, and so when attempting to apply this methodology for obtaining standard error estimates of $\hat{\boldsymbol{\beta}}_{\hat{\lambda}}$, the smoothed estimate $\tilde{\boldsymbol{\beta}}$ may also have fewer non-zero coordinate estimates than the original lasso estimate.

One way of eliminating the possibility of having a bootstrap estimate with a different number of zero coordinate estimates than the original estimate is to use the bootstrap for both model selection and subsequent inference. In the linear regression context, Shao (1996) showed that the bootstrap procedure (where the model that minimizes the bootstrap estimate of mean-squared prediction error is selected) is inconsistent in the sense that the probability of selecting the optimal subset of variables does not converge to 1 as the sample size grows. He corrects the inconsistency by modifying the sampling method. Recently, Gupta and Lahiri (2014) considered a similar idea, where the model is selected based on the bootstrap resamples. In particular, they proposed a maximum frequency (MF) method, where bootstrap-based inference is conducted only on a subcollection of B bootstrap resamples that resulted in the selection of the model with the highest selection frequency among the B replicates. Since all the replicates in this collection correspond to the same model, the extra variability arising from model selection in different resamples is eliminated. This is the approach that we choose to use in our real data application in Section 4.7.

The other challenge in conducting bootstrap-based inference for the vector of partial autocorrelations $\boldsymbol{\pi}$ is in finding the best manner of resampling in the time domain. One approach is to use model-based resampling. The idea of model-based resampling is to first fit a suitable model to the data, to construct residuals from the fitted model, and then to resample from the residuals so that a new series can be obtained by adding the residual resamples to the fitted values. As discussed, when desiring standard error

estimates of selection-based estimators, the residuals from which bootstrap resamples are drawn should not be obtained from the model with shrinkage, but rather from the full model so as to avoid double shrinkage when the selection-based procedure is applied to the bootstrap resample.

Another approach to resampling in the time domain involves resampling not from residuals, but from blocks of consecutive observations. We re-state the procedure outlined in Davison and Hinkley (1997). The idea is to divide the data into N non-overlapping blocks of length L, where we assume that the length of the series is $n = NL$. Then taking $z_1 = (x_1, \ldots, x_l)$, $z_2 = (x_{l+1}, \ldots, x_{2l})$, and so forth, a new series is obtained by sampling from the blocks $z_1, \ldots, z_N$ with equal probability $1/N$, and then placing the blocks end-to-end. Series obtained by resampling schemes based on blocks, however, are less dependent than the original data. The hope is that if the blocks are sufficiently long, then most of the original dependence will be preserved in the resampled series so that statistics calculated from the bootstrap samples will have approximately the same distribution as statistics calculated from resamples of the original series (Davison and Hinkley, 1997). On the other hand, the blocks cannot be too long since then there will be fewer distinct blocks and not enough variability in the sampled blocks. Various authors have discussed optimal block lengths (Hall et al., 1995; Lahiri, 2003). Hall et al. (1995) discuss bootstrap blocking rules and find that the optimal block size depends significantly on context. For our real data application in Section 4.7, we use a block resampling scheme.

4.7 Real Data Analysis

To illustrate the use of our proposed method for estimating the order of an AR model, we apply our method to a data set from Cowpertwait and Metcalfe (2009). The data set consists of the surface height of water, measured in millimetres relative to still water. The measurements were taken using a capacitance probe, positioned at the centre of a wave tank. The continuous voltage signal from the capacitance probe was sampled over 39.6 seconds at a rate of 10 samples per second. As discussed in Cowpertwait and Metcalfe

(2009), there is no trend and no seasonal component and therefore the assumption of a stationary process is reasonable. The time series plot of wave heights is displayed in Figure 4.1 along with its ACF and PACF in Figure 4.2. The ACF for wave heights appears to have a damped cosine wave structure that reflects the behaviour of an AR(p) process with complex roots. The PACF has values significantly different from zero at lags 1, 2, 4 to 10, and 12, suggesting an AR model of order at least 2.

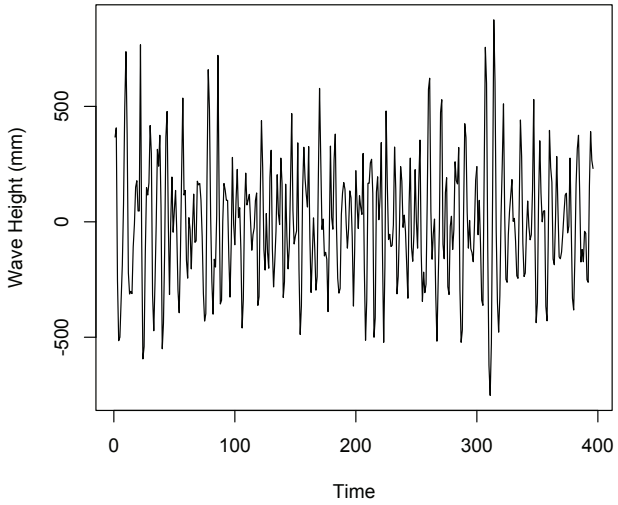

Figure 4.1: Time series plot of the wave heights data, which were sampled at the centre of a wave tank at 0.1 second intervals over a period of 39.6 seconds.

The goal of the analysis of Cowpertwait and Metcalfe (2009) was to find the best ARMA(p, q) model to fit to the time series of wave heights, which would then be used to generate a realistic wave input to a mathematical model in a computer simulation of an ocean-going tugboat. One model considered by Cowpertwait and Metcalfe (2009) for the wave heights data is an ARMA(4,4) model, which was selected based on a minimum variance of residuals. Using their criterion of selection, we find that an ARMA(4,4) model is indeed selected when $p < 6$ and $q < 6$. Rather than using the more general ARMA(p,q) model, it is common to use a high order AR(p) model. Therefore, we fit an AR(p) model to the data, allowing for the possible AR order, p, to be large. Using AIC and BIC to select the AR order results in models of orders 15 and 13, respectively. Applying the adaptive lasso procedure with $\gamma = 1$ yields a model of order 17 with 17

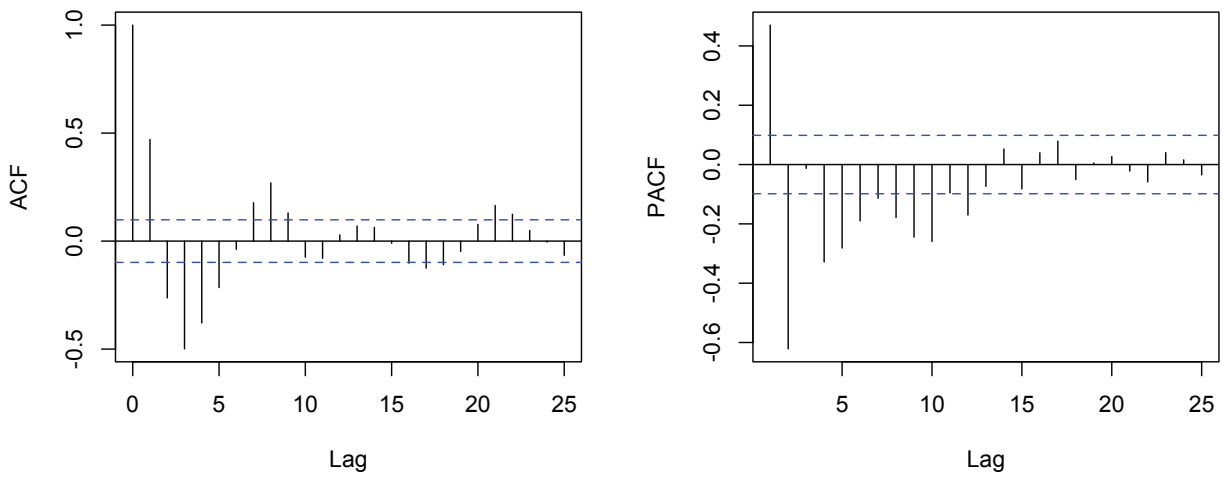

Figure 4.2: Sample autocorrelations (left) and partial autocorrelations (right) of the wave height series.

non-zero PACs, but 3 zero AR coefficients at lags 14 to 16. We next applied our proposed method for order estimation and considered both AIC and BIC for the choice of the tuning parameter. Our method selects an AR model of order 13 with both AIC and BIC agreeing on the choice of the tuning parameter. The maximum penalized likelihood estimates of π_j for $j = 1, \ldots 13$ are provided in Table 4.7. Their corresponding standard errors were obtained by bootstrapping. Various authors (Chatterjee and Lahiri, 2010; Efron, 2014) have discussed variance and confidence interval estimation of selection-based estimators in the linear regression setting via bootstrapping. See Section 4.6 for a review. However, there is no definitive bootstrap approach to date for attaching standard error estimates to penalized likelihood estimates. We use a non-overlapping block bootstrap approach to generate $B = 1000$ bootstrap resamples, where we split the time series into consecutive blocks of length $\ell = 22$ and resample blocks with replacement to obtain a new series by appending blocks end-to-end. The penalized estimates $\hat{\pi}^*_{\hat{\lambda}^*,j}$ are then found for each bootstrap resample, where $\hat{\lambda}^*$ is the BIC-selected tuning parameter and the standard errors are obtained by taking the standard deviation of the bootstrap replications $\hat{\pi}^*_{\hat{\lambda}^*,j}$ for $j = 1, \ldots, 13$, conditional on the selected AR(13) model.

We also employed the maximum frequency (MF) bootstrap (see discussion of Efron,

2014) as a means of performing order selection, where bootstrap-based inference is conducted only on the subcollection of bootstrap resamples that resulted in the choice of the model with the highest selection frequency among the B replicates. The AR model of order 11 was found to have the highest selection frequency among the $B = 1000$ replicates (see Table 4.5). In Table 4.6, we provide summaries of the bootstrap replications of the partial autocorrelations for the AR model of order k based on the subcollection of bootstrap resamples that resulted in the selection of the model of order k.

	AR Order								
	< 10	10	11	12	13	14	15	16	> 16
Nested lasso - PAC (AIC)	1.0	14.8	23.0	15.1	5.3	5.0	8.9	9.4	17.5
Nested lasso - PAC (BIC)	9.9	24.3	35.7	18.3	3.9	2.8	1.8	1.7	1.6

Table 4.5: Percentage each model was selected by the PAC-based nested lasso method with the tuning parameter chosen by AIC and BIC among $B = 1000$ bootstrap resamples, generated using a non-overlapping block bootstrap procedure for the time series of wave heights.

	AR Order			
Parameter	10	11	12	13
π_1	0.457 (0.028)	0.461 (0.026)	0.446 (0.025)	0.432 (0.023)
π_2	-0.615 (0.031)	-0.605 (0.030)	-0.606 (0.026)	-0.596 (0.027)
π_3	-0.035 (0.050)	-0.026 (0.048)	-0.027 (0.044)	-0.013 (0.045)
π_4	-0.302 (0.048)	-0.298 (0.050)	-0.293 (0.056)	-0.290 (0.052)
π_5	-0.264 (0.048)	-0.262 (0.044)	-0.269 (0.044)	-0.292 (0.053)
π_6	-0.163 (0.050)	-0.178 (0.053)	-0.186 (0.048)	-0.182 (0.049)
π_7	-0.083 (0.040)	-0.084 (0.041)	-0.086 (0.044)	-0.109 (0.046)
π_8	-0.101 (0.038)	-0.108 (0.039)	-0.109 (0.042)	-0.106 (0.035)
π_9	-0.203 (0.051)	-0.223 (0.056)	-0.197 (0.051)	-0.219 (0.056)
π_{10}	-0.160 (0.064)	-0.198 (0.063)	-0.196 (0.054)	-0.201 (0.057)
π_{11}		0.011 (0.036)	-0.028 (0.057)	-0.051 (0.053)
π_{12}			-0.064 (0.039)	-0.099 (0.048)
π_{13}				-0.010 (0.036)

Table 4.6: Mean and standard deviation of $\hat{\pi}_j^*$, $j = 1, \ldots, 13$, as a function of the selected order, based on $B = 1000$ bootstrap resamples, for the time series of wave heights.

To assess the goodness-of-fit of the AR(11), AR(13), AR(15) and ARMA(4,4) models, fitted by maximum likelihood, as well as the subset AR(17) model with shrinkage, we partition the $n = 396$ observations into two groups of 356 (past) and 40 (future) observations. The first group will be used to refit the models and compute the forecasts of the next 40 future values. The second group of 40 observations will be used to compute out-of-sample prediction errors. The empirical performance of the models will be assessed

Parameter	MPLE	Standard Error
π_1	0.451	0.023
π_2	-0.630	0.027
π_3	-0.041	0.045
π_4	-0.307	0.052
π_5	-0.282	0.053
π_6	-0.190	0.049
π_7	-0.142	0.046
π_8	-0.157	0.035
π_9	-0.244	0.056
π_{10}	-0.259	0.057
π_{11}	-0.097	0.053
π_{12}	-0.170	0.048
π_{13}	-0.074	0.036

Table 4.7: Maximum penalized likelihood estimates (MPLEs) of the partial autocorrelations π_j, $j = 1, \ldots, 13$ along with their standard errors, obtained from a subcollection of 1000 non-overlapping block bootstrap resamples that resulted in the selection of an AR model of order 13, for the time series of wave heights.

by computing the standard deviation of the 40 prediction errors, that is,

$$\sqrt{\frac{1}{40}\sum_{t=357}^{396}(y_t - \hat{y}_t)^2}, \tag{4.4}$$

where $\hat{y}_t$ are the one-step ahead forecasts. The results are displayed in Table 4.8.

Model	Root-Mean-Squared Prediction Error
AR(11)	119.3
AR(13)	116.1
Penalized AR(13)	115.0
AR(15)	118.2
Penalized subset AR(17)	119.0
ARMA(4,4)	119.3

Table 4.8: Root-mean-squared prediction error for the following models, fitted to the time series of wave heights: AR(11) (using ML estimation), AR(13) (using ML estimation), penalized AR(13) (using PAC-based nested lasso with the BIC-selector), AR(15) (using ML estimation), subset AR(17) (estimated by adaptive lasso) and ARMA(4,4).

From Table 4.8, it can be seen that the AR(13) model has the smallest root-mean-squared prediction error, but there is very little difference between the models. This can also be seen in Figure 4.3, where we plot the one-step-ahead forecasts for the observed wave heights in the test data set for each of the models. The AR models provide similar forecasts to the ARMA(4,4) model. As the forecasts corresponding to the AR models are very similar, the simpler AR(13) model is preferred over the AR(15) and the subset AR(17) models.

The estimated coefficients of the AR(13) model, fitted to the $n = 396$ observed wave heights, are displayed in Table 4.9 along with their standard errors. The estimated residual standard deviation is $\hat{\sigma}_e = 136.3$. An inspection of the ACF, PACF and histogram of the residuals, displayed in Figure 4.4 confirms the adequacy of the fitted model.

Coefficient	Estimate	Standard Error
ϕ_1	0.272	0.050
ϕ_2	-0.924	0.051
ϕ_3	-0.430	0.068
ϕ_4	-0.674	0.067
ϕ_5	-0.711	0.072
ϕ_6	-0.714	0.075
ϕ_7	-0.615	0.078
ϕ_8	-0.584	0.075
ϕ_9	-0.444	0.072
ϕ_{10}	-0.513	0.067
ϕ_{11}	-0.227	0.069
ϕ_{12}	-0.194	0.051
ϕ_{13}	-0.159	0.050

Table 4.9: Estimated coefficients with their corresponding standard errors for the AR(13) model, fitted to the time series of wave heights of length $n = 396$.

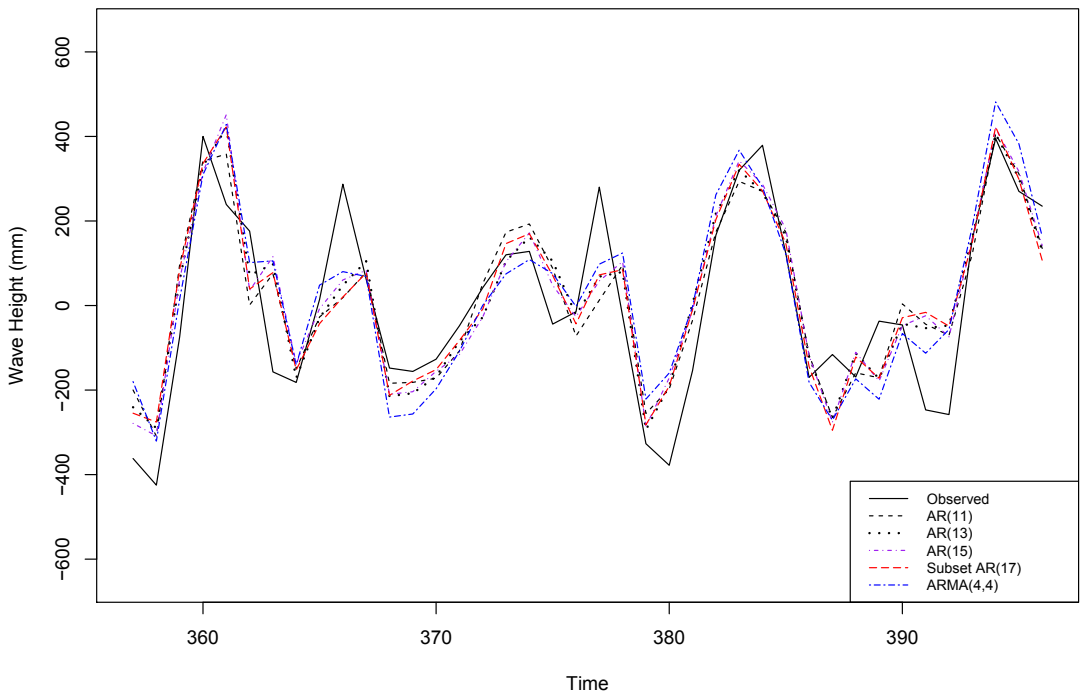

Figure 4.3: Plot of the one-step-ahead forecasts for the AR(11), AR(13), AR(15), subset AR(17) and ARMA(4,4) models over a period of 4 seconds.

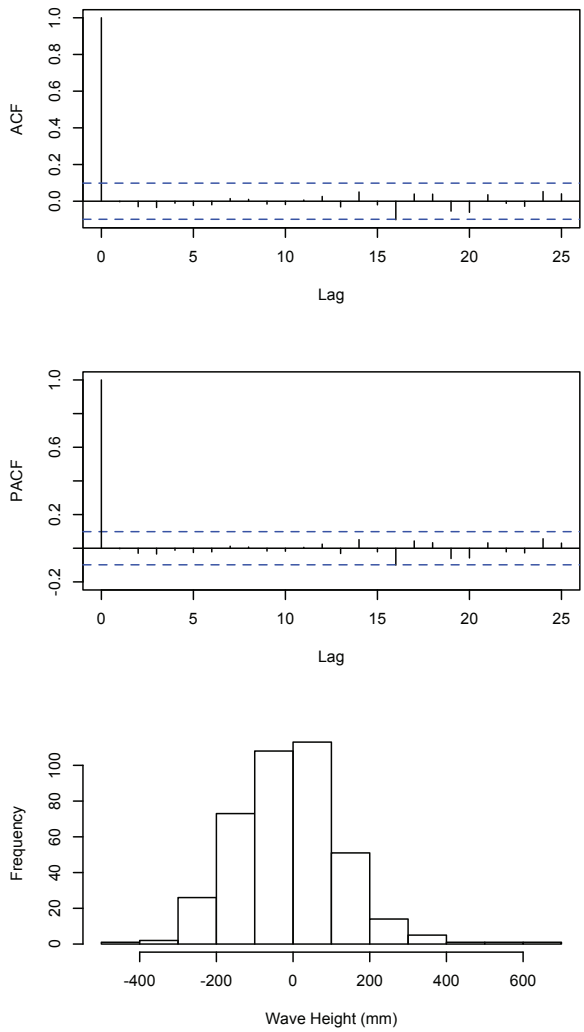

Figure 4.4: Residuals from the fitted AR(13) model to the time series of wave heights: ACF, PACF and histogram.

4.8 Discussion

In this chapter, we proposed penalized likelihood approaches for estimating the order of an autoregressive process based on the partial autocorrelation (PAC) parametrization. It is well known that for observations generated from an autoregressive process, the AR order is identified by the bandwidth of the partial autocorrelation matrix. Therefore, we estimate the bandwidth of the partial autocorrelation matrix using the nested lasso penalty of Levina et al. (2008). While the AR coefficients provide a convenient representation of the autoregressive process, we apply shrinkage to the partial autocorrelations instead as

they better reflect the temporal dependence structure of the AR process. Our proposed method with the nested lasso penalty performs best for estimating the order of non-subset AR models, where partial autocorrelations at all lags $j \leq p$ are non-zero. To handle the case where some PACs at intermediate lags are zero, we apply a lasso penalty to the PACs so that they are penalized independently of each other. Our simulations in Section 4.5 demonstrate that the PAC-based lasso method offers better performance over the lasso and modified lasso methods of Wang et al. (2007b), which are based on penalization of the AR coefficients, for both subset and non-subset AR models. Furthermore, when the true model is a non-subset AR model, it was found in simulation that the PAC-based nested lasso method performs better than various information criteria as well as the lasso and modified lasso methods of Wang et al. (2007b) with smaller percentages of overestimated AR orders. The theoretical properties of the proposed PAC-based methods remain for future work.

Chapter 5

Estimating Networks with Hubs from Microbiome Data

In Chapter 3 of this thesis, we focused on the problem of estimating a sparse precision matrix Θ in the case where variables are *ordered*. In this setting, it is reasonable to assume that the inverse Θ is *banded*. In this chapter, we study sparse inverse covariance estimation in the case where variables are *unordered*, which under multivariate normality, corresponds to estimating a graphical model for the data. We focus on the case where the underlying graphical model has *hubs*, which are highly connected nodes, inspired by a microbiome data application. Methods based on L_1-regularization are widely used for graph estimation. However, while the L_1 penalty encourages sparsity, it does not take into account any structural information. In this chapter, we introduce a new method for estimating networks with hubs that exploits the ability of (inverse) covariance selection methods to include structural information about the underlying network. We propose a weighted graphical lasso approach with novel row/column sum weights that take hub structure into account, which we refer to as the hubs weighted graphical lasso (HWGL). Some asymptotic results are established. Empirically, we then show that the HWGL procedure outperforms competing methods and illustrate the methodology with an application to microbiome data.

5.1 Introduction

A number of biological networks, such as gene regulatory and protein-protein interaction networks, display high-degree or densely connected nodes, called *hubs*. In this chapter, we study the problem of estimating high-dimensional networks with hubs, focusing on an application to microbiome data. We begin by providing some background on the microbiome data application in Section 5.1.1. We then introduce the problem of estimating networks with hubs and briefly review related work in Section 5.1.2.

5.1.1 Motivating Example: Estimating Microbiota Networks

Rapidly developing sequencing technologies and analytical techniques have enhanced our ability to study the microorganisms (such as bacteria, viruses, archaea and fungi) that inhabit the human body as well as a wide range of environments. Large-scale initiatives to analyze microbial communities, such as the Earth Microbiome Project (Gilbert et al., 2010) and the Human Microbiome Project (Turnbaugh, et al., 2007), have made available to the public a growing number of samples from soil, marine, plant, animal and human-associated microbiota.

The microorganisms inhabiting a particular environment do not exist in isolation, but interact with other microorganisms in a range of mutualistic and antagonistic relationships. Beneficial interactions can arise due to reasons such as cross-feeding (which involves the exchange of metabolic products between species) and co-colonization, while harmful interactions can arise due to prey-predator relationships and nutrient competition (Faust and Raes, 2012). One goal of microbiome studies is to model these microbial interactions from population-level data as a network reflecting co-occurrence and co-exclusion patterns between microbial taxa. This is of interest not only for predicting individual relationships between microbes, but the structure of the interaction networks also gives insight into the organization of complex microbial communities. Accurate inference of microbial interaction networks will be key to answering several other questions arising in microbiome studies. In particular, one area of inquiry requiring further elucidation is

the outcome of host-microbe interactions on human health and disease. Recent studies have revealed that microbiome composition and structure varies based on health, diet and environment, and may play a key role in diseases such as obesity (Turnbaugh et al., 2009) and Crohn's disease (Gevers et al., 2014) as well as chronic malnutrition among children (Gough et al., 2015). Therefore, the goal of this chapter is to introduce methodology for accurately reconstructing a microbial interaction network that can also be used in downstream statistical analyses.

In recent studies, networks of pairwise correlations between microbial taxa have been used to model microbe-microbe interactions (Friedman et al., 2012). In this representation, nodes are microbial taxa and an edge between two nodes represents a non-zero association between two taxa. Under multivariate normality, these links represent marginal dependence relationships between taxa. As a pairwise measure of dependence, however, correlation can be limiting in the multivariate setting. As an alternative to computing pairwise correlations, some authors have utilized an approach that estimates a sparse inverse covariance matrix from relative abundance data (Gough et al., 2015; Kurtz et al., 2015), which will also be done in this chapter. Under multivariate normality, non-zero elements in the inverse represent conditional dependence relationships between two taxa and correspond to edges in an undirected graphical model.

Statistical challenges in modelling these graphical models of microbial interactions arise due to data scarcity and the organization of the network's nodes into groups with different levels of connectivity. Specifically, microbial association networks tend to display *hubs*. In ecology, these hubs can represent a few "keystone" species that are vital in maintaining stability of the microbial community (Kurtz et al., 2015).

5.1.2 Estimating Networks with Hubs

When it comes to graph estimation in the sample-starved scenario, methods based on L_1 regularization are widely used, the most popular being the graphical lasso of Friedman et al. (2008). The L_1 penalty, however, implicitly assumes that each edge is equally likely and independent of all other edges, and is therefore inadequate for modelling networks

with a few high-degree nodes in the presence of many low-degree nodes. To accommodate such structures, Tan et al. (2014) had proposed the hubs graphical lasso (HGL), which is a penalization method that encourages solutions of the form $\Theta = \mathbf{Z}+\mathbf{V}+\mathbf{V}^T$, where $\mathbf{Z}$ is a sparse symmetric matrix capturing edges between non-hub nodes and $\mathbf{V}$ is a matrix whose columns are either entirely zero or almost entirely non-zero with the non-zero elements of $\mathbf{V}$ representing hub edges. Their method applies an L_1 penalty to the off-diagonal elements of $\mathbf{Z}$, and L_1 and group lasso (Yuan and Lin, 2007) penalties to the columns of $\mathbf{V}$. However, their method requires a lot of tuning: three tuning parameters are present in their penalized likelihood as well as a user-specified parameter in their BIC-type quantity which is used for tuning parameter selection, and is designed for networks with very densely connected nodes (referred to as "super hubs"). Other authors have proposed methods specifically for estimating scale-free networks (e.g., Liu and Ihler, 2011), which are characterized as having a degree distribution that follows a power law.

In this chapter, we introduce a simpler and more general approach for estimating networks with hubs that exploits the ability of (inverse) covariance selection methods to include structural information about the underlying network. Our proposed method, called the hubs weighted graphical lasso (HWGL), is a weighted graphical lasso approach with informative row/column sum weights that allow for less penalization of hub edges compared to non-hub edges.

Sparse network selection methods have been widely applied to genomic data sets, but are scarcely applied to microbiomic data sets. Thus, this chapter explores a novel application of the statistical methodology developed for modelling high-dimensional networks. In Section 5.2, we continue our discussion of the microbiome data application and provide a brief literature review of methods for estimating networks with hubs. In Section 5.3, we then present the hubs weighted graphical lasso procedure and investigate its theoretical properties in Section 5.4. Simulation studies are provided in Section 5.5, followed by an application to two microbiome data sets in Section 5.6. We then conclude with a discussion in Section 5.7.

5.2 Network Estimation from Microbial Abundance Data

The prediction of microbial associations from abundance data is a network inference problem. These microbe-microbe interactions can be modelled as undirected graphical models. Therefore, we estimate the inverse correlation structure of (transformed) abundance data with the rationale that the inverse correlation structure describes the interactions between microbes that give rise to the observed distribution of abundances.

Estimating graphical models from microbial abundance data poses some technical challenges. First, the relative abundances are compositional, as the counts are normalized to the total number of counts in the sample, and performing conventional correlation analysis may lead to biased results. Second, these networks of microbial interactions tend to be highly structured. In particular, they exhibit many taxa with only a small number of connections and a few highly connected taxa (or hubs). Therefore, like social networks, microbial interaction networks may be scale-free (Faust and Raes, 2012). The networks can also be partitioned into clusters, which are groups of densely interconnected nodes, with only a few connections between clusters. Therefore, the procedures used to model these interaction networks must be able to accommodate highly connected nodes and clustering.

In Section 5.2.1, we address the first of these challenges by reviewing one transformation proposed by Aitchison (1981) for dealing with compositional data. In Section 5.2.2, we then provide a review of existing procedures for estimating networks with hubs.

5.2.1 Transforming Microbial Abundance Data

The collection of samples from microbiomes across a wide range of environments is routinely done using 16S rRNA gene sequencing. In a typical study, bacterial DNA is isolated, and 16S rRNA genes are amplified, sequenced and resulting reads are aligned for the identification of microbial taxa. The 16S rRNA read counts are used as a proxy for taxon abundance and are set by sequencing depth or the amount of genetic material extracted from the community (Friedman and Alm, 2012). Counts $y_1, \ldots, y_p$ are then

normalized by the total number of counts $m = \sum_{j=1}^{p} y_j$ in the sample and the resulting proportions $w_1 = y_1/m, \ldots, w_p = y_p/m$ are compositional as they are constrained to lie within the unit simplex

$$\mathcal{S}^p = \left\{ \mathbf{w} = (w_1, \ldots, w_p) : w_j > 0, \sum_{j=1}^{p} w_j = 1 \right\}.$$

Classical correlation analysis from compositional data, however, can lead to spurious results as proportions tend to be correlated even if the absolute abundances are independent. To overcome the unit-sum constraint of compositional data, log-ratio transformations, proposed by Aitchison (1981), can be used. Here we apply the centered log-ratio (clr) transform

$$\mathbf{x} = \text{clr}(\mathbf{w}) = \left(\log \frac{w_1}{g(\mathbf{w})}, \ldots, \log \frac{w_p}{g(\mathbf{w})} \right), \tag{5.1}$$

where $g(\mathbf{x}) = \left(\prod_{j=1}^{p} w_j \right)^{1/p}$ is the geometric mean of the vector of proportions $\mathbf{w}$ and the components of $\mathbf{x}$ are constrained to sum to zero. The clr transform maps the data isometrically from the unit simplex to a $(p-1)$-dimensional Euclidean vector subspace.

The corresponding covariance matrix $\Sigma = \text{Cov}\{\text{clr}(W)\}$ of the clr-transformed relative abundances is symmetric, but it is also singular. If $\Gamma = \text{Cov}(\log Y)$ is the covariance matrix of the log-transformed abundances, then Σ is related to Γ as follows

$$\Sigma = G\Gamma G,$$

where $G = I_p - \frac{1}{p} J_p$ and J_p is the $p \times p$ matrix of units (Aitchison, 2003). Therefore, for p large, $G \approx I_p$ and an estimate $\hat{\Sigma}$ of Σ can be used as an approximation of $\hat{\Gamma}$ (Kurtz et al., 2015).

5.2.2 Existing Methods for Estimating Networks with Hubs

While maximization of the L_1-penalized likelihood has been a widely used approach for estimating graphical models, it does not typically yield an estimate with hubs. The L_1

penalty applied to the Gaussian log-likelihood can be viewed as an independent double-exponential prior on each edge. Thus, the use of an L_1 penalty assumes that each edge is equally likely and independent of all other edges (Tan et al., 2014). In what follows, we discuss existing methods in the literature for estimating graphical models with hubs that take hub structure into account, such as the hubs graphical lasso (HGL) of Tan et al. (2014) and the reweighted L_1 regularization approach of Liu and Ihler (2011).

Sparse Partial Correlation Estimation (SPACE; Peng et al., 2009):

Peng et al. (2009) proposed a procedure called `space` (Sparse PArtial Correlation Estimation) that is an extension of the neighourhood selection approach of Meinshausen and Bühlmann (2006), where a lasso regression is performed separately for each variable on the rest of the variables. It is designed to address two limitations of the neighourhood selection approach of Meinshausen and Bühlmann (2006). First, `space` employs the symmetry among the partial correlations, which is not done so by neighbourhood selection, resulting in a loss of efficiency. Second, unlike the neighbourhood selection approach of Meinshausen and Bühlmann (2006), their method uses different tuning parameters for the p lasso regressions, making it easy to incorporate prior knowledge about network structure. The authors claim that the degree-reweighted version of `space` performs well in estimating scale-free networks. Since the introduction of the `space` procedure, other regression-based graphical model selection methods have been proposed, such as the symmetric lasso (Friedman et al., 2010) and the CONvex CORrelation selection methoD (CONCORD; Khare et al., 2015).

Power Law Regularization (Liu and Ihler, 2011):

Liu and Ihler (2011) proposed a method for estimating scale-free networks, which are characterized as having a degree distribution that follows a power law: $p(d) \propto d^{-\alpha}$. They estimate Θ by solving the following non-convex optimization problem

$$\underset{\Theta \succ 0}{\arg\min} \left\{ -\log\det\Theta + \operatorname{tr}(S\Theta) + \alpha \sum_{j=1}^{p} \log\left(\|\theta_{\setminus j}\| + \epsilon_j\right) + \sum_{j=1}^{p} \beta_j |\theta_{jj}| \right\},$$

where $\theta_{\setminus j} = \{\theta_{jj'} : j' \neq j\}$, and α, ϵ_j and β_j are tuning parameters. They use $\|\theta_{\setminus j}\|_1 + \epsilon_j$ as a continuous surrogate of the degree d for $\epsilon_j > 0$.

To solve this non-convex optimization problem, they use an MM (majorize-minimization) algorithm, and recast this problem as a sequence of reweighted L_1 regularization problems.

Hub Screening Procedure of Hero and Rajaratnam (2012):

Hero and Rajaratnam (2012) proposed a hub screening procedure that involves thresholding the elements of the sample partial correlation matrix (computed as the Moore-Penrose pseudo-inverse of the sample correlation matrix when $p > n$) based on a z-score representation, where a node is declared a hub if the number of non-zero elements in the corresponding row/column of the thresholded partial correlation matrix is sufficiently large. In their hub screening framework, the user must specify both a minimum partial correlation ρ and a minimum node degree δ. It should be emphasized that this method is designed for hub screening only and does not estimate the edges of the network. Its advantage is its low computational complexity when $p \gg n$.

Hubs Graphical Lasso (Tan et al., 2014):

Tan et al. (2014) considered the problem of studying high-dimensional networks with hub nodes. Rather than using an L_1 penalty on the elements of the precision matrix Θ, the authors introduced a new penalty to accommodate these densely connected nodes and referred to their procedure as the hubs graphical lasso (HGL). They proposed a penalty that encourages a solution of the form $\Theta = \mathbf{Z} + \mathbf{V} + \mathbf{V}^T$, where $\mathbf{Z}$ is a sparse symmetric matrix and $\mathbf{V}$ is a matrix whose columns are either entirely zero or almost entirely non-zero. The non-zero entries in $\mathbf{Z}$ represent edges between non-hub nodes, while the non-zero columns of $\mathbf{V}$ correspond to the edges connecting hubs to other nodes in the network. The authors proposed to estimate Θ by solving the following optimization problem

$$\underset{\Theta \succ 0:\ \Theta = V + V^T + Z}{\arg\min} \left\{ \ell(\Theta; \mathbf{X}) + \rho_1 \|\mathbf{Z}\|_1 + \rho_2 \|\mathbf{V}\|_1 + \rho_3 \sum_{j=1}^{p} \|\mathbf{V}_j\|_2 \right\},$$

which leads to estimation of a network with dense hubs, and where $\ell(\Theta; \mathbf{X}) = -\log\det\Theta + \text{tr}(S\Theta)$ is the negative log-likelihood of the data. Therefore, an L_1 penalty is imposed on the elements of $\mathbf{Z}$ and $\mathbf{V}$ as well as a group lasso penalty on the columns of $\mathbf{V}$ so that each column of $\mathbf{V}$ is either very dense or contains all zero elements. Depending on the tuning parameter ρ_3, many elements in the same column may be removed. Thus, sparsity in $\mathbf{Z}$ is controlled by ρ_1, ρ_2 controls the number of edges connecting hub nodes to other nodes in the network, and ρ_3 controls the selection of hub nodes.

To solve the resulting convex optimization problem, the authors used an alternative direction method of multipliers (ADMM) algorithm. To select the tuning parameters (ρ_1, ρ_2, ρ_3), they considered a BIC-type quantity, given by

$$\text{BIC}^*(\widehat{\Theta}, \widehat{\mathbf{V}}, \widehat{\mathbf{Z}}) = -\log\det\widehat{\Theta} + \text{tr}(S\widehat{\Theta}) + \frac{\log n}{n}|\widehat{\mathbf{Z}}| + \frac{\log n}{n}\left\{\nu + c(|\widehat{\mathbf{V}}| - \nu)\right\}, \qquad (5.2)$$

where $\nu = \sum_{j=1}^{p} 1_{\left\{\|\widehat{\mathbf{V}}_j\|_0 > 0\right\}}$ is the number of estimated hub nodes, $|\mathbf{S}|$ denotes the number of non-zero entries of the matrix $\mathbf{S}$, and $0 < c < 1$ is a user-specified parameter, controlling the number of hub nodes. The set of tuning parameters (ρ_1, ρ_2, ρ_3) for which $\text{BIC}^*(\widehat{\Theta}, \widehat{\mathbf{V}}, \widehat{\mathbf{Z}})$ is minimized are then selected.

5.3 A New Method for Estimating Networks with Hubs

In this section, we present a new procedure for estimating networks with hub nodes. Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent and identically distributed multivariate normal random vectors with mean $\mathbf{0}$ and covariance matrix $\Sigma = \Theta^{-1}$. The sparsity pattern of Θ determines the conditional independence graph. Since the underlying graph has a few hub nodes, the rows/columns of the precision matrix Θ corresponding to the hub nodes are significantly denser than those corresponding to the non-hub nodes. We adopt a weighted lasso approach that uses more informative weights compared to those in the standard adaptive lasso (Zou, 2006; Fan et al., 2009), based on row/column sums. In what follows, we outline our proposed estimation procedure.

Let $\widehat{\Theta}_0 = (\hat{\theta}_{ij}^{(0)})$ be any consistent estimate of the inverse covariance matrix Θ. For

$n > p$, we take $\widehat{\Theta}_0$ to be the inverse of the sample covariance matrix S. For $n < p$, we use the inverse covariance matrix estimate derived from the graphical lasso. Based on the consistent estimate $\widehat{\Theta}_0$, we construct the symmetric matrix $W^{(1)}$ of weights

$$w_{ij}^{(1)} = \begin{cases} \dfrac{1}{|\hat{\theta}_{ij}^{(0)}|^{\gamma_1} \left(\sum\limits_{\substack{k=1 \\ k \neq i}}^{p} |\hat{\theta}_{ik}^{(0)}| \cdot \sum\limits_{\substack{k=1 \\ k \neq j}}^{p} |\hat{\theta}_{kj}^{(0)}| \right)^{\gamma_2}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

for some γ_1, $\gamma_2 > 0$, and define the hubs weighted graphical lasso (HWGL) estimator $\widehat{\Theta}$ of Θ to be

$$\widehat{\Theta} = \underset{\Theta \succ 0}{\arg\max} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \lambda \|W^{(1)} * \Theta\|_1 \right\}, \tag{5.3}$$

where $\lambda > 0$ is a tuning parameter, $*$ denotes Schur matrix product (entrywise multiplication), and $\|\cdot\|_1$ is the L_1 norm (the sum of the absolute values of the elements of $W^{(1)} * \Theta$).

Remarks:

- The weights are designed to allow for less penalization of hub edges compared to non-hub edges. If nodes i and j are hubs, then both $\sum_{k \neq i} |\hat{\theta}_{ik}^{(0)}|$ and $\sum_{k \neq j} |\hat{\theta}_{kj}^{(0)}|$ should be large and therefore the weight $w_{ij}^{(1)}$ will be small. If nodes i and j are non-hubs, then both $\sum_{k \neq i} |\hat{\theta}_{ik}^{(0)}|$ and $\sum_{k \neq j} |\hat{\theta}_{kj}^{(0)}|$ should be small and therefore the weight $w_{ij}^{(1)}$ will be large. If either nodes i or j are hubs, then one of $\sum_{k \neq i} |\hat{\theta}_{ik}^{(0)}|$ and $\sum_{k \neq j} |\hat{\theta}_{kj}^{(0)}|$ should be large and therefore the weight $w_{ij}^{(1)}$ will be moderately sized. When $\gamma_1 > 0$, the additional term $|\hat{\theta}_{ij}^{(0)}|^{\gamma_1}$ in the weights is included to allow for zero entries in columns corresponding to hubs.

- This approach belongs to the family of weighted lasso methods that allow for different penalties on the entries of Θ, which includes the adaptive lasso (Fan et al., 2009). Weighted lasso approaches can result in less bias than the standard lasso by adapting penalties to incorporate information about the location of zeros, based on either an

initial estimate or background knowledge.

- The weights of the HWGL estimator when $p > n$ are constructed from the graphical lasso estimator of Θ due to its estimation consistency. In finite sample, we have found that weights constructed from the ridge-type estimate $\hat{\Sigma}_0 = S + \nu I_p$ of the covariance matrix for some $\nu > 0$, chosen so that $\hat{\Sigma}_0$ is positive definite, tend to yield better performance.

Finite sample improvement through a two-step approach: In Section 5.4, we show that the HWGL estimator is estimation consistent and selection consistent in the fixed dimensional setting, and acts as a finite sample correction to the adaptive lasso (Zou, 2006; Fan et al., 2009) when the true underlying graph has hub nodes. We further observe that better finite sample performance can be obtained by first identifying a set of candidate hub nodes $\hat{H}$ based on the HWGL estimate $\widehat{\Theta}$, allowing for edges to be classified as *hub* or *non-hub* edges, and then penalizing the hub edges separately from the non-hub edges through a second weighted lasso. We outline this approach in what follows.

Based on the one-step HWGL estimate $\widehat{\Theta}$, defined in (5.3), we identify a set of candidate hub nodes $\hat{H}$. We then construct a symmetric weight matrix $W^{(2)} = (w_{ij}^{(2)})$, where

$$w_{ij}^{(2)} = \begin{cases} \lambda_1 & \text{if } i \in \hat{H} \text{ or } j \in \hat{H}, i \neq j \\ \lambda_2 & \text{if } i, j \notin \hat{H}, i \neq j \\ 0 & \text{if } i = j \end{cases}$$

for some tuning parameters $\lambda_1, \lambda_2 > 0$, and solve the weighted lasso optimization problem

$$\tilde{\Theta} = \underset{\Theta \succ 0}{\arg\max} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) - \|W^{(2)} * \Theta\|_1 \right\},$$

where we refer to $\tilde{\Theta}$ as the two-step hubs weighted graphical lasso (HWGL_2) estimator of Θ. The tuning parameter λ_1 controls the number of edges connecting a hub node to any other node in the graph, while the tuning parameter λ_2 controls the number of edges connecting two non-hub nodes.

The tuning parameters λ, λ_1 and λ_2 are chosen using a data-dependent approach. In our simulation studies in Section 5.5, we employ BIC due to its model selection consistency property, which has been established in a variety of settings.

Remarks:

- The set of candidate hub nodes $\hat{H}$ can be obtained by setting a cutoff threshold for a node to be a hub (e.g., in our simulations, we classify a node as a hub if it is connected to at least 10% of all other nodes). The hub findings will then depend on the choice of the cutoff threshold used for classifying a node as a hub. To avoid specifying such a threshold, a clustering approach can be used to classify the nodes based on the first-step estimate $\widehat{\Theta}$ into hub and non-hub groups. From the first-step estimate $\widehat{\Theta}$, the degree of each node can be computed and K-means clustering can then be applied to cluster the nodes into hub and non-hub groups, where the hub group is characterized as the group with the larger mean degree. A similar approach was considered by Charbonnier et al. (2010) in order to cluster nodes in a *directed* graph as hubs and leaves. They had used a Gaussian mixture approach based on the L_1 norms of the columns of the initial estimate of the parameters in the vector autoregressive model of order 1.

5.4 Theoretical Properties

In this section, we study the asymptotic properties of the hubs weighted graphical lasso estimator $\widehat{\Theta}$. We assume that $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are p-dimensional multivariate normal random vectors with mean $\mathbf{0}$ and true covariance matrix Σ_0. The corresponding true precision matrix is $\Theta_0 = \Sigma_0^{-1}$ and the sample covariance matrix is $S = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T / n$. Further, we define the sets

$$\mathcal{A} = \{(i,j) : \theta_{ij,0} \neq 0, i \leq j\} \quad \text{and} \quad \mathcal{B} = \{(i,j) : \theta_{ij,0} = 0, i \leq j\}\,.$$

Therefore, $\mathcal{A}$ is the set of indices of the true non-zero elements in Θ_0 and $\mathcal{B}$ is the set of indices of the true zero elements of Θ_0. We also assume that dimension p is held fixed as

the sample size $n \to \infty$. We show that the HWGL estimator possesses the *oracle property* (Fan and Li, 2001).

Theorem 2. (Oracle property of the hubs weighted graphical lasso estimator) Let $a_n = n^{-1/2} \min_{(i,j)\in\mathcal{A}} \left(|\hat{\theta}_{ij}^{(0)}|^{\gamma_1} D_{ij}^{\gamma_2} \right)$ and $b_n = n^{-1/2} \max_{(i,j)\in\mathcal{B}} \left(|\hat{\theta}_{ij}^{(0)}|^{\gamma_1} D_{ij}^{\gamma_2} \right)$, where $D_{ij} = \sum_{k\neq i} |\hat{\theta}_{ik}^{(0)}| \cdot \sum_{k\neq j} |\hat{\theta}_{kj}^{(0)}|$. If $\lambda_n/a_n \xrightarrow{p} 0$ and $\lambda_n/b_n \xrightarrow{p} \infty$, then $\widehat{\Theta} = (\hat{\theta}_{ij})$ must satisfy

(i) *Selection Consistency:* The HWGL estimator $\widehat{\Theta}$ has the same sparsity pattern, asymptotically, as the true precision matrix Θ_0; that is, $P\left(\hat{\theta}_{ij} = 0\right) \to 1$ as $n \to \infty$ for $(i,j) \in \mathcal{B}$.

(ii) *Asymptotic Normality:* For $(i,j) \in \mathcal{A}$, the entries $\hat{\theta}_{ij}$ are $\sqrt{n}$-consistent and asymptotically normal.

Proof: Before proving Theorem 2, we introduce some notation. As in Fan et al. (2009), we write Θ as a vector of length $d = p(p+1)/2$ by taking $\boldsymbol{\psi} = (\boldsymbol{\psi}_1, \boldsymbol{\psi}_2)$, where $\boldsymbol{\psi}_1 = (\theta_{ij} : (i,j) \in \mathcal{A})$ and $\boldsymbol{\psi}_2 = (\theta_{ij} : (i,j) \in \mathcal{B})$. The precision matrix Θ can be viewed as a function of $\boldsymbol{\psi}$: $\Theta = \Theta(\boldsymbol{\psi})$. We let $\boldsymbol{\psi}_0$ denote the true value of $\boldsymbol{\psi}$, which we can write as $\boldsymbol{\psi}_0 = (\boldsymbol{\psi}_{10}, \boldsymbol{\psi}_{20})$ with $\boldsymbol{\psi}_{20} = \mathbf{0}$ and where $\boldsymbol{\psi}_{10} \neq \mathbf{0}$ has length s.

The log-likelihood function of $\boldsymbol{\psi}$ is given by

$$\begin{aligned}\ell_n(\boldsymbol{\psi}) &= \frac{1}{2} \sum_{i=1}^{n} \left\{ \log |\Theta(\boldsymbol{\psi})| - \log(2\pi) - \mathbf{x}_i^T \Theta(\boldsymbol{\psi}) \mathbf{x}_i \right\} \\ &= \frac{n}{2} \log |\Theta(\boldsymbol{\psi})| - \frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^{n} \mathbf{x}_i^T \Theta(\boldsymbol{\psi}) \mathbf{x}_i\end{aligned}$$

and we let $I(\boldsymbol{\psi}) = \mathbb{E}\left[\left\{ \frac{\partial}{\partial \boldsymbol{\psi}} \ell_n(\boldsymbol{\psi}) \right\} \left\{ \frac{\partial}{\partial \boldsymbol{\psi}} \ell_n(\boldsymbol{\psi}) \right\}^T \right]$ be the Fisher information matrix of $\boldsymbol{\psi}$. We assume regularity conditions as in (A)-(C) of Fan and Li (2001).

(i) Let R_j denote the set of indices k $(1 \leq k \leq d)$ of all elements in the row indexed by the row of the precision matrix entry represented by ψ_j. Also, let C_j denote the set of indices k $(1 \leq k \leq d)$ of all elements in the column indexed by the column of the

precision matrix entry represented by ψ_j. Define the penalized log-likelihood function

$$Q(\boldsymbol{\psi}) = \ell_n(\boldsymbol{\psi}) - n\lambda_n \sum_{j=1}^{d} \hat{w}_j |\psi_j|, \tag{5.4}$$

where $\hat{w}_j = |\hat{\psi}_j^{(0)}|^{-\gamma_1} D_j^{-\gamma_2}$ with $D_j = \sum_{k \in C_j} |\hat{\psi}_k^{(0)}| \cdot \sum_{k \in R_j} |\hat{\psi}_k^{(0)}|$ for some initial $\sqrt{n}$-consistent estimate $\hat{\boldsymbol{\psi}}^{(0)} = (\hat{\psi}_1^{(0)}, \ldots, \hat{\psi}_d^{(0)})$ of $\boldsymbol{\psi}$.

First, we establish estimation consistency of $\hat{\boldsymbol{\psi}}$. Following Fan and Li (2001), we want to show that for any given $\epsilon > 0$, there exists a large constant C such that

$$P\left\{ \sup_{\|\mathbf{u}\|=C} Q(\boldsymbol{\psi}_0 + n^{-1/2}\mathbf{u}) < Q(\boldsymbol{\psi}_0) \right\} \geq 1 - \epsilon,$$

which implies that with probability at least $1 - \epsilon$ there exists a maximum in the ball $\left\{\boldsymbol{\psi}_0 + n^{-1/2}\mathbf{u} : \|\mathbf{u}\| \leq C\right\}$. Hence, there exists a local maximizer such that $\|\hat{\boldsymbol{\psi}} - \boldsymbol{\psi}_0\| = O_p(n^{-1/2})$.

Since only the first s elements of $\boldsymbol{\psi}_0$ are non-zero, we find that

$$\begin{aligned}
D_n(\mathbf{u}) &= Q(\boldsymbol{\psi}_0 + n^{-1/2}\mathbf{u}) - Q(\boldsymbol{\psi}_0) \\
&= \ell_n(\boldsymbol{\psi}_0 + n^{-1/2}\mathbf{u}) - \ell_n(\boldsymbol{\psi}_0) - n\lambda_n \sum_{j=1}^{s} \hat{w}_j \left(|\psi_{j0} + n^{-1/2}u_j| - |\psi_{j0}|\right) \\
&\leq n^{-1/2}\ell_n'(\boldsymbol{\psi}_0)^T\mathbf{u} - \frac{1}{2}\mathbf{u}^T I(\boldsymbol{\psi}_0)\mathbf{u}\left\{1 + o_p(1)\right\} - n\lambda_n \sum_{j=1}^{s} \hat{w}_j \left(|\psi_{j0} + n^{-1/2}u_j| - |\psi_{j0}|\right) \\
&\leq n^{-1/2}\ell_n'(\boldsymbol{\psi}_0)^T\mathbf{u} - \frac{1}{2}\mathbf{u}^T I(\boldsymbol{\psi}_0)\mathbf{u}\left\{1 + o_p(1)\right\} + n^{1/2}\lambda_n \sum_{j=1}^{s} \hat{w}_j |u_j|.
\end{aligned} \tag{5.5}$$

For the first term in (5.5), $n^{-1/2}\ell_n'(\boldsymbol{\psi}_0) = O_p(1)$. Consider the third term in (5.5), which can be expressed as

$$\begin{aligned}
n^{1/2}\lambda_n \sum_{j=1}^{s} \hat{w}_j |u_j| &= n^{1/2}\lambda_n \sum_{j=1}^{s} \left(|\hat{\psi}_j^{(0)}|^{-\gamma_1} D_j^{-\gamma_2} |u_j|\right) \\
&\leq n^{1/2}\lambda_n \left\{ \min_{1 \leq j \leq s} \left(|\hat{\psi}_j^{(0)}|^{\gamma_1} D_j^{\gamma_2}\right) \right\}^{-1} \|\mathbf{u}\| \\
&= \frac{\lambda_n}{a_n}\|\mathbf{u}\| = o_p(1)\|\mathbf{u}\|.
\end{aligned}$$

Finally, the second term in (5.5) is a quadratic term in $\mathbf{u}$. Therefore, by choosing a sufficiently large C, the quadratic term will dominate the other terms with probability $\geq 1-\epsilon$.

Note: For $1 \leq j \leq s$, define $S_j = \{k \in C_j : \psi_{k0} \neq 0\}$ and $S_j^c = \{k \in C_j : \psi_{k0} = 0\}$. Then $\sum_{k\in C_j} |\hat{\psi}_k^{(0)}| = \sum_{k\in S_j} |\hat{\psi}_k^{(0)}| + \sum_{k\in S_j^c} |\hat{\psi}_k^{(0)}|$. Now since $\hat{\psi}_k^{(0)} \xrightarrow{p} \psi_{k0}$ for each $k \in S_j$, we have that $\sum_{k\in S_j} |\hat{\psi}_k^{(0)}| = O_p(1)$. Further, $n^{1/2}|\hat{\psi}_k^{(0)}| = O_p(1)$ for $k \in S_j^c$ by $\sqrt{n}$-consistency of $\hat{\boldsymbol{\psi}}^{(0)}$. Thus, $\sum_{k\in C_j} |\hat{\psi}_k^{(0)}| = O_p(1)$. Similarly, $\sum_{k\in R_j} |\hat{\psi}_k^{(0)}| = O_p(1)$. Now since $\hat{\psi}_j^{(0)} \xrightarrow{p} \psi_{j0}$ for $1 \leq j \leq s$, $|\hat{\psi}_j^{(0)}|^{\gamma_1} D_j^{\gamma_2} = O_p(1)$. Hence, $a_n = n^{-1/2} O_p(1)$ and so the condition $\lambda_n / a_n \xrightarrow{p} 0$ holds if the condition $n^{1/2}\lambda_n \xrightarrow{p} 0$ is satisfied. Thus, the graphical lasso, adaptive lasso and hubs weighted graphical lasso estimators are able to achieve consistency in estimation when $\lambda_n = o_p(n^{-1/2})$, but their performance in finite sample may differ.

We showed that the local maximizer $\hat{\boldsymbol{\psi}}$ of the penalized log-likelihood function $Q(\boldsymbol{\psi})$ is a $\sqrt{n}$-consistent estimator of $\boldsymbol{\psi}$. Therefore, the local maximizer $\hat{\boldsymbol{\psi}} = (\hat{\boldsymbol{\psi}}_1, \mathbf{0})$ also has the property that $\|\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\psi}_{10}\| = O_p(n^{-1/2})$.

Now to show sparsity, i.e. $\hat{\boldsymbol{\psi}}_2 = \mathbf{0}$, it suffices to show that with probability approaching 1, for any $\boldsymbol{\psi}_1$ satisfying $\|\boldsymbol{\psi}_1 - \boldsymbol{\psi}_{10}\| = O_p(n^{-1/2})$ and any constant C,

$$Q\left\{\begin{pmatrix}\boldsymbol{\psi}_1 \\ \mathbf{0}\end{pmatrix}\right\} = \max_{\|\boldsymbol{\psi}_2\| \leq Cn^{-1/2}} Q\left\{\begin{pmatrix}\boldsymbol{\psi}_1 \\ \boldsymbol{\psi}_2\end{pmatrix}\right\}. \tag{5.6}$$

To show (5.6), it suffices to show that for any $\boldsymbol{\psi}_1$ satisfying $\|\boldsymbol{\psi}_1 - \boldsymbol{\psi}_{10}\| = O_p(n^{-1/2})$, $\frac{\partial Q(\boldsymbol{\psi})}{\partial \psi_j}$ and ψ_j have different signs for $\psi_j \in (-Cn^{-1/2}, Cn^{-1/2})$ with $j = s+1, \ldots, d$. We have that

$$\begin{aligned} n^{1/2}\lambda_n \hat{w}_j &= \lambda_n \left(n^{1/2}|\hat{\psi}_j^{(0)}|^{-\gamma_1} D_j^{-\gamma_2}\right) \\ &\geq \frac{\lambda_n}{\max\limits_{s+1\leq j\leq d}\left(n^{-1/2}|\hat{\psi}_j^{(0)}|^{\gamma_1} D_j^{\gamma_2}\right)} \\ &= \frac{\lambda_n}{b_n} \xrightarrow{p} \infty \text{ for } j = s+1, \ldots, d. \end{aligned}$$

Thus, $n^{1/2}\lambda_n\hat{w}_j \xrightarrow{p} \infty$ for $j = s+1, \ldots, d$, and

$$\begin{aligned}\frac{\partial Q(\boldsymbol{\psi})}{\partial \psi_j} &= \frac{\partial \ell_n(\boldsymbol{\psi})}{\partial \psi_j} - n\lambda_n \mathrm{sgn}(\psi_j)\hat{w}_j \\ &= O_p(n^{1/2})\left\{O_p(1) - n^{1/2}\lambda_n \hat{w}_j \mathrm{sgn}(\psi_j)\right\}. \end{aligned} \tag{5.7}$$

Therefore, since $n^{1/2}\lambda_n\hat{w}_j \xrightarrow{p} \infty$ as $n \to \infty$ for $j = s+1, \ldots, d$, the sign of $\frac{\partial Q(\boldsymbol{\psi})}{\partial \psi_j}$ is completely determined by that of ψ_j when n is large.

(ii) Applying Theorem 2 (i), we have that $P(\hat{\boldsymbol{\psi}}_2 = \mathbf{0}) \to 1$ as $n \to \infty$. Hence, the maximizer of $Q(\boldsymbol{\psi})$ is the same as that of $Q\left\{\begin{pmatrix}\boldsymbol{\psi}_1 \\ \mathbf{0}\end{pmatrix}\right\}$ with probability tending to 1. This implies that the penalized estimator $\hat{\boldsymbol{\psi}}_1$ satisfies the equation

$$\begin{aligned} 0 &= \frac{\partial Q(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}_1}|_{\boldsymbol{\psi}=(\hat{\boldsymbol{\psi}}_1^T, \mathbf{0}^T)^T} \\ &= \frac{\partial \ell_n(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}_1}|_{\boldsymbol{\psi}=(\hat{\boldsymbol{\psi}}_1^T, \mathbf{0}^T)^T} - n\lambda_n \left(\hat{w}_1 \mathrm{sgn}(\hat{\psi}_1), \ldots, \hat{w}_s \mathrm{sgn}(\hat{\psi}_s)\right)^T \\ &= \frac{\partial \ell_n(\boldsymbol{\psi}_0)}{\partial \boldsymbol{\psi}_1} - \hat{I}^{(1)}(\boldsymbol{\psi}^*)(\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\psi}_{10}) - n\lambda_n \left(\hat{w}_1 \mathrm{sgn}(\hat{\psi}_1), \ldots, \hat{w}_s \mathrm{sgn}(\hat{\psi}_s)\right)^T, \end{aligned}$$

where $\boldsymbol{\psi}^*$ is between $\hat{\boldsymbol{\psi}}$ and $\boldsymbol{\psi}_0$. Therefore, since

$$\begin{aligned} \frac{1}{\sqrt{n}}\frac{\partial \ell_n(\boldsymbol{\psi}_0)}{\partial \boldsymbol{\psi}_1} &\xrightarrow{d} \mathcal{N}(\mathbf{0}, I_1(\boldsymbol{\psi}_{10})), \\ \frac{1}{n}\hat{I}_n^{(1)}(\boldsymbol{\psi}^*) &\xrightarrow{p} I_1(\boldsymbol{\psi}_{10}) \end{aligned}$$

and $\hat{\psi}_j^{(0)} \xrightarrow{p} \psi_{j0} \neq 0$ for $1 \leq j \leq s$, we have

$$\begin{aligned} 0 &= \frac{1}{\sqrt{n}}\frac{\partial \ell_n(\boldsymbol{\psi}_0)}{\partial \boldsymbol{\psi}_1} - \frac{1}{\sqrt{n}}\hat{I}^{(1)}(\boldsymbol{\psi}^*)(\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\psi}_{10}) - n^{1/2}\lambda_n \left(\hat{w}_1 \mathrm{sgn}(\hat{\psi}_1), \ldots, \hat{w}_s \mathrm{sgn}(\hat{\psi}_s)\right)^T, \\ &= \frac{1}{\sqrt{n}}\frac{\partial \ell_n(\boldsymbol{\psi}_0)}{\partial \boldsymbol{\psi}_1} - \frac{1}{\sqrt{n}}\hat{I}^{(1)}(\boldsymbol{\psi}^*)(\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\psi}_{10}) + o_p(1). \end{aligned}$$

The last equality follows since

$$\begin{aligned} n^{1/2}\lambda_n \hat{w}_j &= n^{1/2}\lambda_n |\hat{\psi}_j^{(0)}|^{-\gamma_1} D_j^{-\gamma_2} \\ &= \frac{n^{1/2}\lambda_n}{|\hat{\psi}_j^{(0)}|^{\gamma_1} D_j^{\gamma_2}} \\ &\leq \frac{n^{1/2}\lambda_n}{\min_{1\leq j\leq s}\left(|\hat{\psi}_j^{(0)}|^{\gamma_1} D_j^{\gamma_2}\right)} \\ &= \frac{\lambda_n}{a_n} \xrightarrow{p} 0 \text{ for } j = 1, \ldots, s. \end{aligned}$$

Thus, by Slutsky's Theorem,

$$\sqrt{n} I_1(\boldsymbol{\psi}_{10})(\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\psi}_{10}) = \frac{1}{\sqrt{n}} \frac{\partial \ell_n(\boldsymbol{\psi}_0)}{\partial \boldsymbol{\psi}_1} + o_p(1) \xrightarrow{d} \mathcal{N}(\mathbf{0}, I_1(\boldsymbol{\psi}_{10}))$$

and so

$$\sqrt{n}(\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\psi}_{10}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, I_1(\boldsymbol{\psi}_{10})^{-1}),$$

as required.

Note: For $s+1 \leq j \leq d$, $\psi_{j0} = 0$. By $\sqrt{n}$-consistency $\hat{\boldsymbol{\psi}}^{(0)}$, we have that $n^{1/2}|\hat{\psi}_j^{(0)}| = O_p(1)$. Hence, $|\hat{\psi}_j^{(0)}|^{\gamma_1} = O_p(n^{-\gamma_1/2})$ and so $n^{-1/2}|\hat{\psi}_j^{(0)}|^{\gamma_1} D_j^{\gamma_2} = O_p(n^{-(\gamma_1+1)/2})$. Now the oracle property of the adaptive lasso (with $\gamma_2 = 0$) requires that $n^{(\gamma_1+1)/2}\lambda_n \to \infty$ as $n \to \infty$ (see Fan et al., 2009 or Section 2.2.1) and if this condition is satisfied, then $\lambda_n / b_n \xrightarrow{p} \infty$.

Remarks:

- The adaptive lasso (Fan et al., 2009) and hubs weighted graphical lasso estimators both have the oracle property under the same conditions on the tuning parameter λ_n, along other regularity conditions. The performance of the two estimators in finite sample may be very different.

5.5 Simulation Studies

In this section, we assess the performance of the graphical lasso (Friedman et al., 2008) using various tuning parameter selection procedures, the graphical adaptive lasso (Fan et al., 2009), the reweighted L_1 regularization approach of Liu and Ihler (2011), the hubs graphical lasso (HGL) of Tan et al. (2014) as well as our proposed hubs weighted graphical lasso (HWGL) for estimating large-scale networks with hubs. We provide results for both the HWGL procedure (HWGL_1), the two-step HWGL procedure (HWGL_2) as well as the two-step HWGL procedure in the case where the hubs are known.

We first introduce some notation. Let TP, TN, FP and FN denote the numbers of true positives, true negatives (or true zero entries), false positives, and false negatives. Further, let $\mathcal{H}$ denote the set of indices of true hub nodes in Θ and $\hat{\mathcal{H}}$ the set of indices of estimated hub nodes. In our simulations, we consider a node to be a hub node if it is connected to more than 10% of all other nodes. The selection methods will be evaluated using the following performance measures:

- True negative rate (specificity):

$$\frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\sum_{i<j} I\left(\hat{\theta}_{ij} = 0, \theta_{ij} = 0\right)}{\sum_{i<j} I\left(\theta_{ij} = 0\right)}$$

- True positive rate (sensitivity):

$$\frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\sum_{i\leq j} I\left(\hat{\theta}_{ij} \neq 0, \theta_{ij} \neq 0\right)}{\sum_{i\leq j} I\left(\theta_{ij} \neq 0\right)}$$

- Percentage of correctly estimated hub edges:

$$\frac{\sum_{i\in\mathcal{H}, i\neq j} I\left(\hat{\theta}_{ij} \neq 0, \theta_{ij} \neq 0\right)}{\sum_{i\in\mathcal{H}, i\neq j} I\left(\theta_{ij} \neq 0\right)} \times 100\%$$

- Percentage of correctly estimated hub nodes:

$$\frac{|\hat{\mathcal{H}} \cap \mathcal{H}|}{|\mathcal{H}|} \times 100\%$$

and the percentage of correctly estimated non-hub nodes:

$$\frac{|\hat{\mathcal{H}}^c \cap \mathcal{H}^c|}{|\mathcal{H}^c|} \times 100\%,$$

where $|\mathcal{H}|$ denotes the cardinality of the set $\mathcal{H}$.

- Frobenius norm: $\frac{1}{p}\|\widehat{\Theta} - \Theta\|^2 = \frac{1}{p}\sum_{i \neq j}(\hat{\theta}_{ij} - \theta_{ij})^2$

To assess the performance of the methods, we consider four generating mechanisms for the adjacency matrix A of the graphical model, similar to those in Tan et al. (2014).

(i) First, we randomly select H hub nodes and set the elements of the corresponding rows and columns of the adjacency matrix A equal to 1 with probability 0.8 and zero otherwise. Next, we set $A_{ij} = A_{ji} = 1$ for all $i < j$ with probability 0.01, and zero otherwise.

(ii) To generate the adjacency matrix A, we use the same setup as in (i) except that each hub node will be connected to another node with probability 0.3.

(iii) The adjacency matrix will be

$$A = \begin{pmatrix} A_1 & B \\ B^T & A_2 \end{pmatrix},$$

where A_1 and A_2 will be generated as in (i), except that all nodes will have a connection probability of 0.04, and $B = (b_{ij})$ has $b_{ij} = 1$ with probability 0.01 and $b_{ij} = 0$ otherwise.

(iv) Scale-free networks: The probability that a node has degree k follows a power law distribution $P(k) \sim k^{-\alpha}$. The scale-free network of Barabási and Albert (1999) is

constructed by progressively adding nodes to an existing network, where each new node is connected to a node i already present in the network with a probability that is proportional to the degree k_i of node i, i.e.,

$$P(\text{linking to node } i) \sim \frac{k_i}{\sum_j k_j}.$$

Therefore, the Barabási and Albert (1999) network model incorporates two important mechanisms: growth and preferential attachment, which are common to a number of real-world networks, such as business networks and social networks. We use the `R` package `igraph` to generate scale-free networks with $\alpha = 2.3$.

For each of the adjacency matrices in (i)-(iv), we then construct a symmetric matrix C such that $C_{ij} = 0$ if $A_{ij} = 0$, and C_{ij} are independent from the uniform distribution on $[-0.8, -0.5] \cup [0.5, 0.8]$ if $A_{ij} = 1$. Finally, we take the precision matrix Θ to be $C + \{0.1 - \lambda_{\min}(C)\} I_p$, where $\lambda_{\min}(C)$ is the smallest eigenvalue of C and I_p is the $p \times p$ identity matrix to ensure that all the eigenvalues of Θ are positive.

For adjacency matrices in (i) and (ii), we take the number of hubs to be $H = \lfloor p/25 \rfloor$. For Simulation (i), the true network model with $p = 100$ has 4 hub nodes with 323 hub edges, 52 non-hub edges and a network density of 7.58%. The true network model with $p = 200$ has 8 hub nodes with 483 hub edges, 198 non-hub edges and a network density of 3.42%. For Simulation (ii), the true network model with $p = 100$ has 4 hub nodes with 115 hub edges, 55 non-hub edges and a network density of 3.43%. The true network model with $p = 200$ has 8 hub nodes with 504 hub edges, 167 non-hub edges and a network density of 3.37%. For Simulation (iii), the true network model with $p = 100$ has 4 hub nodes, 147 hub edges and 96 non-hub edges with a network density of 4.91%. The true network model with $p = 200$ has 8 hub nodes with 634 hub edges, 389 non-hub edges and a network density of 5.14%. Finally, for the scale-free networks in Simulation (iv), the true network model with $p = 100$ has 3 hub nodes with 59 hub edges, 40 non-hub edges and a network density of 2%. The true network model with $p = 200$ has 3 hub nodes with 85 hub edges, 114 non-hub edges and a network density of 1%.

To implement the graphical lasso, we use the `R` function `glasso` and select the tuning parameter λ from a fine grid based on BIC. To implement the graphical lasso using the StARS procedure for tuning parameter selection, we use the function `huge.select`, available in the `R` package `huge`. To implement the hubs graphical lasso of Tan et al. (2014), we use the functions `hglassoBIC` and `hglasso` in the `R` package `hglasso`. Each tuning parameter ρ_i for $i = 1, 2, 3$ in the hubs graphical lasso will be selected from a fine grid. We also consider various values of c in the BIC-type quantity (5.2). The simulation results are displayed in Tables 5.1 to 5.4.

Discussion of Simulation Results:

From Tables 5.1 and 5.2, we see that the two-step HWGL procedure in moderate dimensions outperforms the competitors when the true underlying graph has hub nodes. It is also clear that when the true hubs are known in advance, which is a reasonable assumption in many biological applications, using a weighted lasso that takes into account knowledge of these highly influential nodes, results in substantially better finite-sample performance compared to the lasso and adaptive lasso procedures. The one-step HWGL procedure in the case where the hubs are unknown also performs well; it outperforms competitors in terms of hub edge and hub node identification, better capturing hub structure. As expected, BIC and StARS perform similarly in the case $n = p = 100$ in terms of edge identification, but we observed better performance by StARS in higher dimensions. The scale-free network approach of Liu and Ihler (2011), which is not designed for estimating networks with hubs, does not result in significant improvements over the lasso and adaptive lasso procedures. The HGL with $c = 0.5$ and $c = 0.75$ in the BIC-type quantity for tuning parameter selection lead to much denser graphs compared to the graphical lasso with either BIC or StARS. Recall that a smaller c favours more hub nodes. Using the default value $c = 0.2$ in the `R` function `hglassoBIC` would result in overly dense graphs. Sensitivity to the user-specified parameter c, which controls the number of hub nodes in the graph, is a drawback of the HGL of Tan et al. (2014).

From Table 5.3, again we observe that the one-step HWGL procedure outperforms

the graphical lasso, adaptive lasso, HGL and scale-free network approach of Liu and Ihler (2011). The two-step procedure in the case where the hubs are unknown performs better than the one-step HGL procedure. However, it requires setting a cutoff threshold for a node to be considered a hub. For $n = p = 100$, the HGL of Tan et al. (2014) performs better than the standard lasso and adaptive lasso procedures in terms of hub edge identification. In higher dimensions, the graphical lasso with StARS results in higher true positive rates compared to other methods except the two-step HWGL procedures, but its true negative rate is lower on average. This is not surprising as the goal of StARS is to "overselect"; it allows for false positives but not false negatives.

The scale-free networks generated in Simulation (iv) have hubs that are not as highly connected as those in Simulations (i)-(iii). Therefore, it is not surprising that the graphical lasso with StARS performs well (see Table 5.4). When $n = p = 100$, knowing the true hubs in advance and allowing for different levels of penalization between the hub and non-hub edges results in better performance compared to the other procedures across all performance measures. The one-step HWGL procedure performs well in terms of hub edge identification. When $p > n$, the graphical lasso with StARS performs better than the HWGL procedures in terms of the true positive rate, but its true negative rate is lower on average compared to the HWGL procedures. The results for the HGL of Tan et al. (2014) are omitted as their procedure is not designed for estimating scale-free networks.

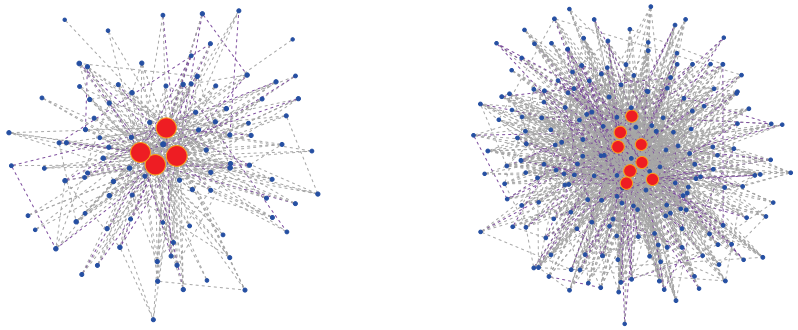

Figure 5.1: Simulation (i) - Networks with hubs for $p = 100$ (left) and $p = 200$ (right), where each hub node is connected to a different node with probability 0.8. Dashed grey lines correspond to hub edges, dashed purple lines correspond to non-hub edges, and the size of each node is proportional to its degree. Hub nodes are shown in red.

Method	Tuning Parameter Selector	True Pos. Rate	True Neg. Rate	Perc. of Correctly Estimated Hub Edges	Perc. of Correctly Estimated Hub/Non-Hub Nodes	Number of Estimated Edges	Frobenius Norm
Simulation (i)							
$n = 100, p = 100$							
Lasso	BIC	48.10 (0.26)	94.40 (0.28)	38.28 (0.33)	99.50 (0.35)/73.70 (1.60)	384.73 (13.73)	7.29 (0.05)
Lasso	EBIC	21.05 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	8.86 (0.06)
Lasso	StARS	46.39 (0.18)	96.22 (0.08)	36.32 (0.25)	99.50 (0.35)/84.08 (0.61)	293.07 (4.31)	7.71 (0.06)
Adaptive lasso	BIC	58.31 (0.19)	96.55 (0.03)	52.97 (0.26)	100 (0)/99.24 (0.10)	334.78 (1.42)	4.49 (0.02)
SF	BIC	53.08 (0.33)	97.94 (0.07)	46.53 (0.46)	99.25 (0.56)/95.05 (0.37)	246.59 (4.50)	5.34 (0.04)
HGL	BIC_1^*	56.12 (0.17)	84.26 (0.29)	47.45 (0.20)	100 (0)/19.91 (1.51)	886.60 (13.72)	6.43 (0.02)
HGL	BIC_2^*	50.81 (0.31)	92.82 (0.33)	42.05 (0.40)	99.50 (0.50)/65.26 (1.67)	469.76 (16.53)	7.30 (0.04)
HWGL_1	BIC	70.55 (0.49)	99.60 (0.01)	72.77 (0.72)	100 (0)/100 (0)	253.23 (2.68)	2.75 (0.03)
HWGL_2	BIC	79.24 (0.36)	99.23 (0.01)	85.56 (0.52)	100 (0)/100 (0)	311.58 (2.17)	2.62 (0.03)
HWGL_2 - Hubs Known	BIC	79.24 (0.36)	99.23 (0.01)	85.56 (0.52)	100 (0)/100 (0)	311.58 (2.17)	2.62 (0.03)
$n = 100, p = 200$							
Lasso	BIC	24.76 (0.22)	99.30 (0.03)	16.01 (0.28)	66.38 (1.11)/99.18 (0.11)	336.06 (9.93)	14.98 (0.09)
Lasso	EBIC	12.18 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	16.26 (0.10)
Lasso	StARS	30.75 (0.09)	96.97 (0.03)	23.27 (0.11)	86.75 (0.30)/91.96 (0.22)	863.51 (7.51)	13.47 (0.06)
Adaptive lasso	BIC	27.30 (0.13)	99.00 (0.03)	19.25 (0.16)	78.75 (0.91)/99.99 (0.01)	432.65 (6.94)	13.28 (0.06)
SF	BIC	28.54 (0.14)	99.50 (0.02)	20.98 (0.17)	68.12 (0.86)/99.81 (0.03)	361.03 (5.05)	11.19 (0.05)
HGL	BIC_1^*	49.69 (0.24)	58.26 (0.35)	41.73 (0.26)	100 (0)/0 (0)	8319.62 (68.81)	73.08 (0.97)
HGL	BIC_2^*	33.69 (0.09)	92.95 (0.21)	26.28 (0.10)	93.12 (0.62)/69.73 (1.29)	1654.97 (38.89)	13.14 (0.05)
HWGL_1	BIC	31.18 (0.16)	99.81 (0.01)	24.53 (0.21)	83.62 (1.11)/100 (0)	347.32 (3.47)	8.91 (0.03)
HWGL_2	BIC	42.15 (0.28)	99.69 (0.001)	38.75 (0.36)	83.62 (1.11)/100 (0)	548.95 (5.10)	8.76 (0.04)
HWGL_2 - Hubs Known	BIC	45.18 (0.18)	99.65 (0.005)	42.67 (0.23)	100 (0)/100 (0)	605.59 (3.38)	8.76 (0.04)

Table 5.1: Networks with hub nodes - True positive rate, true negative rate, percentage of correctly estimated hub edges and hub/non-hub nodes, number of estimated edges and Frobenius norm error, averaged over $N = 100$ replications of size $n = 100$, for the graphical lasso using BIC, EBIC and StARS for tuning parameter selection, the adaptive lasso as well as the scale-free (SF) network approach of Liu and Ihler (2011), HGL of Tan et al. (2014) with tuning parameter selectors BIC_1^* ($c = 0.5$) and BIC_2^* ($c = 0.75$), the HWGL (HWGL_1), and the two-step HWGL (HWGL_2) with hubs unknown and known. The standard errors for the means over the 100 replications are reported in parentheses.

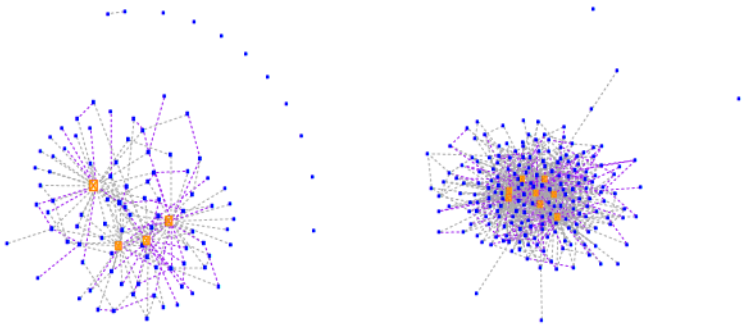

Figure 5.2: Simulation (ii) - Networks with hubs for $p = 100$ (left) and $p = 200$ (right), where each hub node is connected to a different node with probability 0.3. Dashed grey lines correspond to hub edges, dashed purple lines correspond to non-hub edges, and the size of each node is proportional to its degree. Hub nodes are shown in red.

Method	Tuning Parameter Selector	True Pos. Rate	True Neg. Rate	Perc. of Correctly Estimated Hub Edges	Perc. of Correctly Estimated Hub/Non-Hub Nodes	Number of Estimated Edges	Frobenius Norm
Simulation (ii)							
$n = 100, p = 100$							
Lasso	BIC	66.17 (0.36)	97.48 (0.07)	57.84 (0.62)	99.25 (0.43)/92.75 (0.35)	198.98 (4.25)	2.54 (0.01)
Lasso	EBIC	37.04 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	3.12 (0.01)
Lasso	StARS	70.26 (0.21)	96.01 (0.05)	64.17 (0.38)	100 (0)/85.24 (0.34)	280.63 (2.55)	4.08 (0.02)
Adaptive lasso	BIC	72.86 (0.19)	98.57 (0.02)	72.56 (0.37)	99.75 (0.25)/100 (0)	164.85 (0.92)	1.59 (0.01)
SF	BIC	71.31 (0.26)	98.38 (0.04)	72.86 (0.47)	100 (0)/97.74 (0.15)	169.79 (2.34)	1.77 (0.01)
HGL	BIC_1^*	74.01 (0.27)	94.27 (0.16)	69.57 (0.39)	100 (0)/76.81 (0.84)	373.57 (8.07)	2.32 (0.01)
HGL	BIC_2^*	68.34 (0.33)	96.87 (0.09)	62.33 (0.54)	100 (0)/88.56 (0.49)	234.09 (5.24)	2.52 (0.01)
$HWGL_1$	BIC	74.94 (0.24)	99.11 (0.03)	83.49 (0.42)	100 (0)/100 (0)	144.86 (1.77)	1.25 (0.01)
$HWGL_2$	BIC	75.02 (0.16)	97.83 (0.03)	88.98 (0.36)	100 (0)/100 (0)	206.12 (1.93)	1.44 (0.01)
$HWGL_2$ - Hubs Known	BIC	75.02 (0.16)	97.83 (0.03)	88.98 (0.36)	100 (0)/100 (0)	206.12 (1.93)	1.44 (0.01)
$n = 100, p = 200$							
Lasso	BIC	36.77 (0.24)	99.47 (0.02)	23.10 (0.38)	48.00 (1.27)/99.79 (0.03)	222.39 (6.03)	5.71 (0.02)
Lasso	EBIC	22.96 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	6.15 (0.03)
Lasso	StARS	48.07 (0.12)	97.06 (0.03)	40.03 (0.19)	86.12 (0.94)/94.76 (0.19)	784.43 (6.93)	8.68 (0.05)
Adaptive lasso	BIC	41.83 (0.21)	99.30 (0.03)	31.27 (0.33)	60.25 (1.26)/100 (0)	299.40 (6.88)	5.17 (0.02)
SF	BIC	43.25 (0.22)	99.39 (0.02)	34.32 (0.35)	68.38 (1.01)/99.72 (0.03)	294.44 (4.61)	4.47 (0.02)
HGL	BIC_1^*	73.53 (0.21)	57.97 (0.31)	66.61 (0.27)	100 (0)/0 (0)	8522.47 (61.84)	30.42 (0.41)
HGL	BIC_2^*	49.11 (0.22)	96.56 (0.10)	41.35 (0.28)	92.25 (0.94)/92.83 (0.43)	888.32 (21.62)	5.13 (0.02)
$HWGL_1$	BIC	50.98 (0.29)	99.51 (0.01)	47.95 (0.47)	85.88 (0.98)/100 (0)	338.71 (4.53)	3.43 (0.02)
$HWGL_2$	BIC	56.55 (0.30)	98.95 (0.02)	58.04 (0.53)	85.88 (0.98)/100 (0)	493.88 (5.16)	3.75 (0.02)
$HWGL_2$ - Hubs Known	BIC	58.80 (0.26)	98.92 (0.02)	61.92 (0.46)	100 (0)/100 (0)	519.27 (5.20)	3.76 (0.02)

Table 5.2: Networks with hub nodes - True positive rate, true negative rate, percentage of correctly estimated hub edges and hub/non-hub nodes, number of estimated edges and Frobenius norm error, averaged over $N = 100$ replications of size $n = 100$, for the graphical lasso using BIC, EBIC and StARS for tuning parameter selection, the adaptive lasso as well as the scale-free (SF) network approach of Liu and Ihler (2011), HGL of Tan et al. (2014) with tuning parameter selectors BIC_1^* ($c = 0.5$) and BIC_2^* ($c = 0.75$), the one-step ($HWGL_1$), and two-step ($HWGL_2$) HWGL with hubs unknown and known. The standard errors for the means over the 100 replications are reported in parentheses.

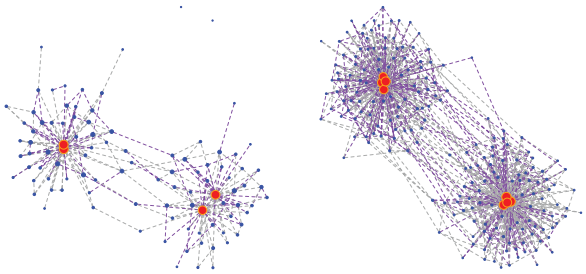

Figure 5.3: Simulation (iii) - Hub networks with clustering for $p = 100$ (left) and $p = 200$ (right). Dashed grey lines correspond to hub edges, dashed purple lines correspond to non-hub edges, and the size of each node is proportional to its degree. The central red nodes in each network indicate hub nodes.

Method	Tuning Parameter Selector	True Pos. Rate	True Neg. Rate	Perc. of Correctly Estimated Hub Edges	Perc. of Correctly Estimated Hub/Non-Hub Nodes	Number of Estimated Edges	Frobenius Norm
Simulation (iii)							
$n = 100, p = 100$							
Lasso	BIC	51.91 (0.30)	97.41 (0.06)	42.35 (0.46)	66.50 (1.95)/87.89 (0.37)	199.94 (3.78)	3.57 (0.02)
Lasso	EBIC	29.15 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	4.40 (0.02)
Lasso	StARS	53.43 (0.21)	97.18 (0.04)	44.71 (0.34)	73.75 (1.93)/86.98 (0.29)	216.04 (2.46)	3.46 (0.02)
Adaptive lasso	BIC	49.45 (0.33)	99.59 (0.03)	43.01 (0.58)	68.50 (1.80)/99.99 (0.01)	88.70 (2.18)	2.54 (0.01)
SF	BIC	49.10 (0.41)	98.76 (0.04)	39.98 (0.80)	64.50 (1.92)/97.00 (0.22)	126.91 (3.14)	2.79 (0.02)
HGL	BIC_1^*	57.68 (0.29)	95.55 (0.10)	53.82 (0.46)	95.25 (1.11)/79.58 (0.48)	307.45 (5.35)	3.45 (0.01)
HGL	BIC_2^*	52.01 (0.49)	96.90 (0.10)	43.84 (0.82)	75.75 (2.29)/85.84 (0.47)	224.18 (6.07)	3.64 (0.02)
HWGL_1	BIC	57.97 (0.34)	99.31 (0.03)	63.07 (0.63)	95.25 (0.99)/99.99 (0.01)	131.19 (2.34)	2.10 (0.01)
HWGL_2	BIC	61.46 (0.35)	98.77 (0.04)	75.27 (0.82)	95.25 (0.99)/99.99 (0.01)	168.87 (2.68)	2.31 (0.02)
HWGL_2 - Hubs Known	BIC	62.65 (0.26)	98.76 (0.04)	78.07 (0.61)	100 (0)/100 (0)	173.14 (2.53)	2.30 (0.02)
$n = 100, p = 200$							
Lasso	BIC	33.57 (0.23)	99.04 (0.04)	30.07 (0.36)	65.38 (0.75)/99.74 (0.05)	392.68 (9.40)	8.87 (0.03)
Lasso	EBIC	16.35 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	9.86 (0.04)
Lasso	StARS	39.10 (0.11)	97.67 (0.04)	37.63 (0.17)	75.75 (0.58)/97.57 (0.19)	717.48 (9.20)	8.10 (0.03)
Adaptive lasso	BIC	36.13 (0.14)	99.13 (0.03)	34.70 (0.21)	69.12 (1.07)/100 (0)	405.32 (6.17)	6.99 (0.02)
SF	BIC	35.82 (0.14)	99.46 (0.02)	35.89 (0.23)	67.25 (0.68)/99.97 (0.01)	340.27 (4.27)	6.73 (0.02)
HGL	BIC_1^*	42.27 (0.10)	95.90 (0.09)	40.85 (0.20)	82.00 (0.74)/92.01 (0.32)	1091.75 (17.25)	7.87 (0.02)
HGL	BIC_2^*	42.34 (0.07)	95.69 (0.03)	40.50 (0.12)	82.25 (0.69)/91.79 (0.29)	1130.89 (6.53)	7.82 (0.02)
HWGL_1	BIC	37.78 (0.17)	99.66 (0.01)	41.03 (0.32)	76.62 (1.25)/100 (0)	325.78 (3.59)	5.37 (0.02)
HWGL_2	BIC	43.17 (0.30)	99.26 (0.02)	51.72 (0.58)	76.62 (1.25)/100 (0)	467.41 (5.80)	5.72 (0.03)
HWGL_2 - Hubs Known	BIC	47.59 (0.15)	99.25 (0.01)	60.26 (0.29)	100 (0)/100 (0)	524.46 (4.29)	5.68 (0.03)

Table 5.3: Networks with hubs and clustering - True positive rate, true negative rate, percentage of correctly estimated hub edges and hub/non-hub nodes, number of estimated edges and Frobenius norm error, averaged over $N = 100$ replications of size $n = 100$, for the graphical lasso using BIC, EBIC and StARS for tuning parameter selection, the adaptive lasso as well as the scale-free (SF) network approach of Liu and Ihler (2011), HGL of Tan et al. (2014) with tuning parameter selectors BIC_1^* ($c = 0.5$) and BIC_2^* ($c = 0.75$), the one-step HWGL (HWGL_1), and the two-step HWGL (HWGL_2) with hubs unknown and known. The standard errors for the means over the 100 replications are reported in parentheses.

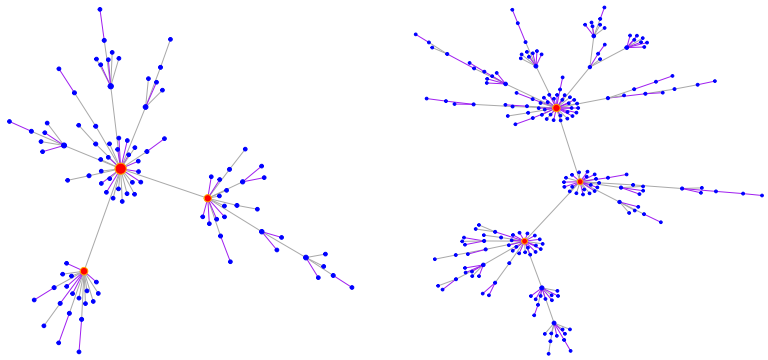

Figure 5.4: Simulation (iv) - Scale-free networks for $p = 100$ (left) and $p = 200$ (right). Grey lines correspond to hub edges, purple lines correspond to non-hub edges, and red nodes in each network indicate hub nodes.

Method	Tuning Parameter Selector	True Pos. Rate	True Neg. Rate	Perc. of Correctly Estimated Hub Edges	Perc. of Correctly Estimated Hub/Non-Hub Nodes	Number of Estimated Edges	Frobenius Norm
Simulation (iv)							
$n = 100, p = 100$							
Lasso	BIC	70.58 (0.31)	98.34 (0.08)	64.92 (0.84)	83.00 (2.39)/97.04 (0.43)	121.01 (4.38)	1.57 (0.01)
Lasso	EBIC	50.25 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	1.83 (0.01)
Lasso	StARS	78.31 (0.34)	95.52 (0.12)	80.64 (0.68)	100 (0)/81.53 (0.73)	273.05 (6.29)	2.56 (0.02)
Adaptive lasso	BIC	78.69 (0.35)	98.85 (0.06)	80.29 (0.71)	79.00 (2.71)/99.99 (0.01)	112.62 (3.28)	0.91 (0.01)
SF	BIC	72.25 (0.32)	99.27 (0.03)	72.19 (0.98)	71.00 (2.67)/99.96 (0.03)	79.14 (2.04)	1.10 (0.01)
$HWGL_1$	BIC	76.94 (0.27)	99.29 (0.03)	80.53 (0.67)	80.00 (2.22)/100 (0)	87.66 (1.64)	1.03 (0.01)
$HWGL_2$	BIC	75.01 (0.46)	98.29 (0.03)	83.10 (1.55)	80.33 (2.23)/100 (0)	132.03 (2.35)	0.92 (0.01)
$HWGL_2$ - Hubs Known	BIC	79.01 (0.08)	98.08 (0.03)	96.58 (0.22)	100 (0)/100 (0)	150.42 (1.35)	0.88 (0.01)
$n = 100, p = 200$							
Lasso	BIC	64.25 (0.18)	99.64 (0.01)	63.59 (0.75)	52.33 (1.66)/100 (0)	128.12 (3.53)	1.50 (0.01)
Lasso	EBIC	50.13 (0)	100 (0)	0 (0)	0 (0)/100 (0)	0 (0)	1.64 (0.01)
Lasso	StARS	74.12 (0.19)	97.58 (0.04)	86.48 (0.41)	98.67 (0.66)/97.57 (0.16)	572.30 (9.02)	2.35 (0.01)
Adaptive lasso	BIC	70.13 (0.29)	99.64 (0.03)	80.73 (0.64)	62.67 (2.69)/100 (0)	150.55 (6.42)	1.08 (0.01)
SF	BIC	67.81 (0.18)	99.74 (0.01)	81.60 (0.79)	80.67 (1.85)/100 (0)	122.53 (2.33)	1.15 (0.01)
$HWGL_1$	BIC	69.47 (0.14)	99.75 (0.01)	86.45 (0.45)	87.33 (1.63)/100 (0)	127.06 (1.74)	0.96 (0.01)
$HWGL_2$	BIC	68.85 (0.27)	99.41 (0.01)	87.81 (1.27)	88.33 (1.60)/100 (0)	191.34 (2.63)	1.07 (0.01)
$HWGL_2$ - Hubs Known	BIC	70.79 (0.05)	99.39 (0.01)	96.86 (0.20)	100 (0)/100 (0)	202.21 (2.38)	1.05 (0.01)

Table 5.4: Scale-free networks - True positive rate, true negative rate, percentage of correctly estimated hub edges and hub/non-hub nodes, number of estimated edges and Frobenius norm error, averaged over $N = 100$ replications of size $n = 100$, for the graphical lasso using BIC, EBIC and StARS for tuning parameter selection, the adaptive lasso, the scale-free (SF) network approach of Liu and Ihler (2011), the hubs graphical lasso (HGL) of Tan et al. (2014), and our one-step ($HWGL_1$) and two-step ($HWGL_2$) HWGL procedures. The standard errors for the means over the 100 replications are reported in parentheses.

5.5.1 Recovery of Global Network Structure

We used the `R` package `igraph` to graphically display the networks and to compute several network measures. In what follows, we summarize some of the network properties of interest, including degree centrality, global clustering coefficient, betweenness centrality, network diameter, average path length, and network density.

Definition 7. (Network Measures)

- **Degree Centrality:** The degree of a node is defined as the number of its edges. The normalized degree divides the degree by the maximum possible degree of the network, yielding a value between 0 and 1. Scale-free networks are characterized by power law degree distributions, in which a few nodes have very high degree while most nodes have low degree. Networks with clusters, on the other hand, have relatively even degree that depends on cluster size (Newman, 2010).

- **Global Clustering Coefficient:** The global clustering coefficient measures the degree to which the nodes' neighbours are also interconnected. It is the ratio of the number of closed triplets to the number of connected triplets (both open and closed), ranging from 0 if the network does not contain triplets to 1 if each two neighbours of all nodes are directly connected as well. A triplet consists of three nodes that are connected by either two (open triplet) or three (closed triplet) undirected edges (Newman, 2010).

- **Betweenness Centrality:** The betweenness centrality of a node is the number of shortest paths between all other nodes in the network that pass through the given node. It can be used to measure the relative importance of the node to the network. Scale-free networks, for example, will have a few nodes with very high betweenness centrality as those nodes will connect most other nodes to each other (Freeman, 1977; Newman, 2010).

- **Network Diameter:** The network diameter is defined as the longest of all the calculated shortest paths in a network (Dorogovtsev and Mendes, 2003).

- **Average Path Length:** The average path length is the average length of all the shortest paths between any two nodes. It is bounded above by the network diameter (Dorogovtsev and Mendes, 2003).

- **Network Density:** The density of a network is defined as the ratio of the number of edges to the total number of possible edges.

5.6 Microbiome Data Analysis

5.6.1 Analysis of the Saliva Microbiomes of Bonobos and Chimpanzees

We illustrate the performance of the methodology on saliva microbiome data sets of two *Pan* species found in Li et al. (2013). We model microbial interactions using undirected graphical models, estimated from relative abundances of genera in the saliva microbiomes of 23 bonobos *(Pan paniscus)* from the Lola ya Bonobo Sanctuary in the Democratic Republic of the Congo (DRC), and 22 chimpanzees *(Pan troglodytes)* from the Tacugama Chimpanzee Sanctuary in Sierra Leone (SL).

For the bonobos, 69 genera were identified along with 2 unknown/unclassified genera. *Enterobacter* (20.8%) was the most abundant genus identified, followed by *Porphyromonas* (10.3%) and *Neisseria* (9.7%). For the chimpanzees, 79 genera were identified along with 2 unknown/unclassified genera. The most abundant genera identified were *Porphyromonas* (16.9%), *Fusobacterium* (14.0%), *Haemophilus* (11.4%) and *Neisseria* (8.1%).

After replacing zero abundance counts by 0.5, we use a centered log-ratio transformation of the data. We then estimate undirected graphical models using the graphical lasso of Friedman et al. (2008). To perform tuning parameter selection, we use StARS (Liu et al., 2010). In Table 5.5, we provide a list of genera for each data set corresponding to high-degree nodes identified by the graphical lasso using StARS for tuning parameter selection.

For each network, we use the `R` package `igraph` to evaluate several network measures, including network density, global clustering coefficient, betweenness centrality, and average path length (Table 5.6). Differences in network measures between the bonobo and chimpanzee groups are assessed for statistical significance by permutation tests with 1000 randomizations. The apes were randomly reassigned to one of two groups 1000 times. For each permutation, a network is estimated for each group and distributions of the differences in network indices were generated for statistical inference. No significant differences were found in terms of the global network structure (measured by global clustering coefficient, betweenness centrality and average path length) between the two groups. Significant differences in degree centrality were found for nodes corresponding to genera *Enterobacter* (0.20 v 0.05, $p = 0.01$), *Escherichia* (0.14 v 0.03, $p = 0.06$), *Eubacterium* (0.20 v 0, $p = 0.05$), *Granulicatella* (0.12 v 0, $p = 0.07$), *Kingella* (0 v 0.12, $p < 0.01$), *Klebsiella* (0.05 v 0, $p = 0.01$), *Neisseria* (0.27 v 0, $p < 0.01$), *Parvimonas* (0.20 v 0, $p < 0.01$), *Pasteurella* (0.05 vs 0.23, $p < 0.01$), *SR1_genera* (0.05 v 0, $p = 0.07$), *Salmonella* (0.08 vs 0.4, $p < 0.01$), and *Schwartzia* (0.17 v 0.35, $p = 0.07$). For both groups, there is a tendency for genera to correlate positively with other genera from the same phylum, especially within Firmicutes and Proteobacteria, which was also found in Li et al. (2013).

Assuming a hub structure for the bonobo microbial interaction network and applying the HWGL procedure, we found nodes corresponding to genera *Neisseria*, *Peptostreptococcaceae* and *Schwartzia* to be highly connected. For the chimpanzee group, nodes corresponding to genera *Erysipelotrichaceae*, *Facklamia*, *Johnsonella*, *Mogibacterium*, *Peptococcus*, *Phocaeicola*, *Salmonella* and *Schwartzia* had high degree. To obtain hub networks reproducible under random sampling, we generated 100 bootstrap samples and applied our HWGL procedure to each sample. Only the edges in the inverse covariance matrix that were reproduced in at least 50 bootstrap replicates were retained. For the bonobo group, the graphical lasso with StARS procedure inferred 205 edges; our procedure inferred 212 edges, and 173 edges were common between the two reconstructed networks. The HWGL-estimated network has 63 nodes. For the chimpanzee group, the graphical

lasso with StARS inferred 247 edges, while our procedure inferred 198 edges; both methods agreed on 182 edges. The HWGL-estimated network has 54 nodes. For both data sets, the HWGL procedures assigned more edges to hubs. The networks produced by HWGL are displayed in Figures 5.5 and 5.6. The edges common to both networks are displayed in Figures 5.7 and 5.8 for the bonobo and chimpanzee groups, respectively. For the bonobo data set, both methods agree on the clusters of nodes corresponding to genera *Actinobacillus*, *Atopobium*, *Coprococcus*, *Eubacterium* and *Parvimonas* as well as *Megasphaera*, *Schwartzia*, *Selenomonas* and *Solobacterium*. For the chimpanzee data set, both methods agree on the cluster of nodes, including *Alloiococcus*, *Erysipelotrichaceae*, *Facklamia*, *Fusobacterium*, *Johnsonella*, *Kingella*, *Peptococcus*, *Salmonella* and *Schwartzia*.

Bonobo			Chimpanzee		
Genus	Phylum	Degree	Genus	Phylum	Degree
Atopobium	Actinobacteria	13	Erysipelotrichaceae	Firmicutes	21
Coprococcus	Firmicutes	13	Facklamia	Firmicutes	21
Enterobacter	Proteobacteria	13	Faecalibacterium	Firmicutes	20
Eubacterium	Firmicutes	13	Johnsonella	Firmicutes	21
Haemophilus	Proteobacteria	16	Peptococcus	Firmicutes	21
Lachnospiracea	Firmicutes	13	Phocaeicola	Bacteroidetes	21
Neisseria	Proteobacteria	18	Ruminococcus	Firmicutes	20
Parvimonas	Firmicutes	13	**Salmonella**	Proteobacteria	24
Peptostreptococcaceae	Firmicutes	13	Schwartzia	Firmicutes	21
Schwartzia	Firmicutes	12			
Solobacterium	Firmicutes	12			

Table 5.5: Genera corresponding to high-degree nodes from the graphical lasso (StARS) reconstruction of the microbial interaction network for the bonobo and chimpanzee groups.

Network Index	Bonobo	Chimpanzee
Global Clustering Coefficient	0.50	0.61
Mean Betweenness Centrality	46.52	43.31
Average Path Length	2.85	2.98
Minimum Degree	0	0
Median Degree	5	2
Mean Degree	5.78	6.10
Maximum Degree	18	24
Network Density	8.2%	7.6%
Network Diameter	6	7

Table 5.6: Network measures from the graphical lasso (StARS) reconstruction of the microbial interaction network for the bonobo and chimpanzee groups.

Figure 5.5: Reconstructed microbial interaction network for the bonobo data set using HWGL. Positive partial correlations are displayed in blue and negative partial correlations are displayed in purple.

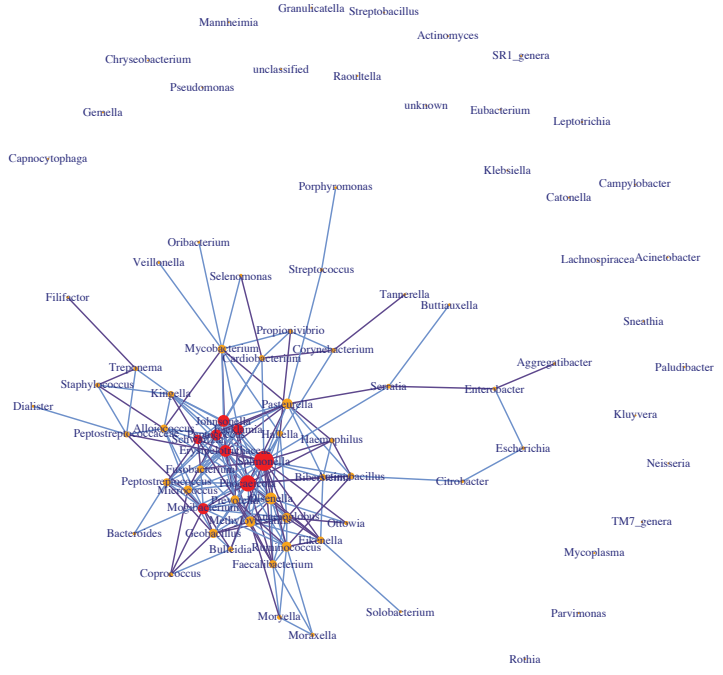

Figure 5.6: Reconstructed microbial interaction network for the chimpanzee data set using HWGL. Positive partial correlations are displayed in blue and negative partial correlations are displayed in purple.

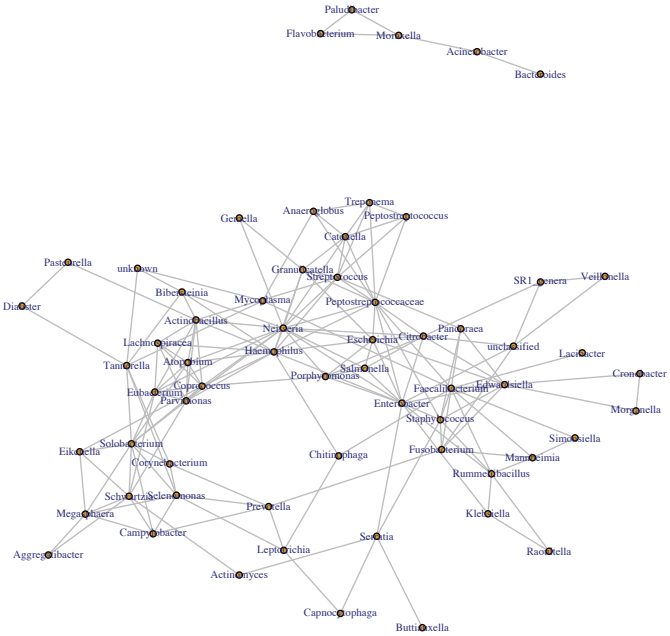

Figure 5.7: Edges inferred by both the graphical lasso with StARS and HWGL for the bonobo data set.

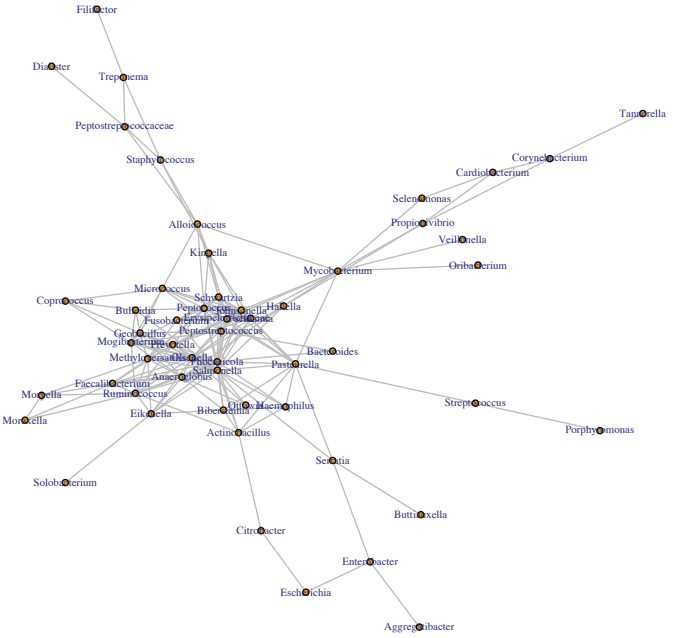

Figure 5.8: Edges inferred by both the graphical lasso with StARS and HWGL for the chimpanzee data set.

5.7 Discussion

Estimating microbial interaction networks from microbial taxon abundance data is an important problem in microbiome studies. Recently, some authors (Faust et al., 2015) have investigated the performance of widely used network inference schemes, such as neighbourhood selection (Meinshausen and Bühlmann) and (inverse) covariance selection (Friedman et al., 2009), in recovering microbial interaction networks from microbiome data, but have found that networks with hub structures elude accurate inference. In this chapter, we addressed this challenge and investigated the network recovery performance of hub network inference schemes, such as the hubs graphical lasso (HGL) of Tan et al. (2014) and the reweighted L_1-regularization approach of Liu and Ihler (2011). The former is designed for estimating networks with very densely connected nodes, referred to as *super hubs*, while the latter is designed for estimating scale-free networks, for which there may be no clear distinction between *hub* and *non-hub* nodes.

In this chapter, we proposed a more general method for estimating networks with hubs that can accommodate both networks with so-called "super hubs" as well as scale-free networks. Our proposed method is a weighted lasso approach with informative weights that take into account hub structure. Empirically, we show that the proposed method performs significantly better than methods that do not explicitly take hub structure into account, but it also outperforms network estimation procedures designed for modelling networks with hubs.

This work focuses on the problem of static network modelling, where the inferred microbial interaction network provides a "snapshot" of the microbial community structure at a single time point. However, it is well known that microbial interaction networks undergo changes over time in response to changes in external conditions (e.g. diet, exposure to antibiotics) and the temporal variation of these networks can be captured with dynamic networks (Faust et al., 2015). Techniques developed for static network modelling will pave the way for the development of new approaches for modelling the dynamics of microbial communities.

Chapter 6

Estimation of Time-Varying Networks

6.1 Introduction

In previous chapters of this thesis, we focused on estimating a single inverse covariance matrix from i.i.d. samples and the *static* network that it encodes. In many applications, however, the network undergoes changes over time in response to internal or external stimuli, and identifying the temporal changes in the network structure is of interest. For example, gene regulatory networks describing temporal processes, such as cell cycle progression or the life cycle of an organism, can undergo systematic rewiring to facilitate regulatory functions changing in response to environmental and genetic stress (Jethava et al., 2013). As another example, in microbiome studies, the inferred static network provides a "snapshot" of the microbial community structure at a single time point. However, microbial interactions evolve over time and numerous phenomena, such as community stability and perturbation (Faust and Raes, 2012), can be studied only if temporality is taken into account.

If enough replications are available at each time point, methods for inferring static networks may be used to estimate the dependence structure of the variables at each time point. On the other hand, with only a single observation available at each time point, some authors in previous analyses have pooled observations from all time points together and inferred a single "average" network (Song et al., 2009, and references therein). Other authors (Zhou et al., 2010; Song et al., 2009) make the assumption that the time-

evolving network is smoothly varying, which allows information to be shared across time by reweighting observations from different time points and then treating those observations as i.i.d. (Song et al., 2009). The weighting is designed so that observations nearby time t are assigned larger weights and the weights become smaller for observations further away from t.

In this chapter, we investigate the problem of estimating time-varying networks. We assume that we have n replications of a T time point longitudinal study for which p variables are measured, and seek to estimate at each time point the dependence structure of the variables. For each replication, the observations are assumed to be independent, but no longer identically distributed, as the underlying graph evolves over time. While methods for estimating static networks may be used to infer the dependence structure at each time point, such methods do not take advantage of the common structure of networks at nearby time points.

In Section 6.2, we review exisiting methods for estimating time-varying networks. The first is the method of Zhou et al. (2010), which estimates a sequence of graphs $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(T)}$ from a single replication of a T time point study. Their procedure makes use of L_1-regularization based on a weighted empirical covariance matrix. The second is the method of Guo et al. (2011), designed for jointly estimating multiple Gaussian graphical models that share common structure.

In Section 6.3.1, we then introduce two procedures for estimating time-varying networks, where we assume that multiple measurements at each time point obtained under similar experimental conditions are available. Working within a penalized maximum likelihood framework, we impose two penalties on Θ_t: a weighted L_1 penalty to encourage sparsity in Θ_t and a Wishart-type penalty that shrinks the network at time point t towards the network at the previous time point. In the literature, an inverse Wishart-type penalty has been used in the penalized maximum likelihood framework for estimating a single covariance matrix (Meyer, 2011), but without imposing the additional constraint of sparsity. We explore two versions of our approach that applies an L_1 penalty and a Wishart-type penalty to Θ_t. The first is a sequential penalized likelihood approach,

where the Wishart-type penalty on Θ_t allows for the "borrowing of strength" from the reconstructed network at only the previous time point. The second estimates $\Theta_1, \ldots, \Theta_T$ jointly. We provide computional details for solving the resulting optimization problem in Section 6.3.2.1. We then perform a simulation study in Section 6.4 to investigate the performance of the proposed methods as well as that of Zhou et al. (2010). The proposed methodology is then illustrated with a microarray time series data set in Section 6.5. We conclude with a discussion in Section 6.6.

6.2 Existing Methods for Estimating Time-Varying Networks

Estimating Smoothly Varying Networks (Zhou et al., 2010): Zhou et al. (2010) investigated the problem of estimating time-varying Gaussian graphical models using L_1 regularization. They assumed that the observations $X_1, \ldots, X_T$ are independent, but no longer identically distributed. Each observation X_t is assumed to be independently drawn from a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix Σ_t with the elements of Σ_t changing smoothly over time. Associated with each X_t is its undirected graph $\mathcal{G}^{(t)} = (V^{(t)}, E^{(t)})$, which is determined by the sparsity pattern of $\Sigma_t^{-1} = \Theta_t$ and where $V^{(t)}$ and $E^{(t)}$ are the sets of vertices and edges of the graph, respectively.

Zhou et al. (2010) proposed to estimate Σ_t by

$$\hat{\Sigma}^{(t)} = \underset{\Sigma \succ 0}{\arg\min} \left\{ \log\det\Sigma + \mathrm{tr}(\Sigma^{-1}\hat{S}^{(t)}) + \lambda \|\Sigma^{-1}\|_1 \right\}, \tag{6.1}$$

where $\hat{S}^{(t)}$ is a weighted covariance matrix given by

$$\hat{S}^{(t)} = \frac{\sum_s w_{st} X_s X_s^T}{\sum_s w_{st}}.$$

At a given time point t, the weight corresponding to observation at time point s is defined

using a symmetric non-negative kernel K as

$$w_{st} = \frac{K\left(|s-t|/h\right)}{\sum_s K\left(|s-t|/h\right)}.$$

The optimization problem in (6.1) can be solved using the graphical lasso algorithm of Friedman et al. (2008). One example of a weighted covariance matrix that can be used is

$$\hat{S}^{(t)} = \frac{1}{2}\left(\frac{1}{2}X^{(t-1)}X^{(t-1)^T} + X^{(t)}X^{(t)^T} + \frac{1}{2}X^{(t+1)}X^{(t+1)^T}\right),$$

which assigns weight $1/2$ to each of observations $X^{(t-1)}$ and $X^{(t+1)}$.

Kolar and Xing (2011) established the model selection consistency of the procedure in Zhou et al. (2010). They showed that the structure of the undirected graphical model can be consistently recovered in the high-dimensional setting, under suitable conditions, when the dimension diverges with the sample size. If $\Sigma_t = (\sigma_{ij}^{(t)})$, where $\sigma_{ij}^{(t)}$ is a smooth function, denoting the covariance between variables i and j at time point t, then $\sigma_{ij}^{(t)}$ must have bounded first and second order derivatives at all times:

$$\max_{ij}\sup_t\left|\frac{\partial}{\partial t}\sigma_{ij}^{(t)}\right| \leq C_1$$

and

$$\max_{ij}\sup_t\left|\frac{\partial^2}{\partial t^2}\sigma_{ij}^{(t)}\right| \leq C_2.$$

Therefore, each element of the covariance matrix Σ_t changes smoothly over time.

Estimating Multiple Graphical Models with Common Structure (Guo et al., 2011): Guo et al. (2011) investigated the problem of jointly estimating multiple graphical models that share the same variables and have common structure. They reparametrized each off-diagonal element $\theta_{ij}^{(k)}$ as $\theta_{ij}^{(k)} = \omega_{ij}\gamma_{ij}^{(k)}$ for $1 \leq i \neq j \leq p$ and $1 \leq k \leq K$, where $\omega_{ij} \geq 0$ to avoid sign ambiguity, and $\omega_{ij} = \omega_{ji}$, $\gamma_{ij}^{(k)} = \gamma_{ji}^{(k)}$ to preserve symmetry. For

diagonal elements, $\omega_{ii} = 1$ and $\gamma_{ij}^{(k)} = \theta_{ij}^{(k)}$. Therefore, if $\omega_{ij} = 0$, then no edge is present between nodes i and j across all networks, and if $\omega_{ij} \neq 0$, some networks may have $\gamma_{ij}^{(k)} = 0$ while others may have $\gamma_{ij}^{(k)} \neq 0$, allowing for differences in structure between networks. To estimate this model, they proposed a penalized likelihood approach, which involves solving the following optimization problem

$$\underset{\Omega, \Gamma^{(k)}}{\arg\min} \sum_{k=1}^{K} \left\{ \operatorname{tr}(S^{(k)}\Theta^{(k)}) - \log\det\Theta^{(k)} + \lambda_1 \|\Omega\|_1 + \lambda_2 \sum_{k=1}^{K} \|\Gamma^{(k)}\|_1 \right\}, \tag{6.2}$$

where λ_1, λ_2 are two tuning parameters. The parameter λ_1 controls the sparsity of the common factors ω_{ij}.

6.3 Proposed Methods

Suppose that we have n replications of a T time point longitudinal study for which p variables are measured. The data can be summarized as an $n \times p \times T$ array $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)^T$, where $\mathbf{X}_i$ is a $p \times T$ matrix with columns $X_i^{(t)} = (X_{i1}^{(t)}, \ldots, X_{ip}^{(t)})^T$. We assume that $X_i^{(1)}, \ldots, X_i^{(T)}$ are independent but not identically distributed, allowing the distribution, and hence the underlying graph, to evolve over time. Assuming that $X_1^{(t)}, \ldots, X_n^{(t)}$ are independently drawn from a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix Σ_t, estimating the graph at time point t from the data amounts to estimating a sparse inverse covariance matrix $\Theta_t = \Sigma_t^{-1}$. While static network inference techniques can be used to infer a network at each time point t, such methods do not leverage information about recurring edges that appear in the interaction networks at nearby time points.

6.3.1 A Sequential Approach for Estimating Time-Varying Networks

The re-scaled log-likelihood of the observations (up to a constant) at time point t is given by $\ell(\Theta_t; S_t) = \log|\Theta_t| - \operatorname{tr}(S_t\Theta_t)$, where $S_t = \frac{1}{n}\mathbf{X}^{(t)^T}\mathbf{X}^{(t)}$. One approach is to estimate T individual networks. For each $t = 1, 2, \ldots, T$, we can compute a separate L_1-regularized

estimator by solving

$$\hat{\Theta}_t = \underset{\Theta_t \succ 0}{\arg\max} \left\{ \ell(\Theta_t; S_t) - p_{\lambda_t}(\Theta_t) \right\},$$

where $p_\lambda(\Theta_t) = \|W_t * \Theta_t\|_1$ is the weighted L_1 penalty for some symmetric weight matrix W_t (e.g. adaptive lasso weights) with * denoting Schur (coordinate-wise) matrix multiplication. Such a procedure, however, does not take advantage of the common structure between networks at consecutive time points.

Therefore, we propose to estimate $\Theta_1, \ldots, \Theta_T$ by solving the following sequential optimization problems

$$\hat{\Theta}_1 = \underset{\Theta_1 \succ 0}{\arg\max} \left\{ \ell(\Theta_1; S_1) - p_{\lambda_1}(\Theta_1) \right\} \tag{6.3}$$

and for $t = 2, \ldots, T$,

$$\hat{\Theta}_t = \underset{\Theta_t \succ 0}{\arg\max} \left\{ \ell(\Theta_t; S_t) - p_{\lambda_t}(\Theta_t) - p_{\nu_1,\nu_2}(\Theta_t \mid \hat{\Theta}_{t-1}) \right\}, \tag{6.4}$$

where $p_{\nu_1,\nu_2}(\Theta|\tilde{\Theta}) = \nu_1 \mathrm{tr}(\tilde{\Theta}^{-1}\Theta) - \nu_2 \log\det\Theta$ for $\nu_1, \nu_2 > 0$. This type of sequential prediction approach has many real-world applications, where data become available sequentially in time. Denote by $\{d_j\}_{j=1}^p$ the eigenvalues of Θ. Then $p_{\nu_1,\nu_2}(\Theta|\tilde{\Theta}) = \nu_1 \mathrm{tr}(\tilde{\Theta}^{-1}\Theta) - \nu_2 \log\det\Theta = \nu_1 \mathrm{tr}(\tilde{\Theta}^{-1}\Theta) - \nu_2 \sum_{j=1}^p \log d_j = \nu_1 \mathrm{tr}(\tilde{\Theta}^{-1}\Theta) + \nu_2 \sum_{j=1}^p \log\left(\frac{1}{d_j}\right)$. Therefore, a positive value of ν_2 will keep the eigenvalues of Θ away from 0.

The case $\lambda = 0$ corresponds to placing a Wishart prior on Θ_t with α degrees of freedom and positive-definite scale matrix $\frac{1}{\alpha}\hat{\Theta}_{t-1}$:

$$\Theta_t|\hat{\Theta}_{t-1} \sim \text{Wishart}\left(\alpha, \frac{1}{\alpha}\hat{\Theta}_{t-1}\right) \implies p(\Theta_t) \propto |\Theta_t|^{\frac{\alpha-p-1}{2}} e^{-\frac{\alpha}{2}\mathrm{tr}(\hat{\Theta}_{t-1}^{-1}\Theta_t)}, \tag{6.5}$$

and $\mathbb{E}(\Theta_t|\hat{\Theta}_{t-1}) = \hat{\Theta}_{t-1}$. Therefore,

$$\nu_1 = \frac{\alpha}{n} \text{ and } \nu_2 = \frac{\alpha - p - 1}{n} = \nu_1 - \frac{p+1}{n},$$

which provides guidance for the choice of ν_1 and ν_2. Further, if $\lambda = 0$, then the penalized maximum likelihood estimator of Θ_t is $\left\{\frac{1}{1+\nu_2}\left(S + \nu_1 \hat{\Theta}_{t-1}^{-1}\right)\right\}^{-1}$.

6.3.1.1 Computational Algorithm

The objective function in (6.4) to be optimized can be written as

$$\log\det\Theta - \operatorname{tr}\left\{c(S + \nu_1\hat{\Theta}_{t-1})\Theta\right\} - \lambda\|W * \Theta\|_1,$$

where $c = \frac{1}{\nu_2+1}$, which can be solved using the efficient graphical lasso algorithm of Friedman et al. (2008). We set $\nu_1 = \nu_2 = \nu$. For each value of ν, we then select λ from a find grid and choose the value of ν that yields the densest graph from a coarse grid. The selection of λ is done based on BIC.

6.3.2 Joint Estimation of Time-Varying Networks

In Section 6.3.1, we explored a sequential penalized likelihood approach for estimating time-varying networks and their corresponding precision matrices $\Theta_1, \ldots, \Theta_T$. We proposed to estimate the network at time point t by encouraging similar structure between Θ_t and the reconstructed precision matrix $\hat{\Theta}_{t-1}$ at the previous time point through the inclusion of a Wishart penalty to the L_1-penalized log-likelihood.

Another possible approach would be to jointly estimate $\Theta_1, \ldots, \Theta_T$ by maximizing the following objective function

$$(\hat{\Theta}_1, \ldots, \hat{\Theta}_T) = \underset{\Theta_1,\ldots,\Theta_T \succ 0}{\arg\max}\left[\sum_{t=1}^{T}\ell(\Theta_t; S_t) - \lambda\sum_{t=1}^{T}\|W_t * \Theta_t\|_1 - \nu\sum_{t=2}^{T}\left\{\operatorname{tr}(\Theta_{t-1}^{-1}\Theta_t) - \log\det\Theta_t\right\}\right], \tag{6.6}$$

for some data-dependent, symmetric weight matrix W_t, where S_t is the sample covariance matrix at time point t, and $\lambda, \nu > 0$ are two tuning parameters. The tuning parameters λ and ν will again be selected using BIC. In what follows, we provide the computational details for solving the optimization problem in (6.6).

6.3.2.1 Computational Algorithm

We solve (6.6) using a block coordinate descent procedure, maximizing (6.6) with respect to one Θ_t at a time, while leaving the rest fixed. Let $\hat{\Theta}_1^{(0)}, \ldots, \hat{\Theta}_T^{(0)}$ be initial estimates of $\Theta_1, \ldots, \Theta_T$. At iteration k, we solve the following optimization problems

$$\hat{\Theta}_1^{(k+1)} = \underset{\Theta_1 \succ 0}{\arg\max} \left\{ \ell(\Theta_1; S_1) - \lambda \|W_1 * \Theta_1\|_1 - \nu \mathrm{tr}(\Theta_1^{-1} \hat{\Theta}_2^{(k)}) \right\}, \tag{6.7}$$

for $t = 2, \ldots, T-1$,

$$\hat{\Theta}_t^{(k+1)} = \underset{\Theta_t \succ 0}{\arg\max} \left[\ell(\Theta_t; S_t) - \lambda \|W_t * \Theta_t\|_1 - \nu \left\{ \mathrm{tr}(\hat{\Theta}_{t-1}^{(k)-1} \Theta_t) - \log\det \Theta_t \right\} - \nu \mathrm{tr}(\Theta_t^{-1} \hat{\Theta}_{t+1}^{(k)}) \right], \tag{6.8}$$

and

$$\hat{\Theta}_T^{(k+1)} = \underset{\Theta_T \succ 0}{\arg\max} \left[\ell(\Theta_T; S_T) - \lambda \|W_T * \Theta_T\|_1 - \nu \left\{ \mathrm{tr}(\hat{\Theta}_{T-1}^{(k)-1} \Theta_T) - \log\det \Theta_T \right\} \right]. \tag{6.9}$$

Each of the optimization problems (6.7)-(6.9) is convex. The optimization problem in (6.9) can be solved using the efficient graphical lasso algorithm of Friedman et al. (2008). We use a generalized gradient descent procedure to solve the optimization problems in (6.7) and (6.8).

The objective functions to be maximized in (6.7) and (6.8) are of the form

$$\log\det \Theta - \mathrm{tr}(\tilde{S}\Theta) - \nu \mathrm{tr}(\Theta^{-1}\tilde{\Theta}) - \lambda \|W * \Theta\|_1 \tag{6.10}$$

for some positive definite matrices $\tilde{S}$ and $\tilde{\Theta}$. We let $g(\Theta) = -\log\det\Theta + \mathrm{tr}(\tilde{S}\Theta) + \nu \mathrm{tr}(\Theta^{-1}\tilde{\Theta})$ and $p(\Theta) = \lambda \|W * \Theta\|_1$. Then solving the maximization problems in (6.7) and (6.8) amount to solving the following minimization problem

$$\mathrm{Minimize}_{\Theta \succ 0} \left\{ g(\Theta) + p(\Theta) \right\}, \tag{6.11}$$

where g is a differentiable function and p is a non-differentiable function. Problem (6.11) is convex since both g and p are convex. Therefore, any local minimum is guaranteed to be the global minimum. To solve (6.11), we use a generalized gradient descent algorithm. In what follows, we provide a description of the general generalized gradient descent algorithm (see Bien and Tibshirani, 2011, and references therein).

Generalized Gradient Descent: Given a non-differentiable function f such that $f(x) = g(x) + p(x)$, where g is convex and differentiable, and $p(x)$ is convex, not necessarily differentiable, suppose that we wish to solve the problem

$$\text{Minimize}_{x \in \mathcal{C}} \left\{ g(x) + p(x) \right\}. \tag{6.12}$$

Define the proximal operator

$$\text{prox}_t(x) = \underset{z \in \mathcal{C}}{\arg\min} \left\{ \frac{1}{2t} \|x - z\|^2 + p(z) \right\}. \tag{6.13}$$

Then generalized gradient descent solves (6.12) by initializing $x^{(0)}$ and then repeatedly updating $x^{(k)}$ as follows

$$x^{(k+1)} = \text{prox}_t \left\{ x^{(k)} - t \nabla g(x^{(k)}) \right\} \tag{6.14}$$

for $k = 1, 2, \ldots,$ until convergence. That is, at each iteration, generalized gradient descent solves

$$x = \underset{z \in \mathcal{C}}{\arg\min} \left[\frac{1}{2t} \|z - \{x - t \nabla g(x)\}\|^2 + p(z) \right]. \tag{6.15}$$

Bien and Tibshirani (2011) had proposed a generalized gradient descent procedure for solving a similar optimization problem to the one in (6.10) using a different convex function g. Following the algorithmic construction of Bien and Tibshirani (2011), if $\tilde{S}$ and $\tilde{\Theta}$ are positive definite, we may tighten the constraint $\Theta \succ 0$ to $\Theta \succeq \delta I_p$ for some $\delta > 0$, which we can compute and will depend on the smallest eigenvalues of $\tilde{S}$ and $\tilde{\Theta}$. We prove this in Proposition 5. Therefore, in our case, we solve

$$\text{Minimize}_{\Theta \succeq \delta I_p} \left\{ -\log\det\Theta + \text{tr}(\tilde{S}\Theta) + \nu \text{tr}(\Theta^{-1}\tilde{\Theta}) + \lambda \|W * \Theta\|_1 \right\}. \tag{6.16}$$

Since the matrix derivative of g with respect to Θ is

$$\frac{dg(\Theta)}{d\Theta} = -\Theta^{-1} + \tilde{S} - \nu\Theta^{-1}\tilde{\Theta}\Theta^{-1}, \tag{6.17}$$

the generalized gradient steps are

$$\Theta = \underset{\Omega \succeq \delta I_p}{\arg\min} \left\{ \frac{1}{2t} \|\Omega - \Theta + t(\tilde{S} - \Theta^{-1} - \nu\Theta^{-1}\tilde{\Theta}\Theta^{-1})\|_F^2 + \lambda \|W * \Omega\|_1 \right\}. \tag{6.18}$$

Without the constraint $\Omega \succeq \delta I_p$, this reduces to the update equation

$$\Theta \leftarrow S\left\{ \Theta - t(\tilde{S} - \Theta^{-1} - \nu\Theta^{-1}\tilde{\Theta}\Theta^{-1}), \lambda t\, W \right\},$$

where S is the elementwise soft-thresholding operator $S(A, B)_{ij} = \text{sgn}(A_{ij})(|A_{ij}| - B_{ij})_+$. If the solution to (6.18) ignoring the constraint $\Omega \succeq \delta I_p$ has minimum eigenvalue greater than or equal to δ, then the above procedure is valid. However, if the minimum eigenvalue is less than δ, then we perform the minimization in (6.18) using the alternating direction method of multipliers (Boyd et al., 2011; Bien and Tibshirani, 2011). The ADMM algorithm for solving the general problem

$$\underset{X \succeq \delta I_p}{\arg\min} \left\{ \|X - A\|_F^2 + \lambda \|W * X\| \right\} \tag{6.19}$$

has been implemented in the `R` package `spcov`. Note that when $p > n$, S is not full rank

and so we set $S = S + \alpha I_p$ for some $\alpha > 0$.

We prove two facts, taken from Bien and Tibshirani (2011), that are needed to establish a convergence rate of the algorithm. In Proposition 4, we show that $dg(\Theta)/d\Theta$ is Lipschitz continuous on $\Theta \succeq \delta I_p$. We then show that the constraint $\Theta \succ 0$ can indeed be tightened to $\Theta \succeq \delta I_p$ in Proposition 5. Therefore, by Propositions 4 and 5, generalized gradient descent is guaranteed to get within ϵ of the optimal value in $O(1/\epsilon)$ steps (Bien and Tibshirani, 2011, and references therein).

Proposition 4. The function $\frac{dg(\Theta)}{d\Theta} = -\Theta^{-1} + \tilde{S} - \nu\Theta^{-1}\tilde{\Theta}\Theta^{-1}$ is Lipschitz continuous over the region $\Theta \succeq \delta I_p$.

Proof: We show that $dg(\Theta)/d\Theta$ is Lipshitz continuous on $\Theta \succeq \delta I_p$ by bounding its first derivative. By the product rule[1] for matrix derivatives, we have that

$$\begin{aligned}\frac{d}{d\Theta}(-\Theta^{-1} + \tilde{S} - \nu\Theta^{-1}\tilde{\Theta}\Theta^{-1}) = (\Theta^{-1}\otimes\Theta^{-1}) - \\ \nu\left\{(\Theta^{-1}\tilde{\Theta}\otimes I_p)(-\Theta^{-1}\otimes\Theta^{-1}) + (I_p\otimes\Theta^{-1})(I_p\otimes\tilde{S})(-\Theta^{-1}\otimes\Theta^{-1})\right\} \\ = (\Theta^{-1}\otimes\Theta^{-1}) + \nu\left\{(\Theta^{-1}\tilde{\Theta}\Theta^{-1})\otimes\Theta^{-1} + \Theta^{-1}\otimes(\Theta^{-1}\tilde{\Theta}\Theta^{-1})\right\}.\end{aligned}$$

Now we obtain a bound on the spectral norm of this matrix as follows

$$\begin{aligned}\left\|\frac{d}{d\Theta}\frac{dg}{d\Theta}\right\|_2 &\leq \|\Theta^{-1}\otimes\Theta^{-1}\|_2 + \nu\|(\Theta^{-1}\tilde{\Theta}\Theta^{-1})\otimes\Theta^{-1}\|_2 + \nu\|\Theta^{-1}\otimes(\Theta^{-1}\tilde{\Theta}\Theta^{-1})\|_2 \\ &\leq \|\Theta^{-1}\|_2^2 + 2\nu\|\Theta^{-1}\tilde{\Theta}\Theta^{-1}\|_2\|\Theta^{-1}\|_2 \\ &\leq \|\Theta^{-1}\|_2^2 + 2\nu\|\tilde{\Theta}\|_2\|\Theta^{-1}\|_2^3,\end{aligned}$$

where the first inequality follows from the triangle inequality, the second follows since $\|A\otimes B\|_2 = \|A\|_2\|B\|_2$, and the third follows from sub-multiplicativity of the spectral norm. Thus, if $\Theta \succeq \delta I_p$, then $\Theta^{-1} \succeq \delta^{-1} I_p$ and so

$$\left\|\frac{d}{d\Theta}\frac{dg}{d\Theta}\right\|_2 \leq \delta^{-2} + 2\nu\|\tilde{\Theta}\|_2\delta^{-3}.$$

[1]Product Rule for Matrix Derivatives: Let $F(X) = G(X)H(X)$, then $F'(X) = \left\{I\otimes H^T(X)\right\}G'(X) + \left\{G(X)\otimes I\right\}H'(X)$.

Therefore, g is Lipschitz continuous on $\Theta \succeq \delta I_p$ with Lipschitz constant $\delta^{-2} + 2\nu\|\tilde{\Theta}\|_2\delta^{-3}$.

Proposition 5. Let Θ^* be an arbitrary positive-definite matrix. The minimization problem in (6.11) is equivalent to

$$\text{Minimize}_{\Theta \succeq \delta I_p} \left\{ -\log\det\Theta + \text{tr}(\tilde{S}\Theta) + \nu\text{tr}(\Theta^{-1}\tilde{\Theta}) + \lambda\|W * \Theta\|_1 \right\} \tag{6.20}$$

for some $\delta > 0$ that depends on $\lambda_{\min}(\tilde{S})$, $\lambda_{\min}(\tilde{\Theta})$ and $f(\Theta^*)$.

Proof: Let $f(\Theta) = g(\Theta) + \lambda\|W * \Theta\|_1$, where $g(\Theta) = -\log\det\Theta + \text{tr}(\tilde{S}\Theta) + \nu\text{tr}(\Theta^{-1}\tilde{\Theta})$ is the differentiable part of the objective function. Using the eigendecomposition of Θ, we can write $\Theta = \sum_{j=1}^{p} d_j u_j u_j^T$, where $d_1 \geq \ldots \geq d_p$ are the eigenvalues of Θ.

Given an abitrary Θ^* with $f(\Theta^*) < \infty$, the problem in (6.20) is equivalent to the problem

$$\text{Minimize}_{\Theta \succ 0, f(\Theta) \leq f(\Theta^*)} f(\Theta).$$

Therefore, we show that the constraint $f(\Theta) \leq f(\Theta^*)$ implies $\Theta \succeq \delta I_p$ for some $\delta > 0$.

We have that

$$g(\Theta) = \sum_{j=1}^{p} \left(-\log d_j + \frac{\nu u_j^T \tilde{\Theta} u_j}{d_j} + d_j u_j^T \tilde{S} u_j \right) = \sum_{j=1}^{p} h(d_j; \nu u_j^T \tilde{\Theta} u_j, u_j^T \tilde{S} u_j), \tag{6.21}$$

where $h(x; a, b) = -\log x + a/x + bx$. The function h has for $a > 0$ and $b > 0$,

$$\lim_{x \to 0^+} h(x; a, b) = +\infty, \qquad \lim_{x \to \infty} h(x; a, b) = +\infty, \tag{6.22}$$

and

$$\lim_{a \to \infty} h(x; a, b) = +\infty, \qquad \lim_{b \to \infty} h(x; a, b) = +\infty. \tag{6.23}$$

We also have that $h'(x; a, b) = -\frac{1}{x} - \frac{a}{x^2} + b$ and $h''(x; a, b) = \frac{1}{x^2} + \frac{2a}{x^3} \geq 0$ for $x > 0$. Therefore, h is convex for all $x > 0$. Further, h attains its minimum value at $(2b)^{-1}(1 +$

$\sqrt{1+4ab}$), which we denote by d^*.

Also, using the fact that the minimum eigenvalues of $\tilde{S}$ and $\tilde{\Theta}$ may be expressed as $d_{\min}(\tilde{S}) = \min_{\|u\|^2=1} u^T \tilde{S} u$ and $d_{\min}(\tilde{\Theta}) = \min_{\|u\|^2=1} u^T \tilde{\Theta} u$, respectively, it follows that

$$g(\Theta) \geq \sum_{j=1}^{p} h(d_j; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) = h(d_p; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) + \sum_{j=1}^{p-1} h(d_p; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S}))$$
$$\geq h(d_p; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) + (p-1)\, h(d^*; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})).$$

Therefore, $f(\Theta) = g(\Theta) + \lambda \|W * \Theta\|_1 \leq f(\Theta^*)$ implies $g(\Theta) \leq f(\Theta^*)$ and so

$$h(d_p; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) + (p-1)\, h(d^*; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) \leq f(\Theta^*). \tag{6.24}$$

Hence,

$$h(d_p; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) \leq f(\Theta^*) - (p-1)\, h(d^*; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})).$$

Thus, the minimum eigenvalue of Θ, d_p, is constrained to lie in an interval $[\delta_-, \delta_+] = \Big\{ d : h(d; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S})) \leq c \Big\}$, where $c = f(\Theta^*) - (p-1)\, h(d^*; \nu d_{\min}(\tilde{\Theta}), d_{\min}(\tilde{S}))$ and $\delta_-, \delta_+ > 0$. Note that $\delta_- > 0$ since h is continuous and monotone decreasing on $(0, d^*)$ and $\lim_{x \to 0^+} h(x; a, b) = +\infty$.

6.4 Simulation Studies

In this section, we investigate the performance of our proposed procedure as well as that of Zhou et al. (2010) using the weighted covariance matrix

$$\hat{S}_t = \frac{1}{2}\left(\frac{1}{2} S_{t-1} + S_t + \frac{1}{2} S_{t+1}\right), \tag{6.25}$$

where $S_u = \frac{1}{n} \mathbf{X}^{(\mathbf{u})^T} \mathbf{X}^{(\mathbf{u})}$. In our simulation studies, we consider smaller values of the number of time points T and only construct the weighted covariance matrix $\hat{S}_t$ at time point t based on the sample covariance matrices at time points $t-1$ and $t+1$. However,

for larger values of T, the weighted covariance matrix may be constructed by leveraging information from multiple time points. We also consider the adaptive lasso version of their method with weights given by $w_{ij} = 1/|\hat{\theta}_{ij}^{(0)}|^{\gamma}$ for some $\gamma > 0$, where $\hat{\Theta}^{(0)} = \hat{S}_t^{-1}$. We refer to a penalized likelihood approach that uses the weighted covariance matrix rather than the standard sample covariance matrix as the ZLW version. The tuning parameter selection procedure used will be BIC.

We consider two generating mechanisms for the graphs $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(T)}$. Let $A^{(t)}$ denote the adjacency matrix of the corresponding graph $\mathcal{G}^{(t)}$. We take the precision matrix $\Theta^{(t)}$ to be $C^{(t)} + \left\{0.5 - \lambda_{\min}(C^{(t)})\right\}$, where $C^{(t)} = (c_{ij}^{(t)})$ is a symmetric matrix whose sparsity pattern is determined by that of $A^{(t)}$.

- **Simulation 1** ($p = 50$, $T = 6$): Starting at time $t = 1$, we generate an Erdös-Rényi graph $\mathcal{G}^{(1)}$ of $p = 50$ nodes with connection probability 0.1. Therefore, we take $A_{ij}^{(1)} = 1$ with probability 0.1. The parameter $c_{ij}^{(1)}$ is chosen uniformly from $[-0.8, -0.5] \cup [0.5, 0.8]$. Then, we randomly select 10 new edges to be added at each time point $t = 2, 3$, where the parameter $c_{ij}^{(t)}$ is chosen uniformly from $[-0.6, -0.5] \cup [0.5, 0.6]$. At each time point $t = 4, 5$, we then take the 10 smallest values of $|c_{ij}^{(t-1)}|$ and remove their corresponding edges to obtain a new graph $\mathcal{G}^{(t)}$. Finally, at time $t = 6$, we add 10 new edges to the graph with $c_{ij}^{(t)}$ chosen uniformly from $[-0.6, -0.5] \cup [0.5, 0.6]$.

- **Simulation 2** ($p = 50$, $T = 6$): At time $t = 1$, we generate a network with one hub by first setting $A_{ij}^{(1)} = 1$ with probability 0.05 and then taking $A_{i_{\max},j} = 1$ with probability 0.2, where $i_{\max}$ is the index of the node with the largest degree. At times $= 2, 3, 4$, we set $A_{ij}^{(t-1)} = A_{ij}^{(t)}$ and add more edges by setting $A_{ij}^{(t)} = 1$ with probability 0.01. Then at time $t = 5$, we introduce clustering by taking $A_{ij}^{(t)} = 1$ with probability 0.25 for $1 \leq i, j \leq 10$. Finally, at time $t = 6$, we introduce more hubs by adding edges to nodes with degree larger than the 90^{th} percentile, denoted by q. In particular, we set $A_{ij}^{(t)} = 1$ with probability 0.25 for all nodes i with degree larger than q. The parameters $c_{ij}^{(t)}$ are chosen uniformly from $[-0.8, -0.5] \cup [0.5, 0.8]$.

The networks generated for Simulations 1 and 2 are displayed in Figures 6.1 and 6.2. To

assess the performance of each of the methods, we evaluate the true positive rate (TPR, sensitivity), true negative rate (TNR, specificity), and F_1 score, given by

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{6.26}$$

where

$$\text{precision} = \frac{TP}{TP+ FP} \quad \text{and} \quad \text{recall} = \frac{TP}{TP+ FN}. \tag{6.27}$$

The F_1 score can be interpreted as a weighted average of the precision and recall. It reaches its best value at 1 and worst value at 0.

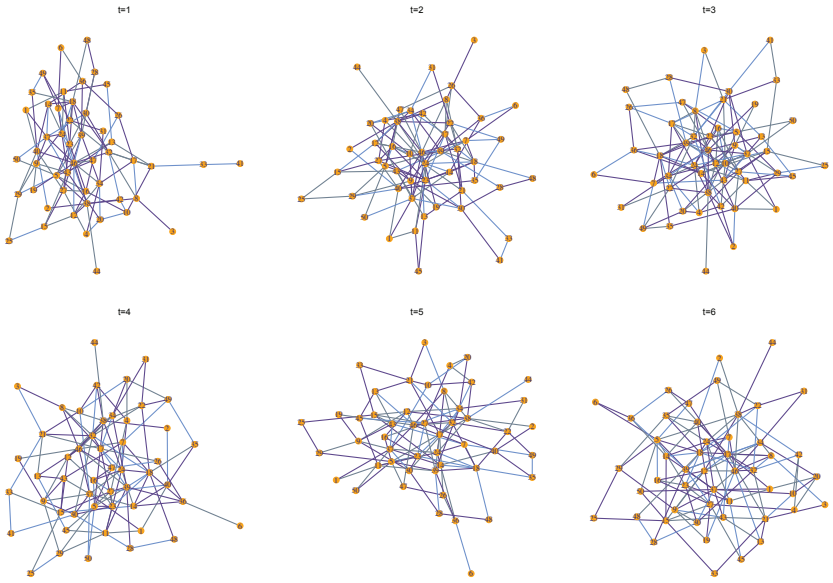

Figure 6.1: Simulation 1 - Networks at time points $t = 1, 2, \ldots, 6$. Edges that are common to all networks are shown in purple.

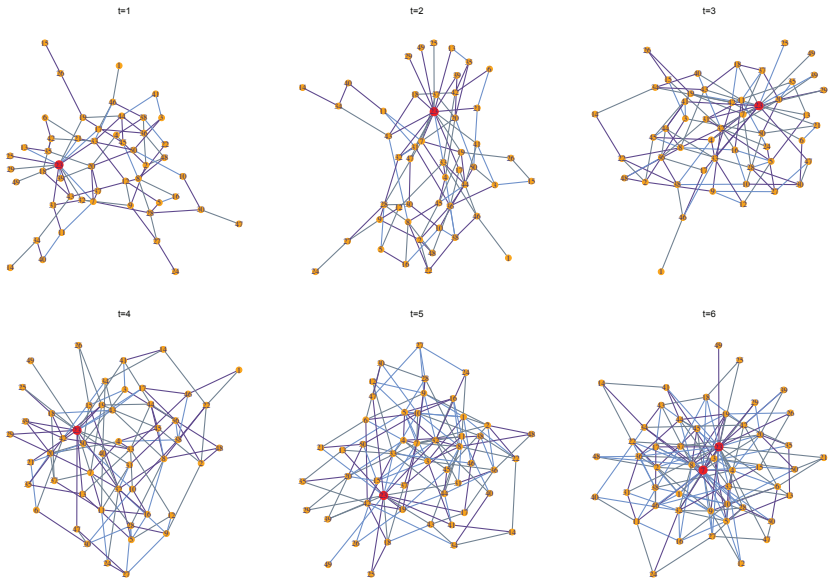

Figure 6.2: Simulation 2 - Networks at time points $t = 1, 2, \ldots, 6$. Edges that are common to all networks are shown in purple.

Discussion of Simulation Results:

When a large number of replications ($n = 50, 100$) are available relative to dimension p, we find that procedures using a weighted covariance matrix, as in (6.25), perform better than those that use the unweighted version. When only a small number of replications ($n = 25$) are available, however, procedures based on the weighted covariance perform similarly to their unweighted covariance counterparts. In our simulation studies, we only considered small values of the number of time points T and the weighted covariance matrix at time t that we used only takes into account information at times $t - 1$ and $t + 1$. However, if more measurements were available, then a more informative kernel can be used to construct the weights in the weighted covariance matrix. For both simulation settings, we find that the joint penalized likelihood approach that makes use of an L_1 penalty and a Wishart-type penalty performs the best, followed closely by the sequential procedure when the weighted covariance matrix is used. The procedures that make use of the Wishart-type penalty are also better able to identify the core edges (edges that occur at all T time points) of the time-evolving network.

	n	TPR	TNR	F_1	TPR (Core Edges)	Sum of Squared Errors
Graphical Lasso	25	0.29	1.00	0.45	0.34	1073.82
	50	0.31	1.00	0.47	0.35	933.94
	100	0.45	0.99	0.60	0.50	808.48
Adaptive Lasso	25	0.36	0.98	0.49	0.40	1031.28
	50	0.43	0.98	0.55	0.47	818.23
	100	0.55	0.98	0.65	0.60	688.75
Sequential Lasso-Wishart	25	0.36	0.98	0.49	0.41	1029.23
	50	0.44	0.98	0.56	0.48	814.66
	100	0.57	0.98	0.66	0.62	679.18
Joint Lasso-Wishart	25	0.32	0.99	0.46	0.37	1117.47
	50	0.42	0.98	0.54	0.46	845.20
	100	0.57	0.98	0.66	0.62	644.49
Graphical Lasso (ZLW)	25	0.29	1.00	0.45	0.34	1070.33
	50	0.30	1.00	0.46	0.34	923.31
	100	0.40	1.00	0.56	0.45	830.79
Adaptive Lasso (ZLW)	25	0.30	1.00	0.46	0.35	1061.11
	50	0.36	1.00	0.53	0.41	859.19
	100	0.70	0.99	0.80	0.75	515.18
Sequential Lasso-Wishart (ZLW)	25	0.31	1.00	0.47	0.35	1056.38
	50	0.41	1.00	0.57	0.46	822.54
	100	0.75	0.99	0.82	0.81	461.81
Joint Lasso-Wishart (ZLW)	25	0.30	1.00	0.46	0.34	1087.55
	50	0.45	1.00	0.60	0.50	775.03
	100	0.76	0.98	0.81	0.81	443.34

Table 6.1: Simulation 1 ($p = 50$, $T = 6$) - Performance as a function of the number of replications n. True positive rate (TPR), true negative rate (TNR), F_1 score, TPR for core edges (edges common across all networks), and sum of squared errors, averaged over 100 replicates.

	n	TPR	TNR	F_1	TPR (Core Edges)	Sum of Squared Errors
Graphical Lasso	25	0.30	1.00	0.46	0.38	934.61
	50	0.31	1.00	0.47	0.39	807.70
	100	0.46	0.99	0.61	0.54	691.30
Adaptive Lasso	25	0.36	0.99	0.50	0.44	899.25
	50	0.43	0.98	0.56	0.50	712.96
	100	0.57	0.98	0.66	0.64	582.13
Sequential Lasso-Wishart	25	0.36	0.99	0.50	0.44	897.95
	50	0.44	0.98	0.56	0.51	708.90
	100	0.59	0.98	0.67	0.66	580.87
Joint Lasso-Wishart	25	0.34	0.99	0.47	0.42	995.76
	50	0.42	0.98	0.54	0.50	739.35
	100	0.60	0.97	0.68	0.67	551.34
Graphical Lasso (ZLW)	25	0.30	1.00	0.46	0.38	937.05
	50	0.30	1.00	0.46	0.38	797.91
	100	0.41	1.00	0.57	0.51	704.42
Adaptive Lasso (ZLW)	25	0.31	1.00	0.47	0.39	927.28
	50	0.36	1.00	0.53	0.45	752.04
	100	0.70	0.99	0.80	0.79	443.866
Sequential Lasso-Wishart (ZLW)	25	0.31	1.00	0.47	0.39	923.41
	50	0.42	1.00	0.58	0.51	710.05
	100	0.76	0.99	0.82	0.84	389.93
Joint Lasso-Wishart (ZLW)	25	0.30	1.00	0.47	0.38	929.75
	50	0.44	1.00	0.60	0.53	672.38
	100	0.77	0.98	0.82	0.85	402.26

Table 6.2: Simulation 2 ($p = 50$, $T = 6$) - Performance as a function of the number of replications n. True positive rate (TPR), true negative rate (TNR), F_1 score, TPR for core edges (edges common across all networks), and sum of squared errors, averaged over 100 replicates.

6.5 Real Data Analysis

We illustrate the proposed methodology with a microarray time series data set, described in Rangel et al. (2004) and found in the `R` package `longitudinal`. The data result from an experiment investigating the response of human T-cells to phorbol myristate acetate (PMA) and ionomicin treatment. After preprocessing, the time course data contain the temporal expression levels of $p = 58$ genes across $T = 10$ time points with 44 replications. The measurements in the experiment were taken at unequally spaced time points: the first one just before treatment, at time point 0, and 9 time points at 2, 4, 6, 8, 18, 24, 32, 48, and 72 hours after treatment. Rangel et al. (2004) used a state space model to estimate a genetic network by combining direct effects and indirect effects via hidden states. Wit and Abruzzo (2015) applied their proposed autoregressive Gaussian graphical model of order 1 to the T-cell dataset, which assumes that genes are conditionally uncorrelated for time lags larger than 1, and which they estimated using an L_1-penalized likelihood approach.

We assume that the observations are independent but not identically distributed across time, and consider the adaptive lasso and lasso-Wishart procedures for estimating a network at each time point t. For each of the procedures, we create 40 bootstrap resamples and retain edges that are reproduced for at least 50% of the resamples.

To estimate the networks and corresponding precision matrices $\Theta_1, \ldots, \Theta_T$, we first apply the adaptive lasso (Fan et al., 2009). In the reconstructed network (obtained after retaining only edges reproducible under random sampling), there is no edge that is common to at least 9 out of the $T = 10$ networks. We next apply the adaptive lasso version of the method in Zhou et al. (2010), using the weighted covariance matrix (6.25). This time we find that common across all $T = 10$ reconstructed networks (after retaining only edges reproducible under random sampling) is an edge connecting genes ITGAM and TCF12. Edges connecting genes APC and PIG3, and IRAK1 and JUNB occur at 9 out of 10 time points.

For the sequential penalized likelihood approach, using the weighted covariance (6.25),

we find the same edge, connecting genes ITGAM and TCF12, common across all networks. Two edges occur at 9 out of 10 time points; they connect genes MAP2K4 and IRAK1, and SMN1 and CCNC. Finally, for the joint estimation approach, none of the edges are common across all $T = 10$ time points. However, 5 edges occur at 9 out of 10 time points. These edges connect genes CD69 and SMN1, ITGAM and TCF12, SMN1 and CCNC, APC and PIG3, and IRAK1 and JUNB. Therefore, as expected, for the lasso-Wishart procedures, there is a greater number of recurring edges.

As in Wit and Abbruzzo (2015), we also find that MCL1, a pro-survival BCL2 family member, is initially a highly connected node, and then loses its connections to other genes over time. This can be explained by the fact that SCF(FBW7) targets MCL1 for ubiquitylation and destruction in order to regulate cellular apoptosis (Wit and Abbuzzo, 2015 and references therein).

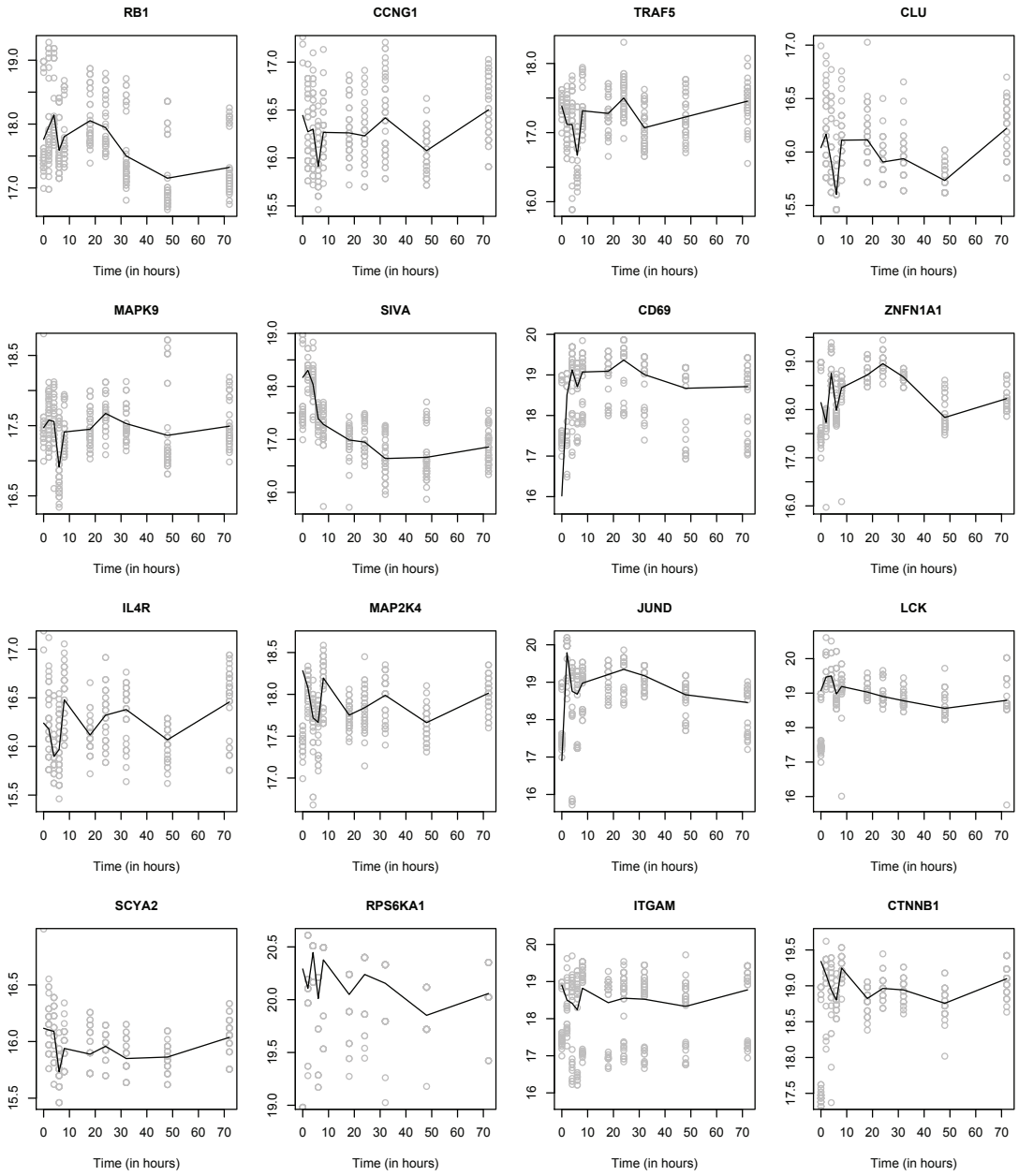

Figure 6.3: Mean gene expression level for 16 of the 58 genes as a function of time (in hours) for the time-course T-cell data set.

6.6 Discussion

In a Bayesian framework, regularization along with the associated degree of uncertainty can be incorporated through the specification of a prior. In the maximum likelihood framework, sparsity and other prior information is incorporated through a penalty applied to the log-likelihood, and a general approach for selecting a penalty is to specify a prior distribution for the parameters. In this chapter, we proposed two penalized likelihood approaches for estimating time-varying networks with a penalty based on a Wishart prior for the precision matrix Θ. We consider a sequential approach, where the estimated precision matrix at time t is taken to be the maximizer of a penalized log-likelihood that encourages sparsity but also shrinkage towards the estimated precision matrix at the previous time point. We also consider a joint estimation approach, where we estimate $(\Theta_1, \ldots, \Theta_T)$ by jointly maximizing a penalized log-likelihood subject to a penalty that encourages shrinkage of consecutive precision matrices toward each other. The joint estimation approach can be used for estimating multiple graphical models with common structure that are not time-varying. We also demonstrate the advantage of borrowing information across time, as suggested by Zhou et al. (2010), through the use of an empirical weighted covariance matrix, constructed by reweighting observations from different time points and then treating those observations as i.i.d. (Song et al., 2009).

Chapter 7

Conclusion and Future Research

7.1 Summary of Thesis Contributions

In recent years, penalized likelihood methodology has been widely used for inducing sparsity in the inverse covariance matrix (or precision matrix) $\Theta = \Sigma^{-1}$. This has been of particular interest because, assuming multivariate normality of the observations, the sparsity pattern of Θ determines conditional dependence relationships between variables. Typically, sparsity is introduced through an L_1 penalty, applied to the elements of the precision matrix. However, often we would like to incorporate more structural information beyond sparsity. In this dissertation, we studied three problems in which L_1-penalization is inappropriate and alternative penalties should be considered.

The first contribution of this thesis was presented in Chapter 3, where we studied the problem of inverse covariance estimation in the case where variables are ordered and considered an alternative parametrization of a covariance matrix, namely the partial autocorrelation (PAC) parametrization. The PAC parametrization has mainly been used in a Bayesian setting for constructing priors for the correlation matrix R, as it removes the positive-definiteness constraint on R, but it has not been considered in the frequentist penalized likelihood framework. Taking advantage of the fact that the PACs are free to vary over the interval (-1,1), we work within the penalized likelihood framework and apply to the Gaussian log-likelihood a nested lasso (Levina et al., 2008) penalty to the PACs. The nested lasso penalty imposes a banded structure in the matrix of PACs, which

corresponds to a banded structure in the inverse covariance matrix, which is appropriate in the ordered data context as it is expected that PACs of large lags are small. An iterative procedure was used to solve the resulting non-convex optimization problem. We also considered another application of the PAC-based methodology in Chapter 4. Specifically, we employed the PAC-based penalized likelihood approach for estimating the order of an autoregressive (AR) model. In the literature, shrinkage is typically applied to the AR coefficients, which provide a convenient representation of the AR process. Rather than imposing shrinkage on the AR coefficients, we instead introduced shrinkage through the PACs, which better reflect the temporal dependence of the AR process. In terms of AR order estimation, we demonstrated the advantage of the PAC-based lasso approach over the lasso and modified lasso methods of Wang et al. (2007b), which apply (weighted) L_1 penalties to the AR coefficients. We also concluded that the proposed PAC-based penalized likelihood methodology performs better in the context of AR order estimation, rather than inverse covariance estimation in the ordered data context.

The conditional dependence relationships between variables determined by the sparsity pattern of Θ can be represented as a graphical model, where vertices correspond to variables and an edge connects two variables if and only if they are conditionally dependent. Such graphical models have been used to model a number of real-world networks, including gene regulatory networks, protein-protein interaction networks, and more recently, microbial interaction networks. The remainder of this thesis focused on the network inference problem, where we considered both the static case (Chapter 5) and the dynamic case (Chapter 6).

The second contribution of this thesis is presented in Chapter 5, where we proposed a method for estimating high-dimensional networks with hubs, inspired by microbiome data. We introduced a weighted lasso approach with novel weights constructed to magnify the differences between rows/columns corresponding to hubs and those corresponding to non-hubs so that hub edges are penalized less compared to non-hub edges. We showed that the proposed estimator possesses the oracle property for fixed p in the sense of Fan and Li (2001) and demonstrated its better finite-sample performance compared to competing

estimators. We then illustrated the performance of the proposed method through a microbiome data application. This application is also relatively new as penalization approaches in microbiome data analysis are only now being considered.

The last contribution of this thesis is presented in Chapter 6, where we studied the problem of estimating time-varying networks from time series or longitudinal data. We assumed that multiple replications, taken under similar experimental conditions, are available, and proposed two new penalized likelihood approaches for estimating time-varying networks in this context, which take into account the common structure between networks at nearby time points. We provided the computational algorithms for solving each of the penalized likelihood problems, and assessed their performance via a simulation study.

7.2 Future Research

To conclude, we suggest some other directions for future work:

- **Selection of the Tuning Parameter:** In a penalized maximum likelihood framework, sparsity is controlled by the tuning parameter. While this thesis does not address in-depth the issue of model selection, the choice of the tuning parameter is crucial to the performance of the penalized likelihood method and in the context of precision matrix estimation has not been adequately addressed in the literature. There are two standard approaches for selecting the tuning parameter. One is to use a resampling scheme, such as cross-validation. The other is to use information criteria, such as AIC or BIC. The tuning parameter selection procedure used should depend on the goal of the study. If the goal is to obtain a model with good predictive power, such as in time series applications, cross-validation and AIC are recommended. On the other hand, if a parsimonious model is desired, then BIC would be more appropriate. The asymptotic properties of CV and BIC in the graphical modelling context was studied by Gao et al. (2012). While BIC-selection of the tuning parameter in the graphical SCAD of Fan et al. (2009) was shown to have desirable asymptotic properties, namely consistency of selection, this asymptotic

result is in some cases not reflected in finite sample. If the goal is to obtain a model reproducible under random sampling, then the stability approach to regularization selection (StARS) procedure should be used. Liu et al. (2010) showed that StARS is partially sparsistent, meaning that the true graph will be contained in the estimated graph with high probability. Such a procedure, however, only performs well (relative to BIC) in high dimensions. Therefore, coming up with a suitable tuning parameter selection procedure is another nontrivial and open-ended problem.

- **Testing for Differences in Graphical Models:** Much of the work has focused on methodology for point estimation of inverse covariance matrices, as studied in this thesis. These methods are typically based on performing sparse selection, and the challenge that remains is statistical inference (obtaining p-values and confidence intervals) after selection (post-model selection inference). Post-model selection methods are two-step procedures resulting from first selecting a model and estimating the parameters in the selected model by, for example, maximum likelihood, and then constructing confidence intervals using asymptotic normality of the MLE. However, some concerns (superefficiency phenomenon, non-uniform convergence) have been raised about post-model selection methods, which are discussed in detail in Leeb and Pötscher (2008).

  Recently, there has been alternative work on quantifying inferential uncertainty for high-dimensional graphical models (e.g. Janková and van de Geer, 2015). In particular, Janková and van de Geer (2015) proposed a de-sparsified estimator based on the graphical lasso, obtained by removing the bias term associated with the penalty, and proved asymptotic normality of the new estimator for sub-Gaussian observations under certain regularity conditions. Thresholding the new estimator will then lead to edge selection guarantees (i.e. thresholding the de-sparsified estimator at some level depending on α will remove all zero entries with probability $1 - \alpha$, asymptotically).

  The development of methodology for quantifying inferential uncertainty is important for instance in differential networks, where the goal is to test equality of net-

works corresponding to two different populations, and it is possible that this can be done by building off of the work of Janková and van de Geer (2015). An application of this problem can be found in Section 5.6 of this thesis, where we had instead contrasted reconstructed networks for case and control groups by testing for significant differences in network indices (e.g. global clustering coefficient, mean betweenness centrality, network density, degree centrality) via a permutation test. Such a test is based on the (weighted) graphical lasso network reconstruction and looks for significant differences in network structure.

- **Regression Analysis Using Network-Structured Predictors with Applications to Microbiome Data:** In Chapters 5 and 6, we focused on the network estimation problem, which has applications to microbiome data. In microbiome studies, one area of inquiry requiring further elucidation is the relationship between the time-evolving microbial interaction network and the health of the human host. The human gut microbiome has been shown to be associated with many diseases, such as obesity (Turnbaugh et al., 2009) and diabetes (Qin et al., 2012). While there has been some work recently on regression analysis for microbiome compositional data (e.g. Shi et al., 2016), where the goal is to identify the microbial taxa that are associated with a continuous response such as body mass index (BMI), there is less work on studying the relationship between a response and the structure of the microbial interaction network.

References

1. Aitchison, J. (1981). A new approach to null correlations of proportions. *Journal of Mathematical Geology*, **13**, 175-189.

2. Aitchison, J. (2003). *The Statistical Analysis of Compositional Data.* Caldwell, New Jersey: The Blackburn Press.

3. Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institue of Statistical Mathematics*, **21**, 243-247.

4. Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki (Eds.), *Second international symposium on information theory* (pp. 267-281). Budapest: Academiai Kiado.

5. Ban, Y., An, L., and Jiang, H. (2015). Investigating Microbial Co-Ocurrence Patterns Based on Metagenomic Compositional Data. *Bioinformatics*, **31**, 3322-3329.

6. Banerjee, O., Ghaoui, L. E. and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, **9**, 485-516.

7. Barabási, A.-L. and Albert R. (1999). Emergence of scaling in random networks. *Science*, **286**, 509-512.

8. Barnard, J., McCulloch, R., and Meng, X.L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, **10**, 1281-1311.

9. Barndorff-Nielsen, O. and Schou, G. (1973). On the Parameterization of Autoregressive Models by Partial Autocorrelations. *Journal of Multivariate Analysis*, **3**, 408-419.

10. Bickel, P.J. and Levina, E. (2008a). Regularized Estimation of Large Covariance Matrices. *Annals of Statistics*, **36**, 199-227.

11. Bickel, P.J. and Levina, E. (2008b). Covariance Regularization by Thresholding. *Annals of Statistics*, **36**, 2557-2604.

12. Bickel, P.J. and Li, B. (2006). Regularization in Statistics. *Sociedad de Estadística e Investigación Operativa Test*, **15**, 271-344.

13. Bien, J. and Tibshirani, R.J. (2011). Sparse estimation of a covariance matrix. *Biometrika*, **98**, 807-820.

14. Bien, J. and Tibshirani, R.J. (2012). Package 'spcov'. https://cran.r-project.org/web/packages/spcov/spcov.pdf.

15. Box, G.E.P., Jenkins, G.M. and Reinsel, G.C. (2008). *Time Series Analysis: Forecasting and Control, 4^{th} edition*, Hoboken, New Jersey: John Wiley & Sons, Inc.

16. Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*, Cambridge: Cambridge University Press.

17. Breiman, L. (1995). Better Subset Regression Using the Nonnegative Garrote. *Technometrics*, **37**, 373-384.

18. Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *Annals of Statistics*, **24**, 2350-2383.

19. Brockwell, P. and Davis, R.A. (2002). *Introduction to Time Series and Forecasting*, 2nd ed. New York: Springer-Verlag.

20. Buckland, S.T., Burnham, K.P. and Augustin, N.H. (1997). Model Selection: An Integral Part of Inference. *Biometrics*, **53**, 603-618.

21. Burg, J.P. (1967). Maximum Entropy Spectral Analysis, *Proceedings of the 37th Meeting of the Society of Exploration Geophysicists*, Oklahoma City, Oklahoma.

22. Cao, Y., Lin, W. and Li, H. (2016). Large Covariance Estimation for Compositional Data via Composition-Adjusted Thresholding. *Unpublished manuscript.*

23. Charbonnier, C., Chiquet, J. and Ambroise, C. (2010). Weighed-Lasso for Structured Network Inference from Time Course Data. *Statistical Applications in Genetics and Molecular Biology*, vol. 9, iss. 1, article 15.

24. Chatfield, C. (1995). Model Uncertainty, Data Mining and Statistical Inference. *Journal of the Royal Statistical Society, Series A*, **158**, 419-466.

25. Chatterjee, A. and Lahiri, S.N. (2010). Asymptotic properties of the residual bootstrap for Lasso estimators. *Proceedings of the American Mathematical Society*, **138**, 4497-4509.

26. Chen, J. and Chen, Z. (2008). Extended Bayesian information criterion for model selection with large model space. *Biometrika*, **94**, 759-771.

27. Chen, C., Davis, R.A., Brockwell, P.J., and Bai, Z.D. (1993). Order Determination for Autoregressive Processes using Resampling Methods. *Statistica Sinica*, **3**. 481-500.

28. Cowpertwait, P.S.P. and Metcalfe, A.V. (2009) *Introductory Time Series with R.* New York: Springer.

29. Csardi, G. (2015). Package 'igraph'. https://cran.r-project.org/web/packages/igraph/igraph.pdf.

30. Daniels, M.J. and Pourahmadi, M. (2009). Modeling covariance matrices via partial autocorrelations. *Journal of Multivariate Analysis*, **100**, 2352-2363.

31. Davison, A.C. and Hinkley, D.V. (1997). *Bootstrap Methods and their Application.* Cambridge: Cambridge University Press.

32. Dempster, A. (1972). Covariance Selection. *Biometrics*, **28**, 157-175.

33. Dey, D.K. and Srinivasan, C. (1985). Estimation of a covariance matrix under Stein's loss. *Annals of Statistics*, **13**, 1581-1591.

34. Donoho, D.L. and Johnstone, I.M. (1994). Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika*, **81**, 425-455.

35. Dorogovtsev, S.N. and Mendes, J.F.F. (2003). *Evolution of Networks: From Biological Networks to the Internet and WWW*, New York: Oxford University Press.

36. Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics*, **7**, 1-26.

37. Efron, B. (2004). Least Angle Regression. *Annals of Statistics*, **32**, 407-499.

38. Efron, B. (2014). Estimation and Accuracy after Model Selection. *Journal of the American Statistical Association*, **109**, 991-1007.

39. Fan, J., Feng, Y. and Wu, Y. (2009). Network exploration via the adaptive lasso and SCAD penalties. *Annals of Applied Statistics*, **3**, 521-541.

40. Fan, J., and Gijbels, I. (1996). *Local Polynomial Modelling and its Applications*, London: Chapman and Hall.

41. Fan, J. and Li, R. (2001). Variable Selection via Noncave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, **96**, 1348-1360.

42. Fan, J. and Lv, J. (2010). A Selective Overview of Variable Selection in High Dimensional Feature Space. *Statistica Sinica*, **20**, 101-148.

43. Fan, J. and Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *Annals of Statistics*, **32**, 928-961.

44. Faust, K. and Raes, J. (2012). Microbial interactions: from networks to models. *Nature*, **10**, 538-550.

45. Faust, K., Lahti, L., Gonze, D., de Vos, W.M. and Raes, J. (2015). Metagenomics meets time series analysis: unraveling microbial community dynamics. *Current Opinion in Microbiology*, **25**, 56-66.

46. Foygel, R. and Drton, M. (2010). Extended Bayesian Information Criteria for Gaussian Graphical Models. *Advances in Neural Information Processing Systems*, **23**, 2020-2028

47. Frank, I.E. and Friedman, J.H. (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics*, **35**, 109-148.

48. Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, **40**, 35-41.

49. Friedman, J. and Alm, E.J. (2012). Inferring Correlation Networks From Genomic Survey Data. *PLoS Computational Biology*, 8, e1002687.

50. Friedman, J., Hastie, T. and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432-441.

51. Friedman, J., Hastie, T. and Tibshirani, R. (2010) Applications of the lasso and grouped lasso to the estimation of sparse graphical models. *Technical Report.* Stanford University, Stanford.

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

53. Friedman, J., Hastie, T., Hoefling, H. and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, **2**, 302-332.

54. Friedman, J., Hastie, T. Simon, N. and Tibshirani, R. (2016). Package 'glmnet'. https://cran.r-project.org/web/packages/glmnet/glmnet.pdf.

55. Gao, X., Pu, D.Q., Wu, Y., and Xu, H. (2012). Tuning Parameter Selection for Penalized Likelihood Estimation of Gaussian Graphical Model. *Statistica Sinica*, **22**, 1123-1146.

56. Gaskins, J.T., Daniels, M.J. and Marcus, B.H. (2014). Sparsity Inducing Prior Distributions for Correlation Matrices of Longitudinal Data. *Journal of Computational and Graphical Statistics*, **23**, 966-984.

57. Gevers, D., Kugathasan, S., Denson, L.A., Vázquez-Baeza, Y., Van Treuren, W., et al. (2014). The treatment-naive microbiome in new-onset crohn's disease. *Cell Host-Microbe*, **15**, 382-392.

58. Gilbert J.A., Meyer F., Jansson J., Gordon J., Pace N., Tiedje J., Ley R., Fierer N., Field D., Kyrpides N., Glockner F.O., Klenk H.-P., Wommack K.E., Glass E., Docherty K., Gallery R., Stevens R., Knight R. (2010). The Earth Microbiome Project: Meeting report of the 1st EMP meeting on sample selection and acquisition at Argonne National Laboratory October 6th 2010. *Standards in Genomic Science*, 3:3.

59. Gough, E.K., Stephens, D.A., Moodie, E.E.M., Prendergast, A.J., Stoltzfus, R.J., Humphrey, J.H., Manges, A.R. (2015). Linear growth faltering in infants is associated with Acidaminococcus sp. and community-level changes in the gut microbiota. *Microbiome*, **3**: 24.

60. Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, **98**, 1-15.

61. Gupta, S. and Lahiri, S.N. (2014). Discussion of the paper, "Estimation and Accuracy After Model Selection" by B. Efron. *Journal of the American Statistical Association*, **109**, 1013-1015.

62. Haff, L.R. (1980). Empirical Bayes estimation of the multivariate normal covariance matrix. *Annals of Statistics*, **8**, 586-597.

63. Hall, P., Horowitz, J.L., and Jing, B.Y. (1995). On blocking rules for the bootstrap with dependent data. *Biometrika*, **82**, 561-574.

64. Hannan, E.J. and Quinn, B.G. (1979). The Determination of the Order of an Autoregression. *Journal of the Royal Statistical Society, B*, **41**, 190-195.

65. Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd. ed.*. New York: Springer.

66. Hastie, T., Tibshirani, R. and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*, Boca Raton: Chapman and Hall.

67. Hero, A. and Rajaratnam, B. (2012). Hub discovery in partial correlation graphs. *IEEE Transactions on Information Theory*, **58**, 6064-6078.

68. Hipel, K.W. and McLeod, A.I. (1994). *Time Series Modelling of Water Resources and Environmental System*, Amsterdam: Elsevier.

69. Hoerl, A.E. and Kennard, R.W. (1970). Ridge Regression: Biased Estimation for Non-Orthogonal Problems. *Technometrics*, **12**, 55-67.

70. Huang, J.Z., Liu, N., Pourahmadi, M. and Liu, L. (2006). Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, **93**, 85-98.

71. Huang, J.Z., Liu, L., and Liu, N. (2007). Estimation of Large Covariance Matrices of Longitudinal Data With Basis Function Approximations. *Journal of Computational and Graphical Statistics*, **16**, 189-209.

72. Hurvich, C.M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, **76**, 297-307.

73. Janková, J. and van de Geer, S. (2015). Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, **9**, 1205-1229.

74. Jethava, V., Bhattacharyya, C. and Dubhashi, D. (2013). Computational Approaches for Reconstruction of Time-Varying Biological Networks from Omics Data. *Systems Biology: Integrative Biology and Simulation Tools*, 209-239.

75. Joe, H. (2006). Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis*, **97**, 2177-2189.

76. Johnstone, I. M. (2001). On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics*, **29**, 295-327.

77. Khare, K., Oh, S.-Y., and Rajaratnam, B. (2015). A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees. *Journal of the Royal Statistical Society, Series B*, **77**, 803-825.

78. Khorshidi, S., Karimi, M. and Nematollahi, A.R. (2011). New autoregressive (AR) order selection criteria based on the prediction error estimation. *Signal Processing*, **91**, 2359-2370.

79. Knight, K. and Fu, W. (2000). Asymptotics for lasso-type estimators. *Annals of Statistics*, **28**, 1356-1378.

80. Kurtz, Z.D., Müller, C.L., Miraldi, E.R., Littman, D.R., Blaser, M.J., and Bonneau, R.A. (2015). Sparse and Computationally Robust Inference of Microbial Ecological Networks. *PLoS Computational Biology*, 11(5): e1004226. doi: 10.1371/journal.pcbi.1004226.

81. Lahiri, S. (2003). *Resampling Methods for Dependent Data*, New York: Springer-Verlag.

82. Lam, C. and Fan, J. (2009). Sparsistency and Rates of Convergence in Large Covariance Matrix Estimation. *Annals of Statistics*, **37**, 4254-4278.

83. Lauritzen, S. (1996). Graphical models. New York: Oxford University Press.

84. Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, **88**, 365-411.

85. Leeb, H. and Pötscher, B.M. (2005). Model Selection and Inference: Facts and Fiction. *Econometric Theory*, **21**, 21-59.

86. Leeb, H. and Pötscher, B.M. (2006). Can one estimate the conditional distribution of post-model-selection estimators? *Annals of Statistics*, **34**, 2554-2591.

87. Leeb, H. and Pötscher, B.M. (2008). Sparse estimators and the oracle property, or the return of Hodges' estimator. *Journal of Econometrics*, **142**, 201-211.

88. Levina, E., Rothman, A. and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. *Annals of Applied Statistics*, **2**, 245-263.

89. Li, J., Nasidze, I., Quinque, D., Li, M., Horz, H.-P., Andre, C., Garriga, R.M., Halbwax, M., Fischer, A. and Stoneking, M. (2013). The Saliva Microbiome of Pan and Homo. *BMC Microbiology*, **13**, 1:204.

90. Lian, H. (2011). Shrinkage tuning parameter selection in precision matrices estimation. *Journal of Statistical Planning and Inference*, **141**, 2839-2848.

91. Liu, H., Roeder, K. and Wasserman, L. (2010). Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models. *Advances in Neural Information Processing Systems*.

92. Liu, Q. and Ihler, A. (2011). Learning Scale Free Networks by Reweighted L_1 Regularization. *Proceeding of the 14th International Conference on Artificial Intelligence and Statistics*, **15**, 40-48.

93. McLeod, A.I., Hipel, K.W. and Lennox, W.C. (1977). Advances in Box-Jenkins Modeling, 2, Applications. *Water Resources Research*, **13**, 577-586.

94. McLeod, A.I. and Zhang, Y. (2006). Partial Autocorrelation Parameterization for Subset Autoregression. *Journal of Time Series Analysis*, **27**, 599-612.

95. McLeod, A. and Zhang, Y., (2008). Improved subset autoregression: With R package. *Journal of Statistical Software*, **28**, 1-28.

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

97. Meyer, K. (2011). Performance of penalized maximum likelihood in estimation of genetic covariances matrices. *Genetics Selection Evolution*, 43:39. doi: 10.1186/1297-9686-43-39.

98. Muirhead, R.J. (1982). *Aspects of Multivariate Statistical Theory*, Hoboken, New Jersey: John Wiley & Sons, Inc.

99. Nardi, Y. and Rinaldo, A. (2011). Autoregressive process modeling via the Lasso procedure. *Journal of Multivariate Analysis*, **102**, 528-549.

100. Newman, M. (2010) *Networks: An Introduction*, New York: Oxford.

101. Nishii, R. (1984). Asymptotic Properties of Criteria for Selection of Variables in Multiple Regression. *Annals of Statistics*, **12**, 758-765.

102. Opgen-Rhein, R. and Strimmer, K. (2015). Package 'longitudinal'. https://cran.r-project.org/web/packages/longitudinal/longitudinal.pdf.

103. Peng, J., Wang, P., Zhou, N. and Zhu, J. (2009). Partial Correlation Estimation by Joint Sparse Regression Models. *Journal of the American Statistical Association*, **104**, 735-746.

104. Pourahmadi, M. (1999). Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika*, **86**, 677-690.

105. Pourahmadi, M. (2001). *Foundations of Time Series Analysis and Prediction Theory*. New York: Wiley.

106. Pourahmadi, M. (2011). Covariance Estimation: The GLM and Regularization Perspectives. *Statistical Science*, **26**, 369-387.

107. Qin, J., Li, Y., Cai, Z., Li, S., Zhu, J., Zhang, F., Liang, S., Zhang, W., Guan, Y., Shen, D., et al. (2012). A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, **490**, 55-60.

108. Quenouville, M.H. (1949). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society: Series B*, **11**, 68-84.

109. Rangel, C., Angus, J. Ghahramani, Z. Lioumi, M., Sotheran, E., Gaiba, A., Wild, D.L., and Falciani, F. (2004). Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics*, **20**, 1361-1372.

110. Rothman, A.J., Levina, E. and Zhu, J. (2010). A new approach to Cholesky-based covariance regularization in high dimensions. *Biometrika*, **97**, 539-550.

111. Schwarz, G. (1978). Estimating the Dimension of a Model. *Annals of Statistics*, **6**, 461-464.

112. Shao, J. (1996). Bootstrap model selection. *Journal of the American Statistical Association*, **91**, 655-665.

113. Shao, J. (1997). An Asymptotic Theory for Linear Model Selection, *Statistica Sinica*, **7**, 221-264.

114. Shi, P., Zhang, A. and Li, H. (2016). Regression Analysis for Microbiome Compositional Data. *Unpublished manuscript*.

115. Shibata, R. (1976). Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika*, **63**, 117-126.

116. Shumway, R. and Stoffer, D. (2006). Time Series Analysis and its Applications: With R Examples. New York: Springer-Verlag.

117. Sokolov, A., Carlin, D.E., Paull, E.O., Baertsch, R. and Stuart, J.M. (2016). Pathway-Based Genomics Prediction using Generalized Elastic Net. *PLoS Computational Biology*, 12(3): e1004790. doi: 10.1371/journal.pcbi.1004790.

118. Song, L., Kolar, M. and Xing, E.P. (2009). KELLER: estimating time-varying interactions between genes. *Bioinformatics*, **25**, 128-136.

119. Statistics Canada. (2015). Monthly Seasonal Adjusted Unemployment Rates by EI Economic Region. Version updated 2015. Ottawa. http://open.canada.ca/data/en/dataset/aad2bcd4-9f45-4013-b2a6-8367106dc0b2. (November 23, 2015).

120. Stone, M. (1974). Cross-validation choice and assessment of statistical predictions (with Discussion). *Journal of the Royal Statistical Society: Series B*, **36**, 111-147.

121. Tan, K.M. (2015). Package 'hglasso'. https://cran.r-project.org/web/packages/hglasso/hglasso.pdf.

122. Tan, K.M., London, P., Mohan, K., Lee, S.I., Fazel, M. and Witten, D. (2014). Learning Graphical Models with Hubs. *Journal of Machine Learning Research*, **15**, 3297-3331.

123. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267-288.

124. Tikhonov, A.N. (1943). On the stability of inverse problems. *C.R. (Doklady) Acad. Sci. URSS (N.S.)*, **39**, 176-179.

125. Tsay, R.S. (2010). *Analysis of Financial Time Series*, 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

126. Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R., Gordon, J. I. (2007). "The Human Microbiome Project". *Nature*, **449**, 804-810.

127. Wang, H. (2012). Bayesian Graphical Lasso Models and Efficient Posterior Computation. *Bayesian Analysis*, **7**, 867-886.

128. Wang, H, Li, R, Tsai, C.-L. (2007a). Tuning Parameter Selectors for the Smoothly Clipped Absolute Deviation Method. *Biometrika*, **94**, 553-568.

129. Wang, H., Li, G. and Tsai, C.-L. (2007b). Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, **69**, 63-78.

130. Wang, Y. and Daniels, M.J. (2013). Bayesian modelling of the dependence in longitudinal data via partial autocorrelations and marginal variances. *Journal of Multivariate Analysis*, **116**, 130-140.

131. Wang, Y. and Daniels, M.J. (2014). Computationally efficient banding of large covariance matrices for ordered data and connections to banding the inverse Cholesky factor. *Journal of Multivariate Analysis*, **130**, 21-26.

132. Wit, E.C. and Abbuzzo, A. (2015). Inferring slowly-changing dynamic gene-regulatory networks. *BMC Bioinformatics*, **16**, (S-6), S5.

133. Wu, W.B. and Pourahmadi, M. (2003). Nonparametric estimation of large covariance matrices of longitudinal data. *Biometrika*, **90**, 831-844.

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

135. Yuan, M. and Lin, Y. (2007) Model selection and estimation in the Gaussian graphical model. *Biometrika*, **90**, 831-844.

136. Zhang, Y., Li, R. and Tsai, C.-L. (2010). Regularization Parameter Selections via Generalized Information Criterion. *Journal of the American Statistical Association*, **105**, 312-323.

137. Zhao, T., Li, X., Liu, H., Roeder, K., Lafferty, J. and Wasserman, L. (2015). Package 'huge'. https://cran.r-project.org/web/packages/huge/huge.pdf.

138. Zhao, P. and Yu, B. (2006). On Model Selection Consistency of Lasso. *Journal of Machine Learning Research*, **7**, 2541-2563.

139. Zhou, S., Lafferty, J. and Wasserman, L. (2010). Time varying undirected graphs. *Machine Learning Journal*, **80**, 295-319.

140. Zou, H. (2006). The Adaptive Lasso And Its Oracle Properties. *Journal of the American Statistical Association*, **101**, 1418-1429.

141. Zou, H., Hastie, T. and Tibshirani, R. (2007). On the "degrees of freedom" of the lasso, *Annals of Statistics*, **35**, 2173-2192.

142. Zou H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models (with discussion). *Annals of Statistics*, **36**, 1509-1533.